Journal of Automation, Mobile Robotics and Intelligent Systems

JAMRIS

WWW.JAMRIS.ORG • pISSN 1897-8649 (PRINT)/eISSN 2080-2145 (ONLINE) • VOLUME 20, N° 2, 2026

Indexed in SCOPUS

AI-generated illustration

Łukasiewicz
Industrial Research Institute for Automation and Measurements PIAP

Journal of Automation, Mobile Robotics and Intelligent Systems

A peer-reviewed quarterly focusing on new achievements in the following fields:
• automation • systems and control • autonomous systems • multiagent systems • decision-making and decision support •
• robotics • mechatronics • data sciences • new computing paradigms •

Typesetting
SCIENDO, www.sciendo.com

Webmaster
TOMP, www.tomp.pl

Editorial Office
ŁUKASIEWICZ Research Network
– Industrial Research Institute for Automation and Measurements PIAP
Al. Jerozolimskie 202, 02-486 Warsaw, Poland (www.jamris.org)
tel. +48-22-8740109, e-mail: office@jamris.org

The reference version of the journal is e-version. Printed in 100 copies.

Articles are reviewed, excluding advertisements and descriptions of products.

Publisher:
Copyright © by Łukasiewicz Research Network - Industrial Research Institute for Automation and Measurements PIAP

Journal of Automation, Mobile Robotics and Intelligent Systems

VOLUME 20, N˚2, 2026

Contents

Communication with the Manipulator Using Gestures to Enhance the Manipulability of Persons with Reduced Mobility

Submitted: 20th January 2025; accepted: 13th May 2025

Sebastian Koryl, Katarzyna Zadarnowska, Krzysztof Arent

DOI: 10.14313/jamris-2026-015

Abstract:
This paper presents a robotic system that assists people with reduced mobility in the activities of picking up and putting down objects out of their reach. Human-robot communication is non-verbal, using gestures that have been specifically selected for the robot's use. Gestures are read out using an RGB-D camera while the commands they express are executed online by a small UR3 cobot. The evaluation of the system has shown that it is useful and safe in the sense of the SUS and GQS, respectively.

Keywords: *assistive robots, manipulators, 3D vision systems, gesture recognition, human-robot interactions*

1. Introduction

Robots as aids for people with disabilities are of wide interest to users, researchers, and engineers [1]. This is a consequence of the strong diversity of needs of people with disabilities, the high societal relevance of this branch of robotics, and the rapid advances in science and technology. A systematic review of assistive robots is included in [1, 2]. The classification proposed in [1] distinguishes manipulation aids. They are categorised into fixed base and wheelchair manipulator arm systems. MySpoon [3], Mico, Jaco [4], iARM [5] and BATEO [6] can serve as examples. In contrast, the classification in [2] is based on applications: assistance in activities of daily living, fetch and bring activities, food and beverage service, medication delivery services, user-related delivery services, telepresence and communication, monitoring safety, and navigation. This article concerns fixed base manipulator arm systems designed for fetch and bring activities.

The potential of assistive robotic manipulation to improve the quality of life for people with motor impairments is a motivation for work [7]. A Body-Machine Interface (BMI) has been proposed. It is based on on a vest that is equipped with four MTx motion trackers in order to capture shoulder movements. BMI uses a person's residual motor capabilities to generate control signals for a robotic arm. In this way, a high-DOF MICO manipulator can assist people with motor impairments to perform everyday tasks. It is noted in [8] that communication with a robot can be done through gestures, and there are circumstances when this is justified. The authors propose a wireless, human gesture based controlled robotic arm system for tool handling (pick and place) and other applications where human reach is elusive. The gestures are captured using IMU and flex sensors placed on the human hand, and they are in sync with the manipulators movement. The solution is intended primarily for industrial applications, not for assistive robotics. The article [9] presents a system for controlling a robot arm by human fingers and hand movements. The sensory hardware is composed of gyroscpes and flex connected to Arduino microcontroller. Sensory data processing is based on fuzzy logic. This is more of a technology demonstrator than a final solution but it is undoubtedly a step in the search for a good system for communicating with a robot using gestures. In [10] the authors propose a vision-based HCI architecture for the robotic arm by identifying somatosensory motion. The input of the model is collected by Kinect sensors from human body movement. The evaluation of the proposed system was preliminary and limited to the virtual environment.

Recently, a number of works have appeared, [11–13], based on the Google MediaPipe [14] that is an open-source framework that offers developers a platform for building real-time multimedia applications. A gesture control interface for laparoscopic surgery is discussed in [11]. Four RGB cameras are used to capture hand movements. Although the study offers a solid mathematical background and includes user evaluation, it lacks real-world testing. A gesture controlled robotic arm for surgical tool assistance is proposed in [12]. The system is based on YOLOv5 for accurate surgical tool detection, MediaPipe Framework for real-time hand gesture tracking, and the UFactory Lite 6 robotic arm. RGB-D and RGB cameras are used for hand movement capture and tool detection, respectively. The recognized gestures have the meaning of instructions to the manipulator, no telemanipulation takes place. The evaluation was carried out in a virtual RobotDK environment and was based exclusively on selected system performance indicators. The concept of a gesture control interface for teleoperation of quadruped robots with a robotic arm is presented in [13]. An RGB-D camera (Intel RealSense) captures human gestures. They have the meaning of motion commands for the Unitree Go 1 mobile platform. The position of the effector of myCobot manipulator can be teleoperated by hand movement using a custom teleoperatin algorithm. However, evaluation of the system is limited to a basic feasibility study.

There are reports in the literature about research on other forms of human-manipulator communication. In [15], the authors present and discuss a tongue drive system in conjunction with augmented reality for an assistive manipulator. A P300-based brain-computer interface is proposed and analysed in [16]. A minimalist interface for a proof-of-concept control system with higher autonomy for head care tasks using the PR2 robot is proposed in [17]. In the article [18], hand configuration in combination with voice commands are used to define the actions of the manipulator. The RGB-D camera is used to segment the scene, detect objects, including the hand, its position, and the object it is pointing at in the scene.

In parallel with the development of human-robot communication technology, the problem of evaluation of such interactive systems has been addressed. An analysis of the previously cited works leads to the conclusion that two types of evaluation parameters can be distinguished: system performance indicators for feasibility study and user-centered UX/HRI measures for users' opinions study. The system performance indicators include task realization success rate [11, 13, 15, 18], task completion time [7, 11, 13, 15, 16, 18–20], response time [11, 12, 16, 18] task execution accuracy [11,18,20]. In turn, user experience has been assessed on the following measures: Quality of Life in Essential Tremor Questionnaire (QUEST; [19]), System Usability Scale (SUS; [11,19,20]), Godspeed Questionnaire Series (GQS) and Attitudes towards Technology Scale (ATTS; [20]), Van Der Laan's technology acceptance scoring, Ergonomics survey [11], the Questionnaire for the Evaluation of Physical Assistive Devices (QUEAD; [21]). This classification provides a structured framework for comparing our system with prior art.

The contribution of this article consists of four elements that are present together.

1) The concept of a manipulator designed to assist individuals with reduced mobility in tasks involving physical objects handling has been proposed. Human-robot interaction is facilitated through gesture-based commands, which are captured by an RGB-D camera and processed using Google MediaPipe Hands software. Crucially, our design eliminates the need for wearable or hand-mounted sensors to indicate to the manipulator effector the position to be tracked and the actions to be performed with the gripper. This sensor-less control mechanism increases user comfort and accessibility.
2) A complete mathematical background is provided to support the gesture-based control logic, enabling robust and responsive interaction between the user and the robotic arm.
3) The gesture control algorithm was implemented and tested on an existing robotics platform comprising hardware and software, confirming its feasibility outside a simulation environment.
4) The evaluation of the gesture control system was carried out in a significant group of 37 naive participants. UX and HRI tests based on live experiments involving a scenario specific to an assistive robotic arm yielded positive results, i.e. the system is sufficiently useful and safe in the sense of SUS and GQS measures.

Initial research for this work was developed in [22] and [20].

The solution proposed in this work has a simpler design and is more user-friendly than the systems discussed in [8–10]. At the same time, it is complementary to the systems presented in [15, 16] and [18], and contributes to the growing field of accessible, sensorless human–robot interfaces.

2. System Idea

The general concept of the manipulation aid system is shown in Figure 1.

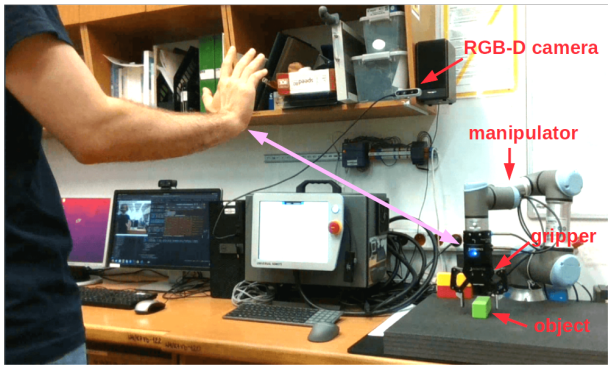


Figure 1. Overall concept of the system

The user can move objects out of their reach into their immediate environment with the support of the manipulator. Commands are given to the manipulator by means of hand gestures, which are recorded using the RGB-D camera.

The set of commands interpreted by the manipulator's sensory system are activation/deactivation, hand tracking, opening and closing of the gripper. Once the gesture command mode is activated, the gripper follows the position of the hand. The gripper can be open or closed, depending on whether the hand is open or closed with a fist. The above makes the manipulator an extension of the user's arm from the user's perspective.

3. System Components

3.1. General System Architecture

The system architecture implementing the concept outlined in Section 3.1 is presented in Figure 2. The user hand is captured by an RGB-D camera. The resulting RGB and depth images are processed by the `/camera_image_processor` module. The extracted hand landmarks positions and recognized gestures are then used to determine the instructions for the robot through the `/robot_controller` module. Consequently, the coordinate system of the gripper follows the coordinate system of the hand and the jaws

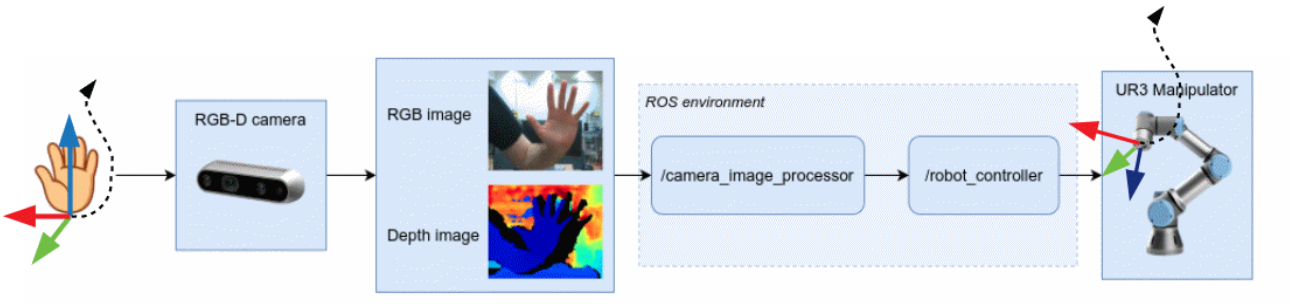


Figure 2. General system architecture

of the gripper follow the instructions expressed by the hand gestures.

3.2. Hardware

The hardware platform of the assistive manipulator under consideration consists of the following components:

- UR3 manipulator with CB3 control box and Robotiq 2F-85 two-finger gripper,
- desktop PC,
- Intel RealSense D435 depth camera (RGB-D).

The camera is connected to the PC via a USB cable while the PC communicates with the CB3 control box via TCP/IP.

The UR3 manipulator is a small cobot and can be adapted as an assistive manipulator. The camera built into the UR3 manipulator is of the RGB type and cannot be used for the gripper to track human hand movement in 3D. For this reason, it was necessary to use an external RGB-D camera.

3.3. Software

Ubuntu 20.04 and ROS Noetic form the software base on the PC for this project. The software of the considered assistive manipulator consists of a number of ROS nodes that are associated with Universal Robots ROS Driver, MoveIt Motion Planning Framework, and gripper. In addition, there are two custom nodes: `/camera_image_processor` and `/robot_controller` mentioned in Section 3.1. A part of the ROS graph with these nodes is included in Figure 3.

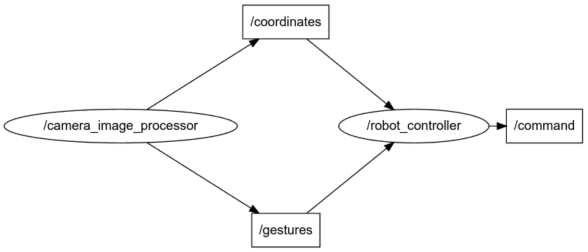
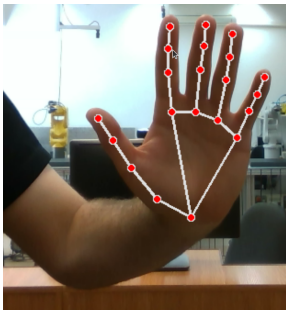
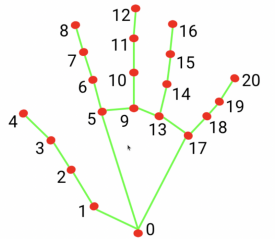


Figure 3. ROS graph

/camera_image_processor is a software which performs RGB image and depth map acquisition and then carries out the data processing.



(a) a human hand and its model in MediaPipe



(b) hand landmarks in the MediaPipe hand model

Figure 4. Google MediaPipe Software Framework: a hand model

The Google MediaPipe Framework [14, 23] was used to implement this node. The work [24] proves that it is a reliable and precise framework for assessing 3D hand movements in clinical applications. Specifically, MediaPipe Hands was used to track hand position and orientation, similar to [11].

The MediaPipe hand landmark recognition model recognizes the hand in the captured data and creates a skeletal model of it, illustrated in Figure 4(a). There are 20 landmarks associated with the hand model shown in Figure 4(b). They can be used to define a coordinate frame $\mathcal{X}_h\mathcal{Y}_h\mathcal{Z}_h$ associated with the hand (a hand coordinate frame).

Let ${}^c p_{h,i}$ denotes a position of the i-th hand landmark in the camera coordinate frame $\mathcal{X}_c\mathcal{Y}_c\mathcal{Z}_c$. The versors $r_{h,x}, r_{h,y}\ r_{h,z}$ of $\mathcal{X}_h\mathcal{Y}_h\mathcal{Z}_h$ are defined as follows:

$$r_{h,z} = \frac{{}^c p_{h,9} - {}^c p_{h,0}}{\|{}^c p_{h,9} - {}^c p_{h,0}\|_2}, \tag{1}$$

$$r_{h,y} = \frac{({}^c p_{h,17} - {}^c p_{h,0}) \times ({}^c p_{h,9} - {}^c p_{h,0})}{\|({}^c p_{h,17} - {}^c p_{h,0}) \times ({}^c p_{h,9} - {}^c p_{h,0})\|_2}, \tag{2}$$

$$r_{h,x} = r_{h,y} \times r_{h,z}. \tag{3}$$

The origin of the hand coordinate frame is located at ${}^c p_{h,0}$. Consequently, a homogeneous transformation matrix from $\mathcal{X}_h\mathcal{Y}_h\mathcal{Z}_h$ to $\mathcal{X}_c\mathcal{Y}_c\mathcal{Z}_c$ takes the following form:

$$T_h^c = \begin{bmatrix} {}^c R_h & {}^c p_{h,0} \\ 0 & 1 \end{bmatrix} \text{ where } {}^c R_h = \begin{bmatrix} r_{h,x} & r_{h,y} & r_{h,z} \end{bmatrix}. \tag{4}$$

The above construct makes some similarities between the coordinate frames of the human hand and the gripper (the latter is denoted by $\mathcal{X}_e\mathcal{Y}_e\mathcal{Z}_e$). In both cases, the z versors have their origin at the wrist and are directed along the fingers to the outside of the arm. In turn, the y versors are directed perpendicular to the plane of the palm on the outside of the inner part of the palm. In the case of a gripper, the inner part of the palm is assumed to be on the camera side of the gripper.

Let T_c^b denotes a homogeneous transformation matrix from $\mathcal{X}_c\mathcal{Y}_c\mathcal{Z}_c$ to the manipulator base coordinate frame $\mathcal{X}_b\mathcal{Y}_b\mathcal{Z}_b$, [25], and $T_h^b := T_c^b T_h^c$. The structure of the resulting matrix is as follows:

$$T_h^b = \begin{bmatrix} {}^bR_h & {}^bp_{h,0} \\ 0 & 1 \end{bmatrix}. \tag{5}$$

The matrix ${}^bR_h$ represents the orientation of $\mathcal{X}_h\mathcal{Y}_h\mathcal{Z}_h$ in $\mathcal{X}_b\mathcal{Y}_b\mathcal{Z}_b$ while ${}^bp_{h,0}$ represents the position of the human wrist in $\mathcal{X}_b\mathcal{Y}_b\mathcal{Z}_b$.

T_h^b is used to determine a reference path for $\mathcal{X}_e\mathcal{Y}_e\mathcal{Z}_e$ in $\mathcal{X}_b\mathcal{Y}_b\mathcal{Z}_b$. Let ${}^b\bar{p}_{h,0}$, ${}^b\bar{R}_h$, ${}^b\bar{p}_e$, ${}^b\bar{R}_e$ are the positions and the orientations of a human hand and a gripper respectively when the control system is activated. Define

$$\bar{p} := {}^b\bar{p}_e - {}^b\bar{p}_{h,0}, \tag{6}$$

$$\bar{R} := ({}^b\bar{R}_h)^{-1} \cdot {}^b\bar{R}_e. \tag{7}$$

Then the reference path for the gripper can be expressed as follows:

$${}^bp_{e,\mathrm{ref}} = {}^bp_{h,0} + \bar{p}, \tag{8}$$

$${}^bR_{e,\mathrm{ref}} = {}^bR_h \cdot \bar{R}. \tag{9}$$

Sometimes it is good to control only the position of the effector, especially when a twitching human palm makes the gripper oscillates and once fixed the orientation of the gripper is sufficient. In this case, the expression (9) takes the form

$${}^bR_{e,\mathrm{ref}} = {}^b\bar{R}_e. \tag{10}$$

Note, that similar mappings are used in direct and bilateral teleoperation [26]. Consequently, two working modes can be distinguished:

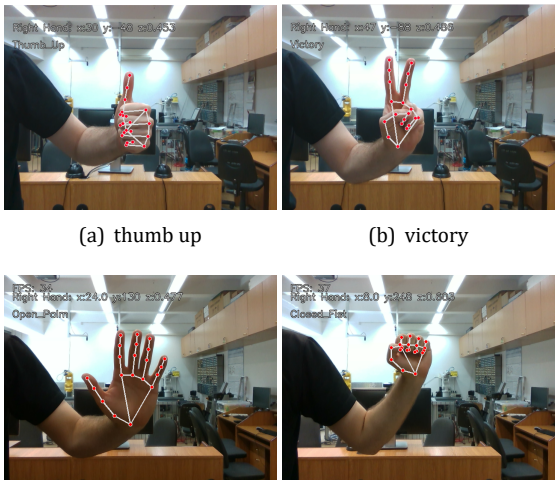
full teleoperation, defined by (1)÷(9);

reduced teleoperation, defined by (1)÷(8), (10).

${}^bp_{e,\mathrm{ref}}$, ${}^bR_{e,\mathrm{ref}}$ are the quantities that are passed to the `/coordinates` topic.

The MediaPipe hand gesture recognition model can recognize several static gestures. The gestures supported by the system are: *closed fist*, *open palm*, *thumb up*, *victory*, *none of them*. They are illustrated in Figure 5. The gesture type is published to the `/gestures` topic once it has been recognized.

Both the position of the wrist and the recognized gestures have an assigned meaning in terms of commands for the manipulator. The position of the wrist determines the position of the gripper. The *open palm* and *closed fist* gestures denote open and closed gripper jaws, respectively. The *thumb up* or *victory* gestures mean activation/deactivation of the gesture teleoperation command mode.



(a) thumb up (b) victory

(c) open palm (d) closed fist

Figure 5. Google MediaPipe Software Framework: recognizable static gestures

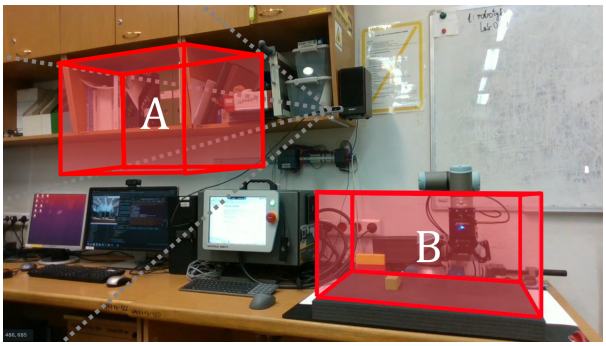


Figure 6. Camera workspace (A) and gripper workspace (B)

The system operates within predefined workspaces in 3D space, as illustrated in Figure 6. Workspace A corresponds to the region within the camera field of view, where user gestures are captured and processed as commands for the robot. Workspace B defines the area within which the gripper operates. Its position remains confined to this designated region. Both workspaces are rectangular in shape, and their exact dimensions can be adjusted to suit specific requirements.

/robot_controller is a high-level software module that processes the data subscribed from the `/coordinates` and `/gestures` topic into commands for the manipulator.

The hand coordinates are converted into coordinates of the robot's base coordinate frame. These coordinates are then set as the target position for the gripper. The manipulator's point-to-point movement is handled by the MoveIt motion planning framework. It provides access to algorithms from the Open Motion Planning Library (OMPL). Robot movement

trajectories are calculated with the Rapidly-exploring Random Trees (RRT) Connect algorithm in this system.

The hand gesture dictates whether the gripper opens or closes, and if the control system is activated or deactivated. The resulting data representing the desired motion is stored in the fields of a variable, which is then published to the `/command` topic.

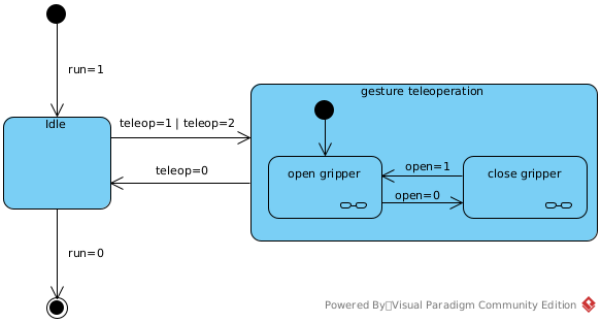


Figure 7. UR3 manipulator gesture control representation using finite state machine

The behaviour of the system is represented by the FSM diagram shown in Figure 7. At startup, the system enters the idle state. The *thumb up* gesture (see Figure 5(a)) toggles the value of the `teleop` variable from 0 to 1, while the *victory* gesture (see Figure 5(b)) toggles it from 0 to 2. A value of 1 corresponds to reduced teleoperation, while a value of 2 corresponds to full teleoperation. When either value is set, the gesture teleoperation state becomes active.

In this state, the manipulator's gripper is teleoperated based on the user's hand gestures according to algorithms (1)–(8), (9), or (10). The system recognises both an *open palm* gesture (see Figure 5(b)) and a *closed fist* gesture (see Figure 5(c)).

Initially, the open gripper state is active. A *closed fist* gesture toggles the value of the open variable between 1 and 0, activating the open gripper and close gripper states, respectively. These states correspond to the open or closed gripper of the manipulator.

The gesture teleoperation state is deactivated when the value of the `teleop` variable is reset to 0 using either the *thumb up* or *victory* gesture. At this point, the system returns to the idle state.

4. System Evaluation

4.1. Research Question

The study aims to answer the following research questions in relation to the system presented in Section 2:

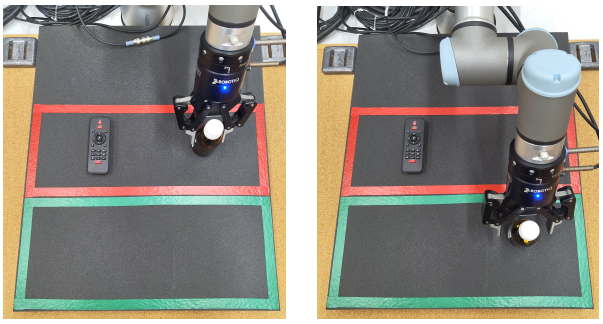
RQ1 Is the system usable in the user's view?

RQ2 Is the system safe from the user's perspective?

4.2. Method

Participants The study involved 37 participants aged 22-25 years old (Med $= 23$, $Q_1 = 22$, $Q_3 = 23$). There were 4 females and 33 males in the group. All of them were students of control engineering and robotics at Wrocław University of Science and Technology.

Informed consent was obtained from all subjects involved in the study after they had been acquainted with the research objectives and procedures used in the study.



(a) the beginning (b) the end

Figure 8. Snapshots of the experimental task being carried out

Scenario The task scenario is illustrated in Figure 8. The participant is asked to move everyday objects, a TV remote control and a syrup bottle, from a further away area to a closer location. The further and closer areas are marked with a red and green border, respectively. A similar task in the evaluation scenario can be found in [7] and [16]. All commands addressed to the robot must be expressed in gestures. The set of commands consists of: activation/deactivation of the system, opening and closing of the gripper, and movement of the gripper (see Fig. 7 in Section 3.3).

The reduced teleoperation was selected to be the working mode during the study.

Procedure The team first trained the participant how to operate the robot. As part of the training, the participant had to practise the task of carrying and bringing back an item, in the form of a plastic block. Then the participant performed the task from the scenario. After the experiment, the participant had to fill out the questionnaires. The experiment was conducted in a fully simulated environment in the Robotic Laboratory at Wrocław University of Science and Technology.

The study was conducted according to the guidelines of the Declaration of Helsinki and was approved by the Research Ethics Committee of Wrocław University of Science and Technology, Opinion No. O-24-63.

Measures The users' impressions have been collected by means of questionnaires. To answer our research questions concerning usability, we used the System Usability Scale (SUS), a well established questionnaire for assessing the usability of the system containing 10 items on a five point Likert scale ranging from totally disagree to totally agree [27]. To assess perceived safety, we used the Godspeed (GQS) questionnaire for perceived safety (3 items using semantic

differential scales) [28]. We also used the Negative Attitude towards Robots scale (NARS-PL) [29], Polish version, in order to gather information about participants' attitude towards technology (containing also 13 items on a five point likert), which could impact the results.

4.3. Results

Internal Reliability of the Questionnaires

- For the 10 SUS items the Cronbach's Alpha was 0.9575 (excellent reliability).
- For the 13 NARS items the Cronbach's Alpha was 0.9561 (excellent reliability).
- For the 3 GS items the Cronbach's Alpha was 0.8917 (very good reliability).

The System Usability Scale It follows from Chi-square goodness-of-fit test that the cummulative SUS score comes from a normal distribution (a 100-point scale, $M = 69.5, SD = 7.0, \alpha = 0.05, p = 0.37$). Furthemore, the one-sample t-test allows to conclude that the mean of SUS score is 70, which can be considered as good according to [30] ($\alpha = 0.05, p = 0.19$).

Godspeed Questionnaire Series: Perceived Safety (PL) It follows from Chi-square goodness-of-fit test that the cummulative GQS score comes from a normal distribution (a 18-point scale, $M = 13.5, SD = 2.3, \alpha = 0.05$, $p = 0.05$). Furthemore, the one-sample t-test allows to conclude that the mean of GQS safety score is 13 ($\alpha = 0.05, p = 0.19$).

Negative Attitude Towards Robots Scale (PL) It follows from Chi-square goodness-of-fit test that the cummulative NARS-PL score comes from a normal distribution (a 48-point scale $M = 16.7, SD = 6.4, \alpha = 0.05$, $p =< 0.001$). Furthemore, the one-sample t-test allows to conclude that the mean of NARS-PL score is 17 ($\alpha = 0.05, p = 0.76$).

4.4. Discussion

With regard to the research questions RQ1 and RQ2, it can be concluded that participants with low negative attitudes toward robots assess the tested system as sufficiently useful and sufficiently safe.

When evaluating a system consisting of a human and a robotic assistive arm, user feedback appears to be most relevant. If the results of the users' study are satisfactory, then it can be tentatively assumed that system performance indicators are also satisfactory. The implication in the oposit direction is not obvious.

The evaluation in most previous work on gesture communication with the assistive robotic arm and with other robots is based on system performance indicators [7, 11, 13, 15, 16, 18–20]. The use of UX and HRI studies can be noted in less numerous and more recent works [11, 19–21]. This makes it challenging to do a reliable comparative analysis with results presented in other works.

The satisfactory result of the SUS test is confirmation of the soundness of the proposed gesture-based control of the manipulator by the users. A satisfactory score of perceived safety means that there was no behavior of the robotic system during the execution of the task by the participant that would interfere with his/her sense of security.

The results obtained, although positive, leave some room for improvement. It would seem that changing the appearance of the manipulator to be less industrial and the gripper to be more hand-like would increase the level of perceived safety. BATEO [6], iARM [5], MICO [7] can serve as examples.

Gesture-based manipulator control systems discussed in [8] and [9] require appropriate sensors to be mounted on the hand. The solution proposed in this paper, which does not require such sensors, is definitely more user-friendly. A similar solution with an RGB-D (Kinect) sensor for gesture identification and arm movement is discussed in [10], but the sensory system proposed there was not integrated into the physical robot and no evaluation was performed with the users (which is the case here).

The works [4, 15, 16] focus on different forms of communication (using tongue, brainwaves, joystick) therefore the gesture-based communication proposed here can be seen as a complement to these. All the previously mentioned forms of communication can form the basis of a simple manual control mode for the robot, but they can also be part of a more elaborate semi-autonomous control system [3].

In the work [18] there is no hand telemanipulation as here, but voice commands are responsible for the movement of the manipulator. The hand is used to indicate target locations for the manipulator's gripper. This solution may be difficult to use for users with speech disabilities. On the other hand, both solutions can coexist and complement each other.

The gesture control system presented in this article uses MediaPipe Hands similary to [11, 12] and [13] and Intel RealSense RGB-D camera similary to [12] and [13]. However, in this paper, the target user is a person with reduced mobility, while in the others, these are doctors or indeterminate users.

In this work the manipulator is meant to be an extension of the user's hand, and for this reason, telemanipulation using gestures is of primary importance from the viewpoint of potential applications. This issue is explored here in more depth compared to [11, 12] and [13]. What's more, the evaluation here is based on live experiments, based on a scenario close to the activities of daily living with a larger group of study participants. Therefore the result obtained here is more reliable compared to the conclusions that can be drawn from [11–13], where the number of study participants was significantly smaller and the studies themselves were conducted in a virtual environment [11, 12].

In our context, MediaPipe Hands, combined with depth data from the RGB-D camera, enables robust detection of complex hand gestures without requiring users to wear any sensors or markers. This significantly improves user comfort and reduces setup

time, while maintaining high gesture recognition accuracy.

5. Conclusion

It has been proposed and studied a robotic system that can assist people with reduced mobility in the activities of picking up and putting down objects out of reach. The manipulator in this system acts as an elongation of a person's arm. The movement of the manipulator's gripper is done by telemanipulation by the user's hand, whose movement is captured by the camera. A few commands given to the robot are expressed using gestures. The sensory part of the system is based on the RGB-D camera (Intel RealSense D435) and Google MediaPipe Software Framework. During the evaluation, the system worked reliably and participants assessed it as useful and safe.

The UR3 manipulator used in this study is not the primary focus of the system. The core software component of the proposed system can be easily adapted for use with other manipulators that are compatible with ROS/ROS2 and the MoveIt software environment.

The results obtained justify the continuation of work at a higher level of technological readiness. This work should be based on another collaborative robot (better suited to the requirements of an assistive robotic arm) a much wider and more diverse group of participants in UX and HRI studies, and a wider spectrum of system performance indicators to evaluate the gesture-based communication with an assistive robotic arm.

AUTHORS

Sebastian Koryl – Faculty of Electronics, Photonics and Microsystems, Wrocław University of Science and Technology27 Wybrzeże Stanisława Wyspiańskiego st., 50-370 Wrocław, Poland, e-mail: sebastian.koryl@gmail.com.

Katarzyna Zadarnowska* – Department of Cybernetics and Robotics, Wrocław University of Science and Technology, 27 Wybrzeże Stanisława Wyspiańskiego st., 50-370 Wrocław, Poland, e-mail: katarzyna.zadarnowska@pwr.edu.pl https://wefim.pwr.edu.pl/pracownicy/wizytowki-pracownikow/profil.html?name=katarzynazadarnowska-1186.

Krzysztof Arent – Department of Cybernetics and Robotics, Wrocław University of Science and Technology27 Wybrzeże Stanisława Wyspiańskiego st., 50-370 Wrocław, Poland, e-mail: krzysztof.arent@pwr.edu.pl https://wefim.pwr.edu.pl/pracownicy/wizytowki-pracownikow/profil.html?name=krzysztofarent-1219.

*Corresponding author

References

[1] H. F. M. Van der Loos and D. J. Reinkensmeyer. *Springer Handbook of Robotics*, chapter "Rehabilitation and Health Care Robotics", 1223–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[2] A. Nanavati, V. Ranganeni, and M. Cakmak, "Physically Assistive Robots: A Systematic Review of Mobile and Manipulator Robots that Physically Assist People with Disabilities", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, no. Volume 7, 2024, 2024, 123–147, https://doi.org/10.1146/annurev-control-062823-024352

[3] R. Soyama, S. Ishii, and A. Fukase, "8 Selectable Operating Interfaces of the Meal-Assistance Device "My Spoon". In: *Advances in Rehabilitation Robotics: Human-Friendly Technologies on Movement Assistance and Restoration for People with Disabilities*, Springer Berlin Heidelberg, 2004, 155–163, 10.1007/10946978_8

[4] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire, and L.-J. C. L'Ecuyer, "Kinova Modular Robot Arms for Service Robotics Applications", *International Journal of Robotics Applications and Technologies*, vol. 5, no. 2, 2017, 10.4018/IJRAT.2017070104

[5] "iARM". Assistive Innovations, 2026. https://www.assistive-innovations.com/robotarm

[6] "BATEO". ACCREA, 2026. https://accrea.com/products/bateo/

[7] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive Robotic Manipulation through Shared Autonomy and a Body-Machine Interface". In: *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015, 526–531, 10.1109/ICORR.2015.7281253

[8] A. Vijay Joseph, A. Mathur, J. Verma, and A. Singh, "Gesture Based Wireless Control for a Robotic Manipulator", *International Journal of Engineering & Technology*, vol. 7, no. 2.31, 2018, 231–234, 10.14419/ijet.v7i2.31.13449

[9] P. Sihombing, R. B. Muhammad, H. Herriyance, and E. Elviwani, "Robotic Arm Controlling Based on Fingers and Hand Gesture". In: *2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*, 2020, 40–45, 10.1109/MECnIT48290.2020.9166592

[10] X. Li, "Human–Robot Interaction Based on Gesture and Movement Recognition", *Signal Processing: Image Communication*, vol. 81, 2020, 115686, https://doi.org/10.1016/j.image.2019.115686

[11] W. Rhee, Y. G. Kim, J. H. Lee, J. W. Shim, B. S. Kim, D. Yoon, M. Cho, and S. Kim, "Unconstrained Lightweight Control Interface for Robot-Assisted Minimally Invasive Surgery Using MediaPipe Framework and Head-Mounted

Display", *Virtual Reality*, vol. 28, no. 114, 2024, https://doi.org/10.1007/s10055-024-00986-1

[12] A. R. Khan, A. Khan, and A. Faisal, "Gesture-Controlled Robotic Arm for Real-Time Surgical Tool Assistance". In: *2025 Fourth International Conference on Power, Control and Computing Technologies (ICPC2T)*, 2025, 886–891, 10.1109/ICPC2T63847.2025.10958621

[13] J. Xie, Z. Xu, J. Zeng, Y. Gao, and K. Hashimoto, "Human–Robot Interaction Using Dynamic Hand Gesture for Teleoperation of Quadruped Robots with a Robotic Arm", *Electronics*, vol. 14, no. 5, 2025, 10.3390/electronics14050860

[14] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann. "MediaPipe: A Framework for Building Perception Pipelines", 2019.

[15] F.-J. Chu, R. Xu, Z. Zhang, P. A. Vela, and M. Ghovanloo, "Hands-Free Assistive Manipulator Using Augmented Reality and Tongue Drive System". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, 5463–5468, 10.1109/IROS.2018.8594508

[16] P. D. Lillo, F. Arrichiello, D. D. Vito, and G. Antonelli, "BCI-Controlled Assistive Manipulator: Developed Architecture and Experimental results", *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, 2021, 91–104, 10.1109/TCDS.2020.2979375

[17] K. P. Hawkins, P. M. Grice, T. L. Chen, C.-H. King, and C. C. Kemp, "Assistive Mobile Manipulation for Self-Care Tasks Around the Head". In: *2014 IEEE Symposium on Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT)*, 2014, 16–25, 10.1109/CIRAT.2014.7009736

[18] B. Kulecki, "Multimodal Robot Programming Interface Based On RGB-D Perception and Neural Scene Understanding Modules", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 17, no. 3, 2024, 29–37, 10.14313/JAMRIS/3-2023/20

[19] C.-S. Chung, K. Williams, S. M. Eckstein, I. Eckstein, R. Cooper, and R. A. Cooper, "Manipulation Performance of an Assistive Robotic Arm Control Interface for Individuals with Cerebral Palsy - A Case Study". In: *Proceedings of the RESNA Annual Conference*, 2022.

[20] K. Arent, J. Jakubiak, M. Drwięga, M. Cholewiński, G. Stollnberger, M. Giuliani, M. Tscheligi, D. Szczęśniak-Stańczyk, M. Janowski, W. Brzozowski, and A. Wysokiński, "Control of Mobile Robot for Remote Medical Examination: Design Concepts and Users' Feedback from Experimental Studies". In: *2016 9th International Conference on Human System Interactions (HSI)*, 2016, 76–82, 10.1109/HSI.2016.7529612

[21] A. Lebrasseur, J. Lettre, F. Routhier, P. S. Archambault, and A. Campeau-Lecours, "Assistive Robotic Arm: Evaluation of the Performance of Intelligent Algorithms", *Assistive Technology*, vol. 33, no. 2, 2021, 95–104, 10.1080/10400435.2019.1601649, PMID: 31070524.

[22] S. Koryl. "Technology for Human-Manipulator Communication Using Gestures". Master's thesis, Wrocław University of Science and Technology, 2024.

[23] Google. *MediaPipe Solutions Guide*, 2024. https://ai.google.dev/edge/mediapipe/solutions/guide

[24] G. Amprimo, G. Masi, G. Pettiti, G. Olmo, L. Priano, and C. Ferraris, "Hand Tracking for Clinical Applications: Validation of the Google MediaPipe Hand (GMH) and the Depth-Enhanced GMH-D Frameworks", *Biomedical Signal Processing and Control*, vol. 96, 2024, 106508, https://doi.org/10.1016/j.bspc.2024.106508

[25] Y. Alemu. "Robotic System Technology to Support the Manipulability of a Person with Motor Disabilities". Master's thesis, Wrocław University of Science and Technology, 2024.

[26] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, "Telerobotics". In: *Springer Handbook of Robotics*, 2016.

[27] R. A. Grier, A. Bangor, P. Kortum, and S. C. Peres, "The System Usability Scale: Beyond Standard Usability Testing", *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, 2013, 187–191, 10.1177/1541931213571042

[28] R. Szczepanowski, T. Niemiec, E. Cichoń, K. Arent, M. Florkowski, and J. Sobecki, "Factor Analysis of the Polish Version of Godspeed Questionnaire (Gqs)", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 16, no. 2, 2023, 30–35, 10.14313/JAMRIS/2-2022/13

[29] G. Pochwatko, J.-C. Giger, M. Różańska-Walczuk, J. Świdrak, K. Kukiełka, J. Możaryn, and N. Piçarra, "Polish Version of the Negative Attitude Toward Robots Scale (NARS-PL)", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 9, no. 3, 2015, 65–72, 10.14313/JAMRIS_2-2015/25

[30] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale", *Journal of Usability Studies*, vol. 4, no. 3, 2009, 114–123.

ARIS: Autonomous Real-Time Interactive Social Robot

Submitted: 21st June 2025; accepted: 29th October 2025

Cesar Minaya-Andino, David Minango, Marcelo Zambrano

DOI: 10.14313/jamris-2026-016

Abstract:
The purpose of this study is to develop and evaluate ARIS (A Real-time Interactive Social Robot), based on a Turtle 4 platform aimed at improving human–robot interactions (HRI) on university campuses. ARIS combines a 3D printed social robot structure with a software architecture based on 2D LiDAR, odometry, and IMU sensors for navigation and mapping, in addition to a voice assistant structured in 3 stages: audio recording, transcriptional processing, reaction, and reproductive synthesis. Experimental results show that the success rate of ARIS is greater than 86.5%, maintaining high accuracy in navigation and obstacle avoidance. The system also offers performance consisting of voice interaction with a total reaction latency of 1500 to 2200 ms. Access to low-cost robotic platforms allows students and researchers access to practical training, customization, and validation in the development of new technologies in social robotics.

Keywords: *ARIS, social Robot, interactions, turtlebot*

1. Introduction

In recent decades, human-robot interaction (HRI) has been one of the main research topics in the field of social robotics. Studying how humans and robots interact in social settings provides key insights for creating robots that interact with people in a natural and empathetic way. In collaborative environments such as hospitals, schools, hotels, offices, and others, these robots assist with daily tasks by engaging in intelligent and responsive conversations. A social robot needs to capture emotions, body language, and language cues, thus becoming a reliable collaborator for both practical work and social relationships to build trust and maintain interest.

Adaptability is also a crucial aspect, as it allows them to integrate into various environments—supporting students, managing hotel guests, collaborating with employees in their daily tasks, or simply serving as a technological attraction.

Several studies have found a variety of environments to incorporate robots as social agents. It is the example of BRIGHTNESS, a bartender robot that gives you a personalized service and is dynamic, adapting to the interactions of customers [1]. Another study has explored the role of the teacher in math activities versus a social robot, tutee, where it was revealed that, the main limitations are the verbal interactions between teacher–student and robot–student, in addition to this feedback, a very important factor in education is that it is best provided by the teacher [2]. In the same way other authors have determined, that the emotional participation and the compression reading cognitive storytelling is higher in children of 5-7 years with the help of a social robot in a comparison to a human [3]. Even to understand the adaptation of the robots socially with children at an early age of 8-9 years, several authors have studied the anthropomorphism of children and their latent growth [4].

A social robot has the characteristic of being stationary or mobile. To be steady-you can reduce a feeling of connection to personal interaction and decrease the experience of social dynamics. By contrast, to be mobile you must have a level of autonomy to adapt in the environment. To integrate into society, these machines must function in complex environments even without direct human intervention. Recent studies have focused on cognition and reasoning, so that CASPER via the development of an architecture for cognitive artificial enables you to observe the actions of humans, understand, and collaborate on tasks in progress [5]. In another study, the authors have developed an architecture hybrid that uses a metaphor of the brain, and a formalist approach [6–8].

Apart from perception, localization and mapping are two key evaluative capabilities that allow a mobile robot to understand and act autonomously in its environment. Since the early 1990s, techniques have been developed to enable a robot to localize itself within an environment. SLAM (Simultaneous, Localization, and Mapping) became a very popular technique starting in the 2000s [9], where, with the help of a LiDAR optical distance sensor, the cross-section of the environment's geometric structure can be measured in order to create a map. With the same goal of improving the accuracy, robustness, computational efficiency, and speed of the SLAM technique, numerous researchers have developed and implemented various methods, such as LOAM, which provides LiDAR odometry computation [10]; ORB-SLAM2, which builds a more accurate map of the environment using a monocular, stereo, or RGB-D camera [11]; RTAB-Map, based on real-time appearance-based mapping with loop closure detection [12,13]; and S-PTAM, which uses stereo cameras and combines the accuracy of the parallel tracking and mapping approach from ORB-SLAM [14].

A key feature of mobile assistant robots is voice recognition, which enables them to understand and respond to human voice commands, making

human-robot communication more intuitive. This technology involves capturing audio through microphones, followed by noise reduction and converting speech to text using automatic speech recognition (ASR) systems [15, 16]. In previous research, various models have been proposed and evaluated to improve the robustness of ASR systems in noisy environments [17, 18]. Other authors have proposed combinations of bidirectional gated recurrent units (Bi-GRU) with convolutional neural networks (CNN) to operate in offline mode [19].

In summary, many interesting results have been reported that highlight the potential of social robotics and the advancements developed over the years. Social robotics brings together various technologies such as artificial intelligence, perception, human–robot interaction, automatic speech recognition, self-localization, computer vision, among others.

The objective of this work is to develop and implement, in a first stage, a social robot named ARIS (Autonomous Real-Time Interactive Social Robot) based on the TurtleBot 4 platform, carrying out human-robot interaction applications and integrating perception, localization, and mapping within the ROS environment, in order to perform tasks on a university campus.

While advanced social robots like Pepper and NAO have demonstrated effectiveness in HRI applications, their high cost ($25,000-$50,000) and closed architectures limit accessibility for educational institutions. Existing low-cost platforms like TurtleBot typically lack integrated social features (expressive structure, natural voice interaction). ARIS bridges this gap by demonstrating that effective social HRI can be achieved using affordable, open-source components while explicitly documenting the trade-offs and limitations.

The manuscript is divided into two parts: The first part includes the title, abstract, and keywords. The second part is the paper's main body including the conclusion section.

2. Related work

2.1. Evaluation of Mobile Robots Designed for Lab-Scale Applications

In the field of mobile social robots, robotics engineers usually choose between two paths: simulations or real hardware. Functional platforms allow testing with sensors in real fields, providing solid and reliable results, although they are more expensive. On the other hand, simulations only allow for quick, inexpensive, and risk-free experimentation with algorithms. The drawback arises when you want to transfer the simulation to hardware, as it is not possible to replicate all the details of the physical world. For this reason, many researchers and engineers have developed affordable educational robotics kits to streamline the creation and implementation of innovative functions.

Examples include MobileCharger, an innovative robot equipped with sensors, actuators, and a combined perception system [20]; Robotis, a humanoid robot that integrates sophisticated sensors and dynamic movement capabilities, ideal for academic research, locomotion algorithm development, human–robot interaction, and experimentation in autonomous robotics [21]; NAO, an autonomous, programmable bipedal robot known for its friendly, expressive design and equipped with 25 degrees of freedom, allowing it to perform natural and complex movements [22]; and Pepper, another humanoid robot built on a holonomic base, equipped with three omnidirectional wheels, developed by Aldebaran to welcome customers in retail environments [23, 24].

Numerous studies have employed the aforementioned platforms, as well as others, with the aim of conducting academic research. In the case of Pepper, methods have been proposed to improve its limited 3D perception capabilities [25]; in Industry 4.0, digital twins have been developed for interaction in smart homes [26]. With NAO, several studies have been presented on gesture recognition using videos recorded solely by the robot's built-in camera while performing clinical procedures [27]. For arm movements, algorithms based on Bayesian Networks (BN) have been proposed and developed to select the appropriate motion model and enhance precision to levels similar to that of humans [28].

On other platforms such as Robear, Ubtech Wassi, and Freivera—robots designed for healthcare, specifically for assisting and caring for the elderly—several studies have proposed improvements in localization and mapping using methods to obtain semantic information and dynamic selection strategies [29].

In terms of personal assistants, DARWIN-OP2 is also noteworthy; it uses reinforcement learning algorithms to detect poor posture in a group of students [30].

The use of robotic platforms can accelerate research and lower the entry barriers for new research groups. Several platforms have been mentioned previously, but there are numerous affordable options in different sizes. However, the more advanced and recently developed platforms are often out of reach for many research groups due to their high cost or closed-source nature. For this reason, there is a need to develop open-source platforms integrated with sufficient sensors and actuators to support research groups in various fields.

Among open-source platforms, we have IGUS, whose robot structure is entirely 3D printed highlighting a lightweight and visually appealing design [31]; NimbRo-OP2X which integrates computer vision, IMU sensors and an Intel processor with GPU, making it ideal for locomotion, perception, and automatic control areas [32]; iCub, standing 104 cm tall, specifically designed to support research in embedded artificial intelligence, with hands engineered to support sophisticated manipulation skills [33–35]; and finally, InMoov, a humanoid robot approximately 180 cm tall that uses accessible components such as

Arduino microcontrollers and servomotors. Its modular design allows it to be built in stages and expanded according to the resources and needs of research groups [36, 37].

2.2. Approach to Real-Time Mapping and Position Tracking

Simultaneous localization and mapping (SLAM) is an indispensable technology which allows wheeled robots to navigate and determine their position accurately. The SLAM algorithm procedure is to collect information from the mains sensors such as cameras, and lidar, along with data from the inertial measurement unit (IMU). With this information, a map is constructed and the robot locates itself within it. A popular choice for mobile robots, Hector-SLAM relies on laser scans and fast estimation methods to deliver high accuracy in real time. Unlike other SLAM approaches, it operates perfectly even without wheel odometry [38].

A SLAM system based on multi-sensor fusion is a robust solution for achieving more accurate and stable localization and mapping in dynamic environments. This system compensates for the individual limitations of each sensor to achieve greater robustness with better estimates. By integrating all the data using fusion algorithms such as extended Kalman filters and optimization graphs, the system operates without disturbances [39].

Visual SLAM is another technique used to construct a map of the environment while a mobile robotic system moves through the environment. Visual information comes from one or more cameras. Unlike traditional SLAM methods that work with sensors such as LIDAR or inertial data, Visual SLAM uses exclusively image data (stereo, RGB-D, or monocular). From this visual information, the system's trajectory is estimated. In environments where GPS sensor data are lost or lacking [40], this technique improves global positioning [41].

Lidar sensors have become a primary tool for data acquisition for map construction, aiding autonomous navigation. Data acquisition from a lidar sensor produces a point cloud containing spatial information (x, y, z), which is invaluable in applications such as object detection, path tracking, and scene reconstruction [42]. In the context of SLAM systems, the point cloud represents robust and accurate data that can be integrated with data from other sensors to improve autolocalization and map construction, even in complex dynamic environments [43].

Low-cost small mobile platforms have also been developed to test algorithms in robotics such as SLAM, autonomous navigation, computer vision, and more. Create is a good example, where many research groups have implemented social functionalities on this mobile base, such as a real-time human–robot interaction system using hand gestures [44], as an office assistant with voice recognition to interact with users [45], and also by proposing a VLM-Social-Nav system with real-time decision-making [46].

Figure 1. Structure of ARIS

3. General architecture of the robot system

The general architecture of ARIS consists of a user-friendly physical robotic structure. The programmable TurtleBot 4 platform was used as the base, with a 3D-printed frame designed to give the appearance of a social robot. A touchscreen is included for human-robot interaction, along with a conference speaker and microphone for automatic speech recognition. The structure is shown in Figure 1.

3.1. Software development environment

TurtleBot 4 is a mobile robotic platform designed to provide an accessible research system for prototyping and robotics training. The mobile base, IROBOT Create 3, is integrated with motors that provide mobility. It is also equipped with proximity sensors, an IMU sensor, a stereo camera, and a 2D LiDAR sensor that provides navigation data. The model uses a Raspberry Pi 4 Model B (4GB RAM) as the main computing unit. The system runs Ubuntu 22.04 LTS with ROS 2 Humble preinstalled.

TurtleBot 4 operates with two main computers: A Raspberry Pi 4B and an integrated Create 3 processor. To visualize sensor data, configure, and control the system, among other functions, we connect another Raspberry Pi 4 Model B (4GB RAM) running the same Ubuntu and ROS 2 as TurtleBot 4. In ROS 2, DDS is integrated as the default communication layer, offering advantages in scalability and performance. Real-time data exchange, sensor reading or control commands are managed using the Simple Discovery configuration, as shown in Figure 2.

3.2. ARIS structure

The structure was manufactured using 3D printing with Fused Deposition Modeling (FDM) and polylactic acid (PLA) filament. With the help of 3D design and Creality Slicer 4.8 software, the printing parameters were adjusted with a 25% internal infill, taking into account the strength of the structure. Figure 3 shows the robot's structure and main components.

The 3D-printed structure was produced using the Creality 3DPrintMill, which offers an infinite Z-axis

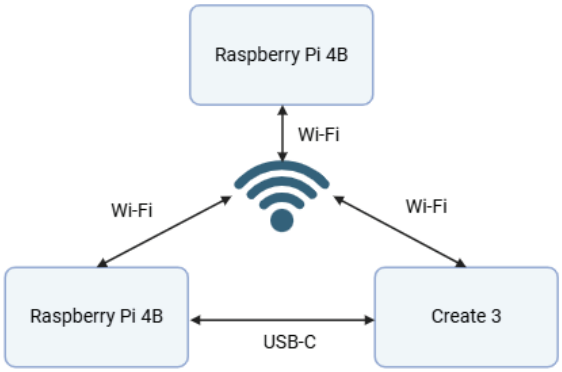


Figure 2. TurtleBot 4 Simple Discovery configuration for ROS 2 Humble

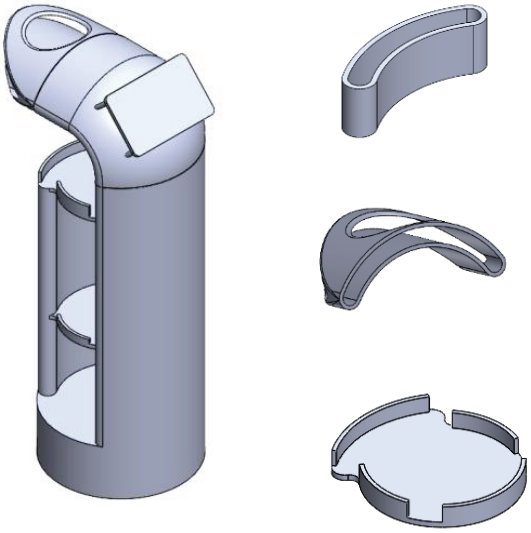


Figure 3. Design structure of the robot in 3D

and a 300 x 300 mm XY plane area. Due to the geometry and varying sizes of the robot components, it was designed and printed as a set of connectable parts rather than a single body. Each piece was oriented to maximize stability and adhesion to the surface of the adjoining piece. The complete assembly reached a weight of 3215 g.

3.3. Architecture of ROS Nodes

The ARIS architecture in ROS 2 is designed for natural and contextual human interaction. The system initiates interaction through the audio input unit, which captures the user's voice commands. This audio stream is processed by a transcription module that converts speech into text and interacts with a large language model (Gemini-1.5-pro) to enhance comprehension and generate harmonized interpretations. The core of the system lies in a central brain node that integrates decoded inputs and generative capabilities. ARIS can also respond using a speech synthesis tool that provides real-time feedback to the user. Additionally, the navigation module provides spatial awareness and mobility, allowing ARIS to physically

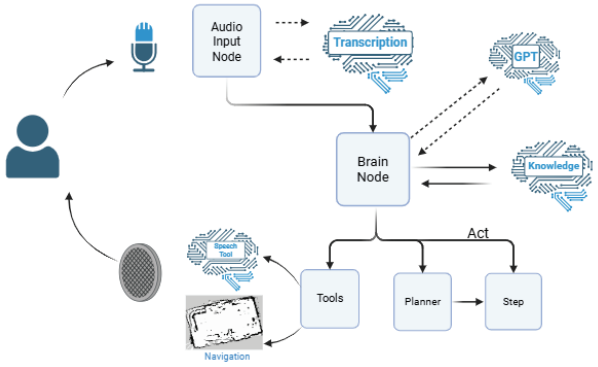


Figure 4. How ROS Nodes Work

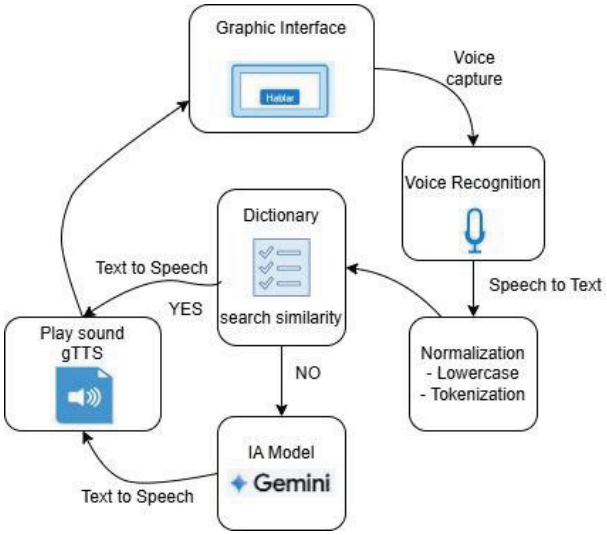


Figure 5. Block diagram assistant voice

interact with its environment. This architecture supports dynamic two-way interaction and enables ARIS to act as a helpful and conversational agent in real-world environments. The ARIS architecture in ROS is illustrated in Figure 4.

3.4. Voice assistant

The voice assistant implemented in ARIS, is organized into five modules as displayed in Figure 5, which are the capture of audio, transcription, processing of the input, the synthesis of the response and the playback of the audio, which is integrated to the graphical interface by clicking a button on the touch screen for human–robot interaction.

The voice assistant implemented in ARIS is organized into five modules that are capturing the audio, transcription, processing of the input, the synthesis of the response and the playback of the audio, which is integrated to the graphical interface by clicking a button on the touch screen for human–robot interaction.

Initialized ARIS, plays an audio file with a welcome message configured manually every 20 seconds, the audio capture was performed using the activation button in the screen called "Talk". When you press the button, it suspends playback of the first message and proceeds to the execution of the method of acquisition of audio. Using the library speech_recognition

automatically adjusts the detection threshold of environmental noise, to improve the quality of the signal in noisy environments. The capture of the audio is done by using the built-in microphone and is stored in an audio file temporarily.

The audio stored is sent to the service of Google Speech Recognition which makes the transcription from voice to text. The service returns a string in the format of a text, then the text is normalized by placing the text to all lowercase letters and using the technique of tokenization to break down the text into fragments of words or tokens. With tokens obtained is carried out a lexical analysis comparative look for the similarity of the tokens with the dictionary of words manually configured in a repository grouped by categories of topics such as (greeting, queries, institutional, control commands and diverse questions).

The detection of similarity between any of the tokens of the input and some of the words stored in the repository of dictionaries, generates the automatic playback of the answers that were previously configured by ensuring an efficient response with respect to queries that correspond to information in the university space. In the absence of match suggestions with the dictionary, the text of the question is sent in the format of a prompt to be answered by the generative model LLM employee who is the Gemini-1.5-pro.

The generative model uses a API_KEY generated from the services of Google AI Studio, additional function of the model is configured with instructions, in which it is established that the duties of a virtual assistant focused on giving answers related to the context of the education system in Ecuador, and in the case of the question requested was not appropriate in this context, is answered in a general way with the base model of a clear and precise way.

The response is generated to be pre-configured or coming from the generative model, converts text to audio format using the library Google Text-to-Speech gTTS, the which creates an MP3 file that is stored in a temporary directory, and is reproduced through the built-in speaker in ARIS using the player mpg123.

At the end of the audio playback, reset the interaction with the activation of the button, and the status message at the interface, indicating that it is ready to listen. This approach allows to ARIS to provide human–robot interaction using the format of a virtual assistant focused in giving answers that were previously configured on information relevant to the university space, combined with the flexibility of a language model to the generative for open consultation.

3.5. Functionality of ARIS

The algorithm described below controls the mobile robot's movement environment with predetermined locations, such as "Point A," "Point A," "Point A," etc. The robot waits for the user to press the touch button that allows it to select its destination. Once selected, the robot begins moving toward this destination. During the route, it visualizes potential obstacles and avoids them if necessary. If there are no obstacles, it will constantly update its movement.

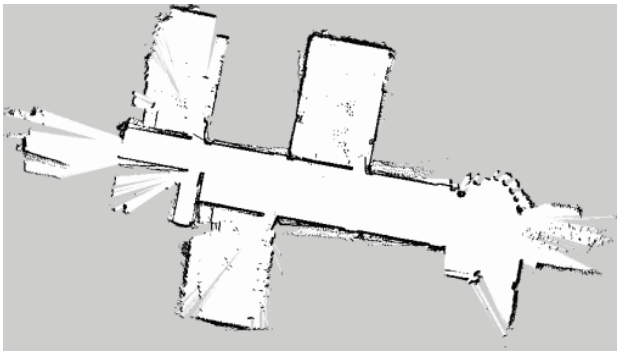


Figure 6. Local map

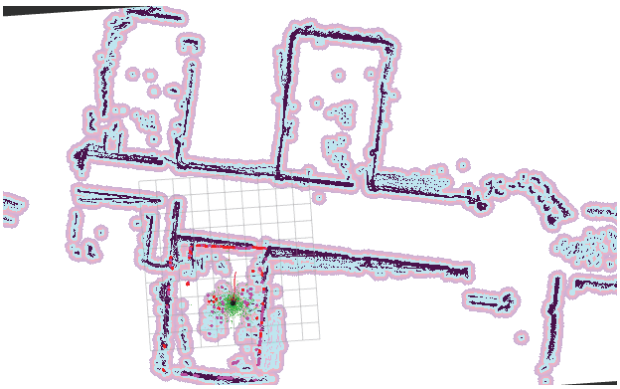


Figure 7. Aris performance with obstacles

When the robot reaches its destination, it interrupts its movement and displays a message indicating that it has arrived. This process is repeated indefinitely so that the robot can obtain new route requests at any time. At the same time, it checks if the battery percentage is greater than 20%. If not, it interrupts its route and heads to the charging point until it reaches an optimal percentage for its operation.

4. EXPERIMENTAL APPROACH

The platform ARIS was tested several times during its development. Firstly, it was tested at laboratory scale and, finally, in a university environment, including offices and corridors. You could measure the accuracy in the achievement of the objectives of the system ARIS.

The platform ARIS was tested with multiple users located in different points within the environment university, as illustrated in Figure 6.

Table 1 presents the default Cartesian coordinates for a set of five offices and the designated ARIS loading point. The X and Y values indicate the respective position of each location within the workspace.

ARIS autonomously determines efficient routes between designated points while performing tasks. The data in Table 1 serve as base positions for navigation.

ARIS is located at the entrance, which is part of the university campus. Users can approach and interact with it or request information about university authorities, academic programs, or simply general

```
start
waypoints := [“Point A”, ” Point B ”, ” Point C ”, ” Point D ”, ” Point H ”]
current_position := ” Point H ”
destination := ””
is_moving := false
battery_level := 100
previous_destination := ””
while true do
   if battery_level < 25 then
      previous_destination := destination
      destination := ”charging station (h)”
      is_moving := true
end if
if battery_level < 70 and current_position = ”charging station (h)” then
      charge_battery()
      continue
end if
   if touch_button_pressed() then
      destination := select_destination(waypoints)
      is_moving := true
end if
   if is_moving then
   if detect_obstacle() then
      avoid_obstacle()
   else
      move_toward(destination)
      update_position()
      update_battery()
      if current_position = destination then
      is_moving := false
      if destination = ”charging station (h)” and battery_level < 70 then
       // stay and keep charging
       continue
      end if
      if destination = ”charging station (h)” and battery_level ≥ 70 then
       destination := previous_destination
       is_moving := true
   else
      display(”arrived at ” + destination)
    end if
  end if
 end if
end if
end while
end
```

Table 1. Default coordinates of the offices and location of ARIS

No.	Offices	Coordinate (X)	Coordinate (Y)
1	Office 1	-0.25	-0.22
2	Office 2	-3.59	-0.41
3	Office 3	6.35	-7.09
4	Office 4	8.72	4.30
5	Office 5	0.59	-14.39
6	Default place of the ARIS to charge	0.0	0.0

information. At the same time, ARIS can guide users to the office they wish to reach.

During the route, ARIS is capable of detecting obstacles in its frontal path. When it detects a distance below a predefined threshold, ARIS stops its movement and performs a turning maneuver to avoid the obstacle, then resumes its trajectory toward the destination as illustrated in Figure 7.

Figure 7 shows the map created by SLAM, in which the dark areas represent permanent barriers such as walls, while the blue sections reflect the live zones that have been explored by the robot’s sensors. The purple color refers to the costmap inflation areas, which alert about the proximity of obstacles. The red band

Table 2. ARIS Performance of ARIS

User	Attempts	Questions Answered Correctly	Reached the Correct Coordinates	Avoided Obstacles (Yes/No)	Travel Time (s)	Success Rate
U1	2	Yes	Yes	Yes	52	100%
U2	2	Yes	Yes	IF	60	100%
U3	3	No	No	No	78	0%
U4	4	Yes	Yes	Yes	49	100%
U5	2	Yes	Yes	Yes	55	100%
U6	2	Yes	Yes	Yes	63	100%
U7	3	Yes	Yes	Yes	50	100%
U8	4	No	Yes	No	70	33%
U9	3	Yes	Yes	Yes	62	100%
U10	4	No	Yes	No	43	33%

Table 3. Performance of ARIS with inflation radius 0.5 m

User	Attempts	Questions Answered Correctly	Reached the Correct Coordinates	Avoided Obstacles (Yes/No)	Travel Time (s)	Success Rate
U1	2	Yes	Yes	Yes	52	100%
U2	2	Yes	Yes	Yes	59	100%
U3	3	No	No	Yes	63	66%
U4	4	Yes	Yes	Yes	49	100%
U5	2	Yes	Yes	Yes	55	100%
U6	2	Yes	Yes	Yes	61	100%
U7	3	Yes	Yes	Yes	50	100%
U8	4	No	Yes	Yes	64	66%
U9	3	Yes	Yes	Yes	60	100%
U10	4	No	Yes	Yes	53	66%

indicates the programmed route the robot will follow to reach its goal.

Autonomous navigation serves as a functional foundation in unknown environments which also allows the system to demonstrate adaptability in dynamic social settings. If ARIS detects a battery level below a predefined threshold, it decides to head toward a predetermined charging station and wait until it reaches a sufficient energy level to resume its activities.

To evaluate the acceptance of ARIS's functionalities, the system is considered successful if it correctly responds to users' questions and reaches the predetermined coordinates while avoiding both dynamic and static obstacles along the way. The travel time during autonomous navigation between the different coordinates was also evaluated to identify any errors.

The success rate of each event was monitored upon completing the described process, conducting 10 random experiments, with each user making a maximum of four attempts.

Table 2 summarizes the performance of ten users in a task-based assessment involving interaction with ARIS. The dataset in Table 2 also provides information on user-robot interaction under task constraints and shows performance as areas requiring improvement in obstacle avoidance and precise positioning in cases of poor performance. To evaluate the overall performance of each user during the navigation task, the level of success was defined based on three key criteria:

- Good, correct coordinates: whether ARIS navigated correctly to the destination.
- Avoid obstacles: whether ARIS avoided any obstacles along the route.
- Answer questions correctly: whether ARIS answered at least one question correctly during the test.

Each criterion received a binary result: 1 if the condition was met and 0 otherwise. The success level was calculated as the percentage

$$Sucess\ rate = \left(\frac{Criteria\ met}{3}\right) \times 100 \qquad (1)$$

Due to the limited processing power of the Raspberry Pi 4, which leads to occasional mapping errors or localization drift, conservative navigation heuristics and obstacle buffer zones were employed in the navigation system to improve the functional performance of ARIS. In the cost map, the inflation radius parameters were set to 0.5 m, providing a buffer zone around all detected obstacles. This increases safety margins.

In conservative heuristics for route planning and control logic, the movement speed of ARIS was reduced. This allowed for early stopping in large environments. Modifying these parameters minimized the likelihood of collisions, mapping errors, and, above all, computational overload on the Raspberry Pi 4 platform. The data obtained are shown in Table 3.

Table 4. Average latency

Stage	Component	Average Latency (ms)	Descriptions
A	ASR (Google Speech-to-Text)	480–650	Varies with noise and internet conditions
B	NLP (Gemini-1.5-pro)	600–900	Depends on prompt complexity and server load
C	TTS (gTTS)	300–450	Measured for responses < 50 words
Total Interaction	**End-to-end delay**	**1500–2200**	End-to-end delay per user query-response c

Table 5. Comparison of interaction latency between standard processing pipeline and FAQ dictionary-based optimization for common queries

Stage	Component	Standard Pipeline Latency (ms)	FAQ Dictionary Pipeline Latency (ms)
A	ASR (Google Speech-to-Text)	480–650	120–180
B	NLP (Gemini-1.5-pro)	600–900	240–360
C	TTS (gTTS)	300–450	180–270
Total Interaction	**End-to-end delay**	**1500–2200**	**600-910**

5. Results and discussion

The experimental phase was conducted to determine the functionality of the entire ARIS system. Functionality was obtained in a dynamic environment of a university campus, with and without external obstacles. The experimental data are shown in Table 2 and Table 3. The accuracy of the ARIS system exceeds 86.5% on average. In particular, an accuracy of 86.5% was achieved in the completion of each action, and ARIS failed only four times out of a total of 93 test cases during the experimental phase. Furthermore, ARIS completed the target point localization process with an accuracy of 90%. The failures were due to various reasons, such as improper operation of the robot's self-localization system and external noise. However, the ARIS did not fail when overcoming static and dynamic external obstacles. Therefore, the ARIS showed no difficulty in autonomously guiding itself while avoiding static and dynamic external obstacles. Overall, the ARIS system achieved its goal, based on user commands, not only in the absence of external obstacles, but also when there are static external obstacles in the university campus environment.

Figure 7 shows the relationship between battery level and system performance. This shows that as the battery level drops, system performance also decreases, but not linearly. When the battery level exceeds 60%, performance remains high with a slight drop. However, the 40% drop becomes much steeper and reaches a critical level below 20% battery level. This trend indicates that the system significantly reduces its performance to save power when the battery is low.

The Table 4 shows the average latency of each component involved in voice-based human–robot interaction which is divided into three stages: speech recognition, natural language processing, and speech synthesis. In Stage A, the speech-to-text conversion system (Google ASR) operates with an average latency of 480 to 650 ms, depending on background noise. Stage B processes natural language using Gemini-1.5-Pro, with a latency of 600 to 900 ms depending on the complexity of the request and server demand. In Stage C, the text-to-speech (GTT) engine converts responses of

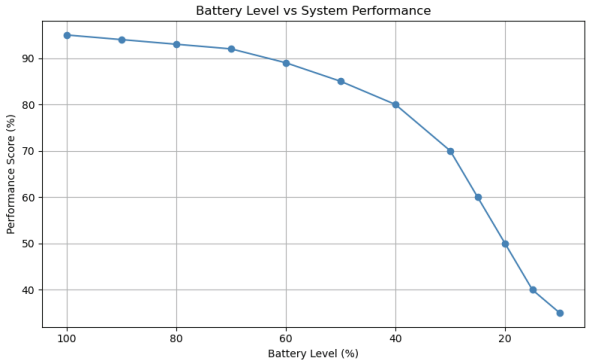


Figure 8. Battery Level vs System Performance

less than 50 words with a latency of 300 to 450 ms. Each user query has a latency in the entire interaction of between 1500 and 2200 ms which represents the total time elapsed from when the user speaks until they receive the response.

To mitigate the latency issues observed in Table 4, where total interaction delays ranged from 1500 to 2200 ms per query-response cycle, a lightweight caching system was implemented that stores frequently asked questions alongside their pre-computed responses. This optimization strategy creates a local repository of the most common user interactions, enabling rapid response retrieval without requiring cloud-based NLP processing or real-time text-to-speech synthesis.

As shown in Table 5, for cached queries, Stage B performs simplified NLP processing using pre-indexed semantic patterns, and Stage C retrieves pre-generated audio files from local storage, reducing synthesis time. ASR processes only the initial trigger phrase for pattern matching.

To measure positioning accuracy as the distance error in the path planning system, the Euclidean distance between the final estimated robot position and the true target position was used. The results in Table 6 show that increasing the inflation radius by 0.5 m significantly improved the accuracy and consistency of navigation. Almost all attempts reduced the calculated condition failure (e.g., from 0.08 m to

Table 6. Position error

Default Inflation Radius				Inflation Radius 0.5 m		
User	**Attempts**	**Reached the Correct Coordinates**	**Error Position (m)**	**Reached the Correct Coordinates**	**Reached the Correct Coordinates**	**Error Position (m)**
U1	2	Yes	0.08	Yes	Yes	0.06
U2	2	Yes	0.09	Yes	Yes	0.07
U3	3	No	0.50	No	No	0.40
U4	4	Yes	0.07	Yes	Yes	0.05
U5	2	Yes	0.08	Yes	Yes	0.06
U6	2	Yes	0.09	Yes	Yes	0.07
U7	3	Yes	0.08	Yes	Yes	0.06
U8	4	Yes	0.10	Yes	Yes	0.08
U9	3	Yes	0.09	Yes	Yes	0.07
U10	4	Yes	0.10	Yes	Yes	0.08

0.06 m U1 and 0.10 m to 0.08 m), while the average error dropped from 0.088 m (default) to 0.072 m with the inflation radius. Furthermore, the number of attempts to achieve high navigation accuracy (error $\leq$ 0.08 m) increased from five to eight. These improvements indicate that the additional buffer space around obstacles provided by the inflation radius improves route planning, reduces localization drift, and enhances overall system reliability, especially in an indoor environment with limited computing resources.

6. Conclusions

The errors occurred for various reasons, one of the main reasons was the background noise, unclear pronunciation, accent variety or variation in the speed of the speech, ASR incorrectly converts the audio into text, which provided misspelled words, substituted or omitted by others with similar sounds. These judgments directly affect the quality of the voice input for later misunderstanding. Another error is the understanding of the transcription of the audio; the model of the natural language processing failed due to limitations in their training.

The limit of processing of the Raspberry Pi 4 is another important factor in the development of the platform, ARIS. There are significant restrictions on intensive tasks simultaneously, such as localization and mapping simultaneously, the voice recognition and the algorithms of social interaction. While the Raspberry Pi 4 is equipped with a Cortex-A processor-72 of four cores with a clock frequency of 1.5 GHz, this does not comply with the requirements of the applications of robotics and high-performance. The algorithms SLAM Gmapping, require the processing of data from multiple sensors in real-time. These tasks require a high memory bandwidth, and a flow of the CPU/GPU of the Raspberry Pi 4 is constantly unable to provide, which causes latency and affects the accuracy of the mapping. The ASR also requires patterns of deep learning and fast processing. These models are located locally on the Raspberry Pi 4 and can result in a delayed or inaccurate recognition.

During the testing of the functioning of ARIS, it was observed that the Raspberry Pi 4 is subjected to prolonged computational loads and is prone to the thermal limitation. This not only reduces performance further, but also leads to unpredictable behavior during longer sessions of SLAM or iteration.

To tackle these issues and boost ARIS's reliability, we adopted cautious navigation strategies. A 0.5 m buffer zone was set around obstacles and the robot's speed was slowed in open spaces to allow quicker reactions. These changes reduce crashes, mapping mistakes and CPU strain.

The research combines proven robotic tools: SLAM, voice recognition, and language processing. The ARIS platform offers a clear methodology for building functional social robots on a tight budget, bridging the gap between high-end laboratory robotic platforms and the real needs of schools and small research laboratories. In addition, the performance of the Raspberry Pi was optimized by adjusting the requirements of the SLAM algorithms and using pre-recorded voice files to reduce latency, enabling smooth real-time operation even with modest hardware. These improvements allow ARIS to operate reliably with low-cost equipment that meets the needs of teaching and research.

AUTHORS

Cesar Minaya-Andino* – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: cesar.minaya@ister.edu.ec.

David Minango – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: david.minango@ister.edu.ec.

Marcelo Zambrano – Departamento de Investigación, Instituto Tecnológico Superior Rumiñahui, Sangolquí, 171103, Ecuador, e-mail: marcelo.zambrano@ister.edu.ec.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by Instituto Tecnológico Superior Rumiñahui through its 2023 research project funding program, ISTER-INV-PRO-D-049.

Data Availability Statement : The data and source code supporting the findings of this study are openly available in the GitHub repository at: https://github.com/cesarandresma/ARIS.

References

[1] A. Rossi et al., "BRILLO: Personalised HRI with a Bartender Robot," *Int. J. Soc. Robot.*, vol. 17, 2025; doi: 10.1007/s12369-025-01228-3.

[2] S. Ekström, L. Pareto, and Y.S. Ljungblad, "Teaching in a Collaborative Mathematic Learning Activity with and without a Social Robot," *Educ. Inf. Technol.*, vol. 30, no. 1, 2025, pp. 1301–1328; doi: 10.1007/s10639-024-12926-2.

[3] S.S. Yeung, M. Ma, and T.T. Law, "Let's listen and Tell a Story Together: Social Robot and Multidimensional Learning Engagement among Young Learners," *AI Brain Child*, vol. 1, no. 1, 2025, p. 5; doi: 10.1007/s44436-025-00006-2.

[4] R. Kühne et al., "How Does Children's Anthropomorphism of a Social Robot Develop Over Time? A Six-Wave Panel Study," *Int. J. Soc. Robot.*, vol. 16, no. 7, 2024, pp. 1665–1679; doi: 10.1007/s12369-024-01155-9.

[5] S. Vinanzi and A. Cangelosi, "CASPER: Cognitive Architecture for Social Perception and Engagement in Robots," *Int. J. Soc. Robot.*, 2024; doi: 10.1007/s12369-024-01116-2.

[6] M.Á. González-Santamarta et al., "A Hybrid Cognitive Architecture to Generate, Control, Plan, and Monitor Behaviors for Interactive Autonomous Robots," *Int. J. Soc. Robot.*, 2024; doi: 10.1007/s12369-024-01192-4.

[7] M.Á. González-Santmarta et al., "MERLIN2: MachinEd Ros 2 pLanINg," *Softw. Impacts*, vol. 15, 2023, p. 100477; doi: 10.1016/j.simpa.2023.100477.

[8] M.Á. González-Santamarta et al., "MERLIN a Cognitive Architecture for Service Robots," *Appl. Sci.*, vol. 10, no. 17, 2020, p. 5989; doi: 10.3390/app10175989.

[9] F. Dellaert and A.W. Stroupe, "Linear 2D Localization and Mapping for Single and Multiple Robot Scenarios," *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), 2002, pp. 688–694; doi: 10.1109/ROBOT.2002.1013438.

[10] C. Gonzalez and M. Adams, "Curvature Scale Space LiDAR Odometry and Mapping (LOAM)," *J. Intell. Robot. Syst.*, vol. 110, no. 2, 2024, p. 67; doi: 10.1007/s10846-024-02096-1.

[11] R. Mur-Artal and J.D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, 2017, pp. 1255–1262; doi: 10.1109/TRO.2017.2705103.

[12] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-based SLAM," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2661–2666; doi: 10.1109/IROS.2014.6942926.

[13] M. Filipenko and I. Afanasyev, "Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment," *2018 International Conference on Intelligent Systems (IS)*, 2018, pp. 400–407; doi: 10.1109/IS.2018.8710464.

[14] T. Pire et al., "S-PTAM: Stereo Parallel Tracking and Mapping," *Robot. Auton. Syst.*, vol. 93, 2017, pp. 27–42; doi: 10.1016/j.robot.2017.03.019.

[15] J.-M. Valin et al., "Robust Recognition of Simultaneous Speech by a Mobile Robot," *IEEE Trans. Robot.*, vol. 23, no. 4, 2007, pp. 742–752; doi: 10.1109/TRO.2007.900612.

[16] S. Alharbi et al., "Automatic Speech Recognition: Systematic Literature Review," *IEEE Access*, vol. 9, 2021, pp. 131858–131876; doi: 10.1109/ACCESS.2021.3112535.

[17] C.T. Ishi et al., "A Robust Speech Recognition System for Communication Robots in Noisy Environments," *IEEE Trans. Robot.*, vol. 24, no. 3, 2008, pp. 759–763; doi: 10.1109/TRO.2008.919305.

[18] J. Davila-Chacon, J. Liu, and S. Wermter, "Enhanced Robot Speech Recognition Using Biomimetic Binaural Sound Source Localization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, 2019, pp. 138–150; doi: 10.1109/TNNLS.2018.2830119.

[19] S. Girirajan and A. Pandian, "Offline Automatic Speech Recognition System Based on Bidirectional Gated Recurrent Unit (Bi-GRU) with Convolution Neural Network," *J. Mob. Multimed.*, 2022; doi: 10.13052/jmm1550-4646.1869.

[20] I. Okunevich et al., "MobileCharger: An Autonomous Mobile Robot with Inverted Delta Actuator for Robust and Safe Robot Charging," *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation* (ETFA), 2021, pp. 1–8; doi: 10.1109/ETFA45728.2021.9613366.

[21] G. Vasilyev et al., "Walking Algorithm for Robotis Op3 Humanoid Robot with Force Sensors," *2019 12th International Conference on Developments in eSystems Engineering* (DeSE), 2019, pp. 20–23; doi: 10.1109/DeSE.2019.00014.

[22] A.D. Ames, "Human-Inspired Control of Bipedal Walking Robots," *IEEE Trans. Autom. Control*, vol. 59, no. 5, 2014, pp. 1115–1130; doi: 10.1109/TAC.2014.2299342.

[23] J. Lafaye, D. Gouaillier, and P.-B. Wieber, "Linear Model Predictive Control of the Locomotion of Pepper, a Humanoid Robot with Omnidirectional Wheels," *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 336–341; doi: 10.1109/HUMANOIDS.2014.7041381.

[24] A.K. Pandey and R. Gelin, "A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, 2018, pp. 40–48; doi: 10.1109/MRA.2018.2833157.

[25] Z. Bauer et al., "Refining the Fusion of Pepper Robot and Estimated Depth Maps Method for Improved 3D Perception," *IEEE Access*, vol. 7, 2019, pp. 185076–185085; doi: 10.1109/ACCESS.2019.2960798.

[26] L. Cascone et al., "DTPAAL: Digital Twinning Pepper and Ambient Assisted Living," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, 2022, pp. 1397–1404; doi: 10.1109/TII.2021.3090363.

[27] G. Ercolano et al., "Gesture Recognition with a 2D Low-Resolution Embedded Camera to Minimise Intrusion in Robot-Led Training of Children with Autism Spectrum Disorder," *Appl. Intell.*, vol. 54, no. 8, 2024, pp. 6579–6591; doi: 10.1007/s10489-024-05477-z.

[28] Y. Wei, "A Comprehensive Approach to the Generation of Human-Like Arm Movements on Robot NAO," *IEEE Access*, vol. 8, 2020, pp. 172869–172881; doi: 10.1109/ACCESS.2020.3025532.

[29] C. Zheng, P. Zhang, and Y. Li, "Semantic SLAM System for Mobile Robots Based on Large Visual Model in Complex Environments," *Sci. Rep.*, vol. 15, no. 1, 2025, p. 8450; doi: 10.1038/s41598-025-90340-5.

[30] K.M. Ahmad Yousef et al., "Personal Assistant Robot using Reinforcement Learning: DARWIN-OP2 as a Case Study," *Intell. Serv. Robot.*, vol. 17, no. 4, 2024, pp. 815–831; doi: 10.1007/s11370-024-00540-7.

[31] P. Allgeuer et al., "The igus Humanoid Open Platform," *KI - Künstl. Intell.*, vol. 30, no. 3, 2016, pp. 315–319; doi: 10.1007/s13218-016-0448-6.

[32] G. Ficht et al., "NimbRo-OP2X: Adult-Sized Open-Source 3D Printed Humanoid Robot," *2018 IEEE-RAS 18th International Conference on Humanoid Robots* (Humanoids), 2018, pp. 1–9; doi: 10.1109/HUMANOIDS.2018.8625038.

[33] T.P. Tomo et al., "A New Silicone Structure for uSkin—A Soft, Distributed, Digital 3-Axis Skin Sensor and Its Integration on the Humanoid Robot iCub," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, 2018, pp. 2584–2591; doi: 10.1109/LRA.2018.2812915.

[34] F. Naveros et al., "VOR Adaptation on a Humanoid iCub Robot Using a Spiking Cerebellar Model," *IEEE Trans. Cybern.*, vol. 50, no. 11, 2020, pp. 4744–4757; doi: 10.1109/TCYB.2019.2899246.

[35] G. Lombardi et al., "Humanoid Facial Expressions as a Tool to Study Human Behaviour," *Sci. Rep.*, vol. 14, no. 1, 2024, p. 133; doi: 10.1038/s41598-023-45825-6.

[36] L. Fortunati et al., "Exploring the Perceptions of Cognitive and Affective Capabilities of Four, Real, Physical Robots with a Decreasing Degree of Morphological Human Likeness," *Int. J. Soc. Robot.*, vol. 15, no. 3, 2023, pp. 547–561; doi: 10.1007/s12369-021-00827-0.

[37] D. Suresh, F. Dax, and D. Borrmann, "Advancing Robotics with an Interactive and Musical Humanoid Robot Based on the InMoov Project," *2024 22nd International Conference on Research and Education in Mechatronics* (REM), 2024, pp. 237–243; doi: 10.1109/REM63063.2024.10735692.

[38] J. Ma et al., "A Multi-LIDAR SLAM Method Based on Hector-SLAM," *2024 China Automation Congress* (CAC), 2024, pp. 345–350; doi: 10.1109/CAC63892.2024.10865468.

[39] S. Huang et al., "A Robust 2D Lidar SLAM Method in Complex Environment," *Photonic Sens.*, vol. 12, no. 4, 2022, p. 220416; doi: 10.1007/s13320-022-0657-6.

[40] N. Abdelaziz and A. El-Rabbany, "LiDAR/Visual SLAM-Aided Vehicular Inertial Navigation System for GNSS-Denied Environments," *2022 5th International Conference on Communications, Signal Processing, and their Applications* (ICCSPA), 2022, pp. 1–5; doi: 10.1109/ICCSPA55860.2022.10019210.

[41] Q. Du et al., "Tightly-coupled Lidar-Visual-Inertial SLAM for Mobile Robot," *2024 International Conference on Image Processing, Computer Vision and Machine Learning* (ICICML), 2024, pp. 842–845; doi: 10.1109/ICICML63543.2024.10957948.

[42] J. Lin et al., "R ^2 LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, 2021, pp. 7469–7476; doi: 10.1109/LRA.2021.3095515.

[43] Z. Wang et al., "SLAM Mapping of Information Fusion between Lidar and Depth Camera," *2022 International Conference on Image Processing, Computer Vision and Machine Learning* (ICICML), 2022, pp. 142–145; doi: 10.1109/ICICML57342.2022.10009758.

[44] R.C. Hsu et al., "Real-Time Interaction System of Human-Robot with Hand Gestures," *2020 IEEE Eurasia Conference on IOT, Communication and Engineering* (ECICE), 2020, pp. 396–398; doi: 10.1109/ECICE50847.2020.9301957.

[45] I. Diddeniya et al., "Human–Robot Communication System for an Isolated Environment," *IEEE Access*, vol. 10, 2022, pp. 63258–63269; doi: 10.1109/ACCESS.2022.3183110.

[46] D. Song et al., "VLM-Social-Nav: Socially Aware Robot Navigation Through Scoring Using Vision-Language Models," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, 2025, pp. 508–515; doi: 10.1109/LRA.2024.3511409.

Review of Hybrid Path Planning Techniques for mobile robots: Integration between AI Techniques and Traditional Methods in known environments

Submitted: 15th July 2025; accepted: 1st October 2025

Mohamed Abdelghafar, Hazlina Selamat, Nurulaqilla Binti Khamis, Anas Aburaya, Mohd Taufiq Muslim

DOI: 10.14313/jamris-2026-017

Abstract:
Mobile robots require effective and secure path planning, especially in complex and dynamic environments. Traditional algorithms such as A, Dijkstra, and Rapidly-exploring Random Trees (RRT) offer dependable and mathematical solutions but face challenges regarding scalability, adaptability, and processing requirements in real-time applications. On the other hand, artificial intelligence (AI) techniques, such as reinforcement learning (RL) and neural networks (NNs) provide flexibility and quick decision-making but face challenges such as data dependency, optimal solution, and computational overhead. This review analyzes hybrid path planning methodologies that integrate traditional algorithms with AI techniques, utilizing the advantages of both to overcome their limitations. Hybrid approaches improve scalability, collision avoidance, and re-planning efficiency by integrating the accuracy and reliability of traditional techniques with the adaptability and learning capabilities of AI. This review categorizes and analyzes research to identify significant gaps and suggests future paths for enhancing hybrid path planning, offering insights for the development of more robust and intelligent navigation systems for mobile robots and autonomous platforms.*

Keywords: *AI, Robotics, Path planning, mobile robots*

1. Introduction

Mobile robots have gained significant attention in areas such as surveillance, agriculture, disaster management, and logistics. An important aspect of mobile robot activities is path planning, which is defined as identifying an optimal pathway from a start point to a destination while minimizing costs in terms of time, energy, and computational resources, and avoiding obstacles. The efficiency of mobile robot path planning is especially crucial in known environments where detailed 3D maps are available, allowing for more precise navigation and optimization.

Traditional methods of path planning, such as the A* algorithm, Dijkstra's algorithm and Rapidly-exploring Random Trees (RRT), have been widely used due to their reliability and mathematical accuracy. These algorithms have proven effective in producing collision-free paths in 2D and 3D environments. However, they often struggle with high computational complexity and lack adaptability in dynamic and complex scenarios [1].

Artificial Intelligence (AI) methods, such as machine learning, deep learning, and reinforcement learning, can enhance path planning performance. AI-based methods offer several advantages, such as learning from data, adaptation to evolving environments, and multi-objective optimization. Despite these strengths, AI methods alone may require extensive training data, high computational power, and may lack interpretability compared to traditional algorithms [2].

The combination of AI and traditional methods for mobile robot path planning presents a promising hybrid approach that leverages the strengths of both paradigms. By integrating AI's adaptability and learning capabilities with the robustness and interpretability of traditional algorithms, researchers aim to develop more efficient and versatile path planning solutions. This hybrid approach has the potential to address the limitations of both AI and traditional methods, particularly in known environments where comprehensive 3D maps provide a structured framework for navigation.

This review aims to explore the current state of research on the combination of AI and traditional methods for mobile robot optimal path planning in known environments. The review will focus on identifying the key AI techniques and traditional algorithms employed, examining their performance, and highlighting the challenges and opportunities associated with their integration. By combining the findings from existing literature, this review aims to provide a comprehensive understanding of the hybrid approaches to mobile robot path planning.

1.1. Fundamentals of Mobile Robot Path Planning

Mobile robot path planning is critical for autonomous moving, as it provides a secure, effective, collision-free path from the starting point to the destination. The difficulty of this task depends on factors such as the environment in operation (urban or rural), obstacles (static or dynamic), energy limitations, and the specific mission goals (surveillance, delivery, mapping, etc.).

The path planning sequence has three phases, environmental modeling, path searching and

optimization, and, finally, path execution and adaptation. Environmental modeling is when a 3-dimensional map of the area is developed and obstacles are modeled similarly using occupancy grids and Voronoi diagrams; path search and optimization is when algorithms identify paths based on parameters such as length and safety; in path execution and modification, mobile robots need to change the path accordingly based on random obstacles and environmental changes [1].

Mobile robot path planning algorithms, both traditional, AI-based or hybrid, are crucial to enabling mobile robots to navigate safely and efficiently in heterogeneous environments. The algorithms are developed for path optimization by balancing objectives such as obstacle avoidance, energy efficiency, and mission-specific requirements that vary based on the use of the mobile robot. Path planning algorithms combine mathematical precision with adaptive functionality as they allow mobile robots to deal with static and dynamic obstacles, environmental changes, and accurate navigation of all types of missions. This holistic consideration of path planning stresses the need for algorithms to develop and adapt as new problems arise in autonomous movement.

1.2. Global and Local Path Planning

Path planning algorithms can be classified into global path planning and local path planning based on their operational scope and planning procedure [2]. Global path planning algorithms, such as Dijkstra's, A*, D* Lite, and Cell Decomposition, are engineered to calculate a path from a start point to the target point for a defined environment. These approaches use a complete or partial representation of the environment to determine an optimal or near-optimal path, accounting for all detected obstacles. These methods are especially efficient in static environments where all information of the environment is known ahead of time and all of what has been learned is used for complete planning [2].

Conversely, local path planning techniques focus on immediate navigation and avoidance of obstacles. Local path planning includes both traditional algorithms, such as Dynamic Window Approach (DWA) and Artificial Potential Field (APF), as well as AI methods, such as Machine Learning (ML) and Reinforcement Learning (RL). Local path planning algorithms are specifically designed to focus on the ability to make quick decisions, from the current state of the mobile robot and its sensors, without requiring all the environmental knowledge. Local path planners can be highly effective in dynamic or partially known environments where the mobile robot must respond quickly to an unexpected change or moving obstacle that was not known prior to encountering [2].

In practical mobile robot operations, the integration of global and local path planning methodologies is frequently beneficial [2]. The global planner will provide a planned long path to the target, and then the local planner can modify the mobile robot's trajectory considering immediate hazards and other changes in the environment. This hybrid method guarantees both optimality and adaptability, improving the mobile robot's overall navigation efficiency.

2. Research Methodology

Previous review papers have focused on algorithm reviews in general or on broad integrations of techniques such as [3–5]. However, this paper specifically reviews the integration between traditional path planning algorithms and artificial intelligence (AI) methods. Papers were retrieved exclusively from Scopus to ensure a comprehensive review. A combination of keywords and Boolean operators was used to maximize the search results. The keywords such as:

- "Mobile robot" AND "path planning" AND "known environment" AND "hybrid" AND "artificial intelligent" OR "AI"
- "Mobile robot" AND "path planning" AND "known environment" AND "combine" AND "artificial intelligent" OR "AI"
- "Mobile robot" AND "path planning" AND "known environment" AND "integrate" AND "artificial intelligent" OR "AI"
- "Robot" AND "path planning" AND "known environment" AND "hybrid" AND "artificial intelligent" OR "AI"

The search range was limited to papers published from January 2019 to December 2024. Using these keywords, a total of 105 papers were retrieved. By applying inclusion and exclusion criteria, 23 papers were identified that fit the requirements. Figure 1 illustrates the number of publications over the years. Some papers focused on the hybrid between two AI methods which fall outside the scope of this review.

The quality of each paper was evaluated based on clarity of objectives and methodology, relevance to hybrid path planning for mobile robots, and robustness of experimental or simulation results. Inclusion criteria ensured the review synthesized the studies discussing the hybrid methods integrating traditional and AI-based methods, and studies of mobile robot research in the known environment. Exclusion criteria ensured studies focused solely on either AI or traditional methods without integration, papers without any experimental or simulation results, and non-English papers. Overall, the review ensures a systematic and unbiased exploration of hybrid path planning methods, offering valuable insights for researchers and practitioners in the field.

3. Hybrid Approaches: Combining AI and Traditional Methods

Traditional path planning algorithms provide reliable and often optimal solutions in structured environments but face scalability issues and struggle with real-time adaptability in dynamic settings [2, 3]. In contrast, Artificial Intelligence (AI) techniques enhance adaptability and decision-making, allowing robots to learn from data and

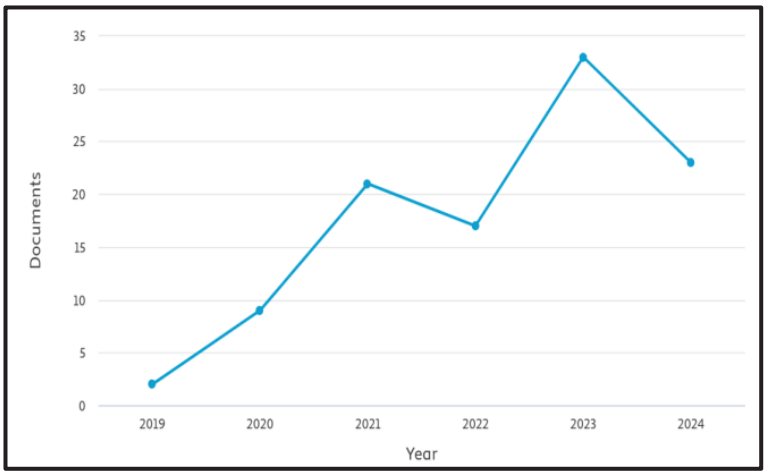


Figure 1. Line graph of number of papers vs. year published

respond to uncertainties, though they remain limited by computational demands, reliance on large datasets, and difficulty in guaranteeing global optimal paths [1, 3].

Hybrid approaches have emerged to combine the strengths of both paradigms. In such frameworks, traditional algorithms are employed as global planners to generate efficient baseline routes in known or partially known maps, while AI methods serve as local planners to adaptively handle real-time changes, dynamic obstacles, and uncertainties. This integration balances the efficiency and optimality of traditional methods with the adaptability and intelligence of AI, resulting in more robust and reliable path planning for mobile robots across diverse environments. Different hybrid models have been discussed, integrating different AI techniques with traditional methods to address the limitations of individual approaches. In the reviewed literature, 11 papers utilized RL as a local planner, including 3 that employed DRL, 8 papers used NN, and 4 utilized Fuzzy Logic, all combined with different traditional algorithms.

3.1. Integration with Reinforcement Learning (RL)

Reinforcement Learning (RL) enhances traditional algorithms by allowing mobile robots to adapt to dynamic environments through learning from interactions. RL is particularly effective for local path optimization, complementing traditional algorithms, which handle global path planning. In [6–8], A* is used with RL because A* provides optimal paths in static environments, but RL helps refine those paths in real time when dynamic obstacles or sudden changes occur. This integration is useful in cases where environmental conditions are semi-known, but real-time changes require constant path adjustment.

On the other hand, in [9, 10] RRT with RL is used because RRT excels in exploring large, unstructured environments, but it lacks refinement in finding optimal paths. RL complements RRT by improving the path's quality, particularly in environments where the mobile robot needs to avoid dynamic obstacles or continuously adapt to unforeseen changes in real-time [7]. RRT's fast exploratory nature is particularly suitable for global exploration, while RL refines local decisions, making this hybrid system ideal for navigating cluttered and rapidly changing environments.

Likewise, in [11, 12], PRM was combined with RL to manage navigation in large-scale, known environments. PRM generates the global path based on the roadmap, and RL optimizes the local trajectory as conditions evolve. PRM's suitability for structured environments makes it an ideal candidate for global path planning, while RL adapts to dynamic elements, ensuring that the mobile robot can handle changing conditions without fully recalculating its route [13].

3.2. Integration with Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) further extends RL by incorporating deep learning models to process complex, high-dimensional data, enabling mobile robots to adapt in real-time. In [19], DRL was integrated with traditional methods like A* or Dijkstra to overcome the challenge of high uncertainty in environments with dynamic obstacles. A* and Dijkstra provide efficient global path planning, but DRL enables the system to process large amounts of sensor data and optimize the mobile robot's local movements. This hybrid model is particularly advantageous in high-complexity environments, where mobile robot must make frequent adjustments based on real-time feedback [15].

The decision to use DRL with A* and Dijkstra stems from DRL's ability to handle vast datasets and improve real-time decision-making, making it well-suited for dynamic settings where frequent adaptations are required.

3.3. Integration with Neural Networks

Neural Networks (NNs) enhance traditional path planning methods by learning from previous experiences and processing real-time sensor data. Integrating NNs with RRT*, as demonstrated in [16–19], accelerate the selection of sampling points and improves the overall path efficiency. The decision to combine NNs with RRT* is based on RRT*'s ability to generate near-optimal paths through

re-wiring, and NNs' strength in refining the path selection by learning from prior experiences. This integration significantly reduces the computational cost of re-planning and improves the mobile robot's ability to navigate highly dynamic environments.

Similarly, integrating NNs with A* enhances the mobile robot's ability to adjust its global path based on real-time sensor data [20]. The proposed Learning Heuristic A* (LHA*) algorithm uses a neural network to model the heuristic function, ensuring faster exploration while maintaining a suboptimality bound. Also, Neural A* reformulates the A* algorithm into a differentiable module, allowing it to be integrated into a neural network and trained end-to-end, which bridges the gap between traditional search algorithms and deep learning [21]. This hybrid approach is particularly effective in environments with dynamic obstacles, where NNs help the mobile robot process sensor inputs and refine the computed path without needing constant recalculation.

Additionally, Deep Neural Networks (DNNs) further improve traditional path planning methods by learning heuristic functions to guide the search process more efficiently. As demonstrated in [22], DNNs can be trained to optimize the search cost in algorithms like A*, reducing computational overhead while maintaining accuracy. Similarly, in [23], DNN-based approach using max-pooling layers efficiently solves large-scale path planning problems without requiring training, demonstrating the power of DNNs in optimizing pathfinding in complex environments.

3.4. Integration with Fuzzy Logic

Fuzzy Logic complements traditional algorithms by providing a flexible decision-making framework in uncertain environments. When integrated with RRT, as seen in [24], Fuzzy Logic allows mobile robots to adjust their path based on imprecise or noisy sensor data, which RRT alone cannot manage effectively. The hybrid approach improves navigation in environments with high uncertainty, such as rescue missions or exploration of unknown areas. The decision to integrate Fuzzy Logic with RRT stems from the need to handle imprecise sensor data, where Fuzzy Logic's ability to manage uncertainty complements RRT's exploratory power.

When integrated with A*, as demonstrated in [25], Fuzzy Logic enhances the efficiency of the search process by dynamically adjusting the heuristic function in response to obstacle density, distance to the goal, and the number of visited nodes. While A* guarantees optimality, it is computationally expensive and memory-intensive in large or maze-like environments. The hybrid Fuzzy A* overcomes these limitations by reducing unnecessary state expansions and memory consumption, while still maintaining near-optimal paths, making it highly effective for large-scale and complex maps. In a similar manner, when combined with Dijkstra, as reported in [26], Fuzzy Logic provides the adaptability required for real-time navigation in partially known environments. Dijkstra ensures a globally optimal path offline but cannot handle unexpected obstacles during execution. Fuzzy Logic fills this gap by enabling reactive obstacle avoidance and smooth control adjustments based on sensor feedback. This integration preserves Dijkstra's reliability in static environments while extending its capability to dynamic scenarios, ensuring safe and efficient navigation where the environment cannot be fully known in advance

3.5. Summary

The reviewed studies focused on integration traditional algorithms with AI techniques for mobile robot path planning. The analysis highlighted how these hybrid approaches combine the systematic efficiency of traditional methods with the adaptability and intelligence of AI to address challenges in dynamic and complex environments. The key findings from the reviewed papers are summarized in Table 1, highlighting the strengths of each approach. Furthermore, the applications of these hybrid methods suggest their suitability for different operational contexts. For instance, A* combined with Reinforcement Learning could be well-suited for long-run navigation in complex outdoor environments and multi-robot coordination, such as autonomous transportation systems, since A* ensures a reliable global path while RL adapts to dynamic traffic or obstacle changes [6–8]. PRM integrated with RL may be advantageous for indoor ground robot navigation and outdoor UAV operations, such as warehouse automation or aerial delivery systems, because PRM efficiently explores high-dimensional spaces while RL refines safe passage in cluttered or dynamic zones [11]. Likewise, A* combined with Deep Reinforcement Learning appears suitable for highly complex and dynamic environments, such as disaster response or search-and-rescue missions, as DRL enhances adaptability to unpredictable hazards while A* provides a stable baseline path [14]. PRM combined with DRL could be applied to large-scale outdoor navigation, with potential uses in autonomous exploration in agriculture or environmental monitoring, where PRM maps vast spaces and DRL manages environmental uncertainty [22]. Meanwhile, RRT* integrated with Neural Networks shows potential for robotic arm manipulation tasks and target tracking systems, for example, in industrial assembly or surveillance applications, since RRT* generates an optimal trajectory while NN supports precision control and adaptive tracking [17]. These insights underline the flexibility of hybrid approaches in tailoring path planning solutions to diverse domains of robotic systems

4. Comparative Analysis

This section evaluates hybrid path-planning methodologies, focusing on two critical aspects: performance and computational complexity. The performance is evaluated based on the method's ability to produce optimal and smooth paths, crucial for effective and secure navigation, whereas

Table 1. Summary of hybrid approaches methods for mobile robot path planning

Paper	Global Planner Algorithm	Local Planner Algorithm	Strength
[6]	IADA*	Reinforcement Learning (RL)	- IADA* algorithm can re-plan paths efficiently in dynamic environments without recalculating the entire path when an obstacle is encountered.
[15]	—	Deep Reinforcement Learning (DRL)	- The human-in-the-loop (HL) training speeds up the convergence of the DRL algorithm, reducing the time required to learn complex navigation policies.
[7]	A*	Reinforcement Learning (RL)	- The RL is allowing the robot to adapt its policy through trial-and-error interactions with its surroundings.
[13]	PRM	Reinforcement Learning (RL)	- PRM is triggered using an updated probabilistic roadmap, if RL fails to find a valid path due to obstacles detected.
[10]	RRT	Reinforcement Learning (RL)	- RL learns to select optimal actions that lead to collision-free paths, while RRT generates collision-free states.
[8]	A*	Reinforcement Learning (RL)	- The approach can be scaled to multi-robot systems without centralized control.
[9]	RRT	Reinforcement Learning (RL)	- RL helps the RRT tree grow toward target point, avoiding computationally expensive steering functions.
[11]	PRM	Reinforcement Learning (RL)	- PRM-RL combines the strengths of PRMs for long-range planning with RL agents that handle short-range.
[12]	PRM	Reinforcement Learning (RL)	- PRM-RL is designed to be robust against sensor noise and unmodeled dynamic environments.
[14]	A*, Dijkstra	Deep Reinforcement Learning (DRL)	- The DRL is specifically trained to navigate around humans, predicting their movements and adjusting robot trajectories accordingly to avoid close encounters.
[17]	RRT*	Back Propagation (BP) Neural Networks (NNs)	- BP-RRT* method uses neural networks to predict the optimal number of samples required in each phase of the search, making it faster and more efficient. - reducing the computational process by optimizing the node selection process
[18]	A*, RRT*	Neural Networks (NNs)	- RNN continuously learns from the environment, making it adaptable and faster in generating paths. - RNN allows it to operate in a relatively constant time regardless of environmental complexity.
[19]	A*, RRT*	Neural Networks (NNs)	- Limitation learning from pre-calculated optimal paths making this method unique on real-time calculations. - The use of R-CNN allows for fast computation by leveraging offline-trained models, reducing the need for heavy real-time computations
[24]	RRT	Fuzzy Logic	- fuzzy logic is computationally light and well-suited for real-time operations. - An extended Kalman filter (EKF) is employed to minimize cross-track errors during path following, ensuring smooth and accurate navigation along the planned trajectory
[22]	PRM	Deep Reinforcement Learning (DRL)	- PMR-Dueling DQN utilizes prioritized replay and dueling networks, improving the learning process by focusing on more critical learning events and better approximating state-action values.
[20]	—	Deep Neural Network (DNN)	- DNN is used to optimize the heuristic function, allowing it to maintain the strengths of traditional search algorithms while improving efficiency.
[27]	A*	Neural Networks (NNs)	- Learning Heuristic A* (LHA*) algorithm uses a neural network to model the heuristic function. - The neural network reduces the number of unnecessary vertex expansions in a graph, speeding up the search process.
[21]	A*	Neural Networks (NNs)	- Neural A* combines learning and search into a unified framework, which allows for both task optimization and improved performance.
[23]	—	Deep Neural Network (DNN)	- OMAP does not require large datasets or neural network training, making it a simple yet powerful alternative for solving complex path-planning problems.
[28]	DWA	Fuzzy logic	- Important points on the global path are selected as key sub-target sites for the local motion planning phase.
[25]	A*	Fuzzy logic	- Significantly reduces computation and memory usage in large, complex environments while maintaining near-optimal paths.
[26]	Dijkstra	Fuzzy logic	- Ensures globally optimal offline planning with adaptive real-time obstacle avoidance in partially known environments

computational complexity analyzes convergence rate, scalability to complex environments, and re-planning efficacy in dynamic situations. Due to the varied experimental configurations and uses in different investigations, direct comparisons are difficult. Therefore, this comparison emphasizes general concepts and trade-offs, providing insights into the strengths of each method in relation to its intended application.

4.1. Performance

The performance of hybrid path planning approaches is evaluated based on two key metrics: (i) optimal and smooth path and (ii) success rate, which together reflect the quality and reliability of the paths generated by these methods.

4.1.1. Optimal and Smooth Path

Creating paths that are optimal in length and smooth in execution is essential for assuring efficient and safe navigation while reducing energy consumption and mechanical wear on robotic systems. Numerous hybrid methodologies proficiently achieve this balance by utilizing the advantages of traditional algorithms alongside AI techniques.

In [14], the methodology was evaluated across many contexts and scenarios, showing that the combination of traditional algorithms with Deep Reinforcement Learning (DRL) always produces optimal paths compared to independent methods. In [15] RRT was integrated with RL, utilizing RRT as the sampling method and RL to link the sampled points in order to formulate the shortest and most secure path. This hybrid methodology utilizes the exploration efficacy of RRT and the flexibility of RL to produce both optimal and smooth paths.

In [16], RRT* was combined as a global planner alongside a neural network as a local planner, significantly improving path-finding efficiency. Experiments comparing this approach with RRT* and improved RRT* (IRRT*) showed that while path length remained comparable, integrating neural networks reduced the number of nodes and computation time required to reach the optimal path. Similar results were observed in [15, 16], where the integration of neural networks refined the sampling and path selection processes, producing optimal solutions more efficiently than RRT* and Improved RRT* as demonstrated in Figure 2.

4.1.2. Success Rate

The success rate is an essential indicator of the reliability of path planning systems, representing the proportion of paths successfully completed without failures or collisions across many scenarios. Hybrid methodologies that combine traditional algorithms with AI techniques have shown considerable enhancements in success rates, especially in dynamic and intricate contexts.

In [11], 50 tests were performed to assess the integration of Anytime Dynamic A* (iADA*) with various reinforcement learning techniques, comparing it with standalone iADA* and iADA* integrated with DQN and DDPG. The results showed that both iADA* with DQN and iADA* with DDPG achieved high success rates in both dynamic and static environments: 94% and 90% in dynamic environments, compared to 74% for standalone iADA*, and 100% in static environments, compared to 94% for standalone iADA* as shown in Table 2.

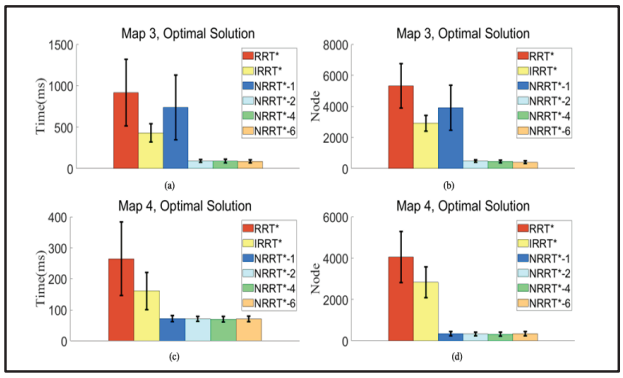


Figure 2. Comparison of the optimal solutions between the integration of RRT with Neural Network, RRT, and Improved RRT* [16]

Table 2. Comparison of the success rates between the integration of iADA with AI techniques and standalone iADA [6]

Static environment	**Algorithms**		
	IADA*	IADA*+DQN	IADA*+DDPG
Success	47	50	50
Failure	3	-	-
Success rate	94%	100%	100%
Dynamic environment	**Algorithms**		
	IADA*	IADA*+DQN	IADA*+DDPG
Success	37	47	45
Failure	13	3	5
Success rate	74%	94%	90%

Likewise, in [14], success rates were emphasized when integrating traditional methods with Deep Reinforcement Learning (DRL). Experiments in various obstacle-laden environments demonstrated that the hybrid technique consistently navigated paths without failure, surpassing traditional algorithms. In [12], Reinforcement Learning was utilized as a global Guide, integrating global planning with localized RL modifications, and attained elevated success rates in complex environments. [22] also exhibited strong performance by achieving higher success rates in both static and dynamic scenarios.

4.2. Computational Complexity

The Computational Complexity of hybrid path planning methods is a crucial factor in assessing their efficiency and practicality, particularly for real-time and resource-constrained applications.

4.2.1. Convergence Speed

One of the key aspects of computational complexity is Convergence Speed, which measures how quickly a method can compute a solution or learn an effective policy.

Table 3. Comparison of convergence time between PRM+DQN, DQN, DDQN, and Q-learning [22]

Method	Algorithm comparison on environment E-2	
	Success rate	Convergence Time/min
Q-learning	26.7	268
DQN	49.3	139
DDQN	56.4	116
PMR-Dueling DQN	84.6	107
Method	**Algorithm comparison on environment E-3**	
	Success rate	Convergence Time/min
Q-learning	21.4	276
DQN	32.6	161
DDQN	41.5	143
PMR-Dueling DQN	79.6	122

Methods that combine reinforcement learning (RL) or deep reinforcement learning (DRL) with traditional algorithms significantly improve convergence speed by enhancing decision-making and minimizing unnecessary computations. [9] employs Globally Guided Reinforcement Learning, which improves convergence by utilizing global planning to guide the RL agent's attention towards pertinent regions of the environment. This reduces the search space and enables the approach to compute optimal pathways more quickly, even in dynamic and complex environments. Similarly, [22] integrates Dueling Deep Q-Networks (Dueling DQN) with PRM, achieving superior performance compared to standalone Q-learning, DQN, and DDQN algorithms. In experiments, the integrated method reduced computation times by significant margins, outperforming the other approaches by 9 and 21 minutes across different test environments, demonstrating its efficiency and scalability as shown in Table 3.

The integration of neural networks (NNs) with traditional algorithms also significantly improves convergence speed by reducing computational overhead. [17] integrates RRT* with a neural network, enhancing the efficiency of sampling nodes and focusing calculations on important regions. Experimental results indicate that this integration achieved a calculation time of 12.6 seconds, in contrast to 21.04 seconds for RRT, 32.31 seconds for RRT*, and 16.07 seconds for improved RRT* (IRRT*). This demonstrates the integration's ability to not only reduce computational time but also minimize the number of nodes required for pathfinding as shown in Table 4.

4.2.2. Scalability

The scalability of hybrid path planning approaches indicates their ability to maintain efficiency and performance as environmental complexity increases, such as larger workspaces, higher obstacle density, or more dynamic environments. Scalability is essential for practical applications requiring dependable navigation in various and challenging environments.

Table 4. Comparison of convergence speed between BP-RRT, RRT*, RRT, and IRRT* [17]

Algorithm name	Avarage search time/s	Avarage number of nodes samples
RRT	21.04	5293.6
RRT*	23.31	1366.7
Improved P-RRT*	16.70	1482.5
BP-RRT*	12.76	1044.7

In [6], the system was tested and evaluated based on success and failure rates in both static and dynamic environments. The test scenario included 17 obstacles of varying sizes, with 7 static and 10 dynamic obstacles moving randomly at velocities ranging from 5 and 25 m/s in unrestricted directions. These conditions simulate complex, dynamic environments, and the method consistently achieved success rates of 90% and above, underscoring its ability to scale effectively in highly dynamic scenarios.

Similarly, [10] showcased the scalability of their method through experiments conducted in six different environments with varying random obstacle distributions. In Environment 5, there were two paths to the destination, whereas in Environment 6, the UAV could reach the goal only by navigating along a specific side of the maze. These experiments demonstrated the method's capability to adapt to both open and constrained environments, maintaining robust performance across all scenarios.

[22] emphasized scalability by evaluating the method in three distinct environments: a regular map, a random map, and a free map. The regular map mimics warehouse environments with a mix of static and dynamic obstacles, while the random map includes randomly distributed static and dynamic obstacles at varying densities. The free map focuses exclusively on dynamic obstacles. Dynamic obstacles were modeled as uncontrollable robots that could move one cell per step in any direction. These experiments demonstrated the method's ability to efficiently navigate environments of varying complexity and obstacle configurations.

4.2.3. Re-planning Efficiency

Re-planning Efficiency evaluates the ability of path-planning techniques to adjust existing paths or create new ones in response to dynamic environmental changes, including the emergence of new obstacles or changes in the goal position. This metric is essential for real-time applications in unstable environments.

[6] and [8] showcase strong re-planning efficiency by integrating Anytime Dynamic A* (iADA*) and reinforcement learning. The iADA* algorithm in [6] can replan paths efficiently in dynamic environments without recalculating the entire path when an obstacle is encountered. Instead, it updates only the affected portion of the path, significantly reducing computational overhead. Similarly, in [8] re-planning efficiency had

been enhanced by using a globally guided reinforcement learning approach that employs a Moving Cost metric.

$$Moving\ Cost = \frac{N_s}{\left|C_{goal} - C_{start}\right|\ Ls} \quad (1)$$

Where N_s is the number of steps taken, and $\left|C_{goal} - C_{start}\right|\ Ls$ is the Manhattan distance between the start and goal cells. This metric indicates the ratio of actual moving steps to the ideal number of steps. The naive reward-based approach achieved success rates ranging between 68% and 89%, ensuring faster re-planning with shorter paths even in complex and dynamic environments.

In [9,10,12], sampling-based methods (RRT, RRT*, and PRM, respectively) were utilized to establish key nodes, while AI techniques connect these nodes to generate optimal paths. In dynamic environments, these established nodes provide a robust framework for quick re-planning. When obstacles appear or configurations change, the AI component adjusts the connections between nodes, allowing for efficient path recalculations without starting from scratch.

4.3. Summary

The assessment of hybrid methods illuminated both the strengths and the limitations of each approach, particularly when examined across key dimensions such as re-planning efficiency, scalability, convergence speed, success rate, and path optimality. Reinforcement Learning (RL) integrations demonstrated superior performance in terms of path smoothness, success rate, and adaptability to dynamic environments, making them highly effective for tasks requiring continuous re-planning. However, this advantage comes at the cost of extensive training times and significant data requirements, which limit their practicality in scenarios where fast deployment is necessary [6,17].

Neural Networks (NN) offered strong performance in processing high-dimensional sensor data and enhancing obstacle detection. Yet, in dynamic and uncertain settings, their reliance on pre-trained models constrained adaptability, often resulting in reduced path smoothness and lower success rates compared to RL-based hybrids. This contrast suggests that NN integrations are better suited for relatively structured environments with predictable features, while RL hybrids remain stronger in highly variable contexts [17].

Deep Reinforcement Learning (DRL) extended the adaptability of RL by handling more complex decision spaces, but its computational burden and slower convergence made it less viable for real-time applications. While DRL may excel in offline or simulation-rich domains where training time is less restrictive, it underperforms in fast-changing environments that require immediate responsiveness. Fuzzy Logic, in contrast, provided resilience in uncertain scenarios due to its rule-based flexibility, but it struggled with scalability and path optimization, limiting its effectiveness in large-scale or multi-robot systems [7].

Overall, the comparative analysis indicates that no single hybrid method provides a universally optimal solution. Each combination exhibits strengths in certain conditions while revealing clear weaknesses in others. This underscores the need for future work to focus not only on improving adaptability and computational efficiency, but also on systematically benchmarking hybrids across diverse scenarios to identify where each approach succeeds or fails.

5. Research Challenges and Future Directions

Although hybrid path-planning techniques have advanced substantially in recent years, many barriers remain that limit their usability and performance in applied settings, particularly in dynamic and uncertain environments. One of the main limits relates to collision avoidance in dynamic scenarios. In [6, 9], due to the local data from LiDAR or sensors, the system does not consider the prediction of future states of moving obstacles [15]. Similarly, it has limitations arising from the Q-learning equation, thus running into potential problems with convergence within dynamic environments, limiting the potential for any true obstacle avoidance strategies. In addition, the need for extensive datasets and training via reinforcement learning (RL) requires considerable computational resources and time to effectively train to act in complex environments [7]. Additionally, Noise in sensor data poses additional challenges, as seen in [12], where environmental uncertainty from noisy inputs impacts the accuracy and reliability of the path planning process. Lastly, combining neural networks with traditional techniques simultaneously in varying applications specifically in high-dimensional environments and concomitant dynamic obstacles [16,18], can often only be done with considerable computational requirements, limiting real-time adaptability.

Future research prioritizes addressing challenges related to obstacle avoidance in dynamic environments and sensor noise. Collision avoidance could also focus on the application of predictive models to calculate projected movements and integrate global and local data for overall situational awareness [6]. In addition, future research may investigate the combination of traditional algorithms, such as D* and D*Lite, with AI models in controlled scenarios to assess their performance in dynamic environments. Another promising direction is hybrid path planning in unknown environments, where prior maps are unavailable or incomplete. In such cases, traditional algorithms can provide exploratory global strategies, while AI techniques—particularly RL and DRL—can adaptively refine navigation based on real-time sensor feedback. This combination could be valuable for applications such as autonomous exploration, planetary rovers, or search-and-rescue missions in disaster zones, where the environment is partially or entirely unknown.

6. Conclusion

This review paper examined various studies published that utilized hybrid models for path planning. Hybrid models leverage the systematic reliability of traditional algorithms while leveraging AI's adaptability to address complex navigation problems in both static and dynamic environments. We concluded with various advantages of each approach including the increase of path optimality, smoothness, scalability, and re-planning. The comparative analysis showed that RL outperformed other methods in path smoothness and successful attempts, especially when complemented by graph-based approaches like A*, which incorporated flexibility to achieve optimal paths. Neural Networks and Deep Reinforcement Learning offered significant adaptations and decision-making capabilities but incurred computational cost. Fuzzy logic was robust to uncertainty but was not successful in path optimization and scalability, which are the main advantages of other associated methods.

Overall, despite additional advancements in hybrid methodology, future work will focus on addressing specific challenges on collision avoidance in highly dynamic environments, reducing issues related to sensor noise, and minimizing computational cost for real-time navigation in autonomous applications. Future research should continue to develop hybrid models that combine automated path-planning with collision avoidance, robust optimization to manage sensor noise, and lightweight analytics to limit computational cost. Future exploration of combinations of AI and traditional methods, using RL with adaptive graph-based approaches, would enhance the adaptability and reliability of hybrid path-planning models. This paper offers an in-depth analysis of hybrid path-planning techniques, providing essential insights for researchers seeking to improve autonomous navigation systems for practical applications.

AUTHORS

Mohamed Abdelghafar – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: mohamedalyabdelghafar@gmail.com.

Hazlina Selamat* – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: hazlina@utm.my.

Nurulaqilla Binti Khamis – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: nurulaqilla@utm.my.

Anas Aburaya – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: ihanas2002@graduate.utm.my.

Mohd Taufiq Muslim – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: mohdtaufiqmuslim@utm.my.

*Corresponding author

ACKNOWLEDGEMENTS

This research received funding under Research Grant Vote 00Q18 by Universiti Teknologi Malaysia and the Malaysian Ministry of Higher Education.

References

[1] S. Aggarwal and N. Kumar, "Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges" Computer Communications, vol. 149, pp. 270–299, Jan. 2020, doi: 10.1016/j.comcom.2019.10.014.10.1016/j.comcom.2019.10.014.

[2] J. R. Sánchez-Ibáñez, C. J. Pérez-Del-pulgar, and A. García-Cerezo, "Path Planning for Autonomous Mobile Robots: A Review," Sensors, vol. 21, no. 23, 2021, doi: 10.3390/s21237898.

[3] C. Cheng, Q. Sha, B. He, and G. Li, "Path Planning and Obstacle Avoidance for AUV: A Review," Ocean Engineering, vol. 235, 2021, doi: 10.1016/j.oceaneng.2021.109355.

[4] K. Katona, H. A. Neamah, and P. Korondi, "Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot," Sensors, vol. 24, no. 11, 2024, doi: 10.3390/s24113573

[5] B. B. K. Ayawli, R. Chellali, A. Y. Appiah, and F. Kyeremeh, "An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning," Journal of Robotics, 2018, doi: 10.1155/2018/8269698.

[6] A. A. Maw, M. Tyan, T. A. Nguyen, and J. W. Lee, "IADA-RL: Anytime Graph-Based Path Planning With Deep Reinforcement Learning for an Autonomous UAV*," *Applied Sciences*, vol. 11, no. 9, 2021, doi: 10.3390/app11093948.

[7] K. Cai, C. Wang, J. Cheng, S. Song, C. W. de Silva, and M. Q-H Meng, "Mobile Robot Path Planning in Dynamic Environments: A Survey," "arXiv preprint arXiv:2002.09197, 2020".

[8] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020, doi: 10.1109/LRA.2020.3026638

[9] H. T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic Motion Planning via Learning Reachability Estimators from RL Policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, Oct. 2019, doi: 10.1109/LRA.2019.2931199.

[10] P. Gao, Z. Liu, Z. Wu, and D. Wang, *A Global Path Planning Algorithm for Robots Using Reinforcement Learning*. IEEE, 2019.

[11] A. F. K. Oslund, O. R. A. Francis, L. Tapia, and M. F. J. Davidson, *PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning*. [IEEE], 2018.

[12] A. Francis et al., "Long-Range Indoor Navigation with PRM-RL," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1115–1134, Aug. 2020, doi: 10.1109/TRO.2020.2975428.

[13] V. Bisht, "Visibility-Constrained Path Planning for Unmanned Aerial Vehicles," M.S./Ph.D. thesis, Year 2023.

[14] R. Guldenring, M. Gorner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning Local Planners for Human-Aware Navigation in Indoor Environments," in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 6053–6060. doi: 10.1109/IROS45743.2020.9341783.

[15] S. Zhang, Y. Li, F. Ye, X. Geng, Z. Zhou, and T. Shi, "A Hybrid Human-in-the-Loop Deep Reinforcement Learning Method for UAV Motion Planning for Long Trajectories with Unpredictable Obstacles," *Drones*, vol. 7, no. 5, May 2023, doi: 10.3390/drones7050311.

[16] J. Wang, W. Chi, C. Li, C. Wang, and M. Q. H. Meng, "Neural RRT∗: Learning-Based Optimal Path Planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, Oct. 2020, doi: 10.1109/TASE.2020.2976560.

[17] Q. Gao, Q. Yuan, Y. Sun, and L. Xu, "Path Planning Algorithm of Robot Arm Based on Improved RRT* and BP Neural Network Algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, Sep. 2023, doi: 10.1016/j.jksuci.2023.101650.

[18] S. Chehelgami, E. Ashtari, M. A. Basiri, M. Tale Masouleh, and A. Kalhor, "Safe Deep Learning-Based Global Path Planning Using a Fast Collision-Free Path Generator," *Rob Auton Syst*, vol. 163, May 2023, doi: 10.1016/j.robot.2023.104384.

[19] Y. Liu, Z. Zheng, F. Qin, X. Zhang, and H. Yao, "A Residual Convolutional Neural Network Based Approach for Real-Time Path Planning," *Knowledge Based Systems*, vol. 242, Apr. 2022, doi: 10.1016/j.knosys.2022.108400.

[20] T. Takahashi, H. Sun, D. Tian, and Y. Wang, "Learning Heuristic Functions for Mobile Robot Path Planning Using Deep Neural Networks," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[21] R. Yonetani, T. Taniai, M. Barekatain, M. Nishimura, and A. Kanezaki, "Path Planning using Neural A* Search," 2021. [Online]. Available: https://omron-sinicx.github.io/

[22] D. A. Deguale, L. Yu, M. L. Sinishaw, and K. Li, "Enhancing Stability and Performance in Mobile Robot Path Planning with PMR-Dueling DQN Algorithm," *Sensors*, vol. 24, no. 5, Mar. 2024, doi: 10.3390/s24051523.

[23] T. Kulvicius, S. Herzog, M. Tamosiunaite, and F. Wörgötter, "Generation of Paths in a Maze using a Deep Network without Learning," 2020. [Online]. Available: https://arxiv.org

[24] Z. Liu, Y. Zhang, C. Yuan, L. Ciarletta, and D. Theilliol, "Collision Avoidance and Path Following Control of Unmanned Aerial Vehicle in Hazardous Environment," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 95, no. 1, pp. 193–210, Jul. 2019, doi: 10.1007/s10846-018-0929-y.

[25] G. Airlangga, "Fuzzy A* for optimum Path Planning in a Large Maze," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 455–466, Dec. 2023, doi: 10.12928/biste.v5i4.9394.

[26] S. Sahloul, D. Ben Halima Abid, and C. Rekik, "An Hybridization of Global-Local Methods for Autonomous Mobile Robot Navigation in Partially-Known Environments," Jul. 01, 2021, *Department of Agribusiness, Universitas Muhammadiyah Yogyakarta*. doi: 10.18196/jrc.2483.

[27] S. Kim and B. An, *Learning Heuristic A*: Efficient Graph Search using Neural Network*, 2020. doi: 10.0/Linux-x86_64.

[28] Y. Sun et al., "Local Path Planning for Mobile Robots Based on Fuzzy Dynamic Window Algorithm," *Sensors*, vol. 23, no. 19, Oct. 2023, doi: 10.3390/s23198260.

Enhanced UAV Path Planning Using the Tangent Intersection Guidance (TIG) Algorithm

Submitted: 6th April 2025; accepted: 23rd April 2025

Hichem Cheriet, Badra Khellat Kihel, Samira Chouraqui

DOI: 10.14313/jamris-2026-018

Abstract:
Efficient and safe navigation of Unmanned Aerial Vehicles (UAVs) is critical for various applications, including combat support, package delivery and Search and Rescue Operations. This paper introduces the Tangent Intersection Guidance (TIG) algorithm, an advanced approach for UAV path planning in both static and dynamic environments. The algorithm uses the elliptic tangent intersection method to generate feasible paths. It generates two sub-paths for each threat, selects the optimal route based on a heuristic rule, and iteratively refines the path until the target is reached. Considering the UAV kinematic and dynamic constraints, a modified smoothing technique based on quadratic Bézier curves is adopted to generate a smooth and efficient route. Experimental results show that the TIG algorithm can generate the shortest path in less time, starting from 0.01 seconds, with fewer turning angles compared to A, PRM, RRT*, Tangent Graph, and Static APPATT algorithms in static environments. Furthermore, in completely unknown and partially known environments, TIG demonstrates efficient real-time path planning capabilities for collision avoidance, outperforming APF and Dynamic APPATT algorithms.*

Keywords: *Path planning, UAV navigation, Elliptic tangent graph, obstacle avoidance, unmanned aerial vehicle*

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) have significantly transformed many fields, such as cooperative combat [1], surveillance and security [2], disaster rescue [3, 4], package delivery [5, 6], traffic inspection [7] and target reconnaissance [8]. However, the path planning problem remains a challenge, hindering UAVs' ability to navigate through complex environments cluttered with threats while considering several factors such as obstacle avoidance, trajectory feasibility, energy consumption, and path length. Addressing this challenge is crucial to facilitate the effective application of UAVs and increase mission success rates. Path planning can be divided into two main approaches: static path planning and dynamic path planning. Static path planning, also known as global path planning, involves determining a path for the UAV from a start point to a destination point before the mission begins. This approach typically relies on a predefined map of the environment and considers static obstacles.

On the other hand, dynamic path planning, also known as local path planning, involves adapting the UAV's trajectory in real-time based on changing environmental conditions and obstacles encountered during the mission. The path calculation is performed onboard the UAV itself, allowing it to react swiftly to unexpected threats or alterations in the environment. This approach requires continuous sensing and decision-making capabilities to navigate safely and efficiently through the changing environments. Dynamic path planning algorithms are crucial for missions where the environment is uncertain or subject to frequent changes, enabling the UAV to autonomously adjust its path to achieve its objectives while avoiding collisions and optimizing performance. However, it is more computationally expensive, and the planned path may not necessarily be the most efficient. In static path planning, the A* algorithm is known for its effectiveness in finding an optimal path to a target while avoiding obstacles. However, in scenarios involving UAVs, the generated paths may not always be suitable. This is due to the unique constraints faced by UAVs, such as limited maneuverability and altitude restrictions. Also, the existing tangent graph methods need to handle the entire map, which can be time-consuming and result in poor solution quality, especially in complex environments. As a result, to efficiently plan collision-free paths in static and dynamic environments, this paper proposes a novel autonomous path planning algorithm based on the Tangent Intersection strategy. This algorithm improves the classical Tangent graph-based method and gives smooth suitability for UAVs.

The primary contributions of this paper can be summarized as follows:

1) We introduce a novel path-planning algorithm named Tangent Intersection Guidance (TIG), which is based on the tangent graph. Initially, the algorithm creates tangent lines for all obstacles in the environment. Subsequently, a heuristic rule is applied to select the best sub-path. Then, the algorithm is iteratively repeated until the target point is reached. This approach significantly reduces path length and gives a higher smoothness.
2) Extensive experiments were conducted in static and dynamic environments using random dense

maps with elliptic obstacles to validate the effectiveness of the proposed Path Planner. Simulation results demonstrate its capability to efficiently plan high-quality collision-free paths, particularly in dense environments.

The rest of this paper is organized as follows. Section 2 reviews related works in the field. Section 3 defines the path planning problem. Section 4 provides a detailed presentation of the proposed algorithm. Computational and comparative results are discussed in Section 5, and the conclusion is presented in Section 6.

2. Related Works

Path planning is one of the fundamental tasks for UAVs. It can be divided into five approaches, namely graph-based approaches, sampling-based methods, potential field methods, meta-heuristic methods, and machine learning techniques [9]. Graph-based methods utilize graph representations of the environment to model feasible paths. They often employ algorithms like A* [10], Dijkstra [11], Voronoi diagram [12], the tangent graph, and so on. The A* algorithm is a commonly used method in path planning that efficiently finds the shortest path. Voronoi's diagram's main idea is to partition the environment into regions based on proximity to obstacles. It can be used to generate paths by connecting the centroids of Voronoi cells. The generated Voronoi paths are far from obstacles, which ensures safety, whereas the planned path is not guaranteed to be optimal. Dijkstra's algorithm can find the shortest path, but it requires more space to store the nodes in dense graphs. These three methods are usually employed in a known static environment and cannot be used to perform UAV path planning in a dynamic environment [13].

Robert [14] proposes an algorithm based on tangent graphs, using common tangents of polygons to determine a feasible path. However, this method proves to be computationally expensive, particularly in high-dimensional environments. Furthermore, it fails to ensure a safe distance between the robot and the obstacles. Chen et al. [15] improve the tangent graph algorithm by enclosing obstacles with circular shapes. However, this approach wastes open space areas, resulting in longer path lengths. Liu et al. [16] enclosed the obstacles in ellipses to address the limitation of circular obstacles. This strategy reduces wasted areas and can result in shorter paths. However, despite this improvement, the path accuracy for UAVs in real-life scenarios remains insufficient. An improved version of the A* algorithm called BAA* was proposed by Wu et al. [17] based on multi-direction. This algorithm outperforms traditional A* in terms of both path length and execution time. Nevertheless, it requires parameter tuning and does not account for low-scale UAVs. Yuan et al. [18] propose an improved lazy theta* algorithm for UAV path planning, utilizing an octree map to reduce search nodes and adjust heuristic weights for precision and speed. Simulation and real-flight tests confirm its effectiveness in complex environments with multiple constraints. Bazeela et al. [19] propose an improved tangent graph intersection algorithm called ATGP-TI, which employs a tangent intersection technique and heuristics to find optimal paths for unmanned aerial vehicles. However, the algorithm is computationally expensive, especially in high-dimensional spaces.

In conclusion, the current methods relying on tangent graphs still encounter computational challenges due to the necessity of constructing tangents for the entire map, and the quality of the planned path remains inadequate for unmanned aerial vehicles.

Visibility graph-based methods have also been explored for UAV path planning due to their ability to generate optimal paths in structured environments. Liu et al. [20] introduced a tangent graph approach for mobile robots, effectively handling polygonal and curved obstacles. Blasi et al. [21] extended visibility graphs to UAV path planning in 3D constrained environments using layered essential visibility graphs, while Shah and Gupta [22] focused on accelerating A* search on visibility graphs over quadtrees for long-range path planning. However, visibility graph methods require preprocessing of all obstacle edges and can become computationally expensive in environments with numerous obstacles. Huan et al. [23] propose an improved tangent graph algorithm called APPATT. The algorithm demonstrates improved results regarding path quality and computational complexity (0.05 seconds). However, the generated path may be infeasible in some situations (Figure 3). The authors also applied a B-spline curve to smooth the generated path, which may lead to collisions(Figure 9). Moreover, these visibility and tangent graphs often generate paths that pass too close to obstacles, necessitating additional safety navigation measures.

Potential field algorithms, such as artificial potential field (APF) [24] and vector field histogram (VFH) [25], have been extensively applied in the field of path planning, particularly in dynamic environments. While these algorithms are known for their ability to generate smooth trajectories, they can also potentially become trapped in local minima.

Sampling-based methods such as the Probabilistic Roadmap Planner (PRM) and Rapidly Exploring Random Trees (RRT) are particularly effective in high-dimensional spaces, which are common in UAV path planning. PRM algorithm succeeds at handling complex environments with obstacles, narrow passages, and dynamic changes. However, building and storing the roadmap can be computationally expensive, especially in high-dimensional spaces, and may require significant memory [26]. On the other hand, the RRT algorithm does not necessitate sampling the entire space and constructing the roadmap before the mission, thereby reducing computational costs. RRT also operates efficiently in complex environments. Nevertheless, the quality of the path remains challenging for UAVs [27]. In their research [28], Zhou et al. propose a Depth Sorting Fast Search (DSFS) algorithm

to enhance path planning efficiency for underwater gravity-aided navigation. This new method improves the Quick Rapidly-exploring Random Trees* (Q-RRT*) algorithm [29], and the comparative experiments show that DSFS improves computational efficiency over Q-RRT*.

Artificial intelligence methods such as genetic algorithms (GA) [30], particle swarm optimization (PSO) [31], ant colony optimization (ACO) [32], and Grey Wolf Optimization (GWO) [33]..etc. Researchers have also proposed variants of these optimization algorithms for addressing UAV path planning problems, particularly in complex environments. However, these methods can become computationally time-consuming, especially when dealing with highly dense environments, and they do not always guarantee an optimal solution.

Also, machine learning methods such as support vector machines [34], neural networks [35], and deep reinforcement learning [36] have been applied to address path-planning challenges. One advantage of these methods is their ability to learn from input data and adapt to the mission environment. However, the training data and the computational resources required for training can be costly and require a lot of time. In addition, many research papers have been proposed. For example, Artificial Neural Networks using Radial Basis Functions (RBF-ANN) [37], improved Deep Q-Network (DQN) [38], and Opportunistic Hamilton-Jacobi-Bellman (oHJB) [39].

The aforementioned survey shows that graph-based methods are computationally demanding, particularly in complex environments, because they need to build the graphs for the entire map. Additionally, it is difficult for the UAV to balance between the planned path quality and the algorithm time.

To address the limitations of traditional tangent graph-based and visibility graph-based approaches, we propose the Tangent Intersection Guidance (TIG) algorithm, a novel tangent-graph path planning method for UAVs. The main contributions of our work are:

- Efficient Graph Construction: Unlike visibility graph-based methods that require precomputing all visibility edges, TIG calculates only the tangents lines of obstacles based on the heuristic function, significantly reducing computational complexity in large environments.
- Collision Detection: Traditional tangent graph methods rely on the line-of-sight algorithm for collision detection, which leads to higher computational costs, especially in the presence of numerous obstacles. TIG instead relies solely on the line-ellipse intersection equation to detect collisions, improving efficiency.
- Adaptability in Dynamic Environments: Visibility graphs and tangent graphs typically require full graph recomputation when obstacles change. TIG efficiently adapts to environmental changes by dividing the environment into smaller sub-environments, enabling real-time path adjustments.

Figure 1. UAV delivery path example

- Safe Navigation: Unlike existing methods, TIG incorporates a safety margin between the planned path and obstacles, ensuring robust collision avoidance and enhancing the reliability of UAV navigation in cluttered environments.
- Effective Waypoint Generation:While tangent-graph methods relying on tangent intersection to create waypoints, the TIG algorithm employs an optimized waypoint generation technique that ensures a smoother trajectory, reducing sharp turns and improving the overall flight efficiency of UAVs.
- Computational Performance: Compared to existing approaches, TIG achieves faster path planning while ensuring collision-free and optimal trajectories, making it highly suitable for UAV navigation in complex environments.

3. Problem Statement

As shown in Figure 1, a drone delivering goods in a low-altitude complex environment cluttered with obstacles, such as buildings, trees, and bridges from a start point S to a target point T. The drone encounters obstacle avoidance challenges in this particular environment, making it necessary to plan a feasible path without collisions between these points. The three-dimensional scenario can be represented in a two-dimensional scenario to reduce algorithm running time. The shapes of obstacles typically represented as polygons (e.g., buildings with sharp corners or irregular shapes) are not uniform, requiring higher computational resources and leading to unsmooth paths for the UAV. Additionally, the planned paths using these shapes are closer to obstacles, which increases the collision rate. To address these challenges, the paper represents obstacles as ellipses rather than exact polygonal representations, ensuring a smooth and efficient path. Ellipses provide a more manageable mathematical form for path planning, allowing the use of tangent-based methods that can quickly compute safe paths while maintaining a smooth trajectory. Additionally, by adding a safety distance into the elliptical representation, the UAV maintains a safety zone around obstacles, which reduces the risk of collision. This safety

distance can be adjusted depending on the UAV size and the specific mission requirements. The obstacles can be represented as:

$$\frac{((x-x_k)\cos\theta+(y-y_k)\sin\theta)^2}{(a+r_{\text{safe}})^2}+\frac{(-(x-x_k)\sin\theta+(y-y_k)\cos\theta)^2}{(b+r_{\text{safe}})^2}=1 \quad (1)$$

where x and y are the coordinates of any point on the ellipse, x_k and y_k are the coordinates of the center of the ellipse, a is the semi-major axis, b is the semi-minor axis, r_{safe} is the safety distance added to both the semi-major and semi-minor axes, and θ is the rotation angle of the ellipse measured counterclockwise from the positive x-axis.

In the previous scenario, there are N obstacles. A set of obstacles is denoted as $P = P_1, P_2, ..., P_N$, and the center coordinate of the obstacles are denoted as (x_1, y_1), (x_2, y_2),...., and (x_N, y_N) respectively. d represents the minimum safe distance required between a UAV and an obstacle.

The UAV may start from S to T, passing through several optimized node positions until it reaches the target point T while avoiding obstacles present in the scene. The path-optimizing process must meet UAV constraints and requirements, including path length, total turning radius, algorithm execution time, and mission duration, which will be detailed later.

4. Design of Tangent Intersection Guidance (TIG)

Most traditional graph-based planning algorithms suffer from complexity and inefficiency because they need to construct paths for the entire map, especially when dealing with high-dimensional environments, leading to more computational resources. To solve this, the Tangent Intersection Guidance algorithm is introduced to generate multiple sub-paths towards the goal point. These sub-paths are derived based on various factors such as sub-path length, collided obstacles, smoothness, and other environmental considerations. However, rather than considering all generated sub-paths, the algorithm employs a heuristic function to select the most promising one, making it easier for the UAV to navigate with low computational resources.

In this paper, we employ two distinct algorithms: a static tangent planner (S-TIG) and a dynamic tangent planner (D-TIG). The environmental map is known in advance in the (S-TIG) planner, and obstacles remain static. This algorithm is suitable for pre-mission planning scenarios (Offline Planner). In contrast, the (D-TIG) planner is used in dynamic environments or situations where the map is partially known or completely unknown, such as scenarios involving exploration or rapidly changing environments where obstacle positions need real-time path planning execution (Online Planner).

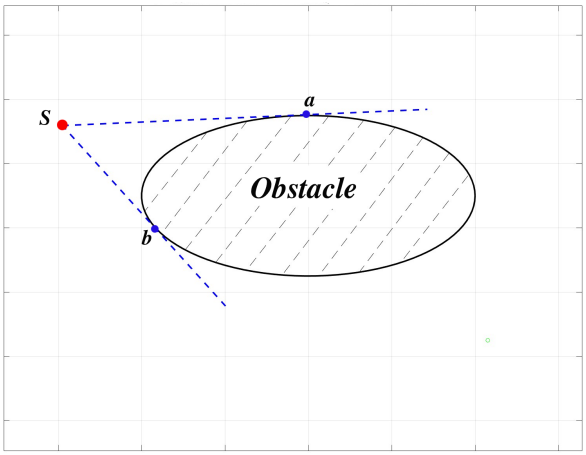


Figure 2. Tangent lines from start point S to elliptic obstacle

Table 1. Definitions of main notations

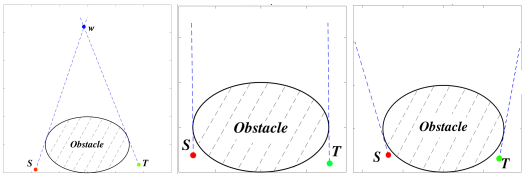
Notations	Description
S	Start-point
T	Target-point
W	A waypoint generated by the tangent planner
CurrentSet	A set to store candidate waypoints
ClosedSet	A set to store visited waypoints
treatedSet	A set that records tangent points that have already been calculated

This section briefly explains both static and dynamic planners. Since obstacles are represented as ellipses, each ellipse has two tangents from a given point x outside the obstacle. For example, to execute a mission from a start point denoted as S, two tangent lines to the elliptic obstacle can be drawn at tangent points a and b, respectively. Figure 2

Like the A* algorithm, the TIG Planner uses two sets: $CurrentSet$ and $ClosedSet$. The first set contains waypoints that are candidates for expansion, while the second set contains all explored waypoints. Each waypoint can result from an intersection of two tangent lines, which will be detailed in the next section. The algorithm procedure can be divided into two main functions. The first function is to get all candidate waypoints from the map. The second function is to choose the best subpath to integrate into the full path. All notations used in the algorithm are explained in Table 1.

4.1. Static Tangent Planner (S-TIG) Algorithm Process

This algorithm generates a smooth, collision-free path for UAVs from S to T. Since the environmental map is already known, we first represent obstacles as ellipses using Equation 1. Each obstacle is defined by its position coordinates and major and minor semi-axes, which are used to calculate the tangent



(a) Scenario 1 **(b)** Scenario 2 **(c)** Scenario 3

Figure 3. 3 Potential scenarios resulting in infeasible paths

lines and generate candidate waypoints. The S-TIG planner procedure consists of two main steps: obtaining candidate waypoints and selecting the best waypoint for path generation. Initially, before the UAV mission begins, the start point is considered the current node in the first loop. Then, a straight line from this current node to the target point is calculated. However, this line is generally infeasible due to obstacles obstructing its way. Hence, two tangent lines are drawn and used as candidate sub-paths from the current node to the first encountered obstacle. The first obstacle encountered is the one closest to the current node intersecting with the direct line to the target node. This process is iteratively repeated until no obstacles are encountered using the tangent lines from the current node.

The generation of waypoints can be defined as the intersection of two tangent lines at obstacle O; the first tangent line originates from the current node, and the second originates from the target point. However, this waypoint may prove insufficient due to multiple scenarios, potentially resulting in a path with poor smoothness or no feasible path. For example, as shown in Figure 3a, the waypoint w is the intersection between the start point S and the Target point T. Nonetheless, the subpath SwT is unfeasible for UAVs to traverse. In the second scenario, as depicted in Figure 3b, the tangent line passing through S is parallel to the tangent line passing through T, indicating no feasible path. In the last scenario, the tangent lines from S and T are impossible to intersect, meaning no waypoint can be added (Figure 3c).

To address this, virtual ellipses around each obstacle are created using the following equations:

$$\begin{aligned} a_{\text{vir}} &= a + d \\ b_{\text{vir}} &= b + d \end{aligned} \tag{2}$$

Where a and b represent the semi-major and semi-minor axes of the obstacle respectively, and a_{vir} and b_{vir} represent the semi-major and semi-minor axes of the virtual ellipses. d represents the safe distance between the obstacle and the virtual ellipse.

The waypoint is then determined as the intersection point between the virtual ellipse's perimeter and the obstacle's tangent line. It's important to note that there are two intersection points, and the selected waypoint is specifically chosen based on a criterion: it is the intersection point where the distance between the current point and the tangent point is smaller than the distance between the current point and the intersection point. Generally, this corresponds to the second intersection point from the current node along the subpath.

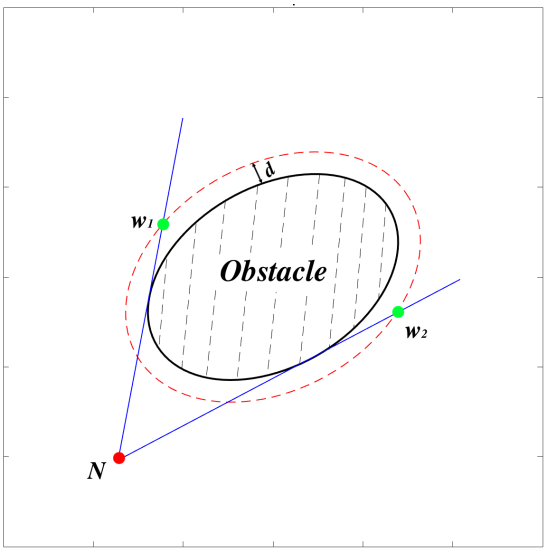


Figure 4. Waypoint generation using virtual ellipse technique

As shown in Figure 4, two tangent lines are generated from the current node N. The first tangent line intersects with the red-dashed virtual ellipse surrounding the obstacle at two points. Our approach selects $w1$ as a new waypoint, which is the farthest intersection point from N. The same criterion applies to the second tangent line, choosing $w2$ as a second waypoint.

This approach ensures the generated path maintains smoothness and avoids unnecessary sharp turns.

After calculating and obtaining all current node candidate waypoints, a heuristic function is applied to select the best sub-path. Details of this heuristic function will be provided later.

As illustrated in Figure 5a, the start point and the target point are denoted by S and T respectively, and $B_1, B_2, \ldots, B_11$ represent elliptic obstacles in the environment. We observe that the straight path from S to T is obstructed by obstacle B_1, which is the first collided obstacle. In this situation, the algorithm, as mentioned before, generates two tangent lines from S to obstacle B_1 at points of tangency P_1 and P_1', respectively. The algorithm also checks if both SP_1 and SP_1' are clear subpaths, which is false in our example, so we handle each case separately. Firstly, we check the first collided obstacle with the tangent line SP_1 and generate two tangent lines, namely SP_2 and SP_2', respectively. We can see now that both tangent lines SP_2 and SP_2' are clear subpaths, so we create two waypoints w_2 and w_2' using the virtual ellipse technique around obstacle B_2(Figure 5b). The same procedure is followed for the subpath SP_1'; drawing two tangent lines for obstacle B_3 to avoid it, we see that the subpath SP_3 is clear, so waypoint W_3 is created(figure. 5c). For the last tangent line SP_3', obstacle B_4 is in its way, so the tangent lines, namely SP_4 and SP_4', are created, and waypoints w_4 and w_4' are generated (Figure 5d). After

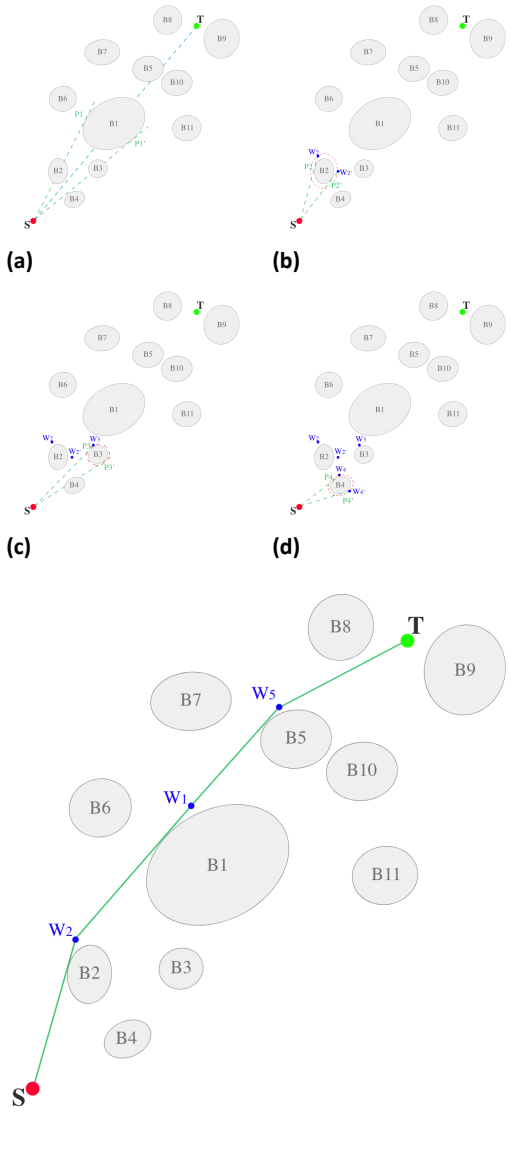


Figure 5. S-TIG algorithm steps

extracting all candidate waypoints, the algorithm uses the heuristic rule to determine the best for the first subpath. After this, the current point is transferred as the selected waypoint in our example, which is w_2. The other subpaths are iteratively created using the same method until the current node reaches the end node. Finally, the algorithm extracts the final path from the closed set using each node with its previous node. The resultant path in this example is $S \rightarrow W_2 \rightarrow W_1 \rightarrow W_5 \rightarrow T$ (Figure 5e).

To choose the best waypoints, we apply the following heuristic rule:

$$H(w) = D(N,w) + \alpha P + D(w,T) \qquad (3)$$

where P represents the number of obstacles intersecting the tangent line from the current point N to the waypoint w, $D(N,w)$ is the path length from N to w, and $D(w,T)$ is the distance from w to the target point T.

Since our algorithm prioritizes a direct path to the goal, incorporating the obstacle count P into the heuristic function encourages waypoint selection with fewer obstacles in way. This approach significantly reduces the number of obstacle interactions, leading to shorter, smoother, and computationally efficient trajectories. In addition, to balance the influence of P while maintaining heuristic admissibility, we introduce the parameter α, which adjusts its weight in the cost function. This refinement ensures that the algorithm remains both optimal and feasible, particularly in complex environments.

The steps of the S-TIG algorithm are detailed as follows:

1) Obstacle Modelization: Model the obstacles in the environment using Equation 1.
2) Initialization: Add the start point S to the currentSet set and initialize the current node as the node with the minimum heuristic value in the currentSet, which is S.
3) Node Initialization: Initialise the current node in the $to\_Explore$ set.
4) Exploration Loop: While the $to\_Explore$ set is not empty, the algorithm selects the first node to be explored as a temporary target node T_{temp}.
5) Subpath Check: Determine whether the direct line from the current node to the temporary target point T_{temp} is clear. If clear, generate a waypoint using the waypoint creation technique and append it to the $waypoints$ set. Otherwise, generate two tangent lines from the current node to the first collided obstacle, and add the points of tangency to $to\_Explore$ set.
6) Node Handling: Remove the temporary target node from $to\_Explore$ and add it to the *Explored* set.
7) Subpath Resolution: Relaunch Step 4) if there are still no clear subpaths.
8) Waypoint Addition: Calculate heuristic values for each waypoint using Equation 3 and add them to *currentSet*.
9) Target Check: Check if N it reaches the target point T. If it does, stop the algorithm and calculate the path; otherwise, return to Step 3) for further exploration.

The algorithm pseudocode is shown in Algorithm 1:

4.2. Dynamic Tangent Planner (D-TIG) Algorithm Process

Unlike the S-TIG algorithm, the D-TIG is suitable for two situations: one in a partially known environment with unexpected obstacles and the other in a completely unknown environment.

4.2.1. Dynamic Tangent Planner In a Partially Known Environment

In such scenarios, the UAV module initially employs the static planner to calculate the path. It then relies on UAV sensors to detect changes in the environment or obstacles' positions. If an obstacle appears or changes its coordinates, the

Algorithm 1 S-TIG planner algorithm

Input:
Start Node **S**, Target Node **T**
Output:
Path **Path**

Generate external ellipse for each obstacle using Equation 1
$currentSet \leftarrow \{S\}, closedSet \leftarrow \emptyset, treatedSet \leftarrow \emptyset$
while $currentSet$ is not empty **do**
  Get the node with the minimum heuristic value from $currentSet$ as the current node N.
  Add N to $closedSet$
  if current node N not equal to T **then**
    Initialise $Explored \leftarrow \emptyset$, $to\_Explore \leftarrow \{T\}$ and $Waypoints \leftarrow \emptyset$
    while $to\_Explore$ is not empty **do**
      Let the temporary target node T_{temp} be the first element in $to\_Explore$
      if the line segment from N to T_{temp} is a clear path and the angle is less than α, and T_{temp} is not in $treatedSet$ **then**
        Set N as the parent of T_{temp}
        Add T_{temp} to $Waypoints$
      else
        Generate two tangent lines of the first collided obstacle from the current node N
        Calculate each waypoint using virtual ellipse strategy using Equation 2 and add them to $to\_Explore$
        Delete T_{temp} from $to\_Explore$ and add it to $Explored$
      end if
    end while
    Add all $Waypoints$ nodes to $currentSet$ with their heuristic values using Equation 3.
  else
    Calculate the path from T to S using nodes in $closedSet$ with their parents.
    Reverse the path to obtain the path from S to T.
    return $Path$
  end if
end while

static path becomes infeasible, and the dynamic planner recalculates the subpath affected by this obstacle. This ensures that the initial path remains unchanged except for the collided subpaths. As shown in Figure 6, the static offline planner generates an initial feasible path from point S to T, which is $S \rightarrow W_2 \rightarrow W_1 \rightarrow W_5 \rightarrow T$ in our example. The UAV begins its mission and travels from point S along the path toward the target point T. Integrated sensors continuously gather real-time information about the environment and detect changes. In the figure, an unexpected obstacle (B12) is detected by the UAV, making the subpath $W_5 \rightarrow T$ collide with B12. This obstacle renders the initial static path infeasible. Here, the dynamic planner's role comes into play: it replans only the collided subpath remains $W_5 \rightarrow T$ using the same static planner technique, providing a new feasible subpath $W_5 \rightarrow W_{12'} \rightarrow T$ to replace the infeasible one.

This operation continues until the UAV safely reaches the target point T. The main steps of the dynamic tangent planner in a partially known environment are as follows:

1) Running S-TIG: Use the Start point S and the target point T to Run the static tangent planner in Algorithm 1
2) Iterate the static planner path nodes

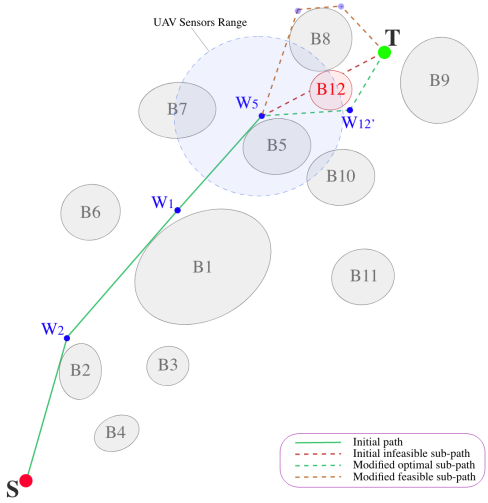


Figure 6. Generated path using dynamic path planner in a partially-known environment

3) UAV Moving: Move UAV to the next node
4) Environment Checking: Check if there are changes in the environment within the current range. If so, the current node N becomes the new start Node N and reruns the static planner; otherwise, continue to the next node in the path.

Algorithm 2 D-TIG planner algorithm in partially known environment

Input:
Start Node **S**, Target Node **T**, Sensor Range radius **R**
Output:
Path **Path**

Run Static Path Planner Using Algorithm 1 and return the initial path set P without the start point S
while true **do**
  $N \leftarrow$ the first element from P.
  Delete N from P
  if N is out of Range R **then**
    $I \leftarrow$ the intersection between the sensor range perimeter R and the line segment from the UAV's current position and N
    $N \leftarrow I$
  end if
  Move UAV to N
  if N reaches the target point T **then**
    Break the loop
  end if
  Update Environment information
  if the environment has changed **then**
    Run Static Path Planner Using Algorithm 1 from N as a new start node and return the updated path P without N
  end if
end while

5) Target Check: Stop the algorithm if the target point is reached; otherwise, go to Step 2)

D-TIG planner algorithm in a partially-known environment pseudocode is shown in Algorithm 2:

4.2.2. Dynamic Tangent Planner in a Completely Unknown Environment

In the second situation, the environment is completely unknown except for the starting and target points. In this scenario, the algorithm plans the path only for the sub-environment detected by UAV sensors, using the S-TIG planner in each update while considering environmental changes. There are two cases when the environment is unknown: firstly, when sensors capture some obstacles within their limited range, and secondly, when no obstacles are in range. As shown in Figure 7a, only the starting point S and the target point T are known. Initially, the UAV captures obstacle positions in the sub-environment depending on the sensor's range (limit distance). The algorithm employs the static planner to avoid the first collided Obstacle $B1$ by creating tangent lines and employs the heuristic function to generate a new waypoint $W1$. While the target point is not yet reached, the algorithm generates a direct line to T. The problem is that the environmental information is missing because T is out of range. In this case, the algorithm creates a maximum range waypoint within the range perimeter. These waypoints are used when the next node is out of range. The maximum range waypoint is defined as the intersection between the sensor perimeter and the direct line from the previous waypoint $W1$ to T. Additionally, the UAV moves to the latest maximum range waypoint, which is $W2$ as shown in Figure 7b. In this case, the sensors detect no intersecting obstacles between $W2$ and the range perimeter. Consequently, the D-TIG planner creates another maximum range waypoint $W3$ and makes the UAV move to it, gathering environmental information along the way. As depicted in Figures 7c and 7d, the D-TIG planner continues avoiding obstacles and creating maximum range waypoints if necessary until the target point is reached. The final planned path is in Figure 7e.

The main steps of the dynamic tangent planner for completely unknown are as follows:

1) Obstacele Modeling: Use Equation 1 to model the obstacles in the sub-environment.
2) Initialization: Begin by initializing sets and variables, including *currentSet* for unexplored nodes, *treatedSet* for calculated waypoints, and T_{temp} as a temporary target node.
3) Exploration Loop Initialization: Begin a loop to explore until the *currentSet* is empty.
4) Select Current Node: Select the node with the minimum heuristic value from the *currentSet* as the current node N.
5) Check Range: If the current node N is exactly in the range perimeter R, update the path, move UAV, and update environment information. Then reinitialize *currentSet* with N.
6) Continue Exploration: If the *currentSet* is not empty, continue exploration.
7) Check Current Node: If the current node N is not reached the target node T, proceed with waypoint exploration.
8) Waypoint Exploration: Explore waypoints from the current node N to the target node T.

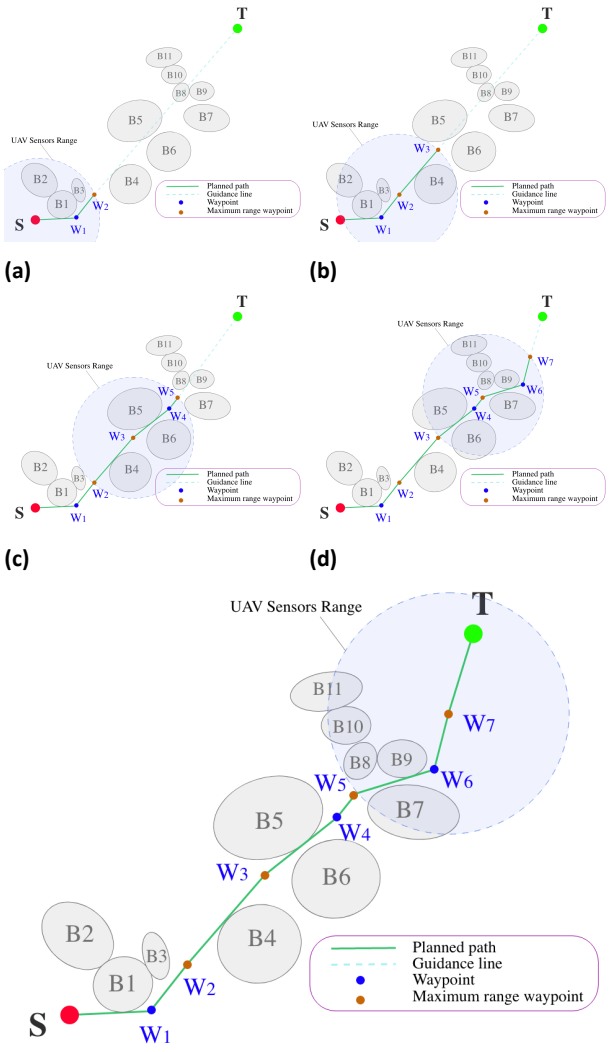


Figure 7. D-TIG planner algorithm steps in an unknown environment

9) Check Clear Subpath: Check if the line segment to the temporary current node N_{temp} is clear and if it has not been treated before.
10) Add Waypoints: Calculate heuristic values for each waypoint using Equation 3 and add them to *currentSet*.
11) Check Target Node: If the current node N is the target node T, terminate the algorithm and extract the final subpath.

D-TIG planner algorithm in an unknown environment pseudo-code is shown in Algorithm 3:

4.3. Path Smoothing Technique

Generated paths using static or dynamic planners often include a series of connected waypoints, but due to their complexity and flight dynamic requirements, they may not be suitable for UAVs. Several research papers have proposed techniques such as Bezier, spline, and Dubin's Curves to smooth these paths and address this issue [40].

However, these methods are effective only in open spaces or situations where the path waypoints are not too close to obstacles presented in the environment, meaning there is a high risk of collision in cluttered and dense environments. As depicted in Figure 8, the initial path is not yet smoothed. A new smoothed path is obtained by applying quadratic Bezier curves to the path, which unfortunately still collides with some obstacles. Figure 9 shows examples of collision areas highlighted within red dashed circles.

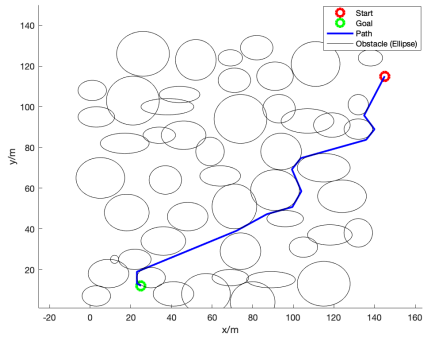


Figure 8. Example of a generated path without smoothing

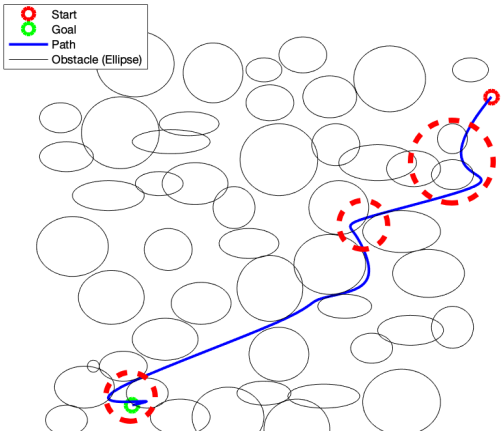


Figure 9. A smoothed path using a quadratic Bézier curve with collision

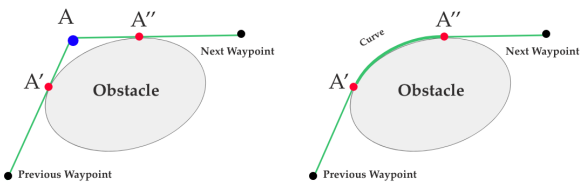
To solve this, the paper proposes using a quadratic Bezier curve technique with every three successive nodes from the start to the target node. Each waypoint, except the start and target nodes, is considered a turn. The idea is to create a curve that smooths these turns, making it suitable for UAVs. Firstly, we extract each waypoint on the path and create two temporary waypoints before and after each waypoint and handle them as the first and last control points for the quadratic Bézier curve, respectively. Secondly, the algorithm uses the quadratic Bezier curve to smooth these waypoints and combines the collected curves in a final smoothed path.

As shown in Figure 10a, the first temporary waypoint A', the waypoint A, and the second temporary point A'' are utilized as control points. After applying the quadratic bezier curve, a smooth subpath is obtained (Figure 10b). This process continues until the target node is reached.

Algorithm 3 D-TIG planner algorithm in unknown environment

Input:
Start Node **S**, Target Node **T**, Sensor Range radius **R**
Output:
Path **Path**

$N \leftarrow S$
while T is out Of Range R **do**
  Initialize $currentSet \leftarrow \{N\}$, $treatedSet \leftarrow \emptyset$, $T_{temp} \leftarrow T$
  while $currentSet$ is not empty **do**
    Get the node with the minimum heuristic value from *currentSet* as the current node N
    Delete N from *currentSet*
    if N is exactly in the range perimeter R **then**
      Calculate in range subpath starting from N
      Move UAV following the obtained subpath
      Update Environment Information
    end if
    Initialize $Explored \leftarrow \emptyset$, $to\_Explore \leftarrow \{N\}$, $Waypoints \leftarrow \emptyset$
    while $to\_Explore$ is not empty **do**
      Let temporary target node T_{temp} be the first element in $to\_Explore$
      Delete T_{temp} from $to\_Explore$ and add it to $Explored$
      if the line segment from N to T_{temp} is a clear path and the angle is less than α, and T_{temp} is not in *treatedSet* **then**
        if T_{Temp} is out of range R **then**
          Move T_{Temp} in range using the intersection between R perimeter and the line NT_{Temp}
        end if
        Set N as the parent of T_{temp}
        Add T_{temp} to $Waypoints$
      else
        Generate two tangent lines of the first collided obstacle from N.
        Calculate each waypoint using virtual ellipse strategy using Equation 2 and add them to $to\_Explore$
        Delete T_{temp} from $to\_Explore$ and add it to $Explored$
      end if
    end while
    Add all waypoints to *currentSet* with their heuristic values using Equation 3
  end while
end while



(a) (b)

Figure 10. TIG smoothing steps

5. Experimental Setup

This section evaluates the path planning algorithm's performance and effectiveness in static and dynamic environments. The paper conducts multiple simulation experiments to compare it with other popular path planning algorithms, such as A*, PRM, RRT*, Tangent Graph, and APPATT(SETG-TG) in static environments and APF and APPATT(DETG-TG) [41] in dynamic environments. These algorithms are coded using MATLAB R2021b on a MacBook Pro 2020 with an Intel i5 processor and 8GB of memory. During the experiment, the path planning process must comply with specific constraints, such as minimizing the path length, execution time, and enforcing a minimum turning radius, to guarantee a low number of curves and an efficient trajectory.

As mentioned in Section 2, this study did not consider the height of obstacles presented in the environment and assumed the map to be two-dimensional. Additionally, to minimize the computation time of the Tangent Graph algorithm, we calculate the waypoints of the algorithm based on the intersection points of the tangent lines from the current and end positions. Moreover, we assume the UAV range distance is 60m.

5.1. Environmental Modeling

UAVs may encounter multiple scenarios in real-world operations. For example, they may need to avoid a single obstacle in some cases, while in others, they must navigate through a densely obstructed area. Similarly, they may have long travel distances in some missions and short travel distances in others.

To address these variations, this paper examines four experiments, as follows:

Short Map vs. Large Map

In these two experiments, we evaluate whether the algorithm can generate feasible paths in environments of different sizes. The short map in this paper measures 500 × 500 m, while the large map is 1000 × 1000 m. These experiments ensure that the algorithms can balance between path and turning angles optimality and computational efficiency.

Sparse vs. Dense Maps

The number of obstacles in a path planning process plays a crucial role. The first experiment involves planning a path in a densely obstructed area, where finding a feasible route for the UAV is significantly challenging. In contrast, the sparse map presents a less demanding environment with fewer obstacles. These experiments assess the algorithm's performance across different urban conditions, highlighting its adaptability and efficiency in navigating environments of varying spatial complexity. In this paper, we define a sparse area as one containing 10% of obstacles, while a dense area contains more than 60% of obstacles.

5.2. Evaluation Metrics

After a literature review of many studies, the proposed algorithms employ only path length minimization as an objective function to maximize the operational range of UAVs. However, this approach can result in paths with sharp turns and unnecessary nodes, negatively impacting energy consumption and path efficiency. Hence, this paper adopts objective functions to minimize the path length, turning radius, and execution time. This approach aims to generate smoother, more adaptable paths that optimize mission efficiency while ensuring operational feasibility.

Short Path Objective Function

Minimizing flight distance is crucial for a UAV path planning algorithm since it directly reduces travel time and energy consumption. Therefore, an objective function is defined to prioritize the shortest path length as follows:

$$minD(x,y) = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (4)$$

where n the length of the path, (x_i, y_i) and (x_{i+1}, y_{i+1}) are the abscissa and ordinates of the i-th node, and (x_{i+1}, y_{i+1}) $(i+1)$-th node respectively. The Euclidean distance is used to calculate the path's length.

Minimum Total Turning Angles Objective Function

Each turning angle impacts the overall smoothness of the path. The lower the sum of turning angles, the smoother the path we get. In contrast, a higher sum results in high energy consumption and requires more motion time. Therefore, an objective function to minimize the turning angles is adopted as follows:

$$\min T_r(\theta) = \sum_{i=1}^{n-2} \theta_i. \quad (5)$$

$$\theta_n = \left| \arctan\left(\frac{\frac{y_{n+1}-y_n}{x_{n+1}-x_n} - \frac{y_{n+2}-y_{n+1}}{x_{n+2}-x_{n+1}}}{1 + \frac{y_{n+1}-y_n}{x_{n+1}-x_n} \cdot \frac{y_{n+2}-y_{n+1}}{x_{n+2}-x_{n+1}}} \right) \right|. \quad (6)$$

Where θ_n is the angle at the nth node, which is calculated based on its previous node $n-1$, and its next node $n+1$.

Shortest Algorithm Execution Time

The execution time of the algorithm also has a significant impact, especially in dynamic environments where UAVs need to adjust their flight paths to avoid obstacles or achieve mission objectives. Adopting a function to minimize algorithm execution time ensures that decisions are made swiftly, contributing to faster and more responsive navigation. The objective function is set as follows:

$$\min T(a) = t \quad (7)$$

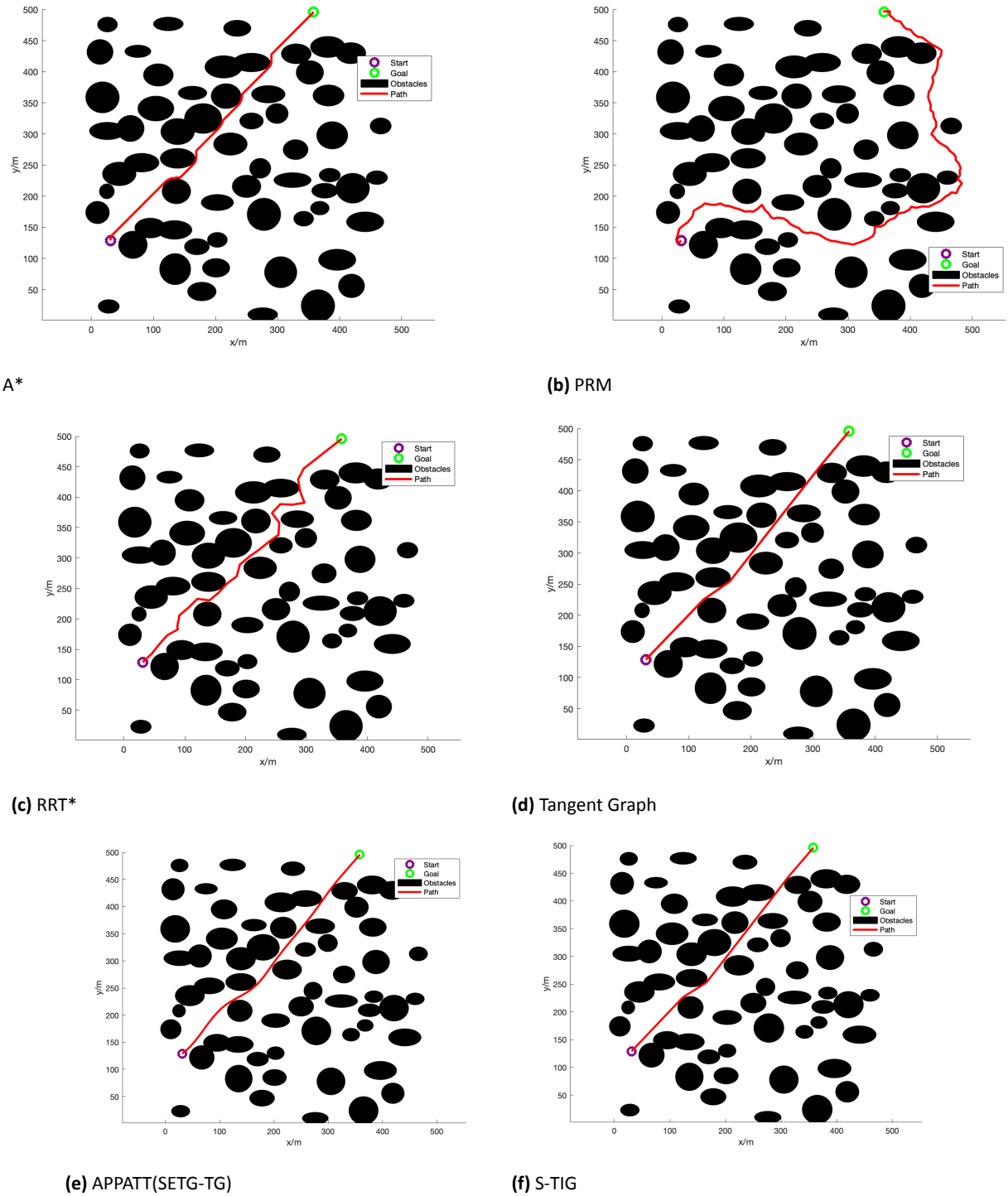
Where t denotes the algorithm a execution time.

5.3. Simulation Experiment

Static Path Planner Simulation Experiment

Based on the simulation results shown in Figures 11 to 14 and the summarized performance metrics in Table 2, it's evident that the S-TIG algorithm consistently outperforms A*, PRM, RRT*, Tangent Graph (TG), and APPATT algorithms across all experiments with randomly generated obstacles of varying numbers and sizes. Firstly, the static planner reduces path length by 5% compared to the A*, 6.98% compared to PRM and a substantial 22.80% compared to RRT*, 7.55% compared to APPATT algorithms, and approximately the same with the tangent graph planner with a difference of 0.18%. Additionally, the execution time of the S-TIG planner algorithm averages a reduction of 98.08%, 98.29%, 47% , 7% and 88.25% compared to A*, PRM, RRT*, Appatt, and Tangent Graph, respectively. In terms of the sum of turning angles, S-TIG demonstrates a reduction of 93.39% compared to A*, 65.15% compared to PRM, 84.27% compared to RRT*, 35.41% compared to APPATT while S-TIG produces the same turning angles as the tangent graph planner in most cases. These results indicate that the S-TIG algorithm consistently generates shorter paths with fewer nodes, requiring less time and producing smoother trajectories due to the minimized total turning angles. From the APPATT (SETG-TG) test results, the algorithm failed due to its structure and the type of environment, especially in C2, C7, and C12, while it also failed in all cases in dense environments.

The APPATT algorithm employs an intersection strategy, which leads to many failures, as represented in Figure 3. In contrast, S-TIG addresses this limitation



(a) A* **(b)** PRM

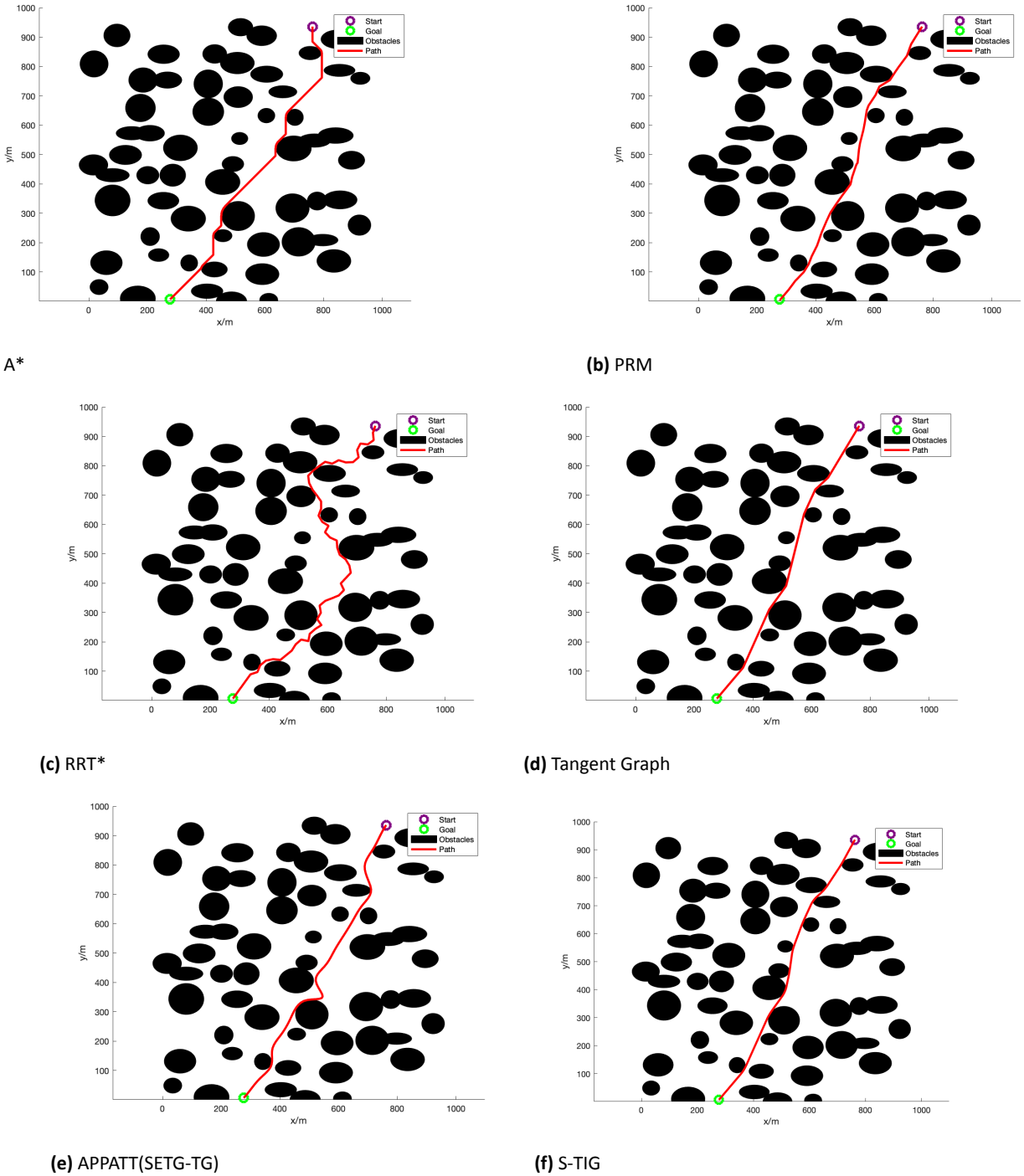
(c) RRT* **(d)** Tangent Graph

(e) APPATT(SETG-TG) **(f)** S-TIG

Figure 11. Generated paths in static environments on a short map (C1)

by adopting the waypoint creation technique. Additionally, APPATT does not implement a strategy for checking created waypoints when they become stuck in overlapping obstacles, which can result in an infinite loop when attempting to choose the best waypoint. Due to these limitations, APPATT failed the tests across five dense maps, whereas the TIG algorithm demonstrated strong capabilities in path generation within static environments.

The A* algorithm can produce an optimal path based on grids with 1m x 1m resolution. However, the quality of the generated path is not satisfactory. In addition, A* and PRM algorithms suffer from high computation times, especially in large areas. For example, in C6, A* needs 98 seconds to generate a feasible path, while PRM takes 5 seconds. RRT*, due to its randomness, generates a large number of turning angles, sometimes exceeding 40 radians. This algorithm cannot guarantee path optimality and smoothness for UAVs. This often results in UAVs having to take sharp turns or unnecessary nodes, which can impact the overall mission time. The tangent graph planner produces the shortest paths with the minimum sum of turning angles; however, this algorithm calculates the



(a) A* **(b)** PRM

(c) RRT* **(d)** Tangent Graph

(e) APPATT(SETG-TG) **(f)** S-TIG

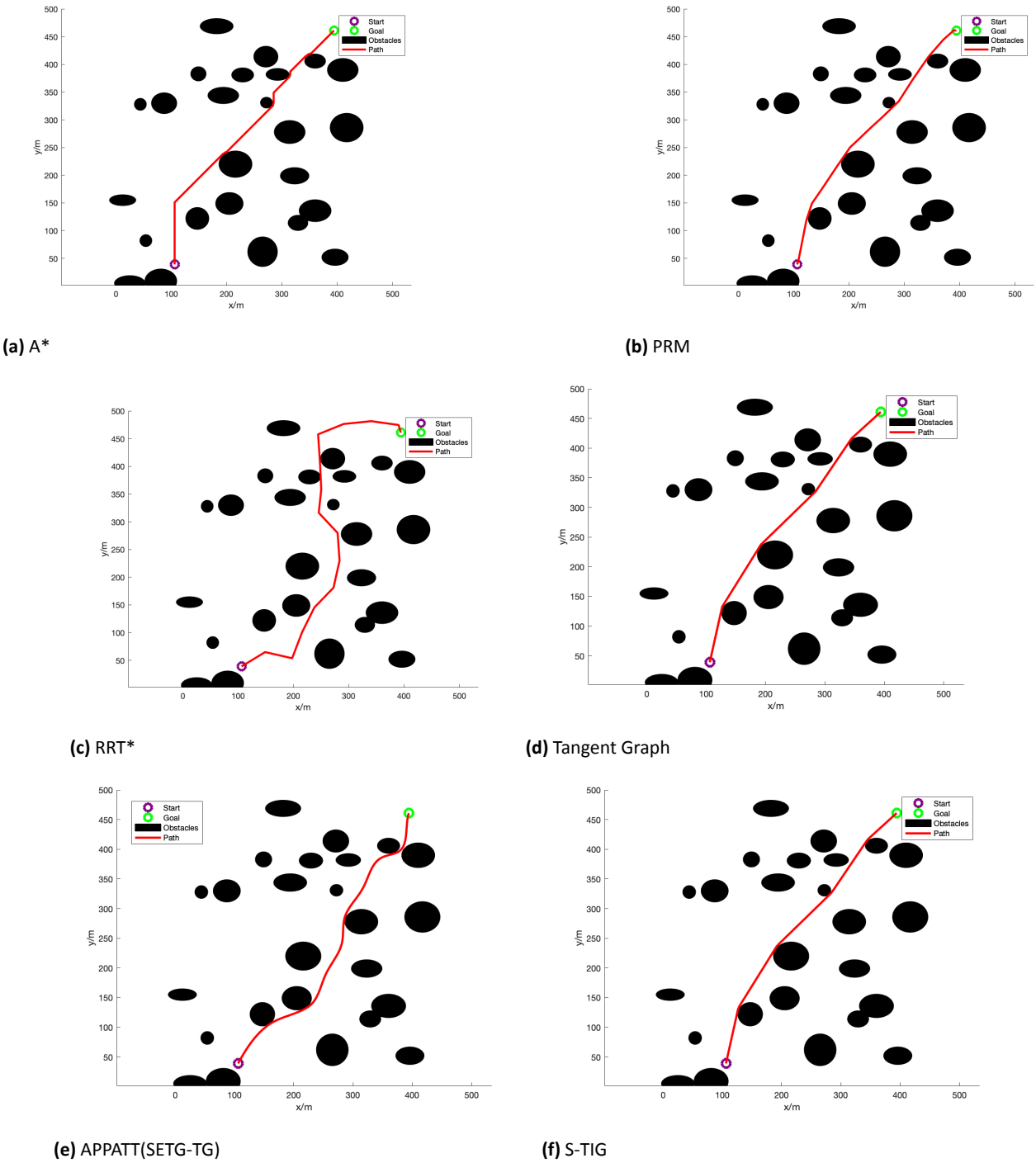
Figure 12. Generated paths in static environments on a large map (C5)

tangents for the entire map, leading to higher computation times, especially in dense areas.

In contrast, the S-TIG algorithm always evaluates the subpath quality using its heuristic rules before making a decision, which eliminates any sharp turns and redundant nodes and produces collision-free paths with shorter times in each of the cases in all scenarios. Next in line is Tangent Graph, followed by Appatt, A*, PRM, and RRT*. To sum up, the S-TIG static planner algorithm shows a superior path planning performance compared to the other five algorithms in terms of path length, time consumption, and turning angles in static environments.

Dynamic Planner Simulation Experiment In Unkonwn Environment

To test the effectiveness of the D-TIG planner in an unknown environment, this paper compares the algorithm with two dynamic planners: APF and APPATT. Figures 15 to 18 show the planned paths for four scenarios with varying obstacle sizes and positions. The test results, summarized in Table 3, show that D-TIG consistently reduced the planned path length by 20.55% compared to the APF algorithm and approximately 0.1% compared to APPATT in all cases.



(a) A* **(b)** PRM

(c) RRT* **(d)** Tangent Graph

(e) APPATT(SETG-TG) **(f)** S-TIG

Figure 13. Generated paths in static environments on a sparse map (C9)

Additionally, the algorithm's execution time is equal to that of APPATT, requiring only 0.01 seconds on average to plan a near-optimal route, unlike APF, which demands significantly more time, sometimes up to 0.14 seconds in high-dimensional environments (e.g., C21). Moreover, the D-TIG dynamic planner produces fewer turning angles than the other algorithms, reducing them by approximately 87% compared to APF and 34.11% compared to APPATT.

Across all test scenarios, the D-TIG algorithm successfully generated paths, whereas the other algorithms failed in multiple cases, such as C20, C24, C29, C30, C31, and C32. In summary, the D-TIG planner is effective in unknown environments, generating shorter paths in less time and with fewer unnecessary turns than the other algorithms.

Dynamic Planner Simulation Experiment In Partially Known Environment With Unexpected Obstacles

Another simulation experiments were conducted to test the effectiveness of the D-TIG algorithm in environments with unexpected obstacles. Figures 19 to 21 illustrate the initially planned path in red before the environmental change, and the green color represents the corrected path after the UAV faced unexpected obstacles, colored in orange. The D-TIG

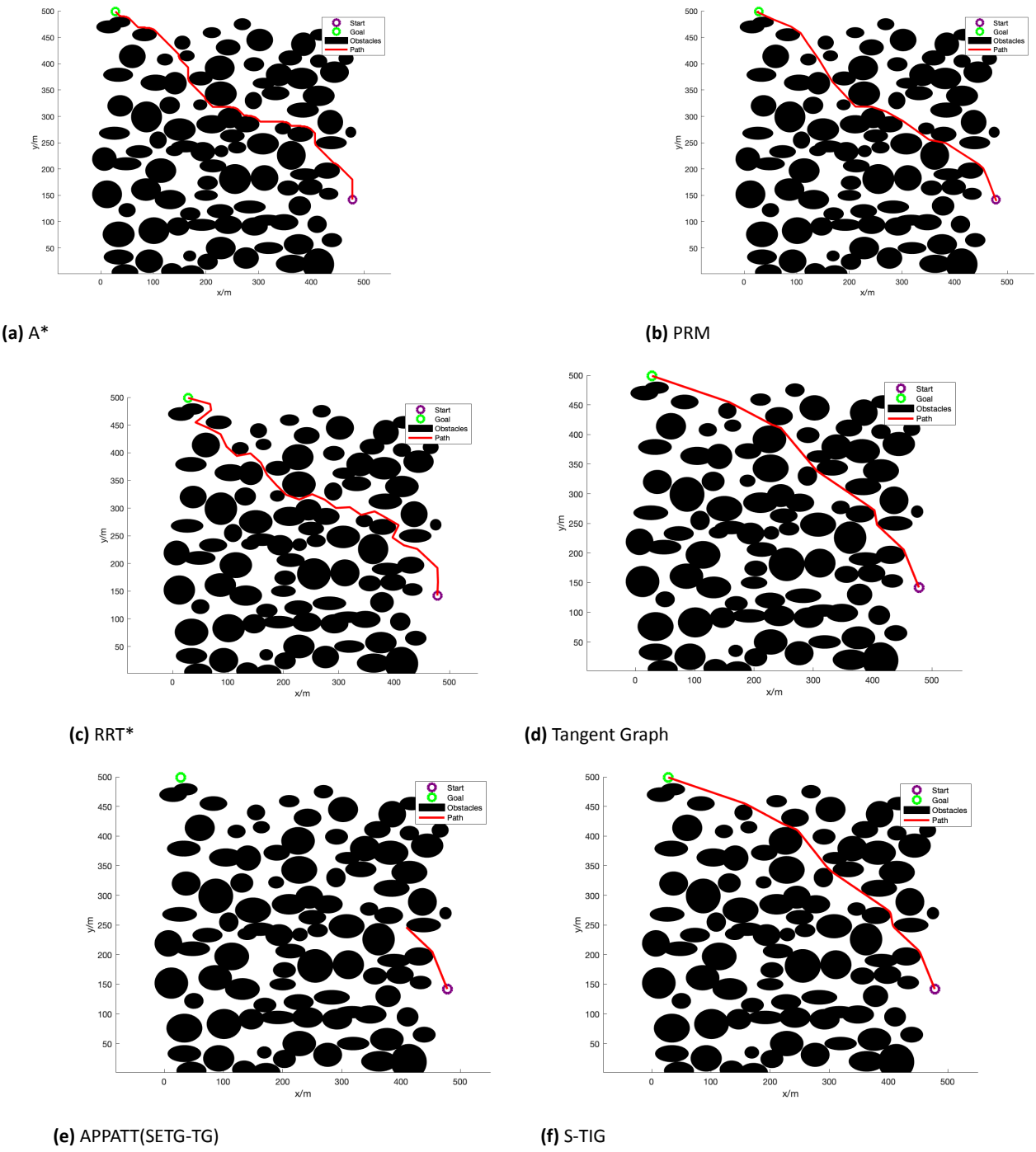


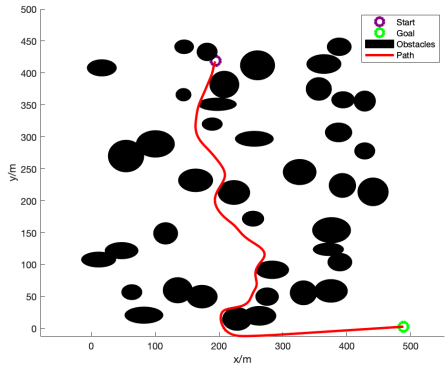
Figure 14. Generated paths in static environments on a dense map (C13)

was compared with the APPATT algorithm since it is designed to calculate new paths based on unexpected obstacles, and the results were collected in Table 4. The results demonstrate that the D-TIG planner can produce shorter, smoother paths in less time. Additionally, the algorithm reduces the planned path length by 5.63% across all test cases and decreases the path turning radius by 25.55% compared to the APPATT algorithm. Moreover, the execution time of both algorithms is approximately the same, which both not exceed 0.08 seconds for replanning, which is good to allow UAVs to make decisions more quickly. Furthermore, the APPATT algorithm fails to generate feasible paths in multiple cases (e.g C21 and C26), which UAVs cannot rely on it in such situations because the risk of collision is too high in real-time scenarios.

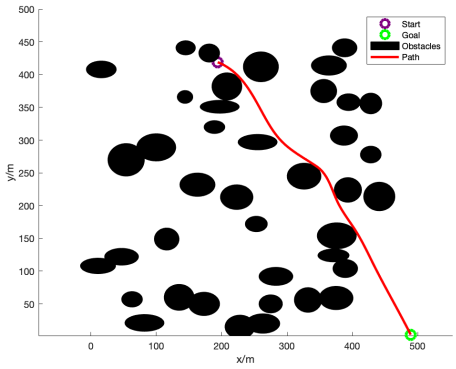
The APPATT algorithm also fails to find feasible paths in unknown environments, as shown in Table 4. Similarly, it cannot find a path in a dense environment with pop-up obstacles. In contrast, the D-TIG algorithm successfully handles dense maps with pop-up obstacles, as shown in Figure 22. These improvements can be attributed to the fundamental differences between the two algorithms.

Table 2. Comparison of different static path planning algorithms across four scenarios

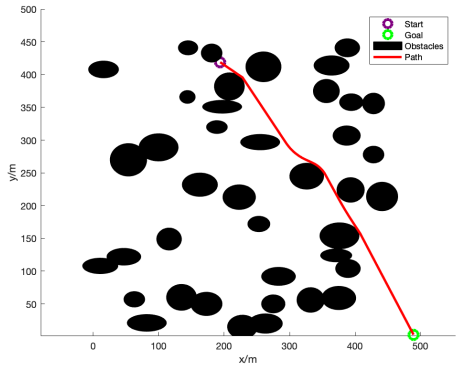
Map Type	Case	Path Length						Time						Turning Radius					
		S-TIG	A*	PRM	RRT	APPATT	TG	S-TIG	A*	PRM	RRT	APPATT	TG	S-TIG	A*	PRM	RRT	APPATT	TG
Short	C1	493.84	510.06	949.33	566.49	495.29	493.59	0.01	8.44	3.65	0.18	0.09	0.20	0.53	32.20	58.35	14.96	1.45	0.53
	C2	511.89	533.16	661.20	595.68	N/A	511.60	0.07	11.72	3.65	0.27	N/A	2.73	0.68	32.20	42.18	16.84	N/A	0.46
	C3	520.53	545.25	587.87	931.31	531.38	514.25	0.06	9.26	3.68	0.64	0.21	2.81	2.81	49.48	30.59	36.17	1.99	1.86
	C4	527.59	557.87	645.85	840.31	584.34	528.50	0.06	10.38	3.36	0.14	0.06	2.02	2.34	46.33	37.94	26.21	2.72	2.34
Large	C5	1063.13	1156.40	1071.65	1429.14	1135.71	1063.76	0.11	73.00	5.25	0.77	0.15	73.48	1.97	55.76	6.43	34.66	9.28	1.72
	C6	1004.27	1022.01	1040.68	1313.74	1064.00	1004.14	0.02	92.65	5.09	0.35	0.10	0.14	0.16	10.21	7.20	34.58	1.87	0.14
	C7	1241.35	1289.83	1253.50	1558.46	N/A	1238.23	0.05	80.09	5.88	0.78	N/A	4.13	1.49	69.90	6.69	44.56	N/A	1.07
	C8	1001.01	1054.11	1019.10	1254.34	1925.46	999.66	0.09	72.85	5.25	0.39	0.14	8.57	1.67	52.62	5.08	35.32	10.29	1.42
Sparse	C9	522.01	545.98	524.81	706.33	543.33	522.24	0.04	16.00	9.85	0.11	0.01	0.83	1.02	18.06	2.30	8.14	5.55	1.02
	C10	515.36	548.42	526.48	714.74	515.30	515.38	0.03	17.47	9.02	0.02	0.03	0.04	0.25	14.92	4.76	12.66	0.24	0.24
	C11	514.74	524.83	525.51	643.16	524.88	514.81	0.01	12.82	9.44	0.11	0.05	0.03	0.37	13.35	6.32	16.09	0.69	0.37
	C12	511.61	520.28	525.26	613.53	N/A	509.00	0.06	16.69	9.36	0.09	N/A	0.16	2.05	16.49	6.08	15.28	N/A	1.14
Dense	C13	601.13	652.94	613.57	713.55	N/A	602.23	0.32	6.71	10.90	0.34	N/A	57.50	2.97	69.90	9.38	17.87	N/A	2.96
	C14	509.77	562.94	515.49	604.97	N/A	509.69	0.29	6.73	9.25	0.47	N/A	5.26	2.46	62.05	6.15	12.70	N/A	2.14
	C15	686.06	698.61	638.75	868.91	N/A	686.05	0.68	6.16	11.60	0.42	N/A	62.59	18.90	68.33	13.63	23.39	N/A	18.90
	C16	572.76	644.14	574.83	711.90	N/A	568.13	0.32	4.35	10.10	0.34	N/A	61.27	11.80	52.62	7.33	17.47	N/A	3.80



(a) APF

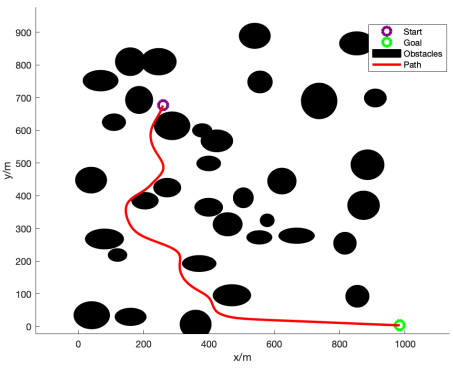


(b) APPATT(DETG-TG)

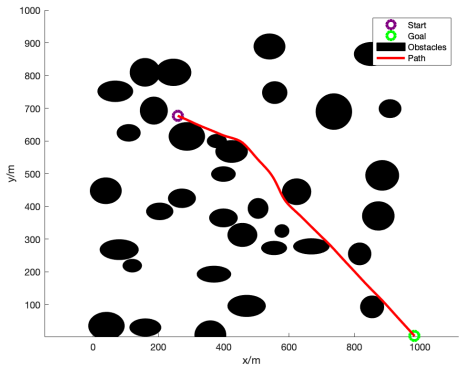


(c) D-TIG

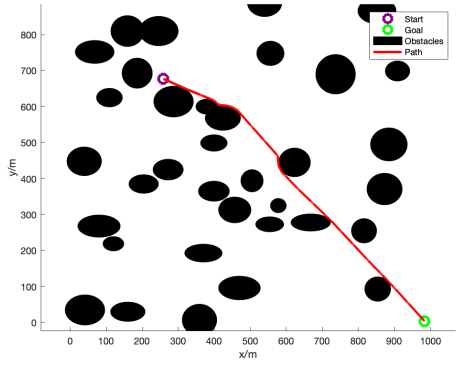
Figure 15. Generated paths in unknown environment on a short map (C17)



(a) APF

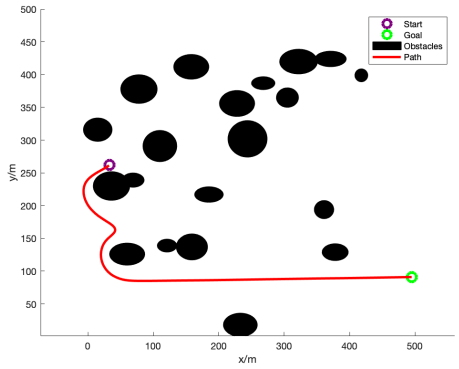


(b) APPATT(DETG-TG)

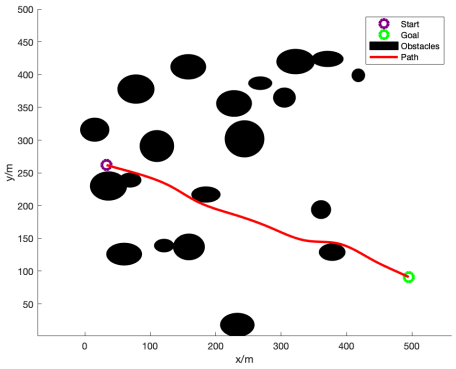


(c) D-TIG

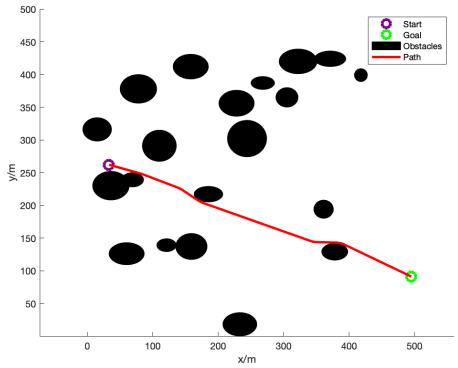
Figure 16. Generated paths in unknown environment on a large map (C21)



(a) APF

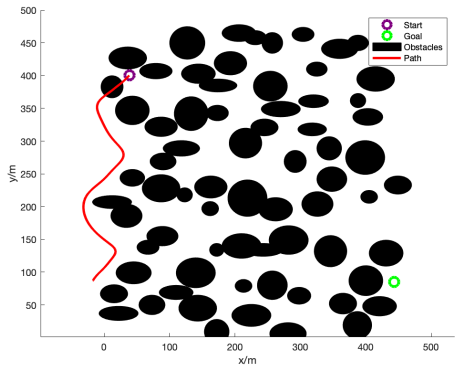


(b) APPATT(DETG-TG)

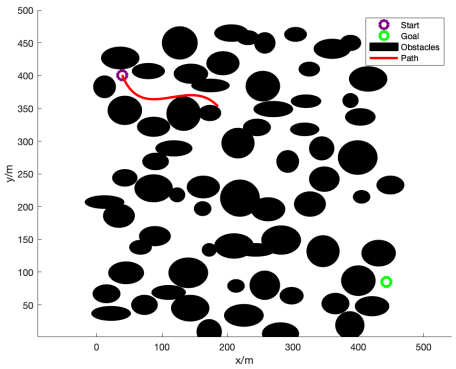


(c) D-TIG

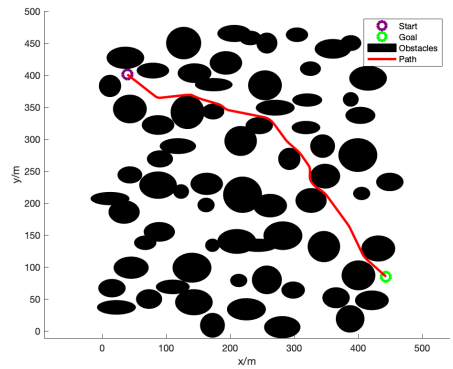
Figure 17. Generated paths in unknown environment on a sparse map (C25)



(a) APF



(b) APPATT(DETG-TG)



(c) D-TIG

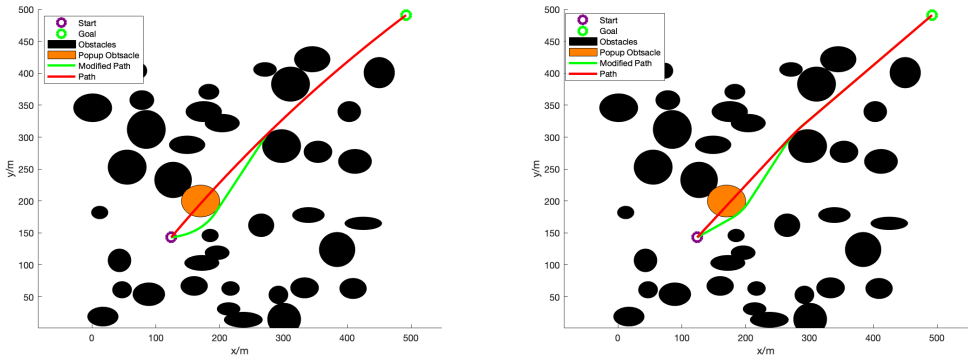
Figure 18. Generated paths in unknown environment on a dense map (C29)

Table 3. Comparison of different dynamic path planning algorithms across different scenarios

Map Type	Case	Path Length			Time			Turning Radius		
		D-TIG	APF	APPATT	D-TIG	APF	APPATT	D-TIG	APF	APPATT
Short	C17	520.46	844.00	519.67	0.03	0.11	0.04	1.72	35.01	2.22
	C18	485.87	718.00	491.07	0.03	0.08	0.03	0.64	147.98	3.45
	C19	509.74	531.00	510.58	0.01	0.07	0.01	0.82	63.21	1.57
	C20	625.68	885.00	N/A	0.04	0.12	N/A	3.08	13.03	N/A
Large	C21	1010.41	1406.00	1010.01	0.07	0.14	0.05	1.38	10.65	1.90
	C22	1125.06	1419.00	1124.14	0.06	0.14	0.04	1.99	9.66	2.28
	C23	1036.91	1401.00	1033.27	0.04	0.14	0.03	3.52	1200.42	2.86
	C24	990.64	N/A	994.53	0.03	N/A	0.04	1.95	N/A	3.40
Sparse	C25	495.73	687.00	496.27	0.02	0.06	0.01	1.08	7.26	1.65
	C26	516.49	532.00	518.80	0.01	0.04	0.01	1.08	2.98	2.17
	C27	489.01	552.00	489.39	0.01	0.05	0.01	0.63	184.25	1.18
	C28	508.54	536.00	513.59	0.01	0.05	0.01	2.21	8.04	3.59
Dense	C29	558.54	N/A	N/A	0.05	N/A	N/A	6.51	N/A	N/A
	C30	562.17	N/A	N/A	0.06	N/A	N/A	5.92	N/A	N/A
	C31	505.38	N/A	505.14	0.04	N/A	0.05	1.86	N/A	2.59
	C32	564.15	N/A	N/A	0.06	N/A	N/A	7.55	N/A	N/A

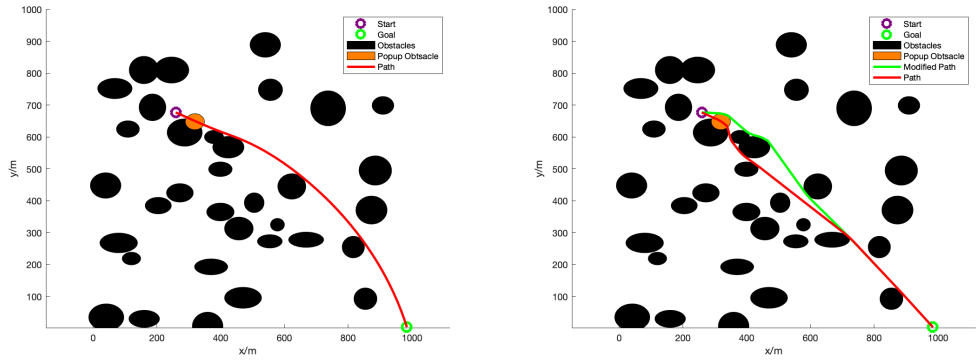
Table 4. Comparison of Different Dynamic Path Planning Algorithms Across Different Scenarios

Map Type	Case	Path Length		Time		Turning Radius	
		D-TIG	APPATT	D-TIG	APPATT	D-TIG	APPATT
Short	C17	520.46	571.56	0.02	0.03	0.55	4.12
	C18	496.99	502.81	0.01	0.01	1.62	1.36
	C19	513.46	520.09	0.005	0.007	0.97	1.47
Large	C21	1017.12	N/A	0.02	N/A	2.03	N/A
	C22	1130.68	1244.94	0.08	0.01	1.56	4.44
	C23	1027.61	1044.94	0.03	0.01	2.70	3.99
Sparse	C25	494.90	522.74	0.001	0.003	0.97	4.27
	C26	534.93	N/A	0.002	N/A	2.2	N/A
	C27	495.09	561.69	0.008	0.002	1.48	4.42



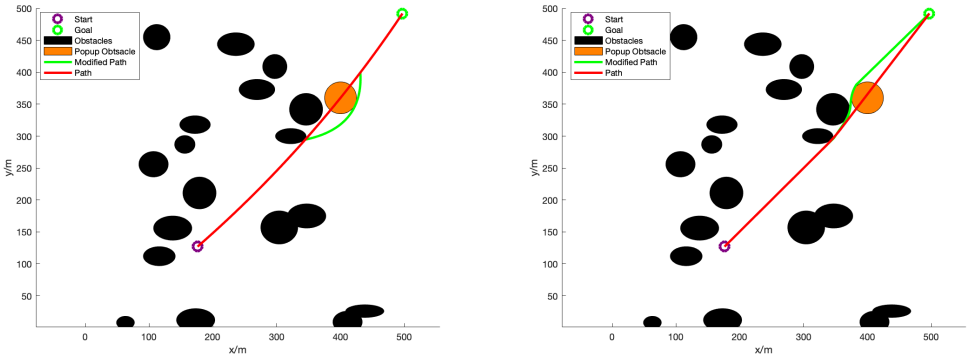
(a) APPATT **(b)** D-TIG

Figure 19. Generated paths in a partially known environment with pop-up obstacles on a short map (C19)



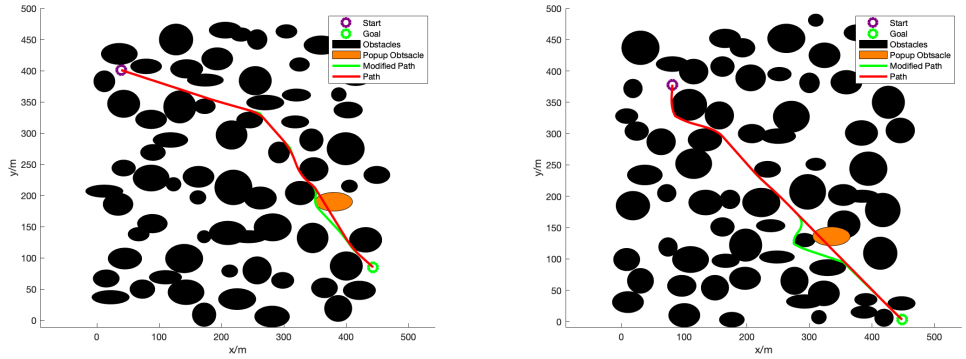
(a) APPATT **(b)** D-TIG

Figure 20. Generated paths in a partially known environment with pop-up obstacles on a long map (C20)

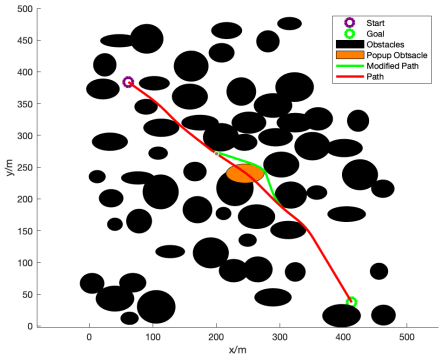


(a) APPATT **(b)** D-TIG

Figure 21. Generated paths in a partially known environment with pop-up obstacles on a sparse map (C25)



(a) D-TIG (C25) **(b)** D-TIG(C26)



(c) D-TIG(C27)

Figure 22. Generated paths using D-TIG in a partially known environment with pop-up obstacles on dense maps

The APPATT algorithm is based on the tangent intersection with the goal position to create feasible waypoints in the search space. In contrast, the dynamic planner (D-TIG) uses a waypoints generation technique, which produces paths closer to pop-up obstacles while maintaining a safe distance to avoid collisions. Based on these results, it is evident that the D-TIG is more effective in a partially known environment with unexpected obstacles compared to the APPATT algorithm.

6. Conclusion

In this paper, we present the Tangent Intersection Guidance (TIG) algorithm, a novel approach for UAV path planning in both static and dynamic environments. The algorithm generates two sub-paths for each elliptic obstacle and selects the optimal one based on a heuristic rule. This process is iteratively repeated until the target point is reached. The static planner (S-TIG) is employed in known environments, and our test results demonstrate that S-TIG generates paths that are, on average, 11% shorter compared to other static methods while reducing the number of turns by 70% and maintaining a low computation time of around 0.1 seconds, even in higher-dimensional environments. Additionally, the dynamic planner (D-TIG) functions as a local planner in partially known environments with unexpected obstacles and completely unknown environments, outperforming existing real-time algorithms by achieving fast replanning times under 0.08 seconds only, while reducing path length and turning angles by approximately 9% and 50%, respectively, compared to other dynamic algorithms. Despite its advantages, TIG has some limitations, particularly in guaranteeing path length optimality in dynamic environments. Moreover, the scalability of TIG to high-dimensional spaces, especially full 3D path planning, requires additional exploration. Future research will focus on extending the TIG framework to three-dimensional environments, optimizing heuristic cost functions to ensure admissibility, and refining path smoothing techniques for a fairer comparison with other algorithms. In conclusion, the Tangent Intersection Guidance algorithm represents a significant step forward in UAV path planning technology. By integrating heuristic-based decision-making and Bézier curve smoothing, TIG enhances UAV navigation efficiency and safety in complex environments. Future studies will focus on addressing the identified limitations, improving computational scalability, and benchmarking against recent advancements in visibility graph-based path planning methods.

7. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8. Data Availability

The data will be provided upon request.

AUTHORS

Hichem Cheriet* – Phd Student,SIMPA Laboratory, Université des Sciences et de la Technologie d'Oran, Bir El Djir 31000, Oran, Algeria, e-mail: hichem.cheriet@univ-usto.dz, N/A.

Badra Khellat Kihel – Department of Economics, Oran 2 Mohamed Ben Ahmed University, Bir El Djir 31000, Oran, Algeria, e-mail: khellat_badra@yahoo.fr, N/A.

Samira Chouraqui – Computer Science Department, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, Bir El Djir 31000, Oran, Algeria, e-mail: samirachouraqui178@gmail.com, N/A.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by the research project "Modeling and Control of Aerial Manipulators" N° C00L07UN310220230004.

References

[1] J. ru Fan, D. guang Li, R. peng Li, and Y. Wang, "Analysis on MAV/UAV Cooperative Combat Based on Complex Network", *Defence Technology*, vol. 16, 2020, 150–157, 10.1016/J.DT.2019.09.002.

[2] A. Sehrawat, T. A. Choudhury, and G. Raj, "Surveillance Drone For Disaster Management and Military Security", *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, vol. 2017-January, 2017, 470–475, 10.1109/CCAA.2017.8229846.

[3] J. Dong, K. Ota, and M. Dong, "UAV-Based Real-Time Survivor Detection System in Post-Disaster Search and Rescue Operations", *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 2, 2021, 209–219, 10.1109/JMASS.2021.3083659.

[4] O. M. Bushnaq, D. Mishra, E. Natalizio, and I. F. Akyildiz. "Unmanned Aerial Vehicles (UAVs) for Disaster Management", 1 2022.

[5] S. Sawadsitang, D. Niyato, P. S. Tan, and P. Wang, "Joint Ground and Aerial Package Delivery Services: A Stochastic Optimization Approach", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, 2018, 2241–2254, 10.1109/TITS.2018.2865893.

[6] D. Sacramento, D. Pisinger, and S. Ropke, "An Adaptive Large Neighborhood Search Metaheuristic for the Vehicle Routing Problem with Drones", *Transportation Research Part C: Emerging Technologies*, vol. 102, 2019, 289–315, 10.1016/J.TRC.2019.02.018.

[7] R. Kumar, N. Kori, and V. K. Chaurasiya, "Real-Time Data Sharing, Path Planning and Route

Optimization in Urban Traffic Management", *Multimedia Tools and Applications*, vol. 82, 2023, 36343–36361, 10.1007/S11042-023-15148-9.

[8] X. Tian, Y. Jia, X. Luo, and J. Yin, "Small Target Recognition and Tracking Based on UAV Platform", *Sensors*, vol. 22, no. 17, 2022, 6579, 10.3390/s22176579.

[9] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, and D. Wu, "Offline and Online Search: UAV Multiobjective Path Planning Under Dynamic Urban Environment", *IEEE Internet of Things Journal*, vol. 5, no. 2, 2018, 546–558, 10.1109/JIOT.2017.2717078.

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, 1968, 100–107, 10.1109/TSSC.1968.300136.

[11] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, vol. 1, 1959, 269–271, 10.1007/BF01386390.

[12] Y. V. Pehlivanoglu, "A New Vibrational Genetic Algorithm Enhanced with a Voronoi Diagram for Path Planning of Autonomous UAV", *Aerospace Science and Technology*, vol. 16, 2012, 47–55, 10.1016/J.AST.2011.02.006.

[13] N. S. Abu, W. M. Bukhari, M. H. Adli, S. N. Omar, and S. A. Sohaimeh, "A Comprehensive Overview of Classical and Modern Route Planning Algorithms for Self-Driving Mobile Robots", *Journal of Robotics and Control (JRC)*, vol. 3, 2022, 666–678, 10.18196/JRC.V3I5.14683.

[14] H. Rohnert, "Shortest Paths in the Plane with Convex Polygonal Obstacles", *Information Processing Letters*, vol. 23, 1986, 71–76, 10.1016/0020-0190(86)90045-1.

[15] H. Chen, K. Chang, and C. S. Agate, "UAV Path Planning with Tangent-Plus-Lyapunov Vector Field Guidance and Obstacle Avoidance", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, 2013, 840–856, 10.1109/TAES.2013.6494384.

[16] Y. Liu, Q. Wang, H. Hu, and Y. He, "A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown Unstructured Outdoor Scenes", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, 2019, 2362–2372, 10.1109/TSMC.2018.2808471.

[17] X. Wu, L. Xu, R. Zhen, and X. Wu, "Bi-Directional Adaptive A* Algorithm Toward Optimal Path Planning for Large-Scale UAV under Multi-Constraints", *IEEE Access*, vol. 8, 2020, 85431–85440, 10.1109/ACCESS.2020.2990153.

[18] M. shun Yuan, T. le Zhou, and M. Chen, "Improved Lazy Theta* Algorithm Based on Octree Map for Path Planning of UAV", *Defence Technology*, vol. 23, 2023, 8–18, 10.1016/J.DT.2022.01.006.

[19] B. Banday, V. Kasula, N. Surwade, S. R. Nagrare, and D. Ghose. "UAV Path Planning for Cave Exploration Using Tangent Based Intersection Method", 1 2024.

[20] Y. H. Liu and S. Arimoto, "Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles", *The International Journal of Robotics Research*, vol. 11, 1992, 376–382, 10.1177/027836499201100409.

[21] L. Blasi, E. D'Amato, M. Mattei, and I. Notaro, "UAV Path Planning in 3-D Constrained Environments Based on Layered Essential Visibility Graphs", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, 2023, 2359–2375, 10.1109/TAES.2022.3213230.

[22] B. C. Shah and S. K. Gupta, "Speeding Up A* Search on Visibility Graphs Defined Over Quadtrees to Enable Long Distance Path Planning for Unmanned Surface Vehicles", *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 26, 2016, 527–535, 10.1609/ICAPS.V26I1.13793.

[23] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, "An Autonomous Path Planning Method for Unmanned Aerial Vehicle Based on a Tangent Intersection and Target Guidance Strategy", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, 2022, 3061–3073, 10.1109/TITS.2020.3030444.

[24] O. Khatib. "The Potential Field Approach and Operational Space Formulation in Robot Control". In: *Robotic Systems for Manipulation and Assembly*, 367–377. Springer US, Boston, MA, 1986.

[25] J. Borenstein and Y. Koren, "The Vector Field Histogram—Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions on Robotics and Automation*, vol. 7, 1991, 278–288, 10.1109/70.88137.

[26] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", *IEEE Transactions on Robotics and Automation*, vol. 12, 1996, 566–580, 10.1109/70.508439.

[27] S. LAVALLE. "Rapidly-Exploring Random Trees: A New Tool for Path Planning", 1998.

[28] X. Zhou, W. Zheng, Z. Li, P. Wu, and Y. Sun, "Improving Path Planning Efficiency for Underwater Gravity-Aided Navigation Based on a New Depth Sorting Fast Search Algorithm", *Defence Technology*, vol. 32, 2024, 285–296, 10.1016/J.DT.2023.04.012.

[29] I. B. Jeong, S. J. Lee, and J. H. Kim, "Quick-RRT*: Triangular Inequality-Based Implementation of RRT* with Improved Initial Solution and Convergence Rate", *Expert Systems with Applications*, vol. 123, 2019, 82–90, 10.1016/J.ESWA.2019.01.032.

[30] S. Forrest, "Genetic Algorithms: Principles of Natural Selection Applied to Computation", *Science*, vol. 261, 1993, 872–878, 10.1126/SCIENCE.8346439.

[31] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948, 10.1109/ICNN.1995.488968.

[32] M. Dorigo and G. D. Caro, "Ant Colony Optimization: A New Meta-Heuristic", *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 2, 1999, 1470–1477, 10.1109/CEC.1999.782657.

[33] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, vol. 69, 2014, 46–61, 10.1016/J.ADVENGSOFT.2013.12.007.

[34] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks", *Machine Learning 1995 20:3*, vol. 20, 1995, 273–297, 10.1007/BF00994018.

[35] I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves. "Artificial Neural Networks: A practical Course", 1 2016.

[36] A. Bessede, M. Gargaro, M. T. Pallotta, D. Matino, G. Servillo, C. Brunacci, S. Bicciato, E. M. Mazza, A. Macchiarulo, C. Vacca, R. Iannitti, L. Tissi, C. Volpi, M. L. Belladonna, C. Orabona, R. Bianchi, T. V. Lanz, M. Platten, M. A. D. Fazia, D. Piobbico, T. Zelante, H. Funakoshi, T. Nakamura, D. Gilot, M. S. Denison, G. J. Guillemin, J. B. Duhadaway, G. C. Prendergast, R. Metz, M. Geffard, L. Boon, M. Pirro, A. Iorio, B. Veyret, L. Romani, U. Grohmann, F. Fallarino, and P. Puccetti, "Deep Reinforcement Learning: An Overview", *arXiv.org*, vol. 511, 2017, 184–190.

[37] I. K. Nikolos, E. S. Zografos, and A. N. Brintaki. *UAV Path Planning Using Evolutionary Algorithms*, volume 70, chapter UAV Path Planning Using Evolutionary Algorithms, 77–111. Springer, Berlin, Heidelberg, 2007.

[38] J. Wu, J. Yi, L. Gao, and X. Li. "Cooperative Path Planning of Multiple UAVs Based on PH Curves and Harmony Search Algorithm", 10 2017.

[39] H. Shiri, J. Park, and M. Bennis, "Remote UAV Online Path Planning via Neural Network-Based Opportunistic Control", *IEEE Wireless Communications Letters*, vol. 9, 2020, 861–865, 10.1109/LWC.2020.2973624.

[40] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C. C. Peng, "Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges", *Sensors 2018, Vol. 18, Page 3170*, vol. 18, 2018, 3170, 10.3390/S18093170.

[41] M. Otte and E. Frazzoli, "RRTX: Real-Time Motion Planning/Replanning for Environments with Unpredictable Obstacles", *Workshop on the Algorithmic Foundations of Robotics*, vol. 107, 2014, 461–478, $10.1007/978-3-319-16595-0_27$.

Multimodal Emotion Detection for Education and Work Environment by Using Improved Artificial Intelligence Machine Vision System

Submitted: July 2024; accepted 6th September 2024

Wan Mohd Bukhari Wan Daud, Adnan Kiral, Mohamed Osman Tokhi, Lee Chung Yee, Muhammad Muzhafar Mohammad Zawawi

DOI: 10.14313/jamris-2026-019

Abstract:
The use of artificial intelligence (AI) has significantly advanced emotion recognition within human-computer interaction (HCI). This paper aims to develop a multimodal emotion detection system for educational and work environments using an enhanced AI machine vision system. The primary focus is on training and testing a multimodal AI model in Python using convolutional neural networks (CNN). The results from the trained facial emotion AI model demonstrated substantial improvements. Training accuracy increased from 30.49% to 72.21%, while validation accuracy improved from 37.6% to 60.58%. Simultaneously, training loss decreased from 180.69% to 73.65%, and validation loss reduced from 172.97% to 107.53%. This CNN-based model can use OpenCV to detect seven emotions: happy, sad, neutral, angry, afraid, disgusted, and surprised. The ECG emotion AI model, also trained with CNN, also successfully recognized patterns for the same seven emotions. When these two models are combined into a multimodal AI system, they can detect facial and ECG-based emotions simultaneously. This comprehensive approach allows for the detection of both visible and hidden emotions, such as stress or anxiety, which may not be easily discernible through facial expressions alone. The integration of these models into a multimodal AI system provides a more accurate and holistic understanding of human emotions, enhancing applications in educational and work settings. The improved detection capabilities can lead to better user experiences and more effective responses to emotional states, ultimately contributing to advancements in HCI.

Keywords: *Artificial Intelligence, Multimodal Emotion Detection, Machine Learning, Convolutional Neural Network, Machine Vision System*

1. Introduction

Emotion detection integrates artificial intelligence, machine learning, and psychology in attempting to identify, analyze, and interpret human emotions through computational methods [8]. Its foundation lies in analyzing various data streams that encompass visual cues, like facial expressions captured through image or video data; linguistic patterns from textual information and tonal inflections in speech; physiological signals such as heart rate or electrodermal activity. They have even been able to analyze behavioral cues. Employing complicated algorithms and techniques like machine learning models, deep neural networks, natural language processing, and computer vision, these systems extract, analyze, and interpret features embedded within these data modalities to differentiate and categorize emotional states [19].

The complexity of this field is amplified by the multifaceted nature of emotions, including slight variations and cultural differences in their expression, as well as the complex interaction between different emotional dimensions. Emotion detection systems continuously evolve, striving for greater accuracy and depth in understanding emotions beyond basic classifications (like happiness, sadness, anger) to address the nuances within each emotional state. These advancements smooth out the way for applications across various sectors, including the workplace, education, healthcare and customer service. Emotion detection, for example, can identify signs of stress or burnout among employees. This allows employers to implement strategies for stress reduction, provide support, and create a healthier work environment at their workplace. In the education field, emotion detection is also core to adaptive learning systems, in which educators to adjust teaching styles and content based on students' emotional states.

Ethical considerations surrounding privacy, biases, and fairness are integral in the development and deployment of these systems. We must ensure their responsible and equitable use while harnessing the potential to build more empathetic and responsive AI technologies that enhance human-machine interactions and enrich our understanding of human emotions and behaviors.

Recently, mental health issues in Malaysia became a major public health concern. The number of individuals experiencing serious long-term mental health issues such as depression, anxiety, stress and so on had increased drastically, and suicides had become more widespread. In Malaysia, there are about 5.5 million adolescents and the statistics have shown that 1 in 5 are depressed, 2 in 5 are anxious and 1 in 10 are stressed. Between 2012 and 2017, the rate of adolescent suicidal behavior—such as suicidal ideation, planning and attempts—had increased. Suicidal ideation increased from 7.9% to 10.0%, suicidal planning increased from 6.4% to 7.3%; and attempts increased

6.8% to 6.9%. As a result, mental health issues at the workplace or education have become a wake-up call to society and need to be solved urgently.

Most people tend to suppress their feelings rather than express their feelings to others. This has made it more difficult for people to recognize emotions in others. Emotion detection using artificial intelligence (AI) has played an important role in order to addressing this problem, as it can detect people's emotion without asking them. Mental health issues in education and the workplace can thus be identified early, and treatment can be started as soon as an abnormal emotional pattern is detected by the AI. The research of X. Li et al. [7] found that emotion detection has garnered significant interest in the field of education during the past few decades, since students' emotions are directly related to their learning effectiveness and academic performance. Based on the timely monitoring of students' emotional states with considerable assistance from AI, instructors can assess the emotions and engagement levels of their students in a way that is difficult to determine from watching them in a big classroom.

2. Literature Survey

To design the AI model, the focus of the literature review was to find:

1. Types of emotional expression.
2. Types of emotion detection systems.
3. Types of artificial intelligence (AI) technologies.
4. Types of machine learning systems.

In 2022, T. Kusunose et al. [3] highlighted that facial expressions are among the most efficient and effective ways to signal emotions, functioning as a near-universal language. This is because the facial cranial nerves that control the muscles involved in facial expressions are highly versatile, surpassing even the vocal cords in conveying prosody and inflection. Specific combinations of facial movements, such as a smile or displaying teeth, can indicate happiness, anger, or concern. Recent advancements in facial recognition technology can detect these expressions in real time, offering valuable applications in healthcare, education, and work environments for assessing emotions and task engagement.

In 2018, L. Shu et al. [29] found that physiological signals like electroencephalogram (EEG), electrocardiogram **(**ECG), and electromyography (EMG) are crucial for emotion detection due to their direct link to emotional states. These signals provide objective measures of physiological responses influenced by emotional arousal and valence, offering insights into emotions that may be difficult to verbalize. The process involves extracting features from these signals, such as EEG frequency bands or ECG heart rate variability, which are then analyzed and associated with emotions using machine learning algorithms. This method enables real-time monitoring and improves the accuracy and reliability of emotion recognition systems in various applications.

In 2021, X. Li et al. [7] wrote that AI-driven emotion detection focuses on automatically analyzing students' emotional states during educational lessons. AI systems can process large amounts of data from text, speech, facial expressions, and body language, recognizing emotions based on eye and head movements in online learning environments. Machine learning algorithms enable these systems to continuously improve their accuracy by learning from new data, making them effective for monitoring user status and mitigating risks. Additionally, as noted by G. Assunção et al. [8] in 2022, AI systems provide consistent and objective analysis, reduce human bias, and can be scaled for large-scale applications such as customer sentiment analysis and mental health monitoring.

In 2023, G. Shi [10] proposed using machine learning technologies to detect emotions by training AI models on extensive data. These algorithms continuously learn and improve their accuracy in recognizing patterns in facial expressions, speech, and body language. Machine learning is adept at identifying subtle emotional cues and can be customized for specific contexts, languages, or cultures. It enables real-time, scalable emotion detection across various applications—such as computer vision, natural language processing, and audio recognition—without requiring constant human intervention. This approach has significantly enhanced the efficiency and effectiveness of emotion detection.

M. U. Khan et al. [19] recommended the use of using deep learning methods, particularly transfer learning with models like MobileNetV2, to train emotion detection models. These methods automatically learn hierarchical representations from raw data, such as images, enhancing their ability to extract complex features crucial for recognizing emotional cues from facial expressions. Pre-training on large datasets like ImageNet allows these models to leverage learned representations to improve performance on emotion-specific datasets, such as Kaggle's emotion dataset. Adjusting learning rates further optimize accuracy, achieving as much as 98.7% accuracy at a learning rate of 0.0001. Deep learning architectures, including CNNs for image data and multimodal networks for integrating multiple data sources, excel in capturing intricate patterns across various modalities, advancing the accuracy and generalization of emotion detection systems without manual feature engineering.

3. Proposed Methodology

The machine learning method, which is uses a convolutional neural network (CNN), is implemented for training and testing the AI model in Figures 1 and 2.

Figure 1 shows a facial emotion recognition process. The input frame, containing images of faces, undergoes pre-processing to enhance and prepare the data. Subsequently, feature extraction identifies crucial characteristics from the processed images. These features are then classified into various categories, leading to the final predicted result—in this example, identification of the "Happy" emotion.

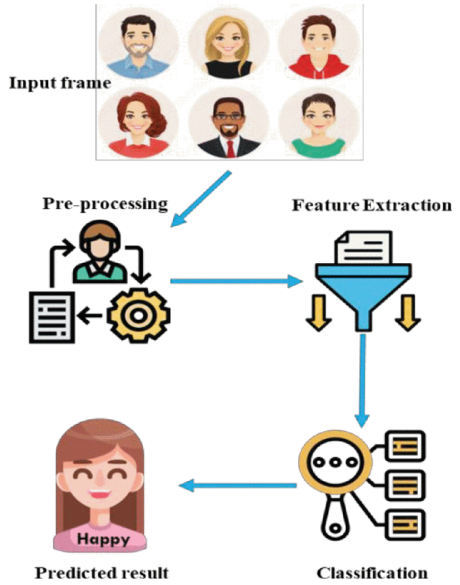


Figure 1. Flow process of training and testing the face emotion detection AI model using machine learning methods

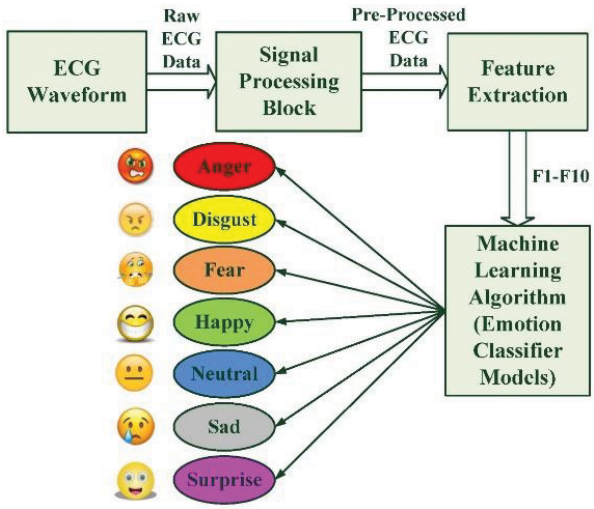


Figure 2. Block diagram of training and testing the ECG emotion detection AI model with machine learning methods

Figure 2 shows an emotion recognition process using ECG data. It begins by capturing the ECG waveform, which is then converted into raw ECG data. This data passes through a signal processing block, resulting in pre-processed ECG data. Feature extraction is performed on this data, generating features, labeled as "F1 score." The features are then fed into a machine learning algorithm that classifies the data into one of seven emotions: happy, sad, neutral, angry, afraid, disgusted, and surprised.

3.1. Training and Testing the Face Emotion Detection AI Model Using Machine Learning Methods

3.1.1. Training the Face Emotion Detection AI Model

Step 1: Importing Libraries

Import various libraries for data manipulation, numerical operations, plotting, and interaction with the OS, such as Matplotlib (plotting), NumPy (numerical operations), Pandas (data manipulation), Seaborn (statistical visualization), and 'os' (OS interaction). Import necessary deep learning libraries from Keras, including tools for image loading and conversion (load_img, img_to_array), image augmentation (Image Data Generator), building neural network layers (Dense, Input, Dropout), defining network architecture (Model, Sequential), and optimization algorithms (Adam, SGD, RMSprop).

Step 2: Displaying Images

Download the face expression recognition dataset from Kaggle. Define picture_size as 48 for input images, and set folder_path to the dataset directory. This prepares the code to work with 48x48 pixel images from the specified folder for facial expression recognition. Set expression to 'happy' and create a 12x12-inch plot for visualizing images related to the 'happy' expression. Load and display a 3x3 grid of images from the dataset using load_img and plt.imshow.

Step 3: Making Training and Validation Data

Set batch_size to 128. Create instances of Image Data Generator for both training and validation datasets, configuring data augmentation for training. Generate batches of training and validation data using flow_from_directory, specifying image directories, target size, color mode, batch size, class mode, and data shuffling.

Step 4: Model Building

Import optimizers from Keras and set no_of_classes to 7. Initialize a sequential model, adding convolutional layers with various filters and sizes, followed by batch normalization, ReLU activation, max-pooling, and dropout. Flatten the output before fully connecting layers. Add fully connected layers with 256 and 512 neurons, including batch normalization, ReLU activation, and dropout. Add output layers corresponding to the number of classes with softmax activation. Compile the model using the Adam optimizer with accuracy as the metric and categorical crossentropy as the loss function. Print the model summary.

Step 5: Fitting the Model with Training and Validation Data

Import Keras optimizers and callbacks. Define the checkpoint for saving the best model, EarlyStopping for stopping training with no improvement, and ReduceLROnPlateau for lowering the learning rate when needed. Set the number of epochs to 48. Assemble the model and start training using fit_generator.

Step 6: Plotting Accuracy and Loss

Set the plot style and create a new figure for accuracy and loss plots. Create subplots to display both loss and accuracy for training and validation.

3.1.2. Training the Facial Emotion Detection AI Model

Step 1: Importing Libraries

Import various libraries for a deep learning model using Keras and OpenCV. These include load_model for loading pre-trained models, sleep for delays, img_to_array for converting images to arrays, image for image preprocessing, cv2 for computer vision tasks, and numpy for numerical operations.

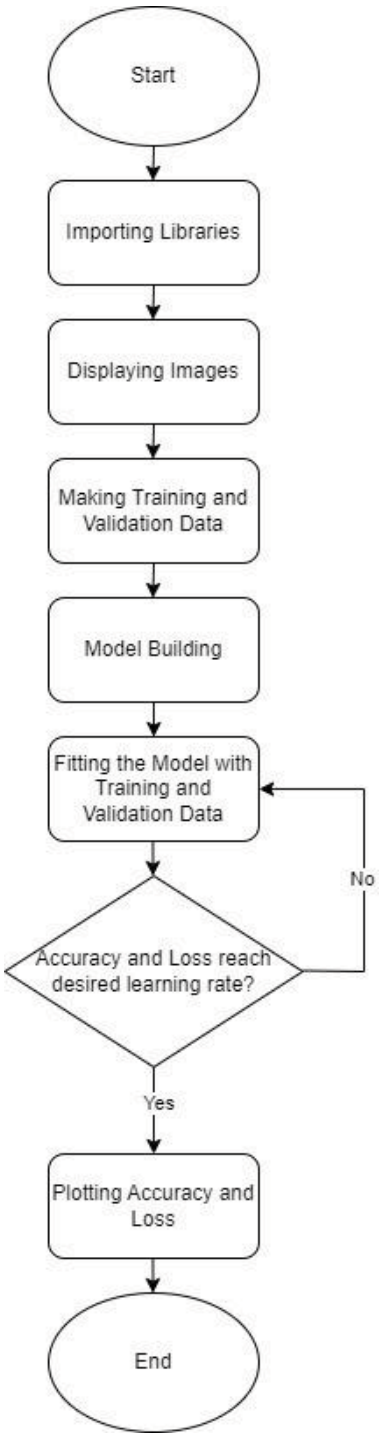


Figure 3. Overall flowchart of facial emotion AI model training process

Step 2: Initialization of AI Model System

Initialize a face cascade classifier using a Haar Cascade XML file, which loads a pre-trained emotion classification model in HDF5 format, defines a list of emotion labels, and initializes a video capture object to start capturing video frames from the default camera.

Step 3: Setting up a Graphical Window for Displaying the Probabilities

Set up a graphical window to display different emotions' probabilities by defining the window dimensions, creating a named window called 'Probabilities' with cv2.namedWindow, and resizing it to the specified dimensions using cv2.resizeWindow.

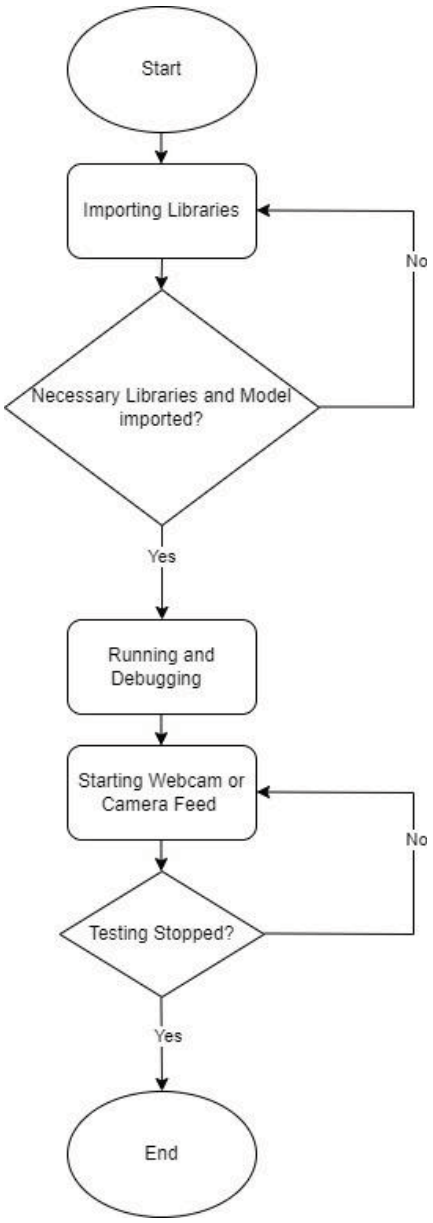


Figure 4. Overall flowchart of facial emotion AI model testing process

Step 4: Running and Debugging the AI Model System

The AI model system runs a real-time video processing loop where it detects faces using the pre-trained face cascade classifier.

3.2. Testing the ECG Emotion Detection AI Model Using Machine Learning Methods

3.2.1. Training the ECG Emotion Detection AI Model

Step 1: Importing Libraries

Import various libraries, including glob, pandas, numpy, os, wget, github, tensorflow.keras, scikit-learn, matplotlib.pyplot, and scipy modules (signal, ndimage, stats, interpolate, and integrate). These libraries provide a wide range of functions for data management, as well as machine learning model creation, evaluation, and signal analysis.

Step 2: Data Processing

Process ECG data files from specified locations by selecting relevant segments and combining metadata

from external dataframes. The code cycles through files, retrieves data up to a predefined length, and parses filenames for session, participant, and video IDs. It filters annotations by comparing IDs with metadata rows, creating database entries with combined ECG data, participant characteristics, and emotional ratings. The processed data is then transformed into a pandas DataFrame, with missing values filled and columns renamed for clarity.

Step 3: Data Visualization

The data will be categorized into one of the seven emotions—happy, sad, neutral, angry, afraid, disgusted, and surprised— using both self-reported and target emotion labels. A custom plotting function iterates through the filtered data, plotting ECG signals for each category and providing a visual representation of ECG signal variations associated with different emotional states.

Step 4:

Create and train a neural network model for emotion identification based on ECG data using TensorFlow and Keras. The data is preprocessed, organized into features and target labels, and standardized. A sequential neural network model is defined with several dense layers, and the model is compiled with categorical cross-entropy loss and the Adam optimizer. Training is performed over 100 epochs with ModelCheckpoint and EarlyStopping callbacks to save the best model and prevent overfitting. The best model is then saved in the HDF5 format.

Step 5: Model Evaluation

Evaluate the pre-trained neural network ECG emotion detection model using a specific test dataset. The model.evaluate() function is used to compute performance metrics, such as accuracy and loss values. The results are stored in a variable and printed, providing a brief overview of the model's predictive performance on unseen test data.

3.2.2. Testing the ECG Emotion Detection AI Model

Step 1: Monitoring Heart Rate in Arduino

Interface with a MAX30100 pulse oximeter sensor to monitor heart rate by establishing serial communication, initializing the sensor, updating readings, and printing the heart rate to the serial monitor every second in Arduino.

Step 2: Real-Time Visualization of Data from Arduino to PyCharm

Capture real-time heart rate data from an Arduino via a serial port, visualize it using Matplotlib, and save the data to a CSV file. The Arduino used to sets up a serial connection (COM4, baud rate 115200) and initializes a Matplotlib plot for dynamic heart rate display. The read_and_process_data() function continuously reads serial port lines, extracts valid heart rate values (60-100 bpm), and appends them to heart_rate_data. The update_plot() function, called by Matplotlib's animation framework every second, updates the plot with the latest readings. At the end of the heart rate measurement, the data is saved to 'heart_rate_data.csv,' and this enables real-time monitoring and logging of heart rate data.

Step 3: ECG Testing

Test the AI model for ECG emotion recognition in Pycharm, importing necessary libraries (NumPy, pandas, TensorFlow's Keras, scikit-learn's StandardScaler, and Matplotlib). The preprocess_ecg function normalizes ECG data using StandardScaler. Load a pre-trained Keras model (ecg_emotion_recognizer.h5) for emotion recognition. ECG data is read from 'heart_rate_data.csv' to simulate real-time data, and the ECG data is normalized using preprocess_ecg. The extract_features function, which returns the normalized ECG data, is defined for feature extraction. The ECG signal is plotted using Matplotlib, and predictions are made by using the loaded model on extracted features. The predicted emotion label is then decoded and printed.

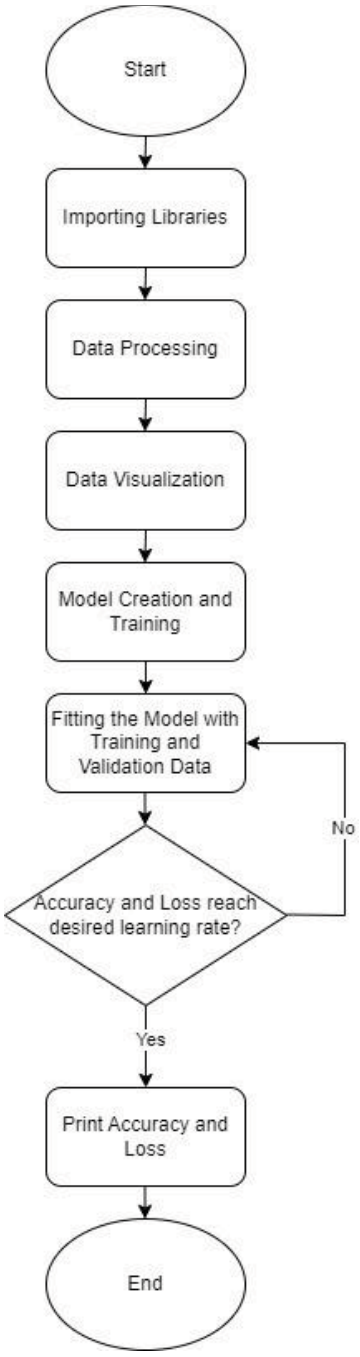


Figure 5. Overall flowchart of ECG emotion AI model training process

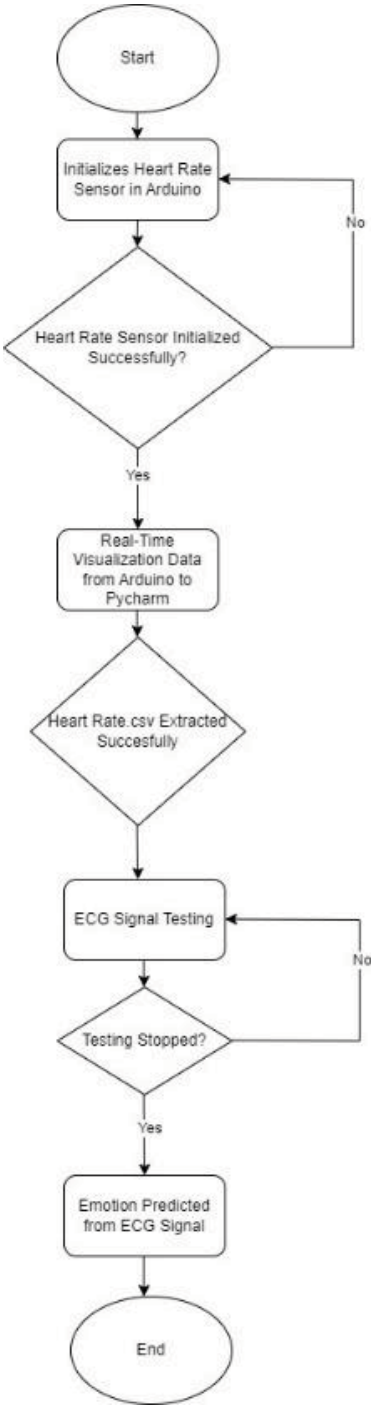


Figure 6. Overall flowchart of ECG emotion AI model testing process

3.3. Testing the Multimodal Emotion Detection AI Model

3.3.1. Testing Multimodal Emotion Detection AI Model Using PyCharm

Step 1: Monitoring Heart Rate in Arduino

Build a multimodal emotion detection AI model using OpenCV for face detection integrating with ECG emotion detection. Facial emotions are predicted using a Haar Cascade classifier, while ECG emotions are predicted from the preprocessed 'heart_rate_data.csv' using a deep learning model. Combined emotions are determined by matching ECG and face predictions, or by defaulting to the ECG emotion if no match occurs. The final combined emotion is then printed for each prediction.

Step 2: Display Combined Emotion in Web Browser

Display a textual description and an emoji in the web browser by the link given, based on the combined emotion from the multimodal emotion detection model.

4. Result with Discussion

In this section, the initial findings from the process of training and testing the multimodal AI model in PyCharm will be discussed.

4.1. Preliminary Results of the Facial Emotion AI Model

Accuracy is a key metric for evaluating convolutional neural networks (CNNs) during both training and validation stages. Training accuracy measures how well the model classifies samples from the training dataset, while validation accuracy assesses the model's ability to generalize to new, unseen data. Both training and validation accuracy increased across epochs, indicating the model's improving performance. The training accuracy reached 72.21%, slightly higher than the validation accuracy, which was 60.58%. A significant gap between these values could indicate overfitting, where the model performs well on training data but fails to generalize to new data. The training process, however, employs an early stopping technique to avoid overfitting.

Loss is a crucial metric for evaluating the performance of convolutional neural networks (CNNs) during the training and validation stages. Training loss measures the discrepancy between the model's predictions and the target values in the training dataset, while validation loss assesses this discrepancy using a separate validation dataset.

The training loss decreased across epochs, indicating that the model's adjustments to its parameters were improving its predictions. The validation loss also decreased, suggesting the model is learning to make accurate predictions on new, unseen data. However, the training loss (73.65%) was significantly lower than the validation loss (107.53%), which could indicate overfitting. This was mitigated, however, by an early stopping technique.

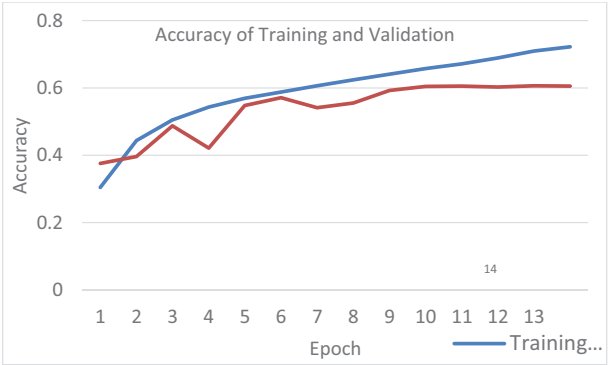


Figure 7. Accuracy of training and validation of facial emotion AI model

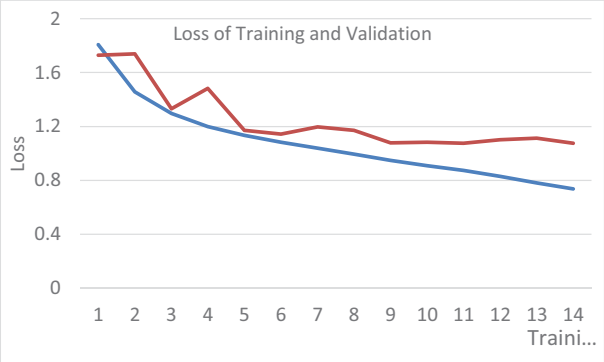


Figure 8. Loss of training and validation of facial emotion AI model

Table 1. Results of experiment.

Emotion	Probability (%)
Happy	98.63
Neutral	56.60
Sad	39.58
Surprise	77.86
Fear	26.36
Angry	41.34
Digust	27.19

Based on Table 1, the highest probability is that the face expression detected by the AI model is "Happy" (98.63%), while the lowest probability is that the face expression detected by the AI model is "Fear" (26.36%). This is because fear often involves subtle facial expressions that can be difficult to capture accurately, especially in real-time or non-controlled environments. Hence, the probability of fear is the lowest, since fear is only expressed in subtle changes like widened eyes or tensed lips, which might be less noticeable, while the "happy" emotion often involves broad smiles and visible changes in facial muscles that are much more noticeable.

4.2. Preliminary Results of the ECG Emotion AI Model

In Figure 9, the model's training accuracy is 26.47%, slightly lower than the validation accuracy of 28.75%, which suggests no overfitting, but indicates poor performance because both accuracies are only around 20%. This low accuracy can be attributed to insufficient and low-quality data, leading to underfitting and ineffective recognition of ECG signals. To enhance the model's performance, it is crucial to collect more and higher-quality data.

In Figure 10, the training loss is 197.367%, which is slightly higher than the validation accuracy of 193.61%. As with the accuracy of training and validation of ECG emotion AI model, the high loss percentage is seen as indicating a poor model. Hence, it is necessary to collect more and better-quality data to improve the performance of the ECG emotion AI model, since the quantity of data used to train the AI model is smaller (only 155 ECG signals for 7 classes of emotions).

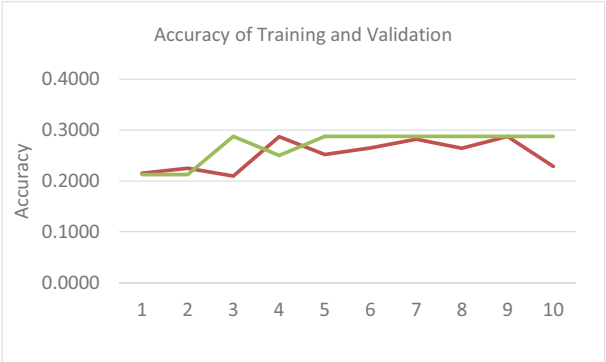


Figure 9. Accuracy of training and validation of facial emotion AI model

Figure 10. Accuracy of training and validation of facial emotion AI model

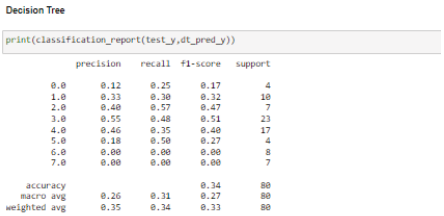


Figure 11. Accuracy of training and validation of facial emotion AI model

4.3. Performance Measures of the ECG Emotion AI Model

4.3.1. F1-Score

The F1-scores can be calculated by using equation (1) below:

$$F1\,Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{1}$$

4.3.2. Decision Tree

The decision tree model for the ECG signal emotion detection has an overall accuracy of 34%, with precision ranging from 0.00 to 0.55, recall from 0.00 to 0.57, and F1-scores from 0.00 to 0.51, indicating inconsistent performance across different emotions. Classes 0.0 and 5.0 show low metrics due to fewer instances, while classes 6.0 and 7.0 have zero precision, recall, and F1-scores. The macro average precision, recall, and F1-scores are 0.26, 0.31, and 0.27, respectively, with slightly higher weighted averages of 0.35, 0.34, and 0.33, highlighting the impact of unbalanced data.

4.3.2. Random Forest

The random forest model for ECG signal emotion detection has an overall accuracy of 39%, with precision levels ranging from 0.00 to 0.59; recall ranging from 0.00 to 0.57; and F1-scores ranging from 0.00 to 0.58. This indicates inconsistent performance across different emotions. Classes 0.0 and 5.0 show low metrics due to fewer instances, while classes 1.0 and 6.0 have zero precision, recall, and F1-scores. The macro average precision, recall, and F1-score are 0.26, 0.35, and 0.29, respectively, with slightly higher weighted averages of 0.34, 0.39, and 0.35, underscoring the impact of unbalanced data.

Based on the comparison between decision tree and random forest for an ECG emotion AI model, referring to Figures 10 and 11, the random forest outperforms the decision tree, achieving a higher total accuracy of 39% compared to 34% for the decision tree. The random forest also exhibits superior weighted averages for precision (0.35 vs. 0.34), recall (0.39 vs. 0.34), and F1-score (0.35 vs. 0.33), indicating better performance across various emotion classes and in managing imbalanced data. Therefore, the random forest model proves more effective and consistent in classification tasks for ECG-based emotion recognition. Further tuning, additional data, and/or alternative modeling methods will be required to improve the AI model.

4.4. Preliminary Results of the Multimodal Emotion AI Model

Figure 13 shows the output in a web browser that includes the combined emotion detected from the multimodal emotion AI model, which is "Happy". It also provides some advice for each emotion. "Happy," for example, has the following text: "Good job! Maintain your positive outlook and keep up the great work."

Random Forest

```
print(classification_report(test_y,rfpred_y))
```

	precision	recall	f1-score	support
0.0	0.22	0.50	0.31	4
1.0	0.00	0.00	0.00	10
2.0	0.40	0.57	0.47	7
3.0	0.59	0.57	0.58	23
4.0	0.43	0.53	0.47	17
5.0	0.22	0.50	0.31	4
6.0	0.00	0.00	0.00	8
7.0	0.25	0.14	0.18	7
accuracy			0.39	80
macro avg	0.26	0.35	0.29	80
weighted avg	0.34	0.39	0.35	80

Figure 12. Classification report using random forest

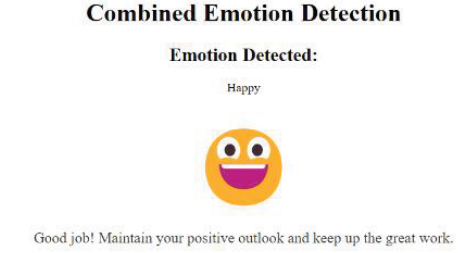


Figure 13. Display combining emotion and advice in the web browser

5. Conclusion

In this study, a multimodal emotion detection system was successfully designed for educational and work environments using convolutional neural networks (CNN). The first objective, involving training and testing the AI model with Python and OpenCV, showed that the model accurately recognized facial emotions from seven classes (happy, sad, neutral, angry, afraid, disgusted, and surprised). The second objective focused on developing the system by integrating OpenCV and ECG signals. The AI model also can detect emotions from seven classes by recognizing the ECG signals. The final objective is validating the multimodal AI system, which accurately combines and displays the results, based on facial and ECG predictions, on a web application.

AUTHORS

Wan Mohd Bukhari Daud* – Faculty of Artificial Intelligence & CyberSecurity at Universiti Teknikal, Malaysia Melaka, Durian Tinggal, 76100, Malaysia. He can be contacted at bukhari@utem.edu.my.
Adnan Kiral – Assistant Professor in the Department of Civil Engineering at Recep Tayyip Erdogan University, Fener, Rize, TR53100, Türkiye. He can be contacted at email adnan.kiral@erdogan.edu.tr
Mohamed Osman Tokhi – A Professor of Engineering at University of London South Bank, London, United Kingdom. He can be contacted at tokhim@lsbu.ac.uk
Lee Chung Yee – Fakulti Teknologi & Kejuruteraan, Universiti Teknikal Malaysia Melaka, Durian Tunggal, 76100, Malaysia, lcyee@student.utem.edu.my
Muhammad Muzhafar Mohammad Zawawi – Fakulti Teknologi & Kejuruteraan,Universiti Teknikal Malaysia Melaka, Durian Tunggal, 76100, Malaysia, m112420023@student.utem.edu.my

ACKNOWLEDGEMENT

Gratitude for Progress Catalysts. We extend our heartfelt appreciation to the Ministry of Higher Education Malaysia and Universiti Teknikal Malaysia Melaka (UTeM) for their invaluable financial support Together, we drive transformative research endeavors.

References

[1] M. Cui, J. Fang, and Y. Zhao, "Emotion Recognition Of Human Body's Posture In Open Environment," *Proc.* 2020. *Chinese Control and Decision Conference (*CCDC), 2020, pp. 3294–3299.

[2] F. Ahmed, A. S. M. H. Bari, and M. L. Gavrilova, "Emotion Recognition From Body Movement," *IEEE Access*, vol. 8, 20210 pp. 11761–11781, 2020.

[3] T. Kusunose, X. Kang, K. Kiuchi, R. Nishimura, M. Sasayama, and K. Matsumoto, "Facial Expression Emotion Recognition Based on Transfer Learning and Generative Model," *Proc. 8th International*

Conference on Systems and Informatics (ICSAI), 2022, pp. 1–6.

[4] D. Shehada, A. Turky, W. Khan, B. Khan, and A. Hussain, "A Lightweight Facial Emotion Recognition System Using Partial Transfer Learning for Visually Impaired People," *IEEE Access*, vol. 11, 2023, pp. 36961–36969,

[5] A. S. Utane and S. L. Nalbalwar, "Emotion Recognition Through Speech," *Proc. 2nd National Conference on Innovative Paradigms in Engineering and Technology* (NCIPET), 2013.

[6] J. Chen, "Speech Emotion Recognition Based On Convolutional Neural Network," *Proc. International Conference on Networking, Communications and Information Technology* (NetCIT*)*, 2021, pp. 106–109.

[7] X. Li, R. Yue, W. Jia, H. Wang, and Y. Zheng, "Recognizing Students' Emotions based on Facial Expression Analysis," *Proc. 11th International Conference on Information Technology in Medicine and Education* (ITME), 2021, pp. 96–100.

[8] G. Assunção, B. Patrão, M. Castelo-Branco, and P. Menezes, "An Overview of Emotion in Artificial Intelligence," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, 2022.

[9] J. M. Foo, S. S. Anhum, S. Islam, and K. H. Keoy, "Facial Emotion Recognition System for Mental Stress Detection among University Students," *Proc. 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering* (ICECCME), 2023, pp. 1–6.

[10] G. Shi, "Technical Analysis of Machine Learning Algorithms in Artificial Intelligence Image Recognition," *Proc. 3rd Asian Conference on Innovation in Technology* (ASIANCON), 2023, pp. 1–5.

[11] S. Ashrita and T. Padmass Machine Learning for Automation Software Testing: Challenges, Use Cases Advantages & Disadvantages," *International Journal of Innovative Science and Research Technology,* vol. 5, no. 9, 2020, pp. 1301–1307.

[12] K. Vinutha, M. K. Niranjan, J. Makhijani, B. Natarajan, V. Nirmala, and T. R. Vijaya Lakshmi, "A Machine Learning based Facial Expression and Emotion Recognition for Human Computer Interaction through Fuzzy Logic System," *Proc. International Conference on Inventive Computation Technologies* (ICICT), 2023, pp. 166–173.

[13] I. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, "Challenges in Representation Learning: A Report on Three Machine Learning Contests," *Neural Networks*, vol. 64, 2013, doi: 10.1016/j.neunet.2014.09.005.

[14] T. Ergin, M. A. Ozdemir, and A. Akan, "Emotion Recognition with Multi-Channel EEG Signals Using Visual Stimulus," Proc. *Medical Technologies Congress* (TIPTEKNO), 2019, pp. 1–4.

[15] L. Huang, K. Yu, T. Song, H. Wang, F. Yuan, Y. Zhang, and H. Yang, "Research on Emotion Recognition Based on Multisource Signals," Proc. IEEE Statistical Signal Processing Workshop (SSP), 2023, pp. 280–284.

[16] Itai Gat, Hagai Aronowitz, Weizhong Zhu, Edmilson Morais, Ron Hoory, "Speaker Normalization for Self-Supervised Speech Emotion Recognition", *Proc. International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2022, pp. 7342–7346.

[17] R. Sato, R. Sasaki, N. Suga, and T. Furukawa, "Creation and Analysis of Emotional Speech Database for Multiple Emotions Recognition," *Proc. 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques* (O-COCOSDA), 2020, pp. 33–37.

[18] P. Yue, L. Qu, S. Zheng, and T. Li, "Multi-task Learning for Speech Emotion and Emotion Intensity Recognition," *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (APSIPA ASC), 2022, pp. 1232–1237.

[19] M. U. Khan, M. A. Abbasi, Z. Saeed, M. Asif, A. Raza, and U. Urooj, "Deep Learning Based Intelligent Emotion Recognition and Classification System," *Proc. International Conference on Frontiers of Information Technology* (FIT), 2021, pp. 25–30.

[20] R. Zhan, J. Lu, Y. Jiang, T. Li, G. Zhong, and W. Lv, "A Multimodal Deep Learning Approach for Typhoon Track Forecast by Fusing CNN and Transformer Structures," *Proc. 4th International Conference on Computer Vision, Image and Deep Learning* (CVIDL), pp. 391–394.

[21] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review Of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, 53.

[22] H. Liu, M. Cocea, and W. Ding, "Decision Tree Learning Based Feature Evaluation and Selection for Image Classification," *Proc. International Conference on Machine Learning and Cybernetics* (ICMLC), 2017, pp. 569–574.

[23] F. Aaboub, H. Chamlal, and T. Ouaderhman, "Analysis of the Prediction Performance of Decision Tree-Based Algorithms," *Proc. International Conference on Decision Aid Sciences and Applications* (DASA), 2023, pp. 7–11.

[24] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2014.

[25] T. Ghirmai, "Generating Laplace process with desired autocorrelation from Gaussian AR processes," *Proc. IEEE Signal Processing and Signal Processing Education Workshop* (SP/SPE), 2015, pp. 113–117.

[26] L. Yang, C. Heiselman, J. G. Quirk, and P. M. Djurić, "Class-Imbalanced Classifiers Using Ensembles of Gaussian Processes and Gaussian Process Latent Variable Models," in *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2021, pp. 3775–3779.

[27] Y. Li, S. Rao, A. Hassaïne, R. Ramakrishnan et al., "Deep Bayesian Gaussian Processes For Uncertainty Estimation in Electronic Health Records," *Scientific Reports*, vol. 11, no. 1, 20685, 2021.

[28] L. Shu, J. Xie, M. Yang, Z. Li, Z. Li, D. Liao, X. Xu, and X. Yang, "A Review of Emotion Recognition Using Physiological Signals," *Sensors,* vol. 18, no. 2018, 2074, doi: 10.3390/s18072074.

Machine Learning–Driven Classification of Text-Based Cybercrime under the Indian IT Act

Submitted: 25th September 2025; accepted: 23rd October 2025

Sukrati Agrawal, Hare Ram Sah, Rajesh Kumar Nagar

DOI: 10.14313/jamris-2026-020

Abstract:
Cybercrimes encompass crime against children, data breaches, and privacy violations. The increased frequency of cybercrimes due to the quick development of technology emphasizes the necessity of complex systems to analyze and categorize these offenses. There are many opportunities to analyze cybercrime data using Machine Learning (ML) techniques because of its enormous accumulation. This study proposes a model that has the potential to automatically analyze text-based reported cybercrime complaints based on the features by use of Random Forest (RF) and Gradient Boosting (GB) algorithms. This model includes a Bag of Words (BoW) approach for feature engineering to analyze reported cybercrime and suggest relevant Indian IT Act sections, such as Section 66E for privacy protection, Section 43A for reported data breach, and Section 72A for disclosure of information, using Natural Language Processing (NLP) for feature extraction and classification. This strategy enhanced the law and enforcement process by timely and accurately categorizing crime. By automating cyber law and providing timely legal answers to various reported cybercrimes, especially those concerning privacy and data protection, the model improves the capabilities of cybercrime units and achieves high accuracy and precision in anticipating pertinent legal sections.

Keywords: *Cybercrime, Machine Learning, NLP, Cyber Law, IT Act, ensemble stacking*

1. Introduction

1.1. Background

The rate of cybercrime has increased as a result of technology globalization, imparting serious risks to both personal safety and national security. In order to defend against these cyberattacks, modern technologies are essential. India stopped 500 million cyberattacks in the first quarter of the year 2023 [1].

As per the data accumulated from the National Crime Records Bureau (NCRB), there has been a persistent rise in reported cybercrime cases in recent years. The number of reported cybercrime cases and ongoing investigations is increasing yearly at an alarming rate, as depicted in the graph shown in Figure 1. This causes an increase in the backlog, which emphasizes how urgently the system must be automated to guarantee that the accused receive just punishment and that victims receive prompt remedies.

Figure 1. Cybercrime complaints reported in India during 2012 and 2022 [2]

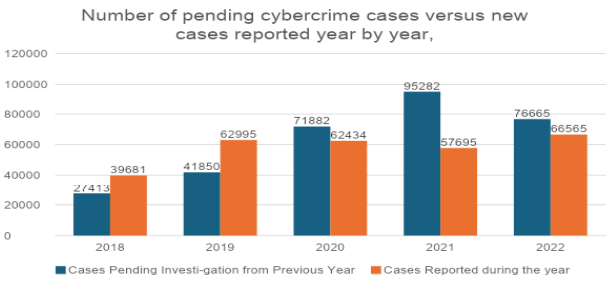


Figure 2. Cybercrime complaints reported and pending during the years 2018–2022 [2]

1.2. Problem Statement

In the present scenario, the manual processing and analysis of reported cybercrime cases delay legal responses, which may cause trauma to victims, as the process is laborious, time-consuming, and subject to human error, as illustrated in Figure 2. Machine learning techniques enable the rapid treatment and categorization of these cybercrime complaints under the applicable Indian IT Act sections. Machine learning also enhances the accuracy of these classifications.

The steady increase in reported cybercrime cases highlights the need for creative and artificial intelligence-based approaches to counter the expanding risks connected to the internet. The drawbacks of conventional case analysis and management techniques result in backlogs and delays in the execution of justice [3, 4].

The sharp increase in reported cybercrime complaints indicates that quick and automated technology is needed to effectively manage these complaints and

respond swiftly. An efficient solution to this problem is to combine the Information Technology (IT) Act with machine learning (ML)-based cybercrime complaint analysis systems to safeguard cyberspace and assist victims [5, 6].

As machine learning algorithms have the potential to analyze large amounts of text-based data and identify trends, they are required for categorizing cybercrime complaints and identifying relevant legal provisions of the Indian IT Act. This technology-driven architecture will enhance the cybercrime response system in terms of time as well as accuracy, ensure efficient classification and legal review, and enable a more effective administration of justice. [7–9].

1.3. Objectives

- To apply RF and GB models for categorizing cybercrime complaints under sections of the Indian IT Act for 66E, 43A, and 72A.
- To evaluate each model's performance in terms of accuracy, precision, recall, and F1-score.

Approach:

In the process of data collection, relevant cybercrime data is gathered from multiple sources like incident reports, legal documents, and cybercrime articles. These sources include a few samples of cybercrime complaints reported at police departments, legal institutions, and online resources. Once the data is gathered, preprocessing is applied to handle missing values and also to convert the data into the required format. Categorical variables are encoded, and outliers in numerical fields are managed to maintain data integrity. Feature engineering includes metrics like complaint duration and resolution status, and text data is cleaned and tokenized. The dataset is split into training, testing and validation sets (70%, 30%) to support thorough model evaluation. Model training utilizes Random Forest and Gradient Boosting algorithms [10, 11], with performance evaluated through precision, recall, F1 score, AUC, and accuracy metrics, ensuring a reliable classification model.

Outline:

The rest of the paper is outlined as follows: Section 2 provides a literature review, including an overview of the relevant provisions for cybercrime, focusing on Sections 66E, 43A, and 72A. It also presents a comprehensive review of legal frameworks for cybercrime and machine learning applications in text classification. Section 3 details the methodology, including data preparation, feature extraction, and the use of Random Forest and Gradient Boosting models. Section 4 presents the results, comparing model performance metrics and classification accuracy. Section 5 discusses the legal implications and limitations of automated classification in cybercrime complaints. It also outlines potential improvements and future research directions. Finally, this section concludes the paper by summarizing the key findings and their significance for enhancing cybercrime management under the IT Act.

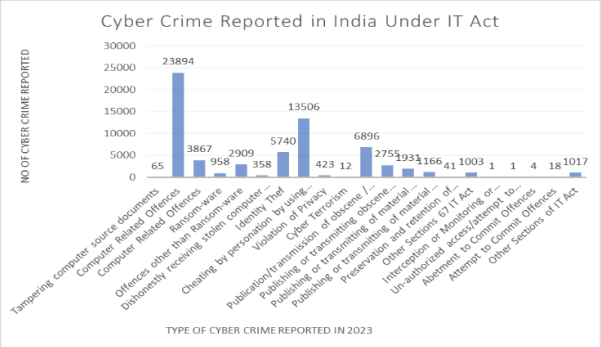


Figure 3. Types of cybercrime registered during the year 2023 under the Indian IT Act [12]

2. Review of the Literature

2.1. Legal Provisions for Cybercrime

The IT Act 2000 addresses various forms of cybercrimes to ensure data protection and privacy, as illustrated in Figure 3. It was amended in 2008 to add stronger provisions, notably Sections 66E, 43A, and 72A (Indian IT Act, 2008). Section 66E criminalizes the unauthorized capture or transmission of private images, focusing on safeguarding personal privacy. Section 43A mandates data controllers to adopt security practices, providing guidelines for reporting data breaches. Section 72A addresses the unauthorized disclosure of personal information by service providers, protecting user confidentiality. The effectiveness of these legal provisions has been widely acknowledged [5, 7].

Geetika Bhardwaj et al. [13] revealed a 46% rise in crimes against women in 2021 compared to 2020, emphasizing the need for proactive action to gather crime data and forecast future trends.

Olena V. et al. [14] highlighted the global increase in cyberattacks, particularly in finance, retail, technology, and communication industries, highlighting the challenges in combating such threats due to geographical disparities.

Gangwar Suraj et al. [15] report a surge in cybercrime due to the Internet of Things, cloud services, and improved connections. Cybercriminals threaten human lives. Targeting industrial control systems, elections, and digital wildfires rank among the top threats worldwide.

The study by Shuai Chenet et al. [16] indicates a positive global correlation between cybercrime and social, economic, and technological variables, with most incidents occurring in urbanized areas with greater infrastructure.

P. Datta et al. [17], in their study, show an increase in fraud cases, primarily affecting 20-29-year-olds, particularly mothers and children, necessitating awareness campaigns to combat cybercrime in India.

The study by Chudasama Dhaval et al. [18] suggests that third-party apps are frequently used by attackers for money transfers, leading to fraud.

Table 1 below summarizes the analysis of already existing algorithms used for cybercrime text detection and related applications.

Table 1. Comparison of existing algorithms

Author(s)	Dataset Used	Algorithm	Efficiency	IT / IPC Detected
Alami & Elbeqqali (2015) [19]	Microblog data	Text mining + SVM	Not detailed	No
Mbaziira & Jones (2016) [20]	Deceptive cybercrime text	Linguistics + ML	Medium	No
Kumari et al. (2018) [21]	Labeled text samples	NLTK, Scikit-learn	Moderate	No
Andleeb et al. (2019) [22]	MySpace bullying texts	Text mining + ML	Not detailed	No
Ch et al. (2020) [23]	State-wise crime stats	SVM, Decision Tree	Good	No
K. veena et al. (2022) [24]	Cybercrime reports	SVM	High	Potential
Pandey et al. (2022) [25]	Custom labeled reports	Ensemble (RF, NB, etc.)	High (noted)	No

Our literature review observed that most prior works focus on classification accuracy without explicitly mapping outputs to IT ACT provisions. This highlights a research gap where existing models are effective in detection but are not efficient in providing legally actionable outcomes, hence motivating the need for more advanced frameworks that provide integration of legal context with efficient machine learning.

2.2. Machine Learning in Legal Text Classification

An automated cybercrime text classification in the legal domain is gaining more attention. This automated cybercrime classification enhances the efficiency of handling a huge number of complaints. For the categorization of text-based reported cybercrime complaints, the most widely used ML models are RF and GB, as they have the potential to process huge datasets and provide strong performance [26]. ML algorithms are proven to be more successful in categorizing and analyzing text-based complaints, especially in cases when it comes to differentiating between several legal categories, as per recent studies [27–29].

ML is updating the classification of legal texts, especially those regarding cybercrime. In order to reduce human error and provide more efficient classification, Ch et al. [23] proposed a sustainable computational framework for classifying cybercrime offenses using ML. Andleeb et al. [22] showed how effective ML is at extracting features, analyzing cybercrime complaints, and classifying offenses through text mining. Patel and Sharma [30] highlighted the wider function of ML in automating legal procedures, especially in cyber law, demonstrating its efficiency in organizing and classifying legal documents. Pandey et al. [25] created a model that outperforms conventional methods by using ensemble learning to increase classification accuracy. Together, these studies highlight how important machine learning is to automate, analyze, and improve the classification of legal texts in cybercrime.

According to our literature review, there are various ways to approach countermeasures and prevent cybercrime. These include analyzing cyber threats, analyzing them with ML, and enhancing law enforcement tactics. The existing background of cybercrime will be examined, along with trends and patterns found in the literature. Proactive cybercrime detection and classification by machine learning is the main emphasis of this work. The research aims to create robust models that identify small irregularities and patterns indicative of cyber threats using various methods and datasets. Because of the increasing complexity of cybercriminals and the expansion of internet connectivity, cybercrime is a growing threat. Anonymity, exponential expansion in digital data and insufficient cybersecurity safeguards are some of the factors that make people and businesses vulnerable.

2.3. Law and enforcement to combat cybercrime

Recent research explores various strategies for combating cybercrime under the Indian IT Act. For instance, ensemble learning can be used to classify cybercrimes under Sections 66 and 67, and models like SVM and Random Forest can be used to improve accuracy, thus offering practical tools for cybercrime cells [26] to analyze cybercrime trends and prevention and providing global insights and statistical data on rising cybercrime patterns [31, 32]. Additionally, study of regional crime patterns using regression can find correlations in cyber offenses such as unauthorized photo sharing and computer theft [29]. Finally, the need to present a computational tool with high accuracy for identifying cybercrime rates at the state level in India underscores the role of machine learning in crime analytics [30].

3. Proposed Methodology

According to recent studies, machine learning techniques are becoming more and more necessary to identify which IT Act section applies to committed cybercrimes. Prior studies have focused on classifying cybercrime, but they have not addressed the crucial part of providing victim justice by determining whether the IT Act is applicable. This facilitates speedier investigations and raises the possibility that perpetrators of cybercrimes may be held accountable. Additionally, machine learning enables proactive enforcement measures to effectively combat cybercrime by keeping law enforcement agencies updated about evolving legal requirements and cyber threats, as presented in Figure 4.

3.1. Dataset

Compiling a Cyber Crime and Law Classification dataset involves gathering relevant and varied information about cybercrime incidents and legal provisions from multiple sources, including FIRs, case studies, news headlines, etc., as presented in Figure 5. Present research includes victim statements, incident reports, court records, and open databases that document cybercrime incidents.

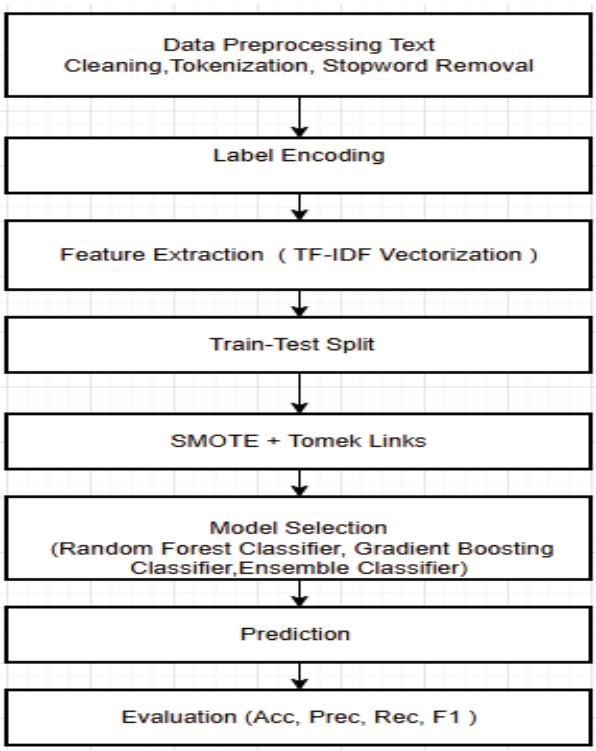


Figure 4. Proposed methodology

Figure 5. Distribution of dataset

3.2. Text Pre-Processing

Some preprocessing steps are applied to the Cyber Crime and Law Classification dataset to ensure the quality of the data. Missing values are addressed, numerical columns are cleaned and converted, and date columns are formatted as a datetime type. Outliers in the numerical data are handled, and categorical features are encoded by label encoder. In text data processing, normalization is done by converting all the text to lowercase. Noise is removed using regular expressions, and stop words are removed. The text is tokenized and lemmatized to ensure consistency.

$$x_i = \text{Clean}(\text{Tokenize}(t_i)) \quad (1)$$

Where:

- t_i = raw text
- x_i = processed token sequence

$$y_i = \text{LabelEncoder}(c_i), y_i \in \{0,1,\ldots,K-1\} \quad (2)$$

Figure 6. Word cloud for Section 43A

Where:

- c_i = category/section name
- y_i = encoded label

3.3. Feature Extraction

Feature extraction is used to change the unprocessed textual input so that the machine learning model can more efficiently use this input. The vectorization approach, TF-IDF (Term Frequency-Inverse Document Frequency), has been used for this investigation. TF-IDF weights words based on how frequently they occur within a certain piece of text compared with the frequency throughout the entire corpus. This will allow the model to discern words used often throughout the corpus versus those used specifically by individual publications. To accomplish this, we use the text data and translate this information into a sparse numerical features matrix with scikit-learn's TfidfVectorizer. We allow algorithms from machine learning to draw inferences from these inputted matrices in search of patterns and relations in the material that is text.

$$TF-IDF(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}} \log\left(\frac{|D|}{1+\left|\{d \in D : t \in d\}\right|}\right) \quad (3)$$

Where:

- $f_{t,d}$ = frequency of term t in document d
- $\sum_{t'} f_{t',d}$ = total terms in document d
- $|D|$ = total number of documents
- $\{d \in D : t \in d\}$ = number of docs containing term t

Figure 6 provides a word cloud that helps in visual analysis of the words frequently occurring in Section 43A of the Indian IT ACT.

Figure 7 provides a word cloud that helps in visual analysis of the words frequently occurring in Section 66E of the Indian IT ACT.

Figure 8 provides a word cloud that helps in visual analysis of the words frequently occurring in Section 72A of the Indian IT ACT.

3.4. Handling Imbalanced Data with SMOTE

To solve the class imbalance in crime datasets, the study employs SMOTETomek, a strategy that combines SMOTE and Tomek Links. While Tomek Links

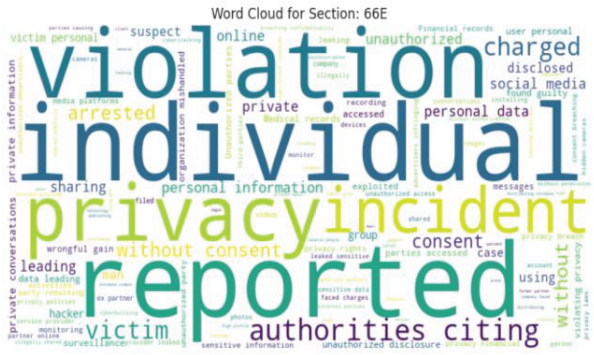


Figure 7. Word cloud for Section 66E

Figure 8. Word cloud for Section 72A

eliminates borderline occurrences, SMOTE creates synthetic examples for the minority class, guaranteeing a balanced class distribution and better predictive performance, particularly for uncommon crime categories.

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i), \quad \lambda \in [0,1] \tag{4}$$

Where:
- x_i = a minority class sample.
- x_{nn} = one of the nearest neighbors of x_i
- λ = a random number.
- x_{new} = newly generated synthetic sample.

3.5. Model Selection

The three text categorization algorithms employed in this study are RF, GB, and Ensemble classifier (stacked), which combines RF and GB for increased accuracy and a soft-voting scheme. The models are evaluated using the following metrics as depicted in Figure 9, which shows that the Ensemble classifier performed the best.

This graphical representation shows that the IT Act's automated classification of cybercrime complaints may accelerate case processing, increasing accuracy and response times. Targeted legal measures are made possible by precise classification under certain provisions, which helps the court and law enforcement better combat cybercrime [32].

For RF:

$$\hat{y} = mode\left\{\widehat{y^{(1)}}(x), \widehat{y^{(2)}}(x), \ldots, \widehat{y^{(B)}}(x)\right\} \tag{5}$$

Where:

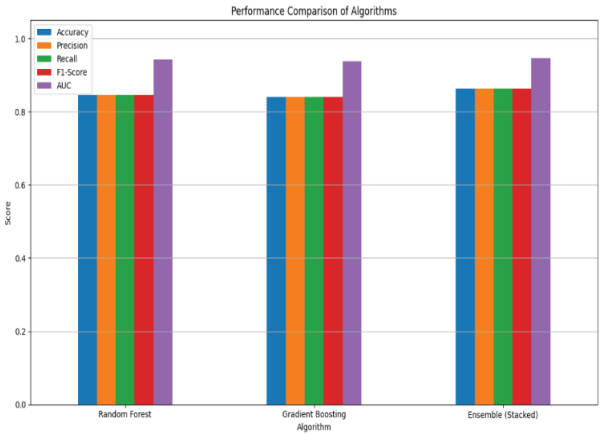


Figure 9. Performance analysis for different models

- $\hat{y}$ = final predicted class.
- $\widehat{y^{(i)}}(x)$ = prediction of the i^{th} base learner for input x.
- B = total number of base learners.
- mode $\{\cdot\}$ = class that appears most frequently among the predictions of all base learners.

For GB:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x) \tag{6}$$

Where:
- $F_m(x)$ = boosted model after m iterations
- $F_{m-1}(x)$ = model from previous iteration (m−1)
- $h_m(x)$ = weak learner at step m
- ν = learning rate
- x = input feature vector

For ECS:

$$\hat{y} = \sigma(w_0 + w_1\widehat{y_{RF}} + w_2\widehat{y_{GB}}) \tag{7}$$

Where:
- $\hat{y}$ = final predicted output
- σ = activation function
- w_0 = bias
- w1i = weights
- y_{RF} = prediction from Random Forest
- $\widehat{y_{GB}}$ = prediction from Gradient Boosting

4. Results and Discussion

4.1. Performance Comparison

The performance of each algorithm is summarized in Table 2, showing the accuracy, precision, recall, F1-score, and AUC based on the training model.

Table 2. Performance comparison of models

Algo	Accuracy	Precision	Recall	F1-Score	AUC
RF	0.84	0.84	0.84	0.84	0.94
GB	0.84	0.83	0.84	0.85	0.93
ECS	0.86	0.86	0.86	0.86	0.94

Table 3. Section-based cybercrime classification result analysis

Section	Algo	Precision	Recall	F1-Score	Support
66E	RF	1.000	1.000	1.000	47
	GB	1.000	1.000	1.000	
	ECS	1.000	1.000	1.000	
72A	RF	0.700	0.903	0.789	31
	GB	0.667	0.903	0.767	
	ECS	0.700	0.903	0.789	
43A	RF	0.927	0.760	0.835	50
	GB	0.923	0.720	0.809	
	ECS	0.927	0.760	0.835	

As per the results shown in Table 2, the Ensemble Classifier performs better than Random Forest and Gradient Boosting in terms of accuracy, precision, recall, F1-score, and AUC.

4.2. Section-Wise Accuracy

A detailed breakdown of accuracy per section (crime category) is provided, showing how each model performs across different crime types. Table 3 illustrates the accuracy of Random Forest, Gradient Boosting, and Ensemble Classifier for each crime section.

In conclusion, the research shows that every algorithm performed exceptionally well, obtaining flawless scores in Section 66E. In Section 72A, RF and Ensemble voting maintained strong recall while marginally outperforming GB in precision and F1-score. Random Forest and Ensemble Voting outperformed Gradient Boosting in Section 43A, achieving better F1 scores. RF is the most dependable and time-efficient algorithm overall since it continuously shows a balance between excellent performance and efficiency throughout all parts.

5. Conclusion

This study demonstrated that ML techniques can be used effectively to classify cybercrime complaints based on textual descriptions. We create a strong feature extraction and preprocessing pipeline by resolving class imbalance with SMOTETomek and applying the TF-IDF vectorization approach. The ensemble classifier outperforms the other models in terms of overall performance across a number of evaluation metrics, demonstrating how well it handles challenging criminal classification tasks.

5.1. Limitations and Model Improvements

The model has been successfully demonstrated for the three sections of the Indian IT Act, including sections 66E, 72A, and 43E. In future models, they can be trained to identify any relevant section of the Indian IT Act based on the reported cybercrime, which requires a huge dataset creation for each section of the Indian IT Act. This model has demonstrated high accuracy, but this can lead to a few small misclassifications in cases where complaints have overlapped phrases between relevant sections. Deeper semantic output may be obtained by improving the models' efficiency by the use of advanced NLP approaches like named entity recognition (NER) and sentiment analysis [28, 32].

5.2. Future Work

In the future, this algorithm can be applied to predict any relevant section of the Indian IT Act. In order to better understand complex patterns of language found in cybercrime complaints, future research must examine advanced approaches such as ensemble algorithms and BERT. These methods could enhance the system's capacity to manage massive volumes of data and adhere to data protection laws when paired with federated learning for privacy-preserving training [26].

AUTHORS

Sukrati Agrawal* – Research Scholar, Department of Computer Science, SAGE University, Indore, Madhya Pradesh, India, +91-8982493161, Kailod Kartal, Indore Rau Bypass Road, Indore, Madhya Pradesh, 452020, e-mail: sukratiagrawalphd@gmail.com.

Hare Ram Sah – Department of Computer Science, SAGE University, Indore, Madhya Pradesh, India, +91-9630689967, Kailod Kartal, Indore Rau Bypass Road, Indore, Madhya Pradesh, 452020, e-mail: ramaayu@gmail.com.

Rajesh Kumar Nagar – Department of Electronics and Communication, IET, SAGE University, Indore, Madhya Pradesh, India, +91-9981850973, Kailod Kartal, Indore Rau Bypass Road, Indore, Madhya Pradesh, 452020, e-mail: errajesh973@gmail.com.

*Corresponding author

References

[1] Indusface Research Team, "The State of Application Security Report Q1 2023," Indusface, 2023. [Online]. Available: https://www.indusface.com

[2] T. Basuroy, "Number of cyber-crimes reported in India 2012–2021," *Statista*, 2022. [Online]. Available: https://www.statista.com

[3] S. Kharat, "Cyber crime – a threat to persons, property, government and societies," *Property, Government and Societies*, Mar. 2017.

[4] S. Vashisth, N. Singh, and S. Dudi, "An in-depth investigation into police operations and their impact on delivering speedy justice in criminal proceedings: Efficiency, challenges, and legal implications," *Library Progress International*, vol. 44, no. 3, 2024, pp. 24434–24446.

[5] A. Kovacs, "Cybersecurity and data protection regulation in India: An uneven patchwork," *CyberBRICS: Cybersecurity Regulations in the BRICS Countries*, 2021, pp. 133–181.

[6] H. Ning, X. Ye, M. A. Bouras, D. Wei, and M. Daneshmand, "General cyberspace: Cyberspace and cyber-enabled spaces," *IEEE Internet of Things Journal*, vol. 5, no. 3, 2018, pp. 1843–1856.

[7] D. M. Katz and J. J. Nay, "Machine learning and law," *Legal Informatics*, 2021, pp. 94–98.

[8] V. Altuglu and R. Schwabe, "Machine learning in litigation," *Handbook of Marketing Analytics*, Edward Elgar Publishing, 2018, pp. 661–670.

[9] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artificial Intelligence Review*, vol. 52, 2019, pp. 273–292.

[10] M. Noguti, E. Vellasques, and L. S. Oliveira, "A small claims court for the NLP: Judging legal text classification strategies with small datasets," *Proc. 2023 IEEE International Conference on Systems, Man, and Cybernetics*(SMC), Oct. 2023, pp. 1840–1845.

[11] P. Payne, R. V. Romould, M. K. Gourisaria, V. Singh, D. K. Behera, and S. Das, "Unveiling bankruptcy risk through advanced data analysis and machine learning techniques," *Proc. 2024 International Conference on Distributed Computing and Optimization Techniques* (ICDCOT), Mar. 2024, pp. 1–6.

[12] S. Agrawal and H. R. Sah, "Study on vulnerability in online social networking: Impact on an individual, community," *Online Social Networks in Business Frameworks*, 2024, pp. 27–46.

[13] G. Bhardwaj and R. K. Bawa, "Performance of machine learning models on crime data," *Proc. International Conference on Computing, Communications, and Cyber-Security*, Springer, Oct. 2022, pp. 811–823.

[14] O. V. Sviatun, O. V. Goncharuk, C. Roman, O. Kuzmenko, and I. V. Kozych, "Combating cybercrime: Economic and legal aspects," *WSEAS Transactions on Business and Economics*, vol. 18, 2021, pp. 751–762.

[15] S. Gangwar and V. Narang, "A survey on emerging cyber crimes and their impact worldwide," *Research Anthology on Combating Cyber-Aggression and Online Negativity*, IGI Global, 2022, pp. 1583–1595.

[16] S. Chen, M. Hao, F. Ding, D. Jiang, J. Dong, and S. Zhang, "Exploring the global geography of cybercrime and its driving forces," *Humanities and Social Sciences Communications*, vol. 10, no. 1, 2023, pp. 1–10.

[17] P. Datta, S. N. Panda, S. Tanwar, and R. K. Kaushal, "A technical review report on cyber crimes in India," *Proc. 2020 International Conference on Emerging Smart Computing and Informatics* (ESCI), IEEE, Mar. 2020, pp. 269–275.

[18] D. Chudasama, D. Patel, A. Shah, and N. Shaikh, "Research on cybercrime and its policing," *American Journal of Computer Science and Engineering Survey*, vol. 8, no. 10, 2020, pp. 14–20.

[19] S. Alami and O. Elbeqqali, "Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts," *Proc. 2015 10th International Conference on Intelligent Systems: Theories and Applications* (SITA), Rabat, Morocco, IEEE, 2015, pp. 1–5.

[20] A. V. Mbaziira, E. Abozinadah, and J. H. Jones Jr., "Evaluating classifiers in detecting 419 scams in bilingual cybercriminal communities," *arXiv* preprint, 2015, arXiv:1508.04123.

[21] S. Kumari, Z. Saquib, and S. Pawar, "Machine learning approach for text classification in cybercrime," *Proc. 2018 Fourth International Conference on Computing Communication Control and Automation* (ICCUBEA), Pune, India, IEEE, 2018, pp. 1–6.

[22] S. Andleeb, R. Ahmed, Z. Ahmed, and M. Kanwal, "Identification and classification of cybercrimes using text mining technique," *Proc. 2019 International Conference on Frontiers of Information Technology* (FIT), IEEE, Dec. 2019, pp. 227–275.

[23] R. Ch, T. R. Gadekallu, M. H. Abidi, and A. Al-Ahmari, "Computational system to classify cybercrime offenses using machine learning," *Sustainability*, vol. 12, no. 10, 2020, pp. 4087–4098.

[24] K. Veena, K. Meena, R. Kuppusamy, Y. Teekaraman, R. V. Angadi, and A. R. Thelkar, "Cybercrime: Identification and prediction using machine learning techniques," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, 2022, p. 8237421.

[25] H. Pandey, R. Goyal, D. Virmani, and C. Gupta, "Ensem_SLDR: Classification of cybercrime using ensemble learning technique," *International Journal of Computer Network and Information Security*, vol. 14, no. 1, 2022, pp. 81–92.

[26] S. Verma and K. Joshi, "Automated text classification for legal applications," *Indian Law Review*, vol. 11, no. 1, 2023, pp. 75–95.

[27] N. Rao and P. Banerjee, "Data classification techniques in cybercrime analysis," *Journal of Information Security*, vol. 29, no. 4, 2022, pp. 410–430.

[28] S. Gupta, "AI in cyber crime management: A study," *Artificial Intelligence and Law*, vol. 28, no. 1, 2020, pp. 102–123.

[29] X. Luo, "Efficient English text classification using selected machine learning techniques," *Alexandria Engineering Journal*, vol. 60, no. 3, 2021, pp. 3401–3409.

[30] K. Patel and V. Sharma, "Machine learning applications in cyber law," *International Journal of Legal Studies*, vol. 8, no. 2, 2019, pp. 56–72.

[31] S. Batra, M. Gupta, J. Singh, D. Srivastava, and I. Aggarwal, "An empirical study of cybercrime and its preventions," *Proc. 2020 Sixth International Conference on Parallel, Distributed and Grid Computing* (PDGC), IEEE, Nov. 2020, pp. 42–46.

[32] S. Verma, S. Nayak, and D. K. Deshmukh, "A survey of emerging cyber crimes and their probable solutions," *Research Journal of Engineering and Technology*, vol. 11, no. 2, 2020, pp. 82–88.

Optimizing Crop Recommendations using Machine Learning: a Comparative Study for Enhanced Yield Prediction

Submitted: 21st July 2024; accepted: 17th September 2024

Sanket Gupta, Trishna Panse, Kailash Chandra Bandhu, Ratnesh Litoriya, Shivani Patnaha, Divya Kumawat, Lishika Pargi, Tisha Modi

DOI: 10.14313/jamris-2026-021

Abstract:
For an important segment of the Indian people, agriculture serves as a primary source of income. Most Indian farmers choose to produce crops in a field using traditional farming methods; hence, one of their biggest issues is that they frequently choose to cultivate the incorrect crop for their soil type. The crop recommendation system proposed in this research would assist farmers and educate them on decision-making regarding which crops to plant on their property. Using soil parameters like potassium, nitrogen, and phosphorus as well as environmental variables like humidity, rainfall, and pH levels, to build this recommendation system, we used ML methods such as Random Forest, KNN, Naïve Bayes, SVM, and Logistic Regression. As a result, we also present comparative performance on the model for the dataset. Therefore, finally, these technologies will be helpful for farming and agriculture. Today's smart agricultural solutions, can address the growing concern about the world population's food consumption and environmental impact. The accuracy of this crop recommendation system will depend on the following: The quality and quantity of our dataset, the relevance and effectiveness of our features, the choice and tuning of our machine learning models, the balance of our dataset and the complexity of the crop prediction task, performing thorough training, validation, and testing will give the accuracy metric we need.

Keywords: *Machine Learning, SVM, KNN, Crop Recommendations*

1. Introduction

Since independence, the primary contributor to the country's GDP has been agriculture. Agronomy and their associated industries constituted 59% of the nation's overall GDP during the fiscal year 1950-1951 [1]. Despite the relatively low agricultural productivity, agriculture remains one of the most prominent economic sectors in India. Precision agriculture is one method we may utilize to boost productivity. Applying precise and suitable amounts of soil, fertilizers, and other elements is what precision farming entails, as the term implies. Globalization has, however, caused a significant shift in the agricultural trend in recent years. Numerous factors have adversely affected India's agricultural sector. In order to restore crop vitality numerous innovative technologies have been developed. Precision farming is one such method, applied at the right time to the crop to boost yields and production. Precision agriculture refers to farming technology that is site-specific. But in agriculture, it's important that the guidance provided be precise and correct.

Machine learning, as defined by Arthur Samuel in 1959, is the study of how computers learn without being explicitly programmed. ML algorithms are trained on vast volumes of facts to generate expectations or findings. Recent years have seen a surge in crop prediction research. For example, using IoT and machine learning (ML) technology to improve agricultural decision-making [2]. Another research proposes employing neural networks to create a robust, precise, and clear recommendation system [3].

To anticipate the most productive crop, this work proposes a crop suggestion method that makes source of machine learning algorithms to evaluate soil (pH, phosphorus, nitrogen, and potassium) and meteorology (temperature, moisture, and rainfall) data. The F1 score, Recall, and Precision have been utilized to assess the performance of the approach suggested for every class and method. Crop recommendation systems can assist farmers in selecting which crops to plant, increasing yields and reducing resource consumption. Crop suggestion systems can also increase agriculture's ability to adapt to climate change. The remaining part of this work includes: The literature review and details on the model are provided in Sections 2 and 3. Sections 4 and 5 cover the Experimental Setup.

2. Literature Review

Including important environmental variables improves the dataset which is used in [4]. The dataset contains Temperature, Humidity, pH, rainfall and label which includes sugarcane, coconut, jute, cotton, papaya, groundnut, maize, graphs, rice, mango, rubber etc. It uses an SVM decision tree (Hybrid approach) which maintains an accuracy rate of 91.8% and Random Forest shows an accuracy of 95%. The Table 1 itself serves as the literature review summary.

The methodology, which integrates machine learning with the IoT, is not well recognized. The authors of [2] proposed it as the Crop Monitoring and Recommendation System utilizing sensors to record certain

Table 1. Crop Recommendation Techniques ML

Authors	Methodology	Features	Accuracy	Dataset	Advantages	Limitations
Srilakshmi A.,Madhumitha K., Geetha K[4]	SVM decision tree (Hybrid approach), Random Forest	Temperature, Humidity, pH, rainfall, label	SVM decision tree (Hybrid approach) - 91.8%. Random Forest - 95%	sugarcane, coconut, jute, cotton, papaya, groundnut, maize, graphs, rice, mango, rubber etc	Predict crop for any type of field	Small dataset
R. Pallavi Reddy, B. Vinitha, K. Rishita, K. Pranavi [2020] [2]	Linear Regression Model	N, P, K, and moisture values			generate recommendations to improve crop production and estimates the price of the yield	Limited in capturing non-linear patterns, Assumes homoscedasticity and independence of errors
S. Mamatha Jajur, Soumya N. G. [2019] [5]	KNN, Decision trees, SVM, CNN and LSTM, ANNs, K-means clustering	Soil Type, pH value, NPK content of the soil, Water holding, Temperature, Average rainfall, Previously Harvested crop	-	wheat, rice, bajra, maize, jawar,	select the optimum crop while keeping a number of variables in mind to boost the output of agriculture, minimise the deterioration of the soil in fields that are under cultivation and use less fertiliser when growing crops.	Many algorithms are used
Mr. Santosh Mahagaonkar, Devdatta A. Bondre [2019] [6]	Random Forest, Support Vector Machine algorithm	crop, crop yield dataset, Location, soil and crop nutrients, fertilizer datasets	soil classification, RF -86.35% crop yield prediction SVM -99.47%	Soybean, Rice, Jowar, Wheat, Sunflower, Cotton, Sugarcane, Tobacco, Onion, Dry Chili, etc.	future prediction of crop yield	Low accuracy in soil classification performance heavily depends on parameter tuning and it is memory intensive, particularly for large datasets
D. Anantha Reddy, Bhagyashri Dadore, Aarti Watekar [2019] [7]	Naïve Bayes, K-NEAREST NEIGHBOUR, RANDOM FOREST, CHAID	Depth, Texture, pH, Soil Colour, Permeability, Drainage, Water holding and Erosion	-	groundnut, pulses, cotton, vegetables, paddy, sugarcane, coriander.	Assist farmers in planting the appropriate seed according to the needs of the soil in order to boost output.	The Naïve Bayes algorithm pretends feature independence, which might not be true when dealing with real-world data., CHAID - Limited to categorical target variables and predictors, making it less versatile for handling continuous data
Nidhi H. Kulkarni [8] 2018	Linear SVM algorithms, Random Forest, Naïve Bayes	Soil type, pH soil, NPK, average rainfall, porosity of soil, sowing season temperature	99.9 1%	Cotton, Sugarcane, Rice, Wheat	Crop productivity has improved exponentially for rice, wheat, cotton, and sugarcane.	restricted to a fairly small number of crops

Table 1. Continued

Authors	Methodology	Features	Accuracy	Dataset	Advantages	Limitations
Zeel Doshi [3] 2018	Neural Network Random Forest, Decision Tree, KNN	Temperature rainfall, Location, soil condition	91%	Jute, sesame, soybean, sugarcane, tobacco, sunflower seeds, ragi, potato, tur, grapeseed, and mustard, bajra, maize wheat, rice gram, barley, cotton, groundnut, and pulses	Neural Networks have the highest accuracy percentage.	predict the crop using the harvest from the previous cycle. Crop supply and demand are not considered
Rohit Kumar Rajak [9] 2017	Random Tree, NB-classifier, ANN, SVM	depth, pH, texture, permeability to store water, color of the soil, and drainage from erosion	-	vegetables, rice, sugarcane, sorghum, coriander, bananas, legumes, and groundnuts	boosts agricultural productivity	larger dataset for model training
S. Pudumalar [10] 2016	Random Tree, Naïve Bayes, KNN, CHAID,	Depth, pH, texture, water-holding permeability, Soil color, erosion drainage,	88%	millet, pulses, groundnut, cotton, banana, vegetables, paddy, sugarcane, sorghum, coriander	Boost productivity	larger dataset for model training
Rakesh Kumar [11] 2015	CSM, Gradient Boosted Decision Tree, and Greedy Forest	soil type, weather, crop type, water density,		ratoi, toria, wheat, potato, sarso, linseed, masoor, khesari, onion, sugarcane, Kanda, mung, til, pumpkin, nenua, ladies' finger, rice, soybean, sweet potato, toor, vegetable seed, and so on	offers a method to select crops while taking into account the yield forecast rate influenced by various factors.	Adopting a prediction technique that performs well and has greater accuracy is necessary.

characteristics of the soil, such as its moisture content and nutrients, and uploading the data to a cloud platform. An Android app receives this data and gives recommendations for crop selection based on soil type among other factors. Furthermore, a price prediction module has been integrated using linear regression. This combined approach is expected to help farmers make good choices and increase farm productivity and profitability aimed at taking into consideration soil health worries.

Authors in [5] have used data including soil type, acidity level, NPK content, permeability, water holding capacity, average rainfall, temperature, as well as previously grown crops. For classification tasks they have supervised learning methods KNN, ensemble learning (EL), and SVM and also unsupervised learning methods (K-means clustering for data analysis).

A technique that farmers throughout India can simply employ is the intelligent crop recommendation

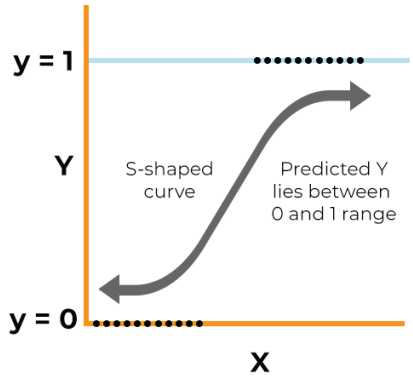


Figure 1. Logistic function [12]

system. Three processes are involved in this research: the classification of the soils, crop yield prediction, and fertilizer suggestion utilizing Random Forest and Support Vector Machine, which provide more robust

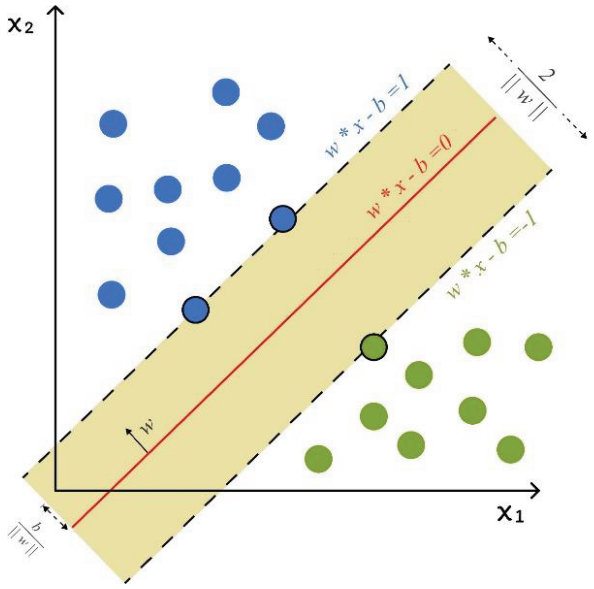


Figure 2. SVM [5]

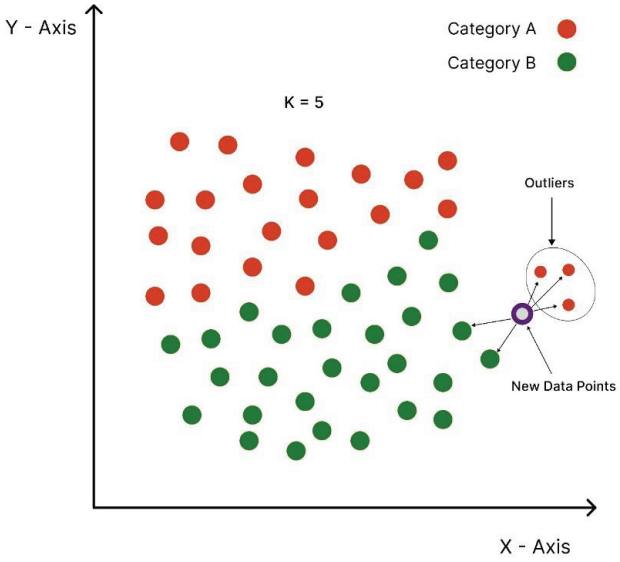


Figure 3. KNN [3, 5]

models than traditional logistic regression model as shown in Figure 1. Additionally, the system has apps from third parties that provide weather data. The result of this experiment reveals that soil classification using Random Forests and crop yield prediction with Support Vector Machines are effective. The future is to improve it by including development of a mobile application for farmer as well as implementing crop disease detection through image processing [6].

The authors in [7] suggested a method so that they can help farmers to make decisions about crops depending on their soil types. It applies soil-specific characteristics such as depth, texture, pH, and water-holding capacity to recommend appropriate crops. It makes use of a group method that incorporates Random Tree, CHAID, Naïve Bayes and KNN machine learning algorithms as demostrated in Figure 3 and 5. This study demonstrates the use of Random Forests for soil categorization and the use of Vector Machines for crop production prediction.

Four crops have been taken into consideration in a crop recommendation system: wheat, cotton, sugarcane, and rice in [8]. Accurate crop selection is provided by crop recommendation systems, which take soil, surface temperature, and rainfall into account. The proposed model uses a high degree of efficiency and accuracy of ensembling approach to predict the crop that will boost yield.

The authors of this work [3] have introduced Agro-Consultant, an intelligent system that Indian farmers may use to make well-informed decisions on which crops to produce by utilizing data on soil properties, geographic location, and climatic parameters like rainfall and temperature. This is accomplished by the utilization of various ML methods, like neural networks, KNNs, Random Forests, and decision trees.

The authors [9] have concluded that crop yield prediction is essential for the country's planned guiding principles made in the field of agriculture development, to provide greater agricultural output and effective use of water resources while assisting farmers in minimising the use of pesticides in crop production and preventing soil deterioration.

Based on the needs of the soil, data mining tools would assist farmers in choosing the best seeds to plant, guaranteeing higher yield to make a profit. To precisely and effectively recommend a crop based on site-specific data, a collaborative recommendation model is built using techniques including the algorithms Random Tree, CHAID, KNN, and Naïve Bayes [10].

To grow the crop's net yield rate, the authors of [11] recommend the Crop Selection Method (CSM), which recommends the series of crops that will be sown throughout the season depending on the crop yield forecast. A solution for crop selection based on factors such as crop type, weather, soil type, and water density is offered by the proposed method. This approach suggests a crop sequence's daily production is maximal for a given season, considering the crop, sowing time, number of days in planting, and season-wise yield rate as inputs.

Machine learning is one of several methods to predict crop production in agriculture. Machine learning techniques such as Random Forests and Support Vector Regression are used, which provide more robust models than traditional linear regression models [12].

3. Model Used

3.1. Logistic Regression

One technique used for binary classification is called Logistic Regression. In this technique we predict the probability of an input example falling into one of two classes. Therefore, the output should be in discrete value i.e., either Yes or No, true or false, etc. It is classified into three basic categories: Binomial, Multinomial and Ordinal.

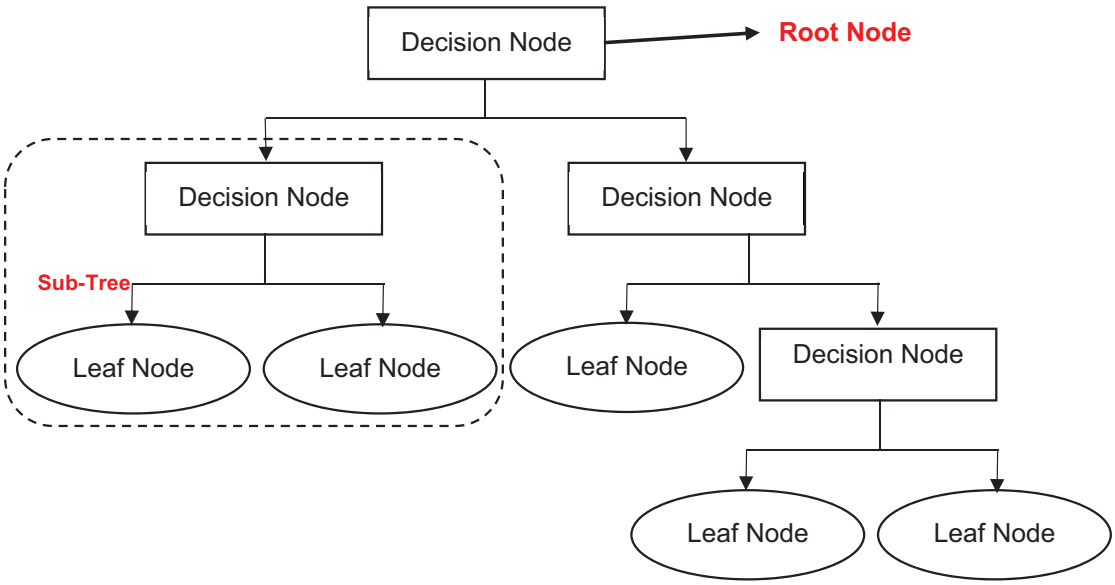


Figure 4. Decision Tree [3]

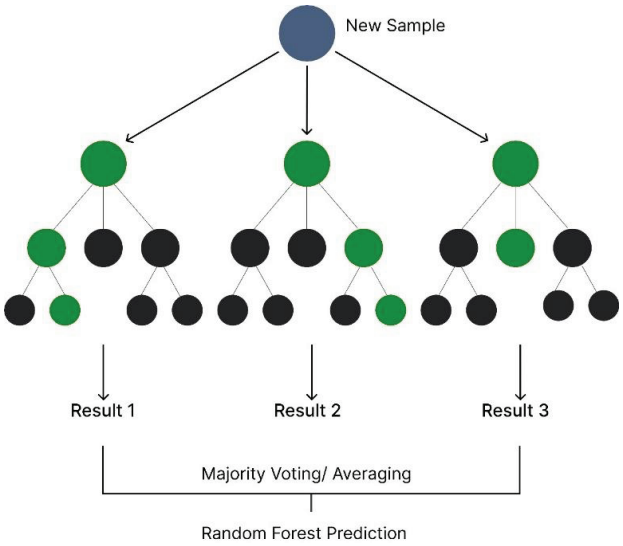


Figure 5. Random Forest [3]

$$y = \frac{1}{1 + e^{-(a_0 + a_1 x)}} \tag{1}$$

- x = input value
- y = predicted output
- a_0 = bias or intercept term
- a_1 = coefficient for input (x)

3.2. Naïve Bayes

One of the most fundamental and successful probabilistic classification methods, Naïve Bayes, is based on the Bayes' theorem. It generates a likelihood table by finding probabilities of the features. Gaussian Naïve Bayes is specifically applied when continuous features follow a Gaussian distribution. It's efficient, simple, and performs well in limited training data.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{2}$$

Eq.(2) is Bayes' theorem in which:
$P(A)$= The probability of A occurring
$P(B)$= The probability of B occurring
$P(A|B)$=The probability of A given B
$P(B|A)$= The probability of B given A

3.3. SVM

Support Vector Machine (SVM) is a supervised machine learning technique that may be applied to both regression and classification applications. The objective of this method is to determine the hyperplane that effectively divides the two classes with the widest possible margin. The SVM algorithm utilizes two crucial elements to choose the most suitable hyperplane for classifying labels: two measurements: the support vectors, or the data points closest to the hyperplane, and the margin, or the distance between the hyperplane and the closest data points. Figure 2 highlights the SVM plan and support vectors.

$$w \cdot x + b = 0 \tag{3}$$

Eq.(3) equation of hyperplane in which:
w = a vector normal to hyperplane
b = an offset

3.4. KNN (K-Nearest Neighbours)

KNN, K-Nearest Neighbours is a supervised ML method, is usually used for division but also for regression. This method starts with selecting the nearest Neighbors. Then, on can try different values for K to find the optimal one. The Euclidean distance between K Neighbors is also calculated. We count the number of data points in each category between these K Neighbors, and we assign a new data point to the category with the highest number of Neighbors. Expanding the training data collection could enhance this technique.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

Eq. (4) is Euclidean distance formula in which:

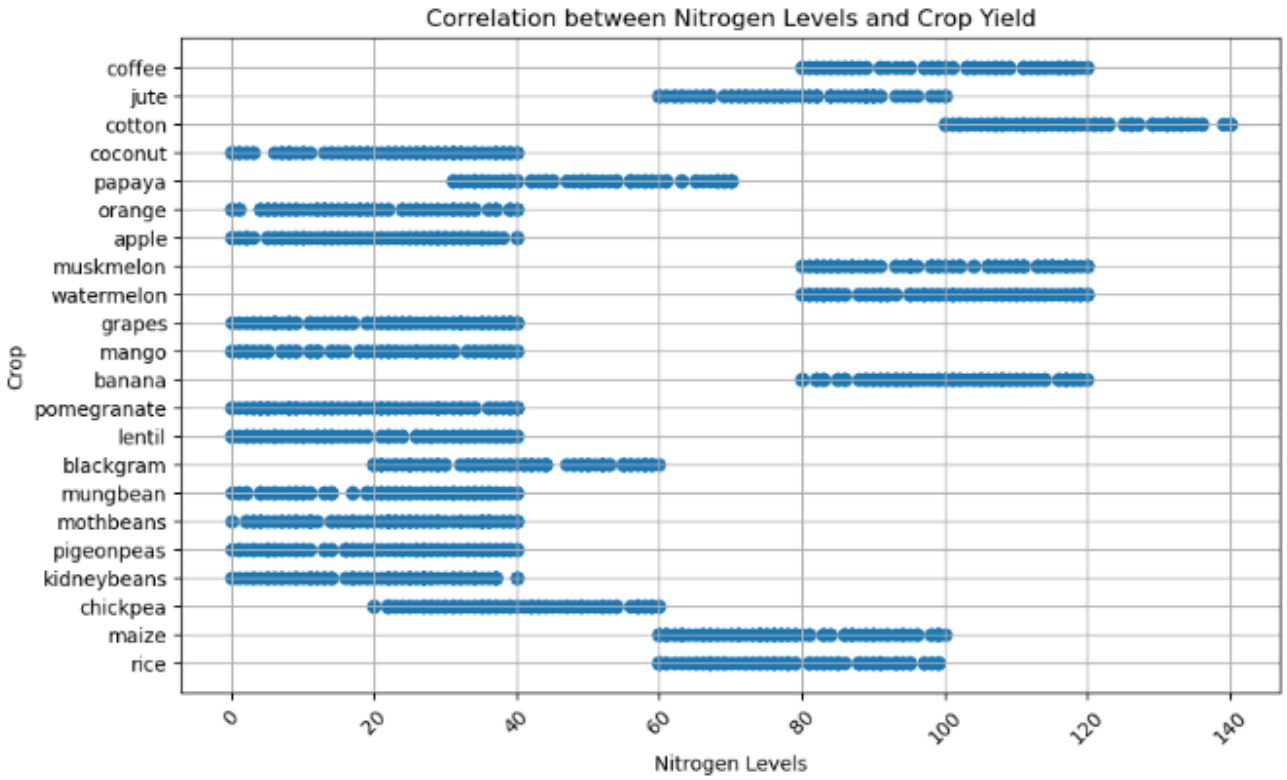


Figure 6. (a). Relationship between Nitrogen Levels and Crop Yield

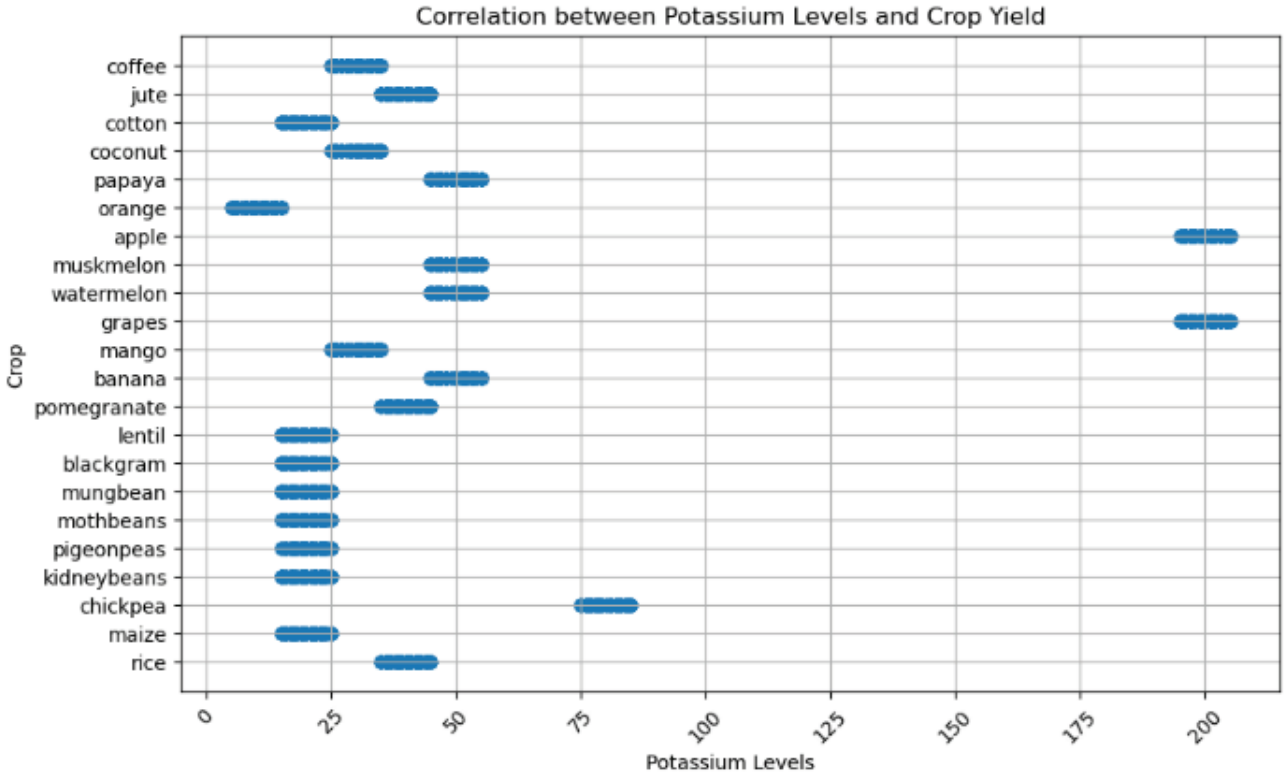


Figure 6. (b). Relationship between Potassium Levels and Crop Yield

The coordinates of one point are (x_1, y_1)
The coordinates of the other point are (x_2, y_2)
Distance between (x_1, y_1) and (x_2, y_2) is d.

3.5. Decision Tree

Although decision trees are among the most powerful tools available, they are typically employed for classification jobs. They can also be utilized for regression assignments. As shown in Figure 4, the Decision

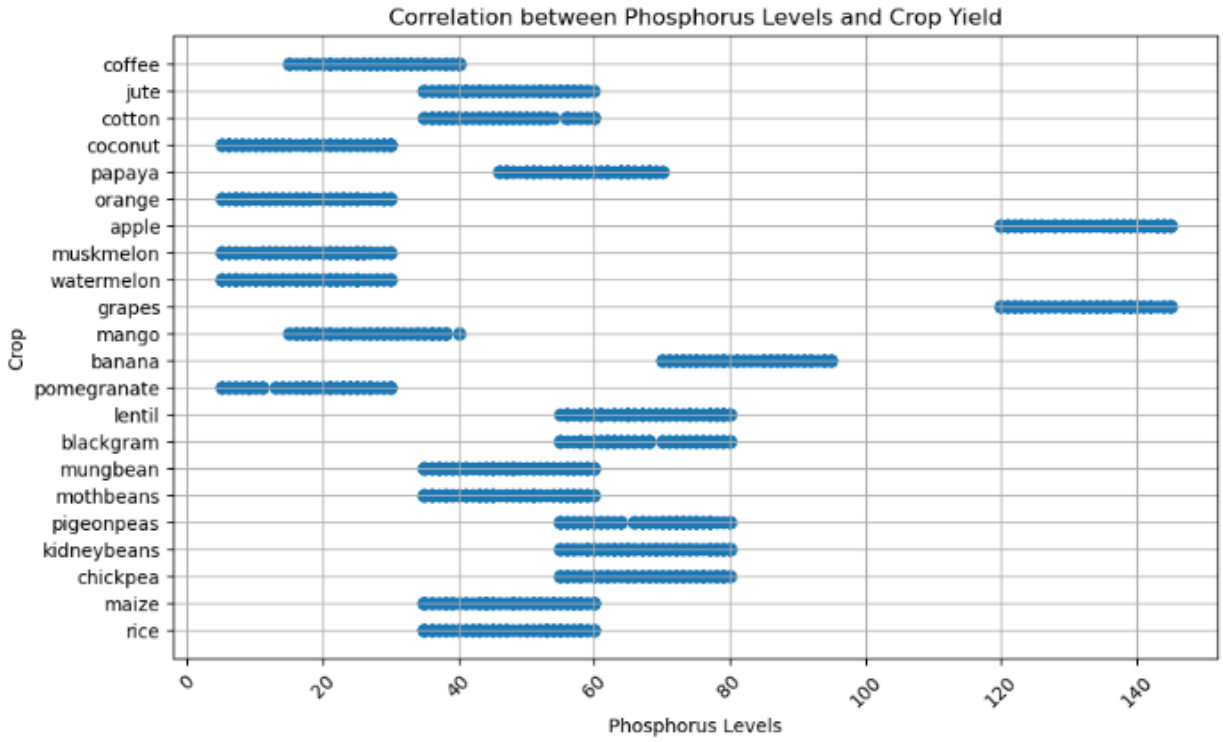


Figure 6. (c). Relationship between Phosphorus Levels and Crop Yield

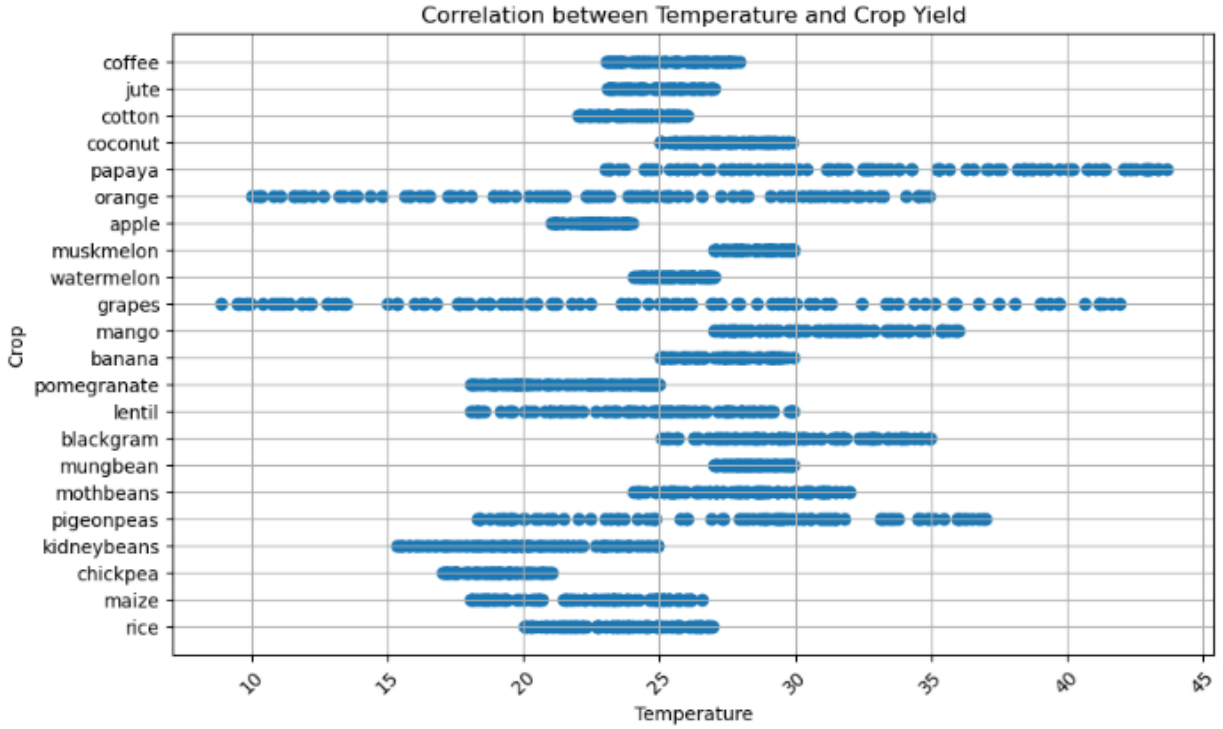


Figure 6. (d). Relationship between Temperature and Crop Yield

Node and Leaf Node are the two nodes that make up this system. Leaf Nodes represent the decision's output, and Decision Nodes are used to make decisions. Decision trees are easy to understand, treatment of both numerical and categorical data though their failure to prune could lead them into over-fit situations especially when noisy datasets are considered.

3.6. Random Forest

Among the widely used ensemble models based on decision trees is Random Forest. Every observation is fed into one of the many decision trees that are generated in Random Forest. Random Forest can deal with high-dimensional data, maintain some level of interpretability, and has less chance of being affected

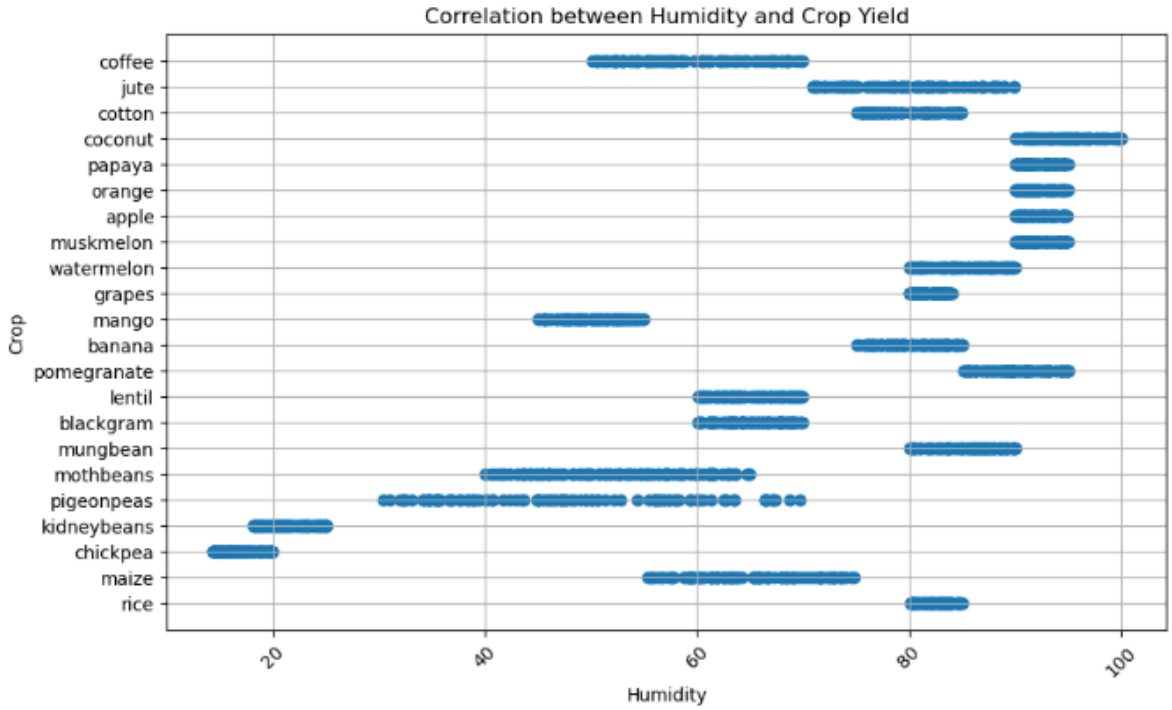


Figure 6. (e). Relationship between Humidity and Crop Yield

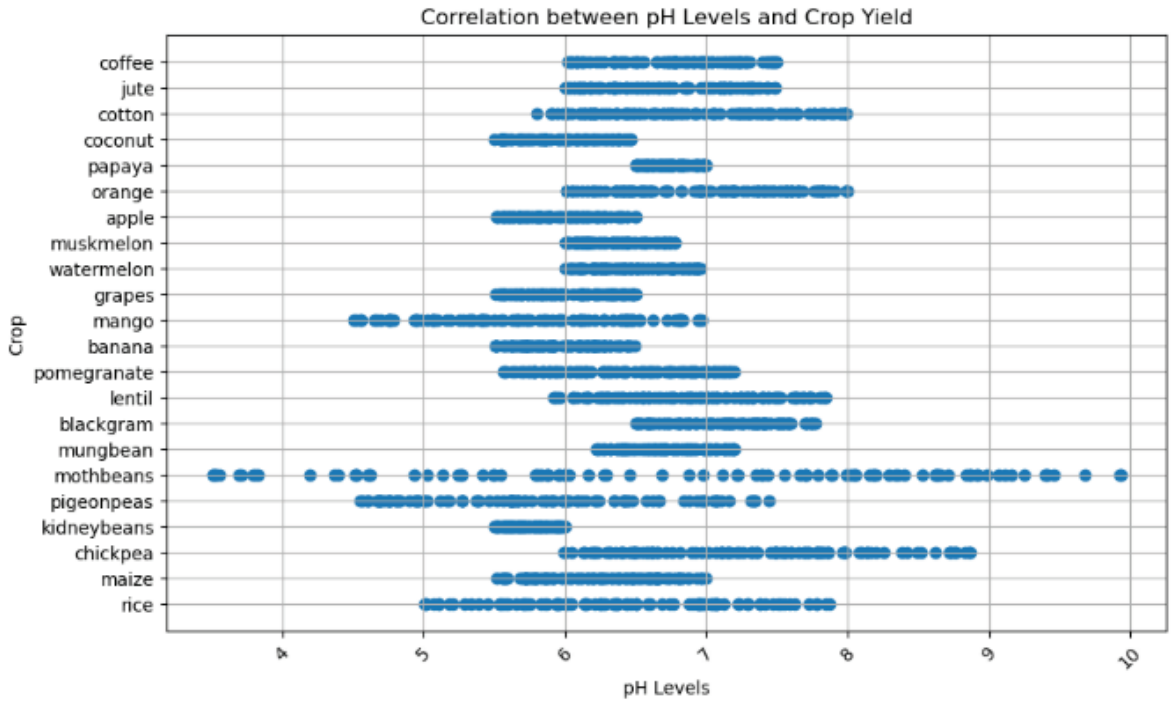


Figure 6. (f). Relationship between pH and Crop Yield

by overfitting than individual decision trees themselves [13].

4. Proposed Method

We aim to develop a CR System by using ML techniques to help farmers select which crop should yield based on soil and climate parameters. The dataset used contains nitrogen (N), phosphorus (P), potassium (K) levels, temperature, humidity, pH, and amount of rainfall, along with the label of crops suitable for the environment and to get a better understanding of the dataset we have visualized the correlation between label and other features.

To ensure data integrity, exploratory data analysis identifies the dataset's dimensions, contents, missing values, and duplicates. Furthermore for smooth flow of data, crop labels were mapped into numerical identifiers. Each crop is associated with a unique numerical value, ranging from 1 to 22. A new column

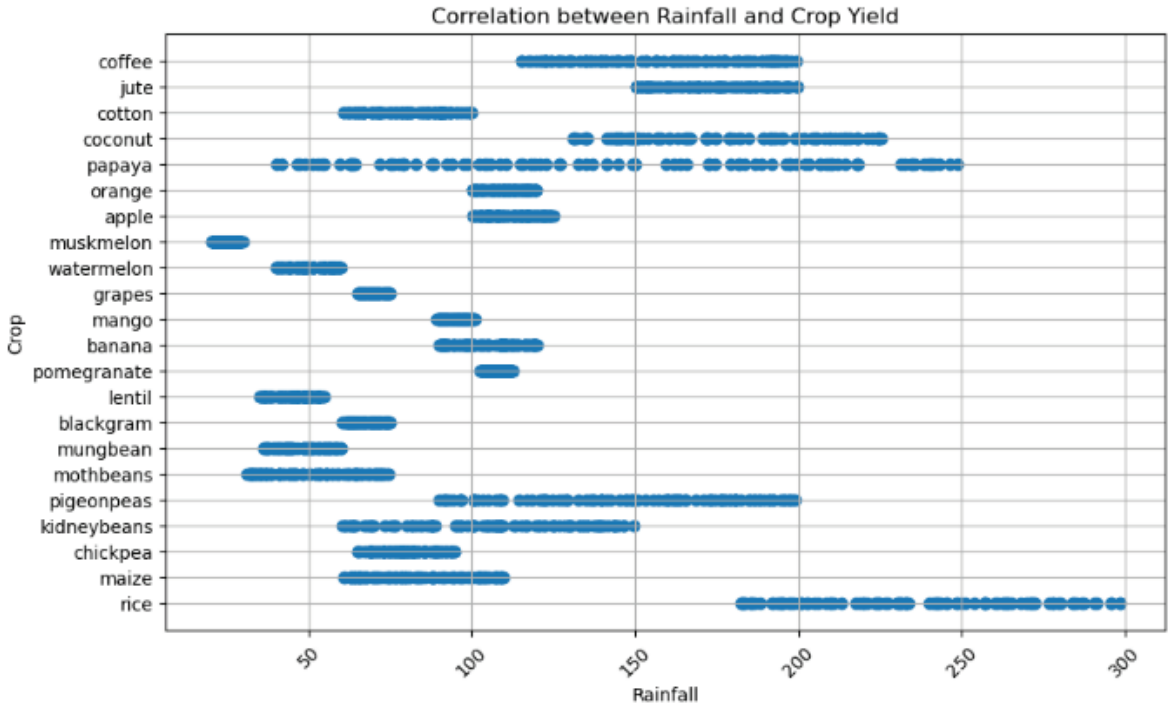


Figure 6. (g). Relationship between Rainfall and Crop Yield

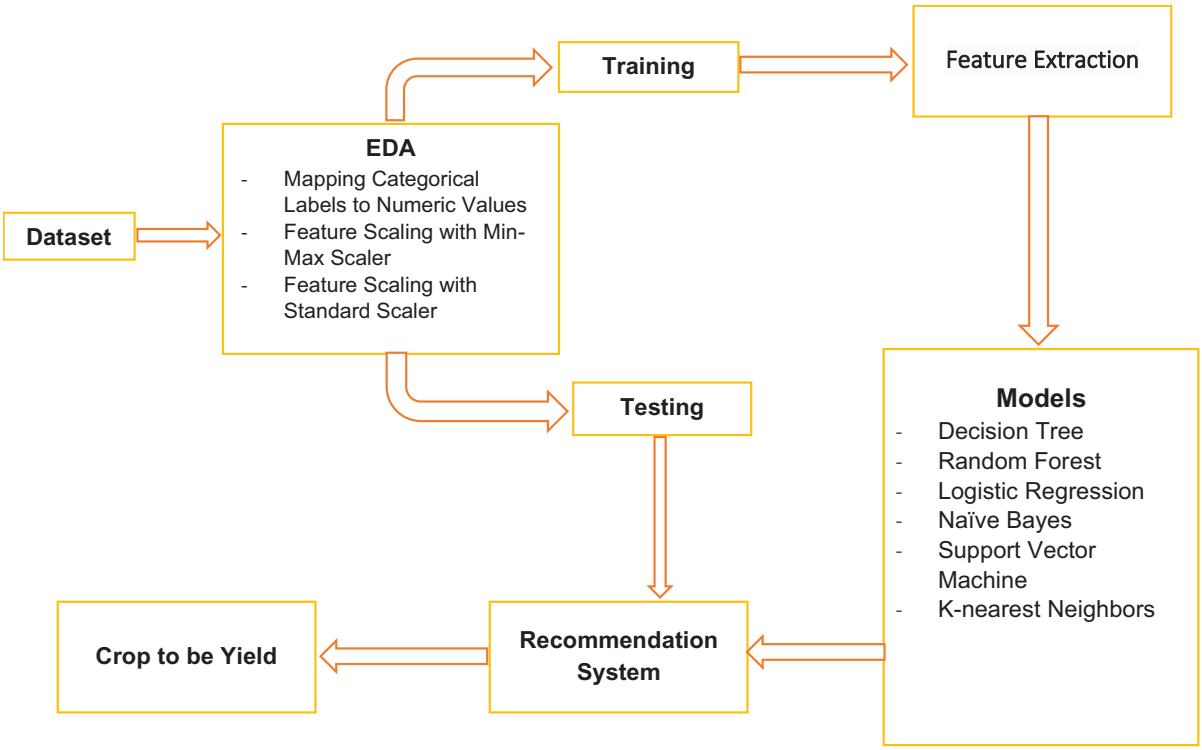


Figure 7. Block Diagram of Crop Recommendation System

'crop_num' is created corresponding to existing column 'label' containing crop names.

The dataset is divided into features (X) containing N, P, K, temperature, humidity, pH and rainfall and labels (y) containing newly created column 'crop_num'. To standardize and normalize the data for model compatibility, we have used Min-Max Scaler. So that it scales and translates each feature individually using the function MinMaxScaler().

The Figure 7 above demonstrates Various Machine learning algorithms were used for the development of the model. Each model's accuracy in crop prediction was evaluate using training dataset (X_train, y_train) after that, the suitable model was trained using a training dataset (X_train, y_train). Naïve Bayes has shown to have the highest accuracy among these models.

For the most accurate crop recommendation as shown in the Figure 8, a Gaussian Naïve Bayes model

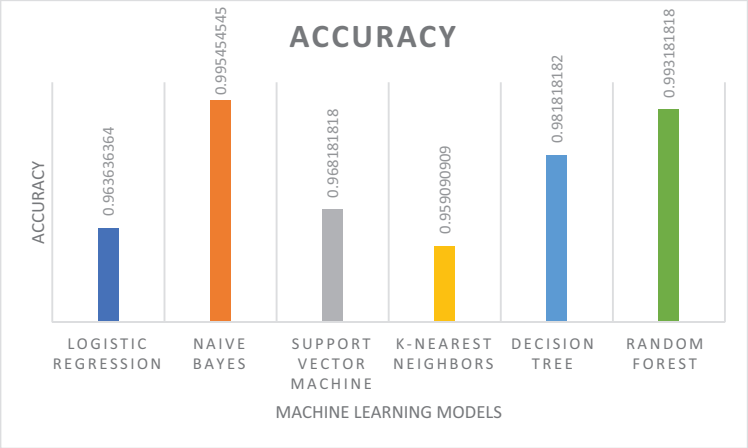


Figure 8. Accuracy Comparison

Table 2. Crop Label and Corresponding Numerical Representation

label	Crop_num
Rice	1
Maize	2
Jute	3
Cotton	4
Coconut	5
Papaya	6
Orange	7
Apple	8
Muskmelon	9
Watermelon	10
Grapes	11
Mango	12
Banana	13
Pomegranate	14
Lentil	15
Blackgram	16
Mungbean	17
Mothbeans	18
Pigeonpeas	19
Kidneybeans	20
Chickpea	21
Coffee	22

is trained on the entire dataset. A recommendation function is developed, which will allow users to input soil and climatic parameters such as nitrogen, phosphorus, potassium levels, temperature, humidity, pH, and rainfall to predict the most suitable crop for cultivation then a trained model is used, and the function simply predicts while mapping the predicted crop number against a predefined dictionary storing its corresponding crop name [14] as displayed in Table 2.

For better understanding of the performance of all models used and to make right decisions we have also generated Confusion Matrix demonstrated in Figures 9 & 9 and Classification report containing Precision, Recall and F1 score of all models (Logistic Regression, Naïve Bayes, SVM, KNN, Decision Tree, and Random Forest) [15].

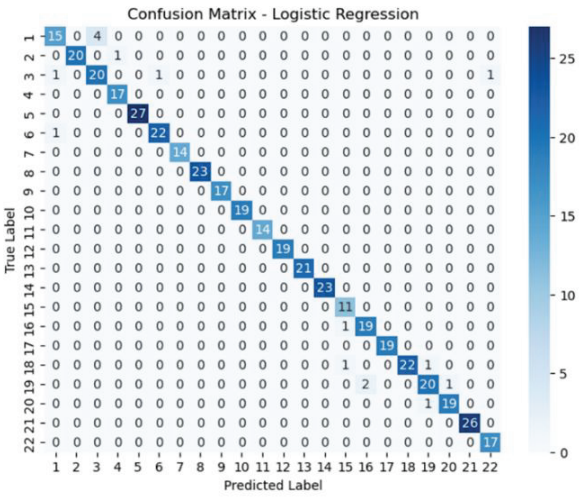


Figure 9. (a). Confusion Matrix - Logistic Regression

Classification Report:

	precision	recall	f1-score	support
1	0.88	0.79	0.83	19
2	1.00	0.95	0.98	21
3	0.83	0.87	0.85	23
4	0.94	1.00	0.97	17
5	1.00	1.00	1.00	27
6	0.96	0.96	0.96	23
7	1.00	1.00	1.00	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	0.85	1.00	0.92	11
16	0.90	0.95	0.93	20
17	1.00	1.00	1.00	19
18	1.00	0.92	0.96	24
19	0.91	0.87	0.89	23
20	0.95	0.95	0.95	20
21	1.00	1.00	1.00	26
22	0.94	1.00	0.97	17
accuracy			0.96	440
macro avg	0.96	0.97	0.96	440
weighted avg	0.96	0.96	0.96	44

Figure 9. (b). Classification Report - Logistic Regression

5. Result

The research presented here compares machine learning approaches used in crop recommendation

Table 3. Before MinMax Scaling

N	P	K	temperature	humidity	pH	rainfall
1656	17	16	14	16.396243	92.181519	6.625539	102.944161
752	37	79	19	27.543848	69.347863	7.143943	69.408782
892	7	73	25	27.521856	63.132153	7.288057	45.208411
1041	101	70	48	25.360592	75.031933	6.012697	116.553145
1179	0	17	30	35.474783	47.972305	6.279134	97.790725

Table 4. After MinMax Scaling

0.12142857	0.07857143	0.045	0.21723408	0.9089898	0.48532225	0.29685161
0.26428571	0.52857143	0.07	0.53710965	0.64257946	0.56594073	0.17630752
0.05	0.48571429	0.1	0.53647858	0.57005802	0.58835229	0.08931844
0.72142857	0.46428571	0.215	0.47446209	0.708898	0.39001747	0.34576958
0.	0.08571429	0.125	0.76468429	0.39318139	0.43145185	0.2783274

Table 5. Accuracy of ML Models

Models	Accuracy
Logistic Regression	0.9636
Naive Bayes	0.9954
Support Vector Machine	0.9681
K-Nearest Neighbors	0.9590
Decision Tree	0.9818
Random Forest	0.9931

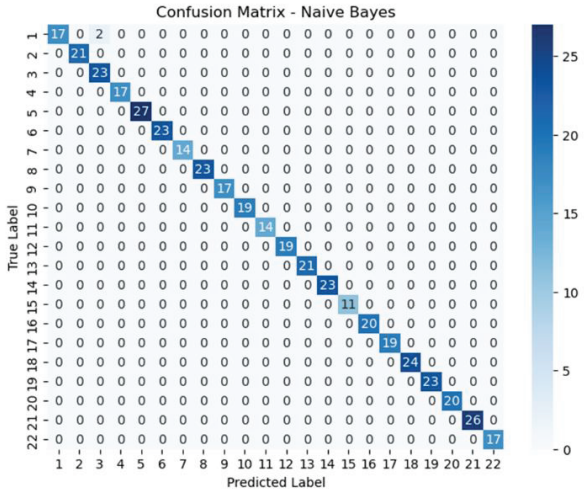


Figure 10. (a). Confusion Matrix - Naïve Bayes

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.89	0.94	19
2	1.00	1.00	1.00	21
3	0.92	1.00	0.96	23
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	27
6	1.00	1.00	1.00	23
7	1.00	1.00	1.00	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	1.00	1.00	1.00	11
16	1.00	1.00	1.00	20
17	1.00	1.00	1.00	19
18	1.00	1.00	1.00	24
19	1.00	1.00	1.00	23
20	1.00	1.00	1.00	20
21	1.00	1.00	1.00	26
22	1.00	1.00	1.00	17
accuracy			1.00	440
macro avg	1.00	1.00	1.00	440
weighted avg	1.00	1.00	1.00	440

Figure 10. (b). Classification Report - Naïve Bayes

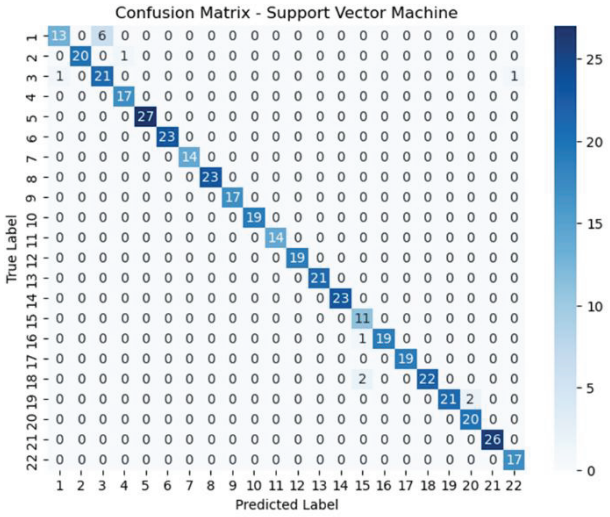


Figure 11. (a). Confusion Matrix - Support Vector Machine

systems to recommend high-yielding crops. To get the better insight of the dataset we have plotted graphs of correlation between each feature (Nitrogen, Phosphorus, Potassium, Temperature, pH and Humidity) and the label. Furthermore, each model is evaluated for accuracy using testing data. Table 3 and 4 displays the standardize and normalize the data for model compatibility, using Min-Max Scaling.

The Figure 8 shown above demonstrates that Random Forest and Naïve Bayes algorithms exhibit the utmost level of accuracy, while Logistic Regression and K-nearest Neighbors methods display the lowest level of accuracy as seen in the Table 5. Metrics such as the confusion matrix, precision, recall, and F1 score offer a more insightful analysis of the prediction results than accuracy alone. Confusion matrices display the true positive, true negative, false positive, and false negative predictions given by each model shown

Classification Report:

	precision	recall	f1-score	support
1	0.93	0.68	0.79	19
2	1.00	0.95	0.98	21
3	0.78	0.91	0.84	23
4	0.94	1.00	0.97	17
5	1.00	1.00	1.00	27
6	1.00	1.00	1.00	23
7	1.00	1.00	1.00	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	0.79	1.00	0.88	11
16	1.00	0.95	0.97	20
17	1.00	1.00	1.00	19
18	1.00	0.92	0.96	24
19	1.00	0.91	0.95	23
20	0.91	1.00	0.95	20
21	1.00	1.00	1.00	26
22	0.94	1.00	0.97	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

Figure 11. (b). Classification Report - Support Vector Machine

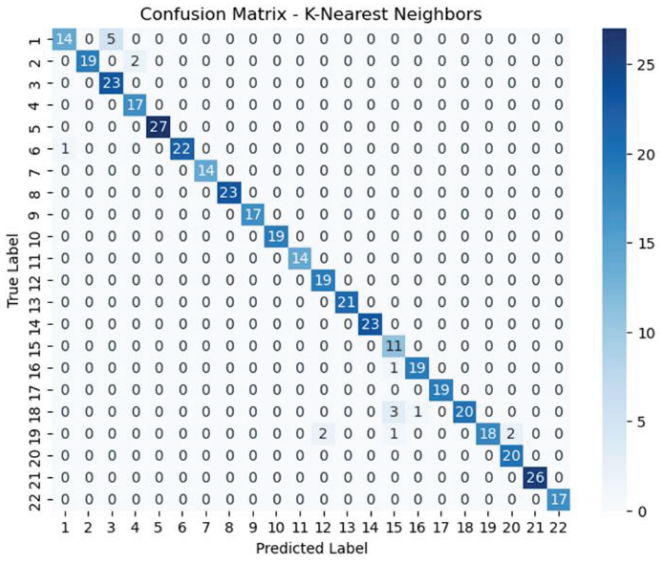


Figure 12. (a). Confusion Matrix - K-Nearest Neighbors

in the Figures 14(a) & 14(b), enabling us to evaluate the precision and efficacy of crop categorization. Additionally, the categorization reports provide a thorough evaluation of performance parameters such as precision, recall, and F1 score for each crop category. Precision assesses the model's accuracy in identifying specific crops by determining the ratio of correctly identified instances to all instances predicted for that crop, for example, correctly identifying wheat crops out of all predicted wheat instances. Recall provides crucial insights into the model's capability to capture and include all relevant events, such as ensuring all instances of a specific crop like rice are correctly identified and included in the predictions. The term refers to the proportion of true positive predictions relative to the combined number of false negative predictions

Classification Report:

	precision	recall	f1-score	support
1	0.93	0.74	0.82	19
2	1.00	0.90	0.95	21
3	0.82	1.00	0.90	23
4	0.89	1.00	0.94	17
5	1.00	1.00	1.00	27
6	1.00	0.96	0.98	23
7	1.00	1.00	1.00	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	0.90	1.00	0.95	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	0.69	1.00	0.81	11
16	0.95	0.95	0.95	20
17	1.00	1.00	1.00	19
18	1.00	0.83	0.91	24
19	1.00	0.78	0.88	23
20	0.91	1.00	0.95	20
21	1.00	1.00	1.00	26
22	1.00	1.00	1.00	17
accuracy			0.96	440
macro avg	0.96	0.96	0.96	440
weighted avg	0.97	0.96	0.96	440

Figure 12. (b). Classification Report - K-Nearest Neighbors

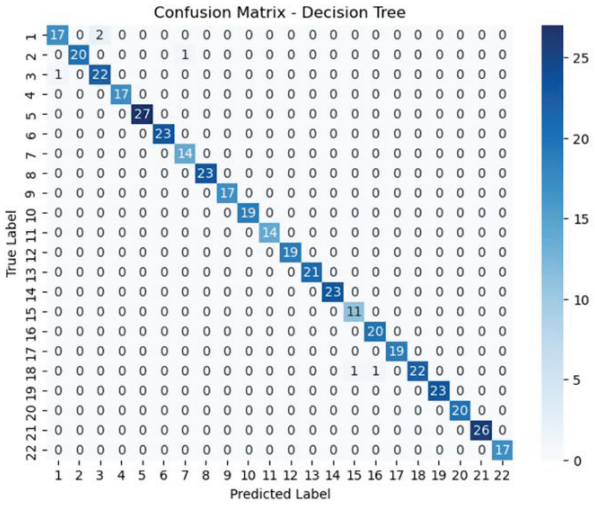


Figure 13. (a). Confusion Matrix - Decision Tree

Classification Report:

	precision	recall	f1-score	support
1	0.94	0.89	0.92	19
2	1.00	0.95	0.98	21
3	0.92	0.96	0.94	23
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	27
6	1.00	1.00	1.00	23
7	0.93	1.00	0.97	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	0.92	1.00	0.96	11
16	0.95	1.00	0.98	20
17	1.00	1.00	1.00	19
18	1.00	0.92	0.96	24
19	1.00	1.00	1.00	23
20	1.00	1.00	1.00	20
21	1.00	1.00	1.00	26
22	1.00	1.00	1.00	17
accuracy			0.99	440
macro avg	0.98	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 13. (b). Classification Report - Decision Tree

and true positive predictions for each category. Taking the weighted harmonic mean of precision and accuracy, one can calculate the F1 score. The number of actual occurrences of the class in the provided dataset is referred to as support. Consequently, we have generated a classification report and visualization for each model. To obtain the most accurate forecast, it is essential to utilize above mentioned metrics provided in the analysis [16].

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{7}$$

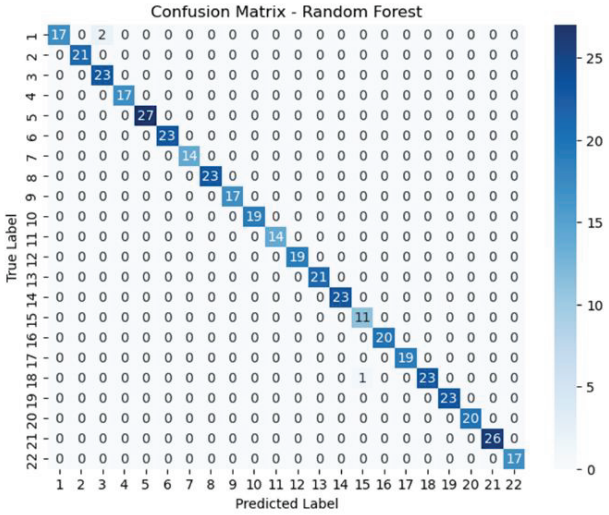


Figure 14. (a). Confusion Matrix - Random Forest

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.89	0.94	19
2	1.00	1.00	1.00	21
3	0.92	1.00	0.96	23
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	27
6	1.00	1.00	1.00	23
7	1.00	1.00	1.00	14
8	1.00	1.00	1.00	23
9	1.00	1.00	1.00	17
10	1.00	1.00	1.00	19
11	1.00	1.00	1.00	14
12	1.00	1.00	1.00	19
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	23
15	0.92	1.00	0.96	11
16	1.00	1.00	1.00	20
17	1.00	1.00	1.00	19
18	1.00	0.96	0.98	24
19	1.00	1.00	1.00	23
20	1.00	1.00	1.00	20
21	1.00	1.00	1.00	26
22	1.00	1.00	1.00	17
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Figure 14. (b). Classification Report - Random Forest

6. Conclusion and Prospective Intent

The proposed study will assist farmers in increasing agricultural output, reducing soil degradation in cultivated fields, and using less fertiliser during crop production by suggesting the optimal crop amount to plant based on a wide variety of criteria. The suggested activity helps farmers sustainability by helping them choose the right crops to cultivate. We have identified the advantages and disadvantages of each model through comparative analysis, providing important information for decision-making in the agricultural sector. We can improve the system later by adding more features to the dataset. Furthermore, with the support of remote sensing technologies and IoT devices, real-time monitoring can be made possible which would allow the system to recommend crops along with climate change. To make the project beneficial to farmers in every corner of our nation, we may also incorporate all regional languages.

Declarations

Funding None

Conflicts of interest The authors confirm that they have no competing interests or conflicts of interest.

Code availability The code for implementation is available upon request, subject to privacy and other restrictions.

Technology used Python programming, Machine learning libraries

AUTHORS

Sanket Gupta – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: sanket.jec@gmail.com.

Trishna Panse – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: trishnapanse@gmail.com.

Kailash Chandra Bandhu – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: kailashchandra.bandhu@gmail.com.

Ratnesh Litoriya – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: litoriya.ratnesh@gmail.com.

Shivani Patnaha – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: patnahashivani45@gmail.com.

Divya Kumawat – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: ddoraya@gmail.com.

Lishika Pargi – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: lishi1146@gmail.com.

Tisha Modi* – Department of Computer Science and Engineering, Medi-Caps University, Indore, India, e-mail: tishamodi0212@gmail.com.

*Corresponding author

References

[1] D. Pal, "Changes that have Taken Place in the Indian Economy after 1951", Dec. 2015.

[2] R. P. Reddy, B. Vinitha, K. Rishita, and K. Pranavi, "Crop Monitoring and Recommendation System using Machine Learning and IOT", *International Journal of Innovative Technology and Exploring Engineering*, 2020, DOI: 10.35940/ijitee.I7008.079920.

[3] Z. Doshi, S. Nadkarni, R. Agrawal, and N. Shah, "AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms", 2018, DOI: 10.1109/ICCUBEA.2018.8697349.

[4] Srilakshmi A., Madhumitha K., and Geetha K, "Machine learning approach: Recommendation of suitable crop for land using meteorological factors", *SSRN Electronic Journal*, 2020, Elsevier BV. DOI: https://doi.org/10.2139/ssrn.3736550

[5] S. Mamatha jajur, Soumya N.G., G.T. Raju, "Crop Recommendation using Machine Learning Techniques", International *Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2S, 2019, 658–661.DOI: 10.35940/ijitee.b1106.1292s19.

[6] D. A. Bondre and M. Santosh Mahagaonkar, "PREDICTION OF CROP YIELD AND FERTILIZER RECOMMENDATION USING MACHINE LEARNING ALGORITHMS," 2019, DOI: 10.33564/IJEAST.2019.v04i05.055

[7] D. A. Reddy, B. Dadore, and A. Watekar, "Crop Recommendation System to Maximize Crop Yield in Ramtek region using Machine Learning," *Int J Sci Res Sci Technol*, 485–489, Feb. 2019, DOI: 10.32628/ijsrst196172.

[8] N. H. Kulkarni, G. N. Srinivasan, B. M. Sagar, and N. K. Cauvery, "Improving Crop Productivity Through A Crop Recommendation System Using Ensembling Technique". in *2018* 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)., 2018, 114–119. DOI: 10.1109/CSITSS.2018.8768790.

[9] Jabed, Md & Murad, Masrah, "Crop Yield Prediction in Agriculture: A Comprehensive Review of Machine Learning and Deep Learning Approaches, with Insights for Future Research and Sustainability", 2024 DOI: 10.1016/j.heliyon.2024.e40836

[10] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika, and J. Nisha, "Crop recommendation system for precision agriculture," in 2016 Eighth International Conference on Advanced Computing (ICoAC)., 2017, 32–36. DOI: 10.1109/ICoAC.2017.7951740.

[11] R. Kumar, M. Singh, P. Kumar, and J. Singh, "Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique", 2015, DOI: 10.1109/ICSTM.2015.7225403.

[12] Ramu, K Priyadarsini, K, "A Review on Crop Yield prediction Using Machine Learning Methods", 2021, DOI: 10.1109/ICOSEC51865.2021.9591764

[13] F. Shahrin, L. Zahin, R. Rahman, A. J. Hossain, A. H. Kaf and A. K. M. Abdul Malek Azad, "Agricultural Analysis and Crop Yield Prediction of Habiganj using Multispectral Bands of Satellite Imagery with Machine Learning", 2020 11th International Conference on Electrical and Computer Engineering (ICECE), 21-24, 2020, DOI: 10.1109/ICECE51571.2020.9393066

[14] M. Kavita and P. Mathur, "Crop Yield Estimation in India Using Machine Learning", 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 220-224, 2020, DOI: 10.1109/ICCCA49541.2020.9250915

[15] P. S. Nishant, P. Sai Venkat, B. L. Avinash and B. Jabber, "Crop Yield Prediction based on Indian Agriculture using Machine Learning", 2020 International Conference for Emerging Technology (INCET)., 1–4, 2020, DOI: 10.1109/INCET49848.2020.9154036

[16] P.S. Maya Gopal and R. Bhargavi, "Optimum Feature Subset for Optimizing Crop Yield Prediction Using Filter and Wrapper Approaches", Applied Engineering in Agriculture., vol. 35, no. 1, 2019, 9-14. DOI: https://doi.org/10.1016/j.compag.2019.104968

Ensemble Learning for Face Recognition in Suspect Identification using Cloud Environment

Submitted: 21st August 2024; accepted: 24th September 2024

Shilpa Chaudhari, Rajarajeswari S, Archana Rane

DOI: 10.14313/jamris-2026-022

Abstract:
Facial recognition technology finds applications in security, surveillance, and social media. Existing research explores the use of machine learning and deep learning for face recognition, emphasizing the need for improved accuracy. This paper proposes a system for suspect identification using facial recognition. The system leverages ensemble learning by integrating seamlessly with OpenAI's advanced technologies and is supported by a robust cloud infrastructure. The comparison of the proposed ensemble model to individual models like VGG-Face, Facenet, Facenet512, Deepface, DeepID, ArcFace, and SFace uses multiple detectors and the Labelled Faces in the Wild (LFW) dataset. The results show that the ensemble model offers the most efficient processing time across all sample sizes. In contrast, models like VGG-Face and DeepID exhibit a steeper increase in processing time, suggesting lower scalability. For instance, at a sample size of 50, the local test completes in 61.3 seconds, while the cloud API test takes 67.2 seconds. This highlights the faster processing speed of the local test across all sample sizes. FaceNet, VGG-Face, and ArcFace models are chosen in ensemble model where in all of them have accuracy above 95% in every face detector test. Facenet512 model has 98.4% among the selected ensemble model whereas ensemble of these models shows 98.8 accuracy.

Keywords: *Face recognition, Accuracy test, Deep learning, ensemble learning, Cloud API*

1. Introduction

Facial recognition technology has become a powerful tool with applications in social media [19], security [20], and surveillance [18]. The traditional method of using hand-drawn sketches for suspect identification is labor-intensive, time-consuming, and often a bottleneck in the investigation process. Despite the spread of modern recognition techniques, the inefficiencies inherent in manual sketching processes—limited scalability, significant time investment, and resource constraints—have hindered the timely and effective identification of suspects [21]. These challenges underscore the urgent need for a revolutionary application that transcends the selection of individual facial features to enable rapid, comprehensive face sketch recognition. Existing research explores various machine learning and deep learning models for face recognition [22, 23].

The proposed method of this paper identifies seven prominent models: VGG-Face [24], FaceNet [25], FaceNet512, DeepFace [26], DeepID, ArcFace [28], and Sface [27]. These models are trained on large datasets (like Labelled Faces in the Wild (LFW)) to achieve high accuracy (above 90%). VGG-Face learns distinctive features for the face recognition process. FaceNet maps face images to high-dimensional features for similarity recognition. FaceNet512 utilizes a higher dimensional space for improved face recognition performance. DeepFace extracts facial features for multi-task learning encompassing face detection, alignment, and recognition. DeepID learns hierarchical representations of faces for high accuracy in face verification and identification. ArcFace and Sface enhance the discriminative power within the face feature space through specific loss functions during training. These models are widely used and have significantly contributed to advancements in face recognition. Their performance varies based on factors like datasets, evaluation protocols, and application (in terms of accuracy, precision, recall, and F1-score).

Ensemble learning combines multiple models to surpass the performance of any single model. It aggregates the diverse predictions from chosen models, leveraging the "wisdom of the crowd." The proposed face recognition model utilizes ensemble learning to improve its performance by harnessing the collective strengths of the seven models while mitigating individual weaknesses. Cloud environments offer convenient data storage and retrieval from anywhere. Services like Amazon S3 provide an interface for cost-effective retrieval of large datasets. Combining these S3 buckets with AWS segmentation services ensures authorized user access to specific data stored within the bucket.

The proposed system designs and develops a recognition model using machine and deep learning-based ensemble learning by leveraging advanced facial recognition against criminal databases with efficient cloud-based storage, management, and security of suspect faces. This approach ensures scalability, reliability, and cost-effectiveness for face recognition application workloads. It establishes a cloud-based centralized database to facilitate easy global access to crucial information for law enforcement agencies. The developed model is tested on AWS cloud for

portability using the respective APIs. The integration of deep learning algorithms and cloud infrastructure for database matching and verification, aiming at a significant improvement in the efficiency of the identification process. Our specific contributions include the following: (1) design and development of recognition model using machine and deep learning-based ensemble learning; (2) establishment of a cloud-based centralized database to facilitate easy global access; (3) ensuring the scalability, reliability, and cost-effectiveness for face recognition application workloads using S3 buckets with AWS segmentation services; and (4) a performance analysis of the developed face recognition model.

This paper is structured as follows: Section II discusses related work on face recognition using deep learning. Section III details the proposed methodology for ensemble learning in a cloud environment. Section IV presents result analysis, followed by the conclusion in Section V.

2. Related Works

The Dual-Scale Markov Network (DSMN) and Multi-Information Fusion algorithm [1] represents an innovative contribution to the field of face sketch-photo synthesis and recognition considering diverse facial features and variations. The algorithm integrates information from multiple sources, enhancing the synthesis and recognition accuracy. The accuracy of the DSMN relies on the quality of the initial sketch. If the sketch is poor, the synthesized photo or recognition accuracy suffers. The information fusion stage, might be computationally expensive, making it challenging for real-time applications.

The authors of [2] achieved promising performance in matching composite sketches generated by eyewitnesses, to mugshot photographs. It relies on three key techniques: Active Shape Model (ASM) for pinpointing facial landmarks on both the sketch and photo, Multiscale Local Binary Patterns (MLBP) to extract distinctive features from each facial component, and component similarity fusion to determine the overall match between the sketch and photo. This component-based approach achieved significant improvements when compared to a leading commercial face recognition system and a simpler method that analyzed the entire face as a single unit. The promising performance suggests that this method has the potential to be a valuable tool for law enforcement agencies in identifying and apprehending suspects.

A comprehensive survey of various 3D face reconstruction techniques is explored in [3], including deep learning, epipolar geometry, one-shot learning, 3D morphable models, and shape-from-shading methods. It delves deeper into the analysis of deep learning-based reconstruction techniques, dominant technique with high accuracy, detail, and robustness. Reconstructing faces from historical photographs or rare medical scans might be challenging due to the scarcity of training data from similar domains. It may require additional cues like depth information for better accuracy for handling complex lighting condition.

The role of facial reconstruction (FFR) identifies unknown individuals by utilizing a combination of scientific principles and artistic skills to recreate a likeness of the deceased based on their skeletal remains [4]. The process involves employing various techniques such as 2D/3D computer-aided methods, manual sculpting, and even clay modeling. An important aspect of FFR is the understanding of facial tissue depth variations those are influenced by factors such as age, sex, and ancestry. However, the accuracy of FFR is currently hampered by limitations in the existing database on tissue depth variations, particularly for non-caucasian populations. This highlights the need for further research and data collection to improve the accuracy of FFR for a wider range of demographics.

An approach to facial image editing [5] leverages a hybrid Convolutional Neural Network (CNN) architecture to manipulate specific facial attributes. Notably, it incorporates a pre-trained facial recognition model to extract key features from the image. This allows the framework to edit aspects like age, gender, expression, and hair color while maintaining a realistic appearance and preserving the underlying facial identity. The current study focuses on editing faces of Asian descent. Further research is needed to determine the generalizability of this approach to a wider range of ethnicities.

3D face reconstruction leverages a neural network to not only predict the 3D face shape but also assess the confidence of the reconstruction [6]. The authors demonstrate that their approach surpasses shape-averaging techniques in terms of reconstruction accuracy, particularly on the MICC dataset. The neural network tends to favor high-quality face images for reconstruction, specifically those with frontal poses, clear visibility, and natural lighting conditions. Conversely, images containing occlusions like sunglasses, hats, or hair can lead to decreased confidence scores in the reconstruction process. This highlights the need for the model to be more robust to variations in image quality and pose for real-world applications.

The study of [7] investigates eye-tracking data of participants viewing freehand sketches. Interestingly, the research identified consistent patterns in how people fixate on various parts of the sketch, both within individual sketches and across sketches of the same category or related groups using a sketch-specific data augmentation technique. This method significantly improves the accuracy of deep learning models in recognizing freehand sketches.

The automation of facial composite production and identification processes focuses on EvoFIT system, a software program that allows witnesses to build facial composites by selecting features from a database [8]. The study compared a standalone version of EvoFIT, which does not require operator guidance, to the full system with a human operator using the Shape Tool. They evaluated the resulting composites using a root mean square error (RMSE) measure

to assess their similarity to the target face, the study explored the potential for matching composites generated using EvoFIT against a database of other composites.

Coupled Information-Theoretic Encoding (CITE) utilizes Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) alongside a Randomized CITE Tree Algorithm to extract informative features from different modalities: photos and sketches [9]. These features are then fed into a Linear Support Vector Machine (SVM) for classification. Notably, the CITE descriptors outperform popular facial recognition features like Local Binary Patterns (LBP) and Scale-Invariant Feature Transform (SIFT). CITE method significantly improves verification rates at low false acceptance rates compared to existing methods. This improvement is further enhanced by incorporating PCA, LDA, and SVM for information fusion, solidifying CITE as a powerful approach for face photo-sketch recognition.

CITE allows the system to capture the essence of facial structure despite the inherent differences between these image types [19]. The technique of synthesizing pseudo photos from query sketches and using RS-LDA for matching demonstrated certain limitations, especially when dealing with significant shape distortion between photos and sketches in the training set.

A real-time deep neural network architecture called DiFRuNNT for disguised face verification [11] consists of two neural networks: CNN to predict 20 facial key points in the image, and a secondary network to classify subjects based on angles and ratios calculated from these predicted points. The achieved accuracies are 67.4% for prediction and 74.8% for classification, respectively.

Conceptual categorization and evaluation metrics given in [12] provides comprehensive survey of relevant publications on face recognition systems under morphing attacks as well as discusses technical considerations, tradeoffs, open issues, and challenges in the field.

Review of deep learning methods in face recognition covers various deep learning architectures, loss functions, databases, protocols, and application scenes [13]. It highlights the rapid evolution and significant impact of deep learning on face recognition, discussing challenges and promising directions.

Discriminative approaches without explicit age modeling achieves excellent performance in face verification across age progression [14]. They find that gradient orientation, particularly in a hierarchical structure called the gradient orientation pyramid (GOP), combined with SVMs, achieves excellent performance. Empirical study on age gaps impact on recognition algorithms, providing insights into age-related challenges.

The conventional pipeline for face recognition involves four stages: detect, align, represent, and classify for improving the alignment and representation steps by incorporating explicit 3D face modeling to apply a piecewise affine transformation. They derive a face representation using a nine-layer deep neural network with over 120 million parameters, employing locally connected layers without weight sharing. Trained on a massive facial dataset containing over four million images from more than 4,000 identities, their method couples accurate model-based alignment with a large facial database, yielding remarkable generalization to faces in unconstrained environments. Even with a simple classifier, their approach achieves an accuracy of 97.35% on the LFW dataset, reducing the error of the current state of the art by over 27% and approaching human-level performance closely.

2D-to-3D integrated face reconstruction approach significantly improves accuracy of face recognition with changing pose, illumination, and expression (PIE) [16]. It offers an efficient and automatic framework for 3D face reconstruction and recognition, overcoming challenges of variant factors like PIE. Experimental results show that their synthesized virtual faces significantly enhance recognition accuracy, especially when dealing with changes in PIE conditions.

Identity-Preserving Face Recovery from Portraits (IFRP) recover photorealistic faces from artistic portraits while preserving identity poses a significant challenge due to potential distortions or loss of fine details. It comprises two main components: the Style Removal Network (SRN) and the Discriminative Network (DN). SRN and DN recover latent photorealistic faces while preserving identity. It introduces a method for recovering realistic faces from unaligned stylized portraits while preserving identity, achieving state-of-the-art results.

3. Proposed Face Recognition

Face recognition process flow involves a user, recognition model, AWS segmentation, and an S3 storage component as shown in Figure 1. The sequence begins with the user providing an image, which is then fed into the recognition model and uploaded to an AWS S3 bucket for storage. The recognition model initiates a recognition task by retrieving the image from AWS S3, facilitated by the AWS Segmentation service that performs pre-processing tasks like image segmentation or feature extraction. Once the recognition process is completed, the model calculates a match percentage and generates metadata related to the image, sending these results back to the user. This workflow integrates cloud storage and processing, with the core functionality residing in the deep learning model deployed on the AWS Lambda cloud platform for on-demand execution. The model focuses on extracting key features from the input image, particularly those corresponding to facial elements like the eyes, nose, and mouth. By comparing these extracted features against a database of facial images, the model attempts to identify potential matches between the sketch and real suspects.

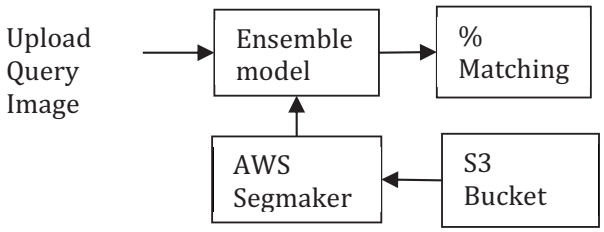


Figure 1. Face Recognition Process

The recognition module receives results from the deep learning model, which typically include similarity scores for each potential suspect in the database, indicating how closely the sketch resembles a particular suspect's facial features. Acting as a bridge between the user-query image and the suspect database, the recognition module leverages the power of deep learning to identify potential matches based on facial recognition. The accuracy of seven existing algorithms in Deepface, including VGG-Face, FaceNet, FaceNet512, DeepFace, DeepID, ArcFace, and SFace, is tested. This process enhances recognition tasks by effectively leveraging cloud computing capabilities for scalable and efficient image analysis. The evaluation begins with meticulous dataset selection. Each image undergoes standard preprocessing steps such as normalization of size and color intensity, and alignment using detectors like Retinaface, MTCNN, fast MTCNN, dlib, yolov8, yunet, centerface, mediapipe, ssd, and OpenCV. Testing of the seven models involves comparing an input image to each existing image, with predictions recorded to calculate the models' accuracy. These seven models run in an ensemble model processed in parallel for the same input image. The most frequently occurring output image generated by each individual model is then put through a voting system, where the mode of all output images is taken as the final result. Given that each facial recognition model was trained on different datasets by different developers at different times, varying results are expected. Thus, this method of taking the mode value of the results provided by each model ensures greater accuracy and consistency. A facial recognition model broadly works on the same pipeline as shown in Figure 2.

Facial Recognition Process Flow is as follows: (1) Image Input: The process begins with an image provided by the user as shown in Figure 2; (2) Storage: The image is uploaded to a designated storage location, an Amazon S3 bucket in this case; (3) Image Preprocessing: The Amazon Segmentation service retrieves the image from storage and performs preprocessing tasks like segmentation or feature extraction; (4) Recognition Model: A deep learning model, deployed on the AWS Lambda platform, analyzes the preprocessed image. The model extracts key features from the image, particularly those corresponding to facial elements. These features are compared against a database of facial images to identify potential matches; (5) Results Generation: The model calculates a match percentage for each potential match in the database. Metadata related to the image is also generated; and (6) Output: The results, including match percentages and image metadata, are sent back to the user.

Following Deep Learning Model Details are considered. (1) The core functionality lies in a deep learning model trained to extract facial features. (2) The model is evaluated against various existing algorithms to ensure optimal accuracy. (3) An ensemble model approach is used, where multiple models process the image in parallel and the final result is determined through a voting system.

This workflow leverages cloud storage (S3) and processing (Lambda) for scalable and efficient image analysis. It demonstrates a typical facial recognition pipeline where an image is uploaded, processed, analyzed for facial features, compared against a database, and results are delivered.

3.1. Face Recognition and Alignment

Face detection involves identifying and aligning faces within an image. Alignment is straightforward once the face and eyes are detected. Various algorithms used for face detection are as follows: (1) RetinaFace: Discusses its architecture and the specific features that enable RetinaFace to handle different scales and orientations in face detection; (2) MTCNN: Explains the multi-task cascaded framework and its efficiency in detecting detailed face attributes alongside face detection; (3) FastMTCNN: Focuses on the optimizations that make FastMTCNN a faster alternative to MTCNN while maintaining comparable accuracy; (4) Dlib: Compares the HOG+SVM-based approach and the CNN-based detector in Dlib, highlighting scenarios where each is preferable; (5) YOLOv8: Describes how YOLOv8 adapts the YOLO object detection framework for fast and effective face detection; (6) YuNet: Provides details on YuNet's architecture and its effectiveness in real-time face detection applications; (7) CenterFace: Analyzes the method's approach to detecting face centers and scales, particularly in crowded environments; (8) MediaPipe: Explores the integration of MediaPipe's face detection in multimodal pipelines and its real-world applications; (9) SSD: Discusses the application of the SSD framework for face detection and its performance across various datasets; and (10) OpenCV: Outlines the use of Haar feature-based cascade classifiers and their suitability for entry-level face detection tasks. A detector detects an image and aligns it as shown in Figure 3.

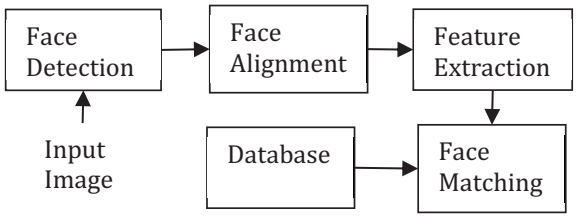


Figure 2. Face Recognition Stages

Figure 3. Facial Detection and Alignment

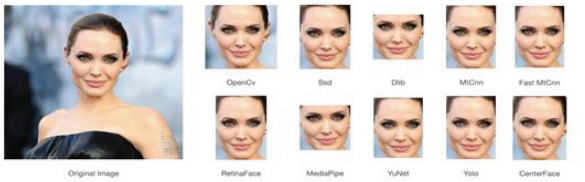


Figure 4. Facial Detection by Different Detectors

Figure 5. Nodal Points on Two Different People

Subsequently, different detectors perceive faces in distinct ways. For instance, Figure 4 illustrates how each of the previously mentioned detectors perceives the same face from a single image. While each cropped image includes the entire face, the variation lies in the amount of background retained in the image.

3.2. Feature Extraction

Features such as eyes, nose, mouth, etc. are extracted from the aligned face. This is achieved by using nodal points [14]. As seen in Figure 5, the ratio of the nodal points of both the humans are different, it is not possible for them to be similar unless they are identical twins.

Face Representation: Each face is represented as a high-dimensional vector in a feature space. This vector is obtained from the output of a deep neural network (e.g., VGG Face, FaceNet) that has been trained to map facial images into a compact embedding space.

3.3. Feature Matching

Encoded representations are compared to a database or gallery of known identities, using methods such as cosine similarity and Euclidean distance to determine the similarity between two vectors in a multi-dimensional space.

Distance Metric: To determine how similar or dissimilar two facial feature vectors are, the Euclidean distance between them is computed. For two feature vectors x and y, each of dimension n, the Euclidean distance d is calculated using the formula given in Equation 1 where *Sum of Squared Differences* is computed as given in Equation 2.

$$Euclidean\ distance: d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

$$\text{Sum of Squared Differences} = \sum_{i=1}^{n}(x_i - y_i)^2 \quad (2)$$

Cosine Similarity in Face Recognition

1. Direction Over Magnitude: Cosine similarity focuses on the orientation of the vectors rather than their magnitude. This is useful for face recognition because it measures how similar the patterns of the features are, regardless of their scale. This can help in scenarios where the length of the feature vector may vary due to different factors like the scale of images or variations in lighting conditions.
2. Normalization: Cosine similarity normalizes the feature vectors, making it robust to variations in the length of the vectors. This normalization can be beneficial when comparing face embeddings that may be affected by varying image conditions.
3. Similarity Measurement: Cosine similarity is used to compute the similarity score between feature vectors. A higher cosine similarity indicates that the vectors are more similar, which can be interpreted as the faces being more likely to belong to the same individual.

Feature Vectors: For two facial feature vectors x and y, each of dimension n, the cosine similarity is defined as given in Equation 3.

$$\text{Cosine Similarity}(x.y) = \frac{x.y}{||x||\ ||y||} \quad (3)$$

where:

1) x·y is the dot product of the vectors.
2) ‖x‖ and ‖y‖ are the magnitudes (or norms) of the vectors.

Dot Product: The dot product x·y is calculated as given in Equation 4.

$$x \cdot y = \sum_{i=1}^{n} x_i.y_i \quad (4)$$

where xi and yi are the components of vectors x and y, respectively.

Vector Norms: The norm (or magnitude) of a vector x and y is calculated as given in Equation 5.

$$||x|| = \sqrt{\sum_{i=1}^{n} x_i{}^2} \ and \ ||y|| = \sqrt{\sum_{i=1}^{n} y_i{}^2} \qquad (5)$$

Combining these components, the cosine similarity is computed as given in Equation 6.

$$\text{Cosine Similarity}(x, y) = \frac{\sum_{i=1}^{n} x_i . y_i}{\sum_{i=1}^{n} x_i{}^2} \sum_{i=1}^{n} y_i{}^2 \qquad (6)$$

This value ranges from -1 to 1, where 1 indicates that the vectors are identical in direction, 0 indicates orthogonality (no similarity), and -1 indicates opposite directions.

The ensemble model includes seven recognition models, each running simultaneously. The focus on each model individually is given as follows. (1) VGG-FACE: The VGG Face model extracts discriminative features from facial images, significantly enhancing facial recognition technology. Trained on a vast dataset, these networks learn complex feature hierarchies crucial for facial identification, effectively handling varying expressions and lighting conditions. VGG Face emphasizes features critical for accurate individual identification, supported by rigorous training and optimization. Additionally, it employs a joint Bayesian framework to model these features' distribution, enabling robust verification of face pairs in diverse real-world scenarios. This approach ensures VGG Face's reliability and effectiveness in high-security applications, setting a new standard in the field of facial recognition.

(2) ARCFACE: ArcFace is an advanced facial recognition system distinguished by its innovative use of an angular margin penalty added to the loss function during training. This method maximizes the distinctiveness between the learned features of different individuals (inter-class discrepancy) while maintaining consistency in features of the same individual across various images (intra-class compactness). ArcFace is trained on extensive datasets containing a wide array of facial images, allowing the neural networks to effectively learn and abstract complex hierarchies of facial features into higher layers. By integrating the angular margin penalty, ArcFace enhances feature embeddings by decisively separating embeddings of different classes and bringing those of the same class closer together. This mechanism significantly improves the discriminative power of the model, making it exceptionally capable of handling challenges in facial recognition, such as variations in lighting, expression, and other dynamic environmental factors.

(3) FACENET: FaceNet, developed by Google, utilizes deep CNNs to map face images into a compact Euclidean space where distances represent facial similarity. This model measures similarity by calculating the distance between points representing faces. The innovation of FaceNet lies in its use of the triplet loss function during training, which minimizes the distance between an anchor image and a positive image (same person) while maximizing the distance between the anchor and a negative image (different person). This approach enables FaceNet to accurately distinguish between individuals under varying conditions such as changes in expression, lighting, and camera angles. Its ability to maintain high accuracy despite these variations makes it highly effective for applications requiring reliable facial recognition.

(4) Facenet512 has a high value in accuracy calculation compared to Facenet.

(5) DEEPFACE: Deepface [15] Developed by Facebook, DeepFace represents a significant advancement in facial recognition technology. It utilizes a sophisticated nine-layer neural network with over 120 million connection weights, trained on an extensive dataset comprising four million facial images from more than 4,000 distinct identities. This system employs a novel approach by aligning faces using three-dimensional models, allowing it to adjust for variations in head position, orientation, and lighting conditions before processing the images through its deep neural net. The core functionality of DeepFace lies in extracting and utilizing detailed facial features from aligned images. It learns a compact representation of each face, simplifying the comparison and identification of faces across various conditions. By focusing on these representations, DeepFace achieves a high level of accuracy in facial recognition tasks, effectively capturing and analyzing subtle facial features crucial for reliable identification.

(6) DEEPID: DeepID represents a significant advancement in facial recognition technologies, focusing on the extraction and utilization of deep hidden identity features. Developed at The Chinese University of Hong Kong, DeepID employs deep CNNs to learn a hierarchy of features from a substantial dataset of facial images. These features, which become increasingly abstract at higher layers of the network, enhance the discriminative power of the model, making it particularly effective in distinguishing between different identities. Additionally, DeepID's application of a joint Bayesian framework to model these features allows for more accurate face pair verification, providing robustness against typical variations encountered in real-world scenarios, such as changes in expression and lighting conditions. This approach improves the accuracy of facial recognition systems and extends their applicability in security and personal identification, addressing pressing challenges faced by current technologies.

(7) SFACE: SFace is a lesser-known term in the context of popular face recognition technologies and might refer to a specific implementation or model within the research community. If it follows the conventions of other deep learning-based face recognition systems, SFace would likely employ a deep neural network to learn a representation of facial features useful for recognition tasks. The "S" could potentially stand for a specific feature of the model, such

as secure, simple, or scalable, indicating a focus on those aspects of the facial recognition process. Since SFace is not a widely recognized term like DeepFace or DeepID, the specifics of its architecture and functionality would depend on the context in which it is referenced, including the particular research paper or implementation that defines it.

4. Results and Discussion

The development environment leverages AWS as the cloud infrastructure to achieve scalability, elasticity, and potential cost reductions. The following AWS services are considered for implementation. (1) Amazon S3: For storing the facial image dataset and potentially the application code. (2) AWS Lambda: To execute the deep learning model for facial recognition in a serverless environment, enabling on-demand processing without server management. (3) Amazon SageMaker: As a managed service for building, training, and deploying machine learning models. SageMaker may be explored for deep learning model development as needed. (4) Amazon EC2 Instance: If AWS Lambda proves insufficient for deep learning tasks, an EC2 instance can be considered for model training or inference.

The system requires integration with a deep learning library (e.g., TensorFlow, PyTorch) to facilitate facial feature recognition, matching, and image generation capabilities. Deep learning serves as the core technology underpinning facial recognition functionality. Pre-trained deep learning models are utilized to analyze constructed sketches and extract relevant features for comparison against a facial image database. A deep learning model for facial recognition is to be developed using TensorFlow or PyTorch and trained on a dataset of facial images with labeled features. Model training can be executed on an EC2 instance or potentially through SageMaker.

A database API interface is used to connect to a criminal database containing suspect information and facial recognition data. Snowflake serves as the database management system for storing suspect information and potentially sketch data, providing a scalable and secure solution.

This study compares the accuracy of several models, including Facenet512, Facenet, VGG-Face, ArcFace, SFace, DeepFace, and DeepID, using the LFW dataset. The outcome of this research is a comparison of the accuracy of each model. Multiple face detectors, such as Retinaface, MTCNN, fastMTCNN, dlib, yolov8, yunet, centerface, mediapipe, ssd, and OpenCV, are used for testing to determine the best-performing model on the dataset [1]. The testing method involves verifying the accuracy of each model by comparing each image with another, yielding two outcomes: threshold and distance. Threshold values can be modified based on the model: 0.68 for VGG-Face, 0.4 for FaceNet, 0.3 for FaceNet512, 0.23 for DeepFace, 0.015 for DeepID, 0.68 for ArcFace, and 0.593 for SFace. Depending on the threshold, the distance value determines whether the result is true or false; if the distance exceeds the threshold, the result is false, and vice versa.

After testing, the results shown in Table 1 indicate that the Cosine metric was used. The results demonstrate that the FaceNet512 model achieved the highest accuracy value of 0.984 or 98.4% with the RetinaFace [11, 17] detector and surpassed all other models in every detector. The FaceNet model achieved an accuracy value of 0.974, the VGG-Face model achieved 0.960, the ArcFace model achieved 0.967, the SFace model achieved 0.924, the DeepFace model achieved 0.677, and the DeepID model achieved 0.659.

The Euclidean metric was used to obtain the test results in Table 2. The results indicate that the FaceNet512 model achieved the highest accuracy value of 0.976 or 97.6% with the centerface detector, outperforming all other models in every detector. The highest accuracy values for other models are as follows: FaceNet model at 0.938, VGG-Face model at 0.960, ArcFace model at 0.886, SFace model at 0.814, DeepFace model at 0.690, and DeepID model at 0.665.

From these results, it can be concluded that the Facenet512 model is superior to every other model in terms of accuracy.

Cosine similarity distances of various facial recognition models when tested with a set of 20 and 50 sample images. Is given in Table 3. Lower values indicate closer or more accurate matches. For 20 sample images, ArcFace demonstrates the best performance with the lowest distance at 0.012, suggesting highly accurate model predictions. In contrast, VGG-Face has the highest distance at 0.46, indicating less precision. Other models like Facenet, Facenet512, Sface, DeepFace, and Ensemble exhibit varied performances, with distances ranging between 0.12 and 0.46.

For 50 sample images, VGG-Face starts with the highest distance at 0.42, indicating less precision. Facenet512 shows the best performance with the lowest distance at 0.17. The distances for other models like Deepid, ArcFace, Sface, DeepFace, and Ensemble range from 0.10 to 0.31, reflecting varying levels of accuracy across these models.

Further, Table 4 compares the highest accuracy obtained from each model from these research results and the accuracy of the models previously studied by the creators. The accuracy obtained is lower than what is been declared for models, this occurred because there were differences in the type of dataset used where models used the LFW dataset for the training process as this study used a celebrity dataset of varying ages and ethnicity. This condition has occurred in previous studies where racial differences in the dataset affected the level of accuracy [7, 13]. Ensemble model shows 98.8 accuracy.

The time taken by various facial recognition models (VGG-Face, Facenet, Facenet512, Deepid, Arc Face, Sface, Deepface, Ensemble) as a function of sample size, ranging from 10 to 150 samples is shown in Figure 6. All models show increasing time with larger sample sizes. The Ensemble model is the most time-efficient across all sample sizes, while models like

Table 1. Cosine metric accuracy

Detector	FaceNet512	Face Net	VGG-Face	ArcFace	SFace	DeepFace	DeepID
retinaface	98.4	96.4	95.8	96.6	92.4	67.7	64.4
mtcnn	97.6	96.8	95.9	96	90.5	66.3	63
fastmtcnn	98.1	97.2	95.8	96.4	90	67.4	63.6
dlib	97	92.6	94.5	95.1	69.8	66.5	58.7
yolov8	97.3	95.7	95	95.5	91.9	67.5	65.9
yunet	97.9	97.4	96	96.7	91	66.5	63.5
centerface	97.7	96.8	95.7	96.5	89.3	67.8	63.6
mediapipe	96.1	90.6	92.9	90.3	75.4	64.8	63
ssd	88.7	87.5	87	86.2	84.5	63.8	62.6
opencv	87.6	84.9	87.2	84.6	83.6	63.7	60.1

Table 2. Euclidean metric accuracy

Detector	FaceNet512	FaceNet	VGG-Face	ArcFace	SFace	DeepFace	DeepID
Retinaface	95.9	93.5	95.8	85.2	80.2	67	65.6
MTCNN	95.2	93.8	95.9	83.7	77.4	66.5	63.5
FastMTCNN	96	93.4	95.8	83.5	77.7	66.7	64
Dlib	96	90.8	94.5	88.6	66.3	63.4	60.4
Yolov8	94.4	91.9	95	84.1	73.4	69	66.5
Yunet	97.3	96.1	96	84.9	79.4	65.8	65.2
CenterFace	97.6	95.8	95.7	83.6	77.4	65.5	62.8
MediaPipe	95.1	88.6	92.9	73.2	72.5	61.8	62.2
SSD	88.9	85.6	87	75.8	76.9	63.4	62.5
OpenCV	88.2	84.2	87.3	73	81.1	65.5	59.6

Table 3. Cosine Similarity distance

Sample	FaceNet512	FaceNet	VGG-Face	ArcFace	SFace	DeepFace	DeepID	Ensemble
20	0.19	0.12	0.46	0.35	0.29	0.11	0.12	0.12
50	0.17	0.2	0.42	0.31	0.22	0.2	0.27	0.2

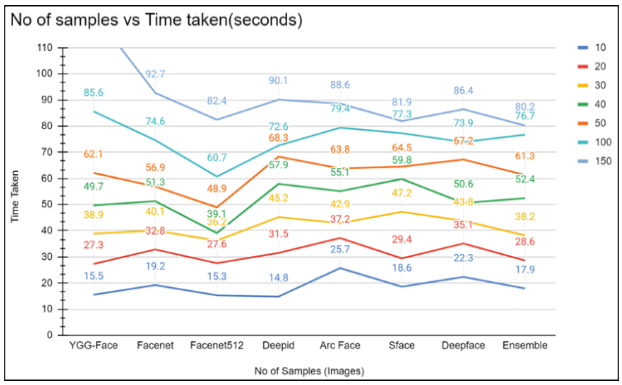


Figure 6. Measured and Declared Accuracy Comparision

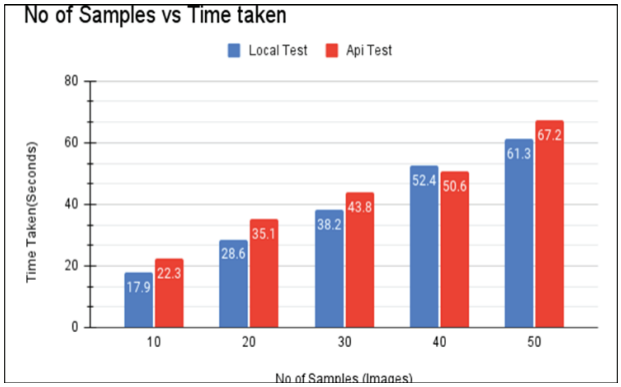


Figure 7. Time Taken local and cloud environment.

VGG-Face and Deepid have steeper time increases, indicating lower scalability.

Figure 7 compares the time taken for processing varying numbers of samples (from 10 to 50) between Local Tests and Cloud API Tests. As the number of samples increases, both testing methods show a gradual rise in processing time. Local Tests consistently take less time than API Tests for the same number of samples, indicating higher efficiency. For instance, at 50 samples, the Local Test completes in 61.3 seconds while the API Test takes 67.2 seconds, highlighting the faster processing of the Local Test across all sample sizes.

Figure 8 depicts responses by various facial recognition models (VGG-Face, Facenet, Facenet512, Deepid, Arc Face, Sface, Deepface, Ensemble) across different sample sizes from 10 to 50 images. The majority of the models consistently deliver between two and three responses, regardless of sample size. Interestingly, Facenet512 reaches a maximum of three responses at the smallest sample size, highlighting an exceptional case in its response pattern.

Table 4. Measure and Declared accuracy

Detector	FaceNet512	FaceNet	VGG-Face	ArcFace	SFace	DeepFace	DeepID	Ensemble
Measured	98.4	97.4	96.7	96.7	93	69	66.5	98.80
Declared	99.6	99.2	98.9	99.5	99.5	97.3	97.4	NA

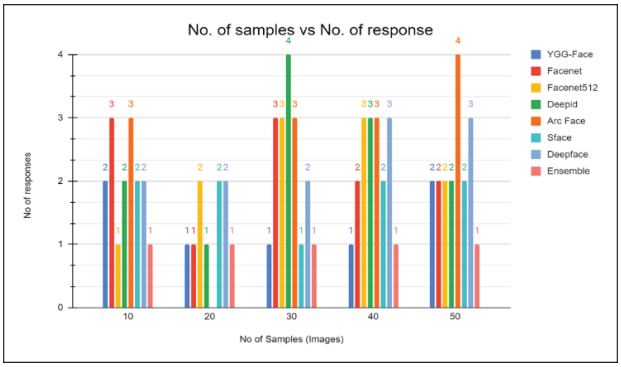


Figure 8. Time Taken local and cloud environment.

5. Conclusion

A system for suspect identification using facial recognition based on ensemble learning integrates seamlessly with OpenAI's advanced technologies and is supported by a robust cloud infrastructure. The comparison of the proposed ensemble model with individual models like VGG-Face, Facenet, Facenet512, Deepface, DeepID, ArcFace, and SFace uses multiple detectors and the LFW dataset with varying ages, gender, and ethnicity. Tests were conducted using multiple face detectors on each model, with the technicality of each image compared to one another. Cosine similarity distances for various facial recognition models across tests with 20 and 50 sample images highlight significant performance variations among the models. When tested, the Facenet512 model has higher accuracy than every other model, which is 0.984 or 98.4% which means it can predict actual faces at 100%. From the results of this study, we concluded that the FaceNet, VGG-Face, and ArcFace models could be used if desired as these have accuracy above 95%. Ensemble of these models shows a 98.8% accuracy. For instance, at a sample size of 50, the local test completes in 61.3 seconds, while the cloud API test takes 67.2 seconds. This highlights the faster processing speed of the local test across all sample sizes.

AUTHORS

Shilpa Chaudhari* – Dept. of CSE, M S Ramaiah Institute of Technology (Affiliated to VTU), Bangalore-560054, India, e-mail: shilpasc29@msrit.edu.

Rajarajeswari S – Dept. of CSE, M S Ramaiah Institute of Technology (Affiliated to VTU), Bangalore-560054, India, e-mail: raji@msrit.edu.

Archana Rane – K. K. Wagh Institute of Engineering Education and Research, Nashik- 422003. India, e-mail: alrane@kkwagh.edu.in.

*Corresponding author

References

[1] S.I. Serengil and A. Ozpinar, "LightFace: A Hybrid Deep Face Recognition Framework," *2020 Innovations in Intelligent Systems and Applications Conference* (ASYU), 2020, pp. 23–27; doi: 10.1109/ASYU50717.2020.9259802

[2] H. Han et al. "Matching Composite Sketches to Face Photos: A Component-based Approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, 2012, pp. 191–204; doi: 10.1109/TIFS.2012.2228856

[3] Y. Zhong et al., "SFace: Sigmoid-Constrained Hype rsphere Loss for Robust Face Recognition," *IEEE Transactions on Image Processing*, vol. 30, 2021, pp. 2587–2598; doi: 10.1109/tip.2020.3048632.

[4] D. DeepInsight, "deepinsight/insightface," GitHub, 4 Jun. 2021; https://github.com/deepinsight/insightface

[5] X. Ning et al., "Face Editing Based on Facial Recognition Features," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 2, 2023, pp. 774-783; doi: 10.1109/TCDS.2022.3182650

[6] J. Xiang and G. Zhu, "Joint Face Detection and Facial Expression Recognition with MTCNN," In *2017 4th International Conference on Information Science and Control Engineering* (ICISCE), 2017, pp. 424–427; doi: 10.1109/ICISCE.2017.95

[7] K. Vangara et al., "Characterizing the Variability in Face Recognition Accuracy Relative to Race," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019; doi: 10.1109/CVPRW.2019.00281

[8] J. Deng et al., "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR), 2019, pp. 4690–4699; doi: 10.1109/cvpr.2019.00482.

[9] A.A. Poeloemgam et al., "Web-based Face Detection and Recognition using YOLO and Dlib," In *2023 17th International Conference on Telecommunication Systems, Services, and Applications* (TSSA), 2023, pp. 1–6; doi: 10.1109/TSSA59948.2023.10366984

[10] Q. Cao, "Vggface2: A Dataset for Recognising Faces across Pose and Age," In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition* (FG 2018) 2018, pp. 67–74; doi: 10.1109/FG.2018.00020

[11] J. Hu et al., "Rapid Face Detection in Complex Environments based on the Improved RetinaFace," In *Proceedings of the 4th International Conference on Advanced Information Science and System*, 2023, pp. 1–7; doi: 10.1145/3573834.3574552

[12] W. Wu, H. Peng, and S. Yu, "Yunet: A Tiny Millisecond-Level Face Detector," *Machine Intelligence Research*, vol. 20, no. 5, 2023, pp. 656–665; doi: 10.1007/s11633-023-1423-y

[13] J.G. Cavazos et al., "Accuracy Comparison Across Face Recognition Algorithms: Where are We on Measuring Race Bias," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, 2020, pp. 101–111; doi: 10.1109/TBIOM.2020.3027269.

[14] M.S.S. Bobde and S.V. Deshmukh, "Face Recognition Technology," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 10, 2014, pp.192–202.

[15] Y.Taigman et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2014, pp. 1701–1708; doi: 10.1109/**CVPR**.2014.220

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2015; doi: 10.1109/cvpr.2015.7298682.

[17] J. Deng et al., "Retinaface: Single-shot Multi-level Face Localisation in the Wild," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5203–5212; doi: 10.1109/CVPR42600.2020.00525

[18] S. Solarova et al., "Reconsidering the regulation of Facial Recognition in Public Spaces," *AI Ethics*, vol. 3, 2023, pp. 625–635; doi: 10.1007/s43681-022-00194-0

[19] M. Mortensen, "Sneaking AI through the Back Door: Constructing the Identity of Capitol Hill Rioters through Social Media Images and Facial Recognition Rechnologies," *Information, Communication & Society*, 2024, pp. 1–17; doi: 10.1080/1369118X.2024.2358164

[20] D. Utegen and B.Z. Rakhmetov, "Facial Recognition Technology and Ensuring Security of Biometric Data: Comparative Analysis of Legal Regulation Models," *Journal of Digital Technologies and Law*, vol. 1, no. 3, 2023, pp. 825–844; doi: 10.21202/jdtl.2023.36

[21] S. Gokulakrishnan et al., "An Optimized Facial Recognition Model for Identifying Criminal Activities using Deep Learning Strategy," *International Journal of Information Technology*, vol. 15, no. 7, 2023, pp. 3907–3921; doi: 10.1007/s41870-023-01420-6

[22] V. Munusamy and S. Senthilkumar, "Face Identification of Suspects Using Sequential-Deep Convolutional Neural Network," In *2024 Second International Conference on Emerging Trends in Information Technology and Engineering* (ICETITE) 2024, pp. 1–3; doi: 10.1109/ACCESS.2024.3523101

[23] N.K. Sharma et al., "Enhancing Facial Geometry Analysis by DeepFaceLandmark Leveraging ResNet101 and Transfer Learning," *International Journal of Information Technology*, 2024, pp. 1–21; doi: 10.1007/s41870-024-01872-4

[24] L. Yu et al.,, "Facial Expression Recognition Based on Improved VGG-face Model and Transfer Learning," In *Proceedings of the 2023 International Conference on Computer, Vision and Intelligent Technology*, 2023, pp. 1–7; doi: 10.1145/3627341.3630376

[25] K.L. Sailaja et al., "Facial Detection and Recognition in Drone Imagery Using FaceNet," In *International Conference on Advances in Distributed Computing and Machine Learning*, 2024, pp. 183–197; doi: 10.1007/978-981-97-1841-2_13

[26] M. Gulhane et al., "Advancing Facial Recognition: Enhanced Model with Improved Deepface Algorithm for Robust Adaptability in Diverse Scenarios," In *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering* (UPCON), vol. 10, pp. 1384–1389; doi: 10.1109/UPCON59197.2023.10434721

[27] S. Srinivas and M.P. Selvan, "E-CNN-FFE: An Enhanced Convolutional Neural Network for Facial Feature Extraction and Its Comparative Analysis with FaceNet, DeepID, and LBPH Methods," In *International Conference on Data Management, Analytics & Innovation*, 2024, pp. 339–354; doi: 10.1007/978-981-97-3245-6_23

[28] M.A. Altaha et al., "Facial Expression Recognition based on ArcFace Features and TinySiamese Network," In *2023 International Conference on Cyberworlds* (CW), pp. 24–31; doi: 10.35784/jcsi.7973

Towards Accurate Glaucoma Identification: GAN-Enhanced Synthesis and Classification Using Pretrained MobileNetV2

Submitted: 19th February 2024; accepted: 15th May 2024

Govindharaj I, G. Karthick, G. Michael

DOI: 10.14313/jamris-2026-023

Abstract:
Irreversible vision loss, which often develops slowly and with no outward signs of illness, is most commonly caused by glaucoma. Because it may slow the disease's progression, the initial stages of glaucoma detection are of the utmost importance. Ordinary procedures and manual assessments are based on traditional diagnostic techniques, which are notoriously imprecise. It follows that automated glaucoma analysis is critically important for the early and precise detection of glaucoma. Also, on the other hand, the medical image dataset is mostly imbalanced in nature. To overcome all these issues, the present research work developed an effective framework by utilizing Generative Adversarial Networks (GAN) to synthesize images to balance out the dataset. For example, when dealing with fundus images, conventional methods, such as translation from image-to-image operations, are used. In particular, these techniques are employed to produce synthetic fundus images and the associated vessel networks. Improving the quality of the synthetic images as a whole and capturing finer details is the main goal. The goal of this effort is to improve the accuracy and authenticity of synthetic fundus images, leading to new developments in fundus image synthesis. Initially, a raw dataset has been preprocessed using the Gaussian filtering technique, which helps to minimize the unnecessary noise in the images. Then, a GAN is used to balance out the dataset, which helps to produce synthetic images and produce reliable outcomes in classification tasks. The next segmenting optic cup is done using the Enhanced Level Set Algorithm. Finally, Pretrained MobileNetV2 is used for the accurate classification of glaucomatous images into normal and abnormal. Experimental results show that our proposed frameworks perform well compared to existing approaches with an accuracy of 98.9%.

Keywords: *Enhanced Level Set Algorithm (ELSA), Gaussian Filtering Technique (GFT), Generative Adversarial Networks (GAN), Pretrained MobileNetV2*

1. Introduction

Glaucoma has become one of the leading causes of visual disability worldwide. According to recent statistics, approximately 85.9 million people were expected to be affected by glaucoma in 2023 [1]. This progressive eye disease results in irreversible vision loss due to damage to the optic nerve, which is responsible for transmitting visual information from the eye to the brain. Among the constraints that face the management of glaucoma, it is highly asymptomatic at its initial stages, and thus, detecting the disease is quite challenging. The optic nerve head (ONH), also known as the optic disc, undergoes structural changes as the optic nerve begins to degenerate. This is a structure with two parts: the neuroretinal rim, which is at the periphery, and the optic cup, which is at the center.

These structural components are compromised as glaucoma progresses, resulting in thinning of the neuroretinal rim and enlargement of the optic cup. Thus, the precise identification of changes in the optic disc and optic cup and constant monitoring are the key to detecting glaucoma earlier and its course. Nonetheless, optic cup automatic detection is more difficult than optic disc automatic detection, primarily because it is less defined in boundaries and may have a different appearance in different retinal fundus images [2].

Historically, the diagnosis of glaucoma has been conducted with the help of multiple clinical tools and metrics like Dynamic Contour tonometry (DCT), air-puff non-contact tonometry, and other tonometry-related procedures of intraocular pressure measurement [3]. Nonetheless, the traditional diagnostic measures have several shortcomings, which include low accuracy, the time-consuming nature of the examination process, and a high level of manual intervention on the part of clinicians.

To overcome these issues, there has been an increase in the use of computerized methods for diagnosing glaucoma. These systems will facilitate the aims of minimizing the challenges encountered by clinicians when carrying out a holistic assessment of glaucoma through more efficient, objective, and automated diagnostic assistance. The described transition is an indication of the growing need for quicker and more dependable diagnostic tools in clinical practice [4].

Some automated tools have been created in recent years to enable glaucoma diagnosis. The techniques primarily use retinal fundus images to examine structural alterations associated with the disease. The use of these methods is intended to improve efficiency in the diagnostic process, enhance detection accuracy, and save time on time-consuming manual procedures [5]. The introduction of the automated glaucoma detection systems is a crucial improvement in the ophthalmic diagnostics since it enhances the

efficiency of detecting this sight-threatening disease, and its reliability and accuracy [6].

Computer-Aided Diagnosis (CAD) can be a valuable resource for large-scale disease exploration and screening across diverse populations. CAD systems are more efficient, consistent, and scalable than traditional clinical examinations conducted by medical professionals in detecting a disease. Such technologies, when integrated, can improve, streamline, and facilitate the existing healthcare systems. This comes in handy, especially in the resource-constrained and developing world, where skilled and experienced optometrists or ophthalmologists may be in short supply [7].

Computer-aided diagnosis (CAD) systems have been extensive in ophthalmology, where they have been used to analyze and segment optic nerves in retinal fundus images to identify the early signs of glaucoma [8]. The studies of the available literature have proven that image processing methods are effective in assessing the structural changes of the optic disc and optic cup, which are some of the major indicators of glaucomatous damage. Glaucoma is a chronic and progressive eye disease and is among the leading causes of blindness in the world, and the United States is not an exception. Despite the number of automated diagnostic methods already developed, the existing methods also contain a few drawbacks, which include unreliable segmentation accuracy and sensitivity to image quality changes. These issues demonstrate the necessity of creating stronger and more efficient models to address the drawbacks of the existing methods and enhance the quality of the glaucoma detection systems [9].

Retinal image databases have been widely used in the initial years of glaucoma research to analyze and develop models. Medical practitioners have traditionally done eye examination by hand when examining images of the retinal fundus to detect glaucomatous abnormalities. In the process, clinicians determine structural characteristics of the Cup-to-Disc Ratio (CDR), as well as changes in the diameter and limits of the optic disc and optic cup. Nevertheless, one of the biggest problems is related to the inadequate number of qualified specialists, which may cause certain delays in terms of the timely identification of ocular abnormalities.

Diagnosis and treatment of glaucoma at early stages are of high importance because early treatment is expected to prevent the progression of the disease and minimize the chances of total blindness [10]. Consequently, this brings about the urgent need to come up with more efficient and reliable automated models that can help overcome such shortcomings and help in the early diagnosis of glaucoma and in the end lead to better patient outcomes.

The models of deep learning, specifically the Convolutional Neural Networks (CNNs), have demonstrated excellent performance in a variety of applications in computer vision, including image classification, object detection, and image segmentation. Deep learning methods have greatly improved medical image analysis, as they now enable more successful and efficient disease detection and diagnosis [11].

Nevertheless, medical imaging data are usually associated with such issues as the imbalance in classes and the scarcity of expert-labeled data. To overcome these challenges, several methods have been proposed to expand training datasets by generating synthetic medical images. These techniques increase the efficiency and strength of computer-aided diagnosis (CAD) systems by increasing the size and variety of the training data. Synthetic images are significant in medical imaging because they improve the performance of classification and segmentation tasks by supplementing datasets and enabling models to learn more representative features. The method is especially useful in cases of small or unbalanced datasets, which eventually results in the creation of more precise and valid deep learning models used in medical image analysis [12].

Generative Adversarial Networks (GANs) represent a type of unsupervised machine learning model that has proven very successful with both synthetic image generation as well as image-to-image translation in the real-world image space. The common GAN architecture is a two-player zero-sum game, and it involves two neural networks, one called the generator and one called the discriminator, which are both trained simultaneously [13].

Here, the generator is trained to generate candidate images that are distributed according to the target data through a latent variable. In the meantime, the discriminator tries to distinguish generated images from real samples using the actual data distribution. In this adversarial training, the generator gets better at making realistic-looking synthetic images, whereas the discriminator is getting better at detecting generated samples [14].

Various versions of Generative Adversarial Network (GAN) models have been built, such as WGAN, InfoGAN, DCGAN, CGAN, Pix2Pix, and CycleGAN, among others [15]. These models have shown great potential in solving image-to-image translation problems and synthetic image generation problems. Consequently, GANs keep evolving and become more flexible in computational intelligence and image processing. These various architectures have been created due to the continued improvements in the field of generative modeling, which allows more efficient solutions to complex image analysis tasks.

The novelty of the proposed work lies in the integration of multiple advanced techniques to improve the accuracy and reliability of automated glaucoma detection. The proposed framework, unlike the current approaches that only use the traditional deep learning frameworks, also involves Generative Adversarial Networks (GANs) to address the issue of class imbalance by creating synthetic retinal images. Moreover, Precise optic cup segmentation is achieved using an Enhanced Level Set Algorithm, which improves the extraction of important structural characteristics

of fundus images. Lastly, classification is performed using a pretrained MobileNetV2 model, enabling efficient feature extraction and high detection rates. The proposed framework has better segmentation accuracy, better dataset balancing, and better classification performance when compared to the current methods, including CNN-based, graph-based, and ensemble models, with an overall accuracy of 98.9% on the ORIGA dataset.

2. Literature Survey

Several studies examined the automated method of detecting glaucoma in machine learning, computer vision, and deep learning. The techniques are mainly aimed at analyzing retinal fundus images to detect the structural changes in the optic disc and optic cup areas, which are major indicators of the glaucomatous damage.

The extraction of multi-scale features of the retinal images is a recent technique called M-LAP that has been suggested to enhance the precision of glaucoma identification. It is an effective technique to bridge the gap between global semantic analysis and the localization of glaucomatous regions, thereby more accurately identifying patterns associated with the disease [16]. Also, it identifies abnormal areas in fundus images, thereby making glaucoma analysis more interpretable and clinically significant [17]. Experimental results have shown promising performance, particularly in terms of Area Under the Curve (AUC). Similarly, the EAMNet framework demonstrates high sensitivity, which contributes to improved diagnostic accuracy. However, these methods still face certain limitations, including challenges in accurately detecting the optic cup region. Additionally, the use of Global Average Pooling (GAP) may restrict the representation of high-resolution feature maps, thereby affecting fine-grained feature extraction [18].

In another study, glaucoma detection was performed by analyzing blood flow patterns in retinal arterial and venous networks. In this method, Support Vector Machines (SVMs) were used to classify vascular features extracted from fundus images [19]. The approach achieved high accuracy and sensitivity in identifying glaucomatous changes. Nevertheless, the model requires a large amount of data for effective training and analysis, thereby increasing computational complexity and limiting its practical applicability.

Optimization-based techniques have also been investigated for glaucoma detection. For instance, a Group Search Optimization (GSO) model has been developed for automatic detection of the optic cup in retinal fundus images. This approach constructs the solution based on the intensity gradient within the cup region. The GSO algorithm incorporates adaptive neighborhood behavior, which improves search capability and allows accurate detection even in cases with weak cup boundaries or low contrast. Although the method achieves a high F-score and reduced detection errors, it requires careful estimation of the Cup-to-Disc Ratio (CDR). Moreover, when applied to low-resolution images, the model may suffer from insufficient pixel information, which affects the accurate delineation of optic cup boundaries [20].

Recent research has also focused on computer vision-based techniques for automated glaucoma detection using digital fundus images. One such approach utilizes a geometric feature-based model for optic disc segmentation. This method improves system robustness against image noise and illumination variations, thereby enhancing detection accuracy [21]. However, the approach still faces challenges such as class imbalance in the dataset and significant variability in pixel intensity within the optic disc and surrounding blood vessels, which can negatively affect segmentation performance [22].

With the rapid advancement of deep learning techniques, Convolutional Neural Networks (CNNs) have been widely applied to glaucoma detection. An interpretable Computer-Aided Diagnosis (CAD) model was proposed to enable glaucoma detection directly from mobile devices. The framework integrates multiple datasets to construct CNN-based models for both classification and segmentation tasks [23]. The suggested pipeline performs thorough segmentation and characterizes retinal structures associated with glaucoma, resulting in improved glaucoma analysis. Experimental findings prove less computational complexity, high accuracy, and F-score [24]. Nevertheless, this method is also characterized by the shortage of training samples and the coverage of pathological areas of attention maps.

Another important contribution is the Attention-based Convolutional Neural Network (AG-CNN), developed specifically for glaucoma detection using the LAG database and other fundus images [25]. The AG-CNN architecture improves model convergence and robustness while reducing classification errors. Despite these advantages, the model exhibits certain drawbacks, including a decrease in Area Under the Curve (AUC) and the inability of the attention mechanism to fully capture the entire pathological region [26].

Furthermore, deep learning frameworks such as the disc-aware ensemble network have been proposed for glaucoma screening using retinal fundus images. This architecture integrates contextual information from both the global fundus image and the local optic disc region. The network consists of several components, including a global image stream, a segmentation-guided network, a local disc region stream, and a disc polar transformation stream [27]. Although the method achieves high specificity and accuracy in glaucoma detection, it requires significant computational resources and longer processing time due to its complex architecture.

Despite the significant progress achieved by existing approaches, several challenges remain in automated glaucoma detection. These include dataset imbalance, variability in optic disc and optic cup structures, sensitivity to image-quality variations, and

Table 1. Comparative analysis of existing glaucoma detection approaches

Ref	Method / Model	Dataset	Key Contribution	Limitations
[16]	M-LAP Model	Fundus Images	Multi-scale feature extraction for glaucoma detection	Difficulty in accurate optic cup segmentation
[19]	SVM-based Classification	Retinal fundus images	Uses retinal blood flow features for glaucoma classification	Requires large datasets for effective training
[20]	GSO Algorithm	Fundus Images	Automatic optic cup detection using intensity gradients	Performance affected by low-resolution images
[21]	Geometric Feature Model	Digital fundus images	Optic disc segmentation using computer vision methods	Sensitive to illumination and intensity variations
[23]	CNN-based CAD Model	Fundus Images	Automated glaucoma detection using deep learning	Limited training samples and incomplete attention mapping
[25]	AG-CNN	LAG Dataset	Attention-based CNN improves convergence and robustness	Reduced AUC and incomplete attention coverage
[27]	Disc-aware Ensemble Network	Fundus Images	Integrates global and local contextual information	High computational complexity

limited segmentation accuracy. Therefore, there is a need for a more robust framework that can effectively address these challenges. To overcome these limitations, the present study proposes a GAN-based synthetic image generation approach combined with Enhanced Level Set segmentation and a pretrained MobileNetV2 classification model for accurate and efficient glaucoma detection.

Table 1 provides a comparative summary of existing glaucoma detection approaches reported in the literature. The table highlights the key methodologies, datasets used, main contributions, and limitations of each study. From the comparison, it can be observed that many existing methods focus on improving classification accuracy using machine learning and deep learning techniques. However, several challenges remain, including dataset imbalance, sensitivity to image quality variations, and difficulties in accurately segmenting optic disc and optic cup regions. These limitations indicate the need for more robust frameworks that can address these issues effectively. To overcome these challenges, the proposed method integrates GAN-based data augmentation, an enhanced level set algorithm for accurate segmentation, and a pretrained MobileNetV2 model for reliable glaucoma classification.

3. Methodology

Manual glaucoma diagnosis can be time-consuming and costly, and many existing automated glaucoma detection techniques either do not achieve satisfactory performance or lack comprehensive statistical validation. Therefore, this study proposes an improved Convolutional Neural Network (CNN)-based detection model for glaucoma identification. The detailed architecture and workflow of the proposed model are presented in the following sections, as illustrated in Figure 1.

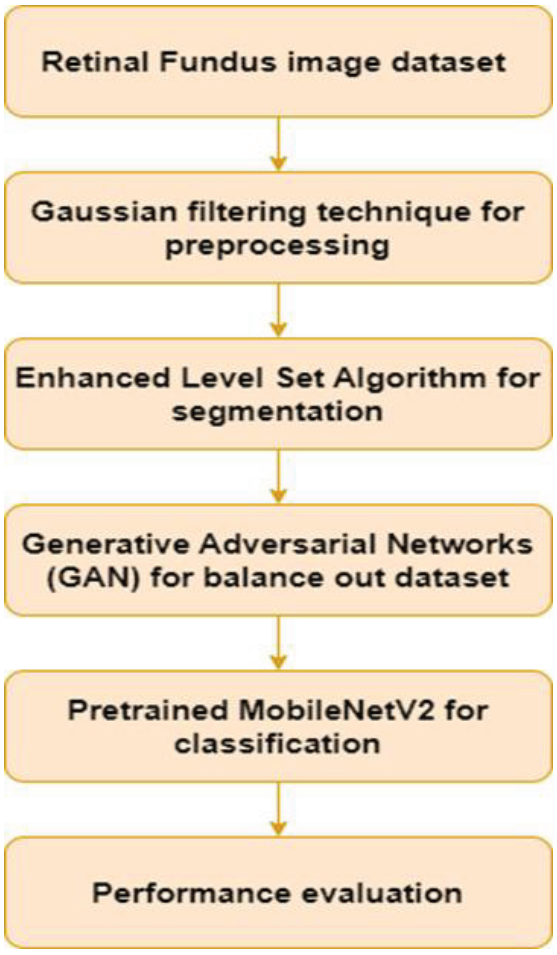


Figure 1. Workflow of Proposed Work

3.1. Pre-Processing Phase

Following the application of a Gaussian Filter (GF) during the preprocessing step, the quality of the input image is improved by the application of a filtering process. After going through the process of scaling, the

original image, which had dimensions of (1154, 1600, 3), has been changed to (512, 512, 3).

The Modified Level Set Algorithm is utilized to accomplish the task of optic cup segmentation. Two categories of features are utilized in the process of feature extraction: morphological and non-morphological features. Morphological features, also known as Femf, are derived through the applications of closing and dilation operations. These features include disc region, cup region, and RNFL thickness. The extraction process also includes the extraction of non-morphological aspects, often known as Fenmf, which include color, form, and Modified Local Binary Pattern (LBP) [28].

After this, the features are sent to a Convolutional Neural Network (CNN), which has been optimized. The weights of CNN are subsequently fine-tuned using the simulated annealing and Biogeography-Based Optimization Algorithm (SA-BOA) method. A graphical representation of the entire workflow is given in Figure 1.

Pre-processing of the image under use is the first step and involves Gaussian filtering, a necessary process to enhance image quality. A blurred image can be well refined through the application of Gaussian filtering, which is applied to take away noise and blur that is caused by the reduction of high frequencies in an image. This filtering process is particularly useful when the task is to minimize Gaussian noise in an image where the noise has been introduced by speckle noise or by brain MR images in ultrasound imagery. Under this technique, the noise pixel is replaced with the mean of the pixels that are adjacent to the noise pixel, with the help of a Gaussian filter. Using a 5 x 5 window, the Gaussian denoising approach is applied. Following this, the pre-processed image that was produced, which is referred to as ImPr, is segmented in order to conduct additional analysis.

3.2. Generative Adversarial Networks (GAN)

Within the context of game theory, the Generative Adversarial Nets (GAN) function according to the principles of Nash equilibrium. There are two components that make up the model's fundamental structure: G and D. G's primary objective is to learn the mapping relationship between the retinal image x and the corresponding optic disc and optic cup. The ultimate objective is to separate the optic disc and optic cup based on the retinal picture that is being input, x, while making certain that the representation D(G(x)) of the segmentation outcome G(x) on D coincides with the representation D(y) of the ground truth y on D. Only then will the goal be accomplished. During this procedure, it is D's responsibility to learn the differences between y and G(x), as well as to accurately differentiate between the source data of the input optic disc and the label on the optic cup (whether it is the ground truth or G). D directs G to reduce this discrepancy as much as possible, which ultimately results in an improvement in the precision of the partitioned optic disc and optic cup. In the course of model training, it is essential to perform periodic optimization of G and

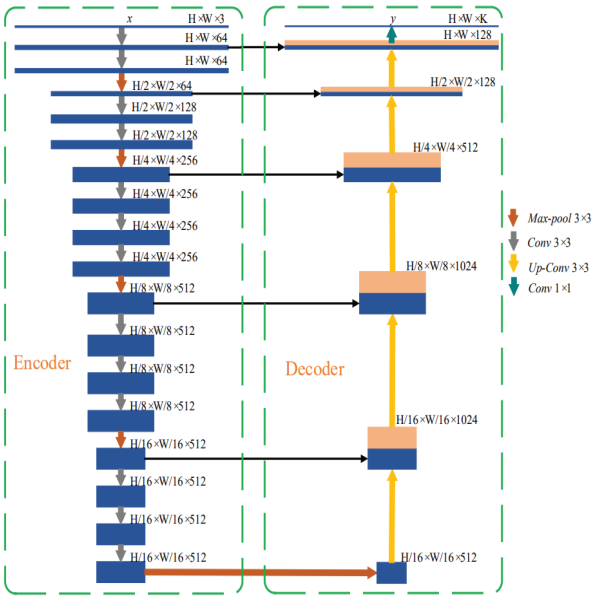


Figure 2. Structure of generator

D in order to enhance their capacity for segmentation and discrimination. The objective is to locate the Nash equilibrium between these two variables. The Nash equilibrium is reached when the resultant of D equals half, which indicates that the origin of the optic disc and optic cup label are unable to be reliably differentiated because they are both identical. Since G is now able to precisely separate the optic disc and optic cup, the training is regarded as finished at this time. The ultimate objective of the architecture is to achieve the following objective function, which was given in Equation (1).

$$\min_G \max_D \{\mathbb{E}_{x\in T, y\in L}[\log D(x,y)] + \mathbb{E}_{x\in T, y\in L}[\log(1 - D(x, G(x)))]\} \quad (1)$$

3.2.1. Generator Network (GE)

The G is a complete convolutional network framework with 19 layers. Figure 2 shows the structure of its network. The retinal picture x is fed into this model. In this study, we set H = W = 512 and C1 = 3. There is an encoder (on the left) and a decoder (on the right) in the network layout. To get characteristics from the retinal image, the encoder uses the VGG16 network layout. Even though the original VGG16 network has a downsampling ratio of 32, we found that too much downsampling causes important characteristic data to be lost, especially in areas that are smaller than 32 pixels, like the optic disc. To fix this, we got rid of the last two downsampling levels and raised the reduction factor to 16. This cuts down on data loss, model settings, and computations.

The ReLU activation function is used to turn on the output feature map of each convolutional layer. We allow skip links to improve the model's ability to get back low-level information and get full context information. The feature map input from the layer that pools data within the encoder is sent straight to the decoder through those connections.

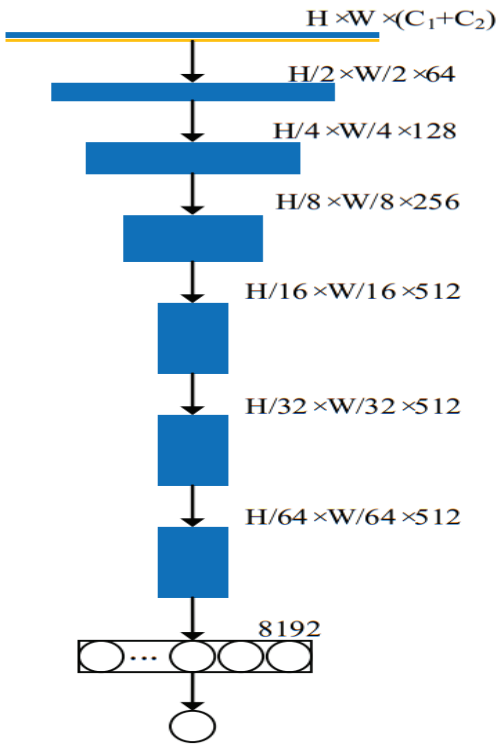


Figure 3. Structure of discriminator network

Using deconvolution techniques for up sampling, the decoder brings back feature knowledge. The output segmentation map with features is then joined with the pooling layer's characteristic map of the same size in the encoder, based on channel measurement. The decoding layer does four downsampling operations. The encoder then does four upsampling operations to match both the width and height of the feature list with the input image.

The encoder's last predictor uses a 1x1 convolutional layer with softmax activation to classify pixels one by one and make a probability map. Optical disc, optic cup, and backdrop segmentation give us G(x), which is a K-channel probability map. K is the class number, and in our case, it is 3. The forecasted probability map shows which group has the highest chance of occurring for each pixel. This makes it possible to separate the optic disc and optic cup at the same time. Each time, dropout = 0.8 is given to the last two stages of the encoder to stop overfitting and make the model more general. The same is done for every single component of the decoder.

3.2.2. Discriminator Network (DN)

Figure 3 shows that the discriminator is built with an 8-layer network model. Input data for D is made up of the retinal picture x and the ground truth y, which is written as G(x). The amount of info that D needs is H × W × (C1 + C2). The chance (D(x, y)) of the optic disc and optic cup label for the retinal image x is shown by D. This makes it easier to find the best values for the network structure's parameters. Each layer uses strided convolution (strided = 2) instead of a pooling layer. After each convolution, the size of the feature map is cut in half, to 1/4 of its original size.

Batch normalization is used to make the information of each layer in the convolutional layer more normal (mean s = 0, variance ε = 1). This speeds up convergence and lessens the effect of weight introduction on the network model. In every single layer, the activation function is Leaky-ReLU with α = 0.2. The main image's resolution drops to H64 × W64 after six downsampling steps. The link is made by the last fully connected layer. D sends either 0 or 1 as the discriminant answer. The gradient descent method is used to change the parameters βg of G and the value of the parameter θd of D based on the difference between the D value and the labeled result. The goal of this method is to get the best model improvement effect.

Algorithm 1 : TGA Algorithm

Input: *U → Number users*
C → Number of servers
Output: *AS → List of Allocated server*

Start

Define RTF = [SCPUsMems BW]//Assign the RTF (Resource Threshold Factor) using the basic parameters of servers like their CPU, RAM and bandwidth
For each U
TimInt = random//Requests at Time Interval
DemR = max([uCPUuMemuBW]) // User demand for resources
UtilRate = DemR × TimInt// Resource Utilization Rate
If max (UtilRate) ≤ max? *(RTF)*
AvgPTimeSer (i) = DemR(i)/(1-
UtilRate (i) AvgPTime AllSeri (i) = (AvgPTimeSer (i) × TimInt (i))/TimInt (i))
AS LIST = ceil(cserver × rand)
End - If
End - For
Return : AS LIST as a list of allocated servers
End - Algorithm

3.3. Segmentation of Optic Cup using Enhanced Level Set Algorithm

Historically, many prior techniques have been categorized under edge-based frameworks [29, 30]. These approaches primarily rely on image gradient information to construct edge detection functions for identifying object boundaries. The mathematical formulation of the level set model used in such methods is illustrated in Equation (2).

$$\frac{\partial \phi}{\partial t} = IG\,|\nabla \phi| \left(\mathrm{div}\left(\frac{\nabla \phi}{|\nabla \phi|} \right) + u \right) \qquad (2)$$

From Equation (2), $\mathrm{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right)$ forecasts the curvature of the mean, u denotes the forces with respect to the balloon and ϕ indicates the different levels of set operation.

In level set schemes, the reliability function, denoted by the symbol ϕ, plays a crucial role in ensuring stable evolution and accurate computations during

the segmentation process. To address this challenge, a fast level set formulation has been developed to improve computational efficiency and stability. The mathematical representation of this formulation is given in Equation (3).

$$\frac{\partial \phi}{\partial t} = \mu P(\phi) + \eta(IG, \phi) \tag{3}$$

The penalizing function, which is denoted by the symbol $P(\varphi)$, places an emphasis on the difference between the sign separation operation and the deviation of φ during its evolution. As shown in Equation (4), the succeeding function $\eta(IG, \varphi)$ is responsible for integrating the data pertaining to the image gradient.

$$\eta(IG, \phi) = \lambda\delta(\phi)\mathrm{div}\left(IG\frac{\nabla\phi}{|\nabla\phi|}\right) + uIG\delta(\phi) \tag{4}$$

The Dirac function is represented by the symbol $\delta(\phi)$. Parameters such as λ, u, and μ are used to determine the individual contributions of the constraints within the model. In conventional approaches, the parameter u is typically maintained at a constant value. However, in the proposed model, $u$is designed to vary adaptively, as defined in Equation (5).

In this formulation, H_σ denotes a Gaussian filter with standard deviation σ, while $\Delta H_\sigma \times I$ represents the Laplacian operator applied to the filtered image. The parameter w denotes the controlling constraint, where $w > 0$. Furthermore, I_G represents the function $u(I_m)$, enabling $u(I_m)$ to adapt dynamically according to the information present in the image. For region detection, conventional methods generally employ a Gaussian kernel. In contrast, the proposed model replaces this with a bilateral kernel, which helps preserve edge information while performing region-based analysis.

3.4. Accurate Classification of Glaucoma using MobileNetV2 Pretrained Model

Transfer learning is the process of utilizing a model that has already been trained on an extensive data set in order to subsequently adapt to and carry out tasks on various data sets. This method is particularly popular because it successfully classifies small datasets. It does this by addressing the difficulty of getting high precision while training models using deep learning from scratch on limited data. The MobileNetV2 network, which has been pre-trained on the ImageNet dataset, is the basis for this methodology. It acts as the foundation.

Within the framework of the proposed method, we augment the convolutional layers of MobileNetV2 by incorporating a collection of calculations that are referred to as the head model. A feature map with dimensions 7x7x1280 pixels is produced when the base model's output is fed into the first layer of the topmost model, which is a global pooling layer. A pooling operation is performed by the global pooling layer, producing a one-dimensional feature vector. This is done in order to greatly reduce the dimensionality of the data. In the overall pooling layer, the suggested method makes use of an average pooling technique with a kernel size of 7x7 pixels. This results in the generation of an output feature map that is 1x1x1280 pixels in size [31].

The global pooling layer is followed by two fully connected layers. Within these entirely interconnected layers, the ReLU activation function is responsible for activating 128 and 64 nodes, respectively. When the two subclasses of the dataset (glaucoma and non-glaucoma) and the application of one-hot encoding are taken into consideration, the output layer is made up of two nodes that are activated by softmax technology. Particularly noteworthy is the fact that the 1x1 convolutional layer diverges from the standard by not having a layer for batch normalization and an activation function (ReLU6). This is because its low-dimensional output is only subjected to batch normalization.

3.5. Algorithmic Representation of Proposed Framework

The proposed algorithm 1 describes the complete workflow of the glaucoma detection framework. Initially, retinal fundus images are collected and preprocessed using Gaussian filtering to remove noise and enhance image quality. To address the issue of dataset imbalance, Generative Adversarial Networks (GANs) are employed to generate synthetic retinal images. The augmented image data is then subjected to the Enhanced Level Set Algorithm, which correctly splits the optic cup region based on the fundus images. Lastly, the segmented images are fed into a pre-trained MobileNetV2 model to extract and classify the images into glaucomatous and normal, classifying the images into glaucomatous and normal. This step-wise process enhances the accuracy and reliability of the glaucoma detection procedure.

4. Experimentation and Result Discussion

This section presents the experimental results obtained to evaluate the effectiveness of the proposed glaucoma detection framework. To assess the robustness and classification capability of the model, experiments were conducted using the ORIGA (Online Retinal Fundus Image Database for Glaucoma Analysis) dataset, a widely used benchmark dataset for glaucoma detection studies [32]. The ORIGA dataset contains 650 retinal fundus images, including 168 glaucomatous images and 482 normal images. The experiments were conducted and analyzed in MATLAB to evaluate the performance of the proposed method.

Glaucoma classification using the ORIGA dataset is challenging due to the presence of various artifacts and structural variations. These present a considerable range in the size, color, position, and texture of the optic disc (OD) and optic cup (OC). Moreover, the dataset has several image distortions, including noise, blur, hue, and intensity variation, which makes it a

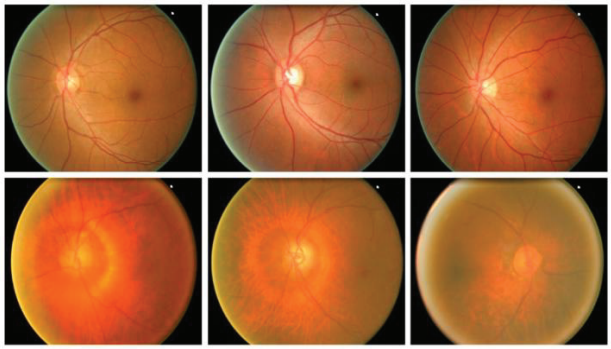


Figure 4. Sample images from the dataset

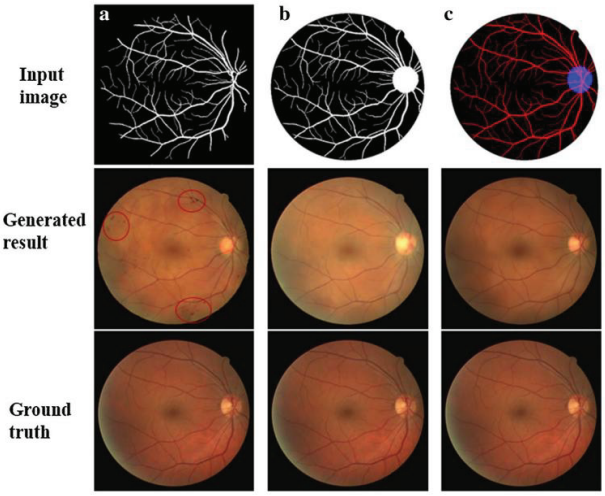


Figure 5. Synthetic output after GAN applied

complicated and realistic test set for glaucoma detecting models. In Figure 4, several sample data points in the dataset are presented.

To assess the performance of the proposed method on the ORIGA dataset, the dataset was split into training and testing subsets in an 80:20 ratio, respectively.

Figure 5 shows the output of the GAN model when the input images have been fed into the model. After this, it was followed by a segmentation process that was done with the enhanced level set algorithm, which is an effective method of dividing the foreground and background area of glaucomatous fundus images. The findings of this type of segmentation are shown in Figure 6.

This study was conducted to demonstrate the effectiveness of the proposed glaucoma diagnosis and

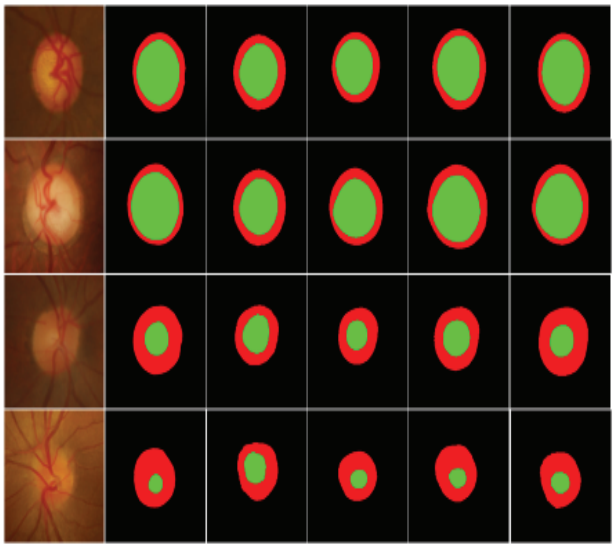


Figure 6. Segmented output

classification method through a thorough comparative study with several modern methods, all of which were tested on the same dataset. To achieve this, the work of the proposed method was compared with other contemporary methods to ensure fair and unbiased coverage.

The quantitative data for this comparison are presented in Table 2, which contains performance measures for evaluating the effectiveness and superiority of the various methods. Moreover, the corresponding comparative graphic representations are shown in Figures 7–11 which provide a clearer visualization of the differences in the performance of the investigated approaches.

All figures and tables in this study were generated by the authors to illustrate the proposed framework, experimental setup, and obtained results.

5. Conclusion

The results of this research are relevant to the crucial issue of glaucoma being detected early and at the earliest opportunity, as one of the largest causes of permanent blindness on a global scale. The common diagnostic techniques that use routine clinical examination and manual tests are usually poorly accurate and have sluggish processes. To address these shortcomings, this study will introduce a powerful automated glaucoma diagnosis system based on Generative Adversarial Networks (GANs)

Table 2. Performance comparison of the proposed approach over the existing technique

Methods	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	Sensitivity (%)
Deep CNN [32]	92.5	92.3	91.3	90.4	90.3
Graph CNN [33]	92.9	92.9	91.2	91.3	90.8
Ensembling [34]	93.1	93.2	92.3	92.3	91.1
U-Net + Inception V3 [35]	93.6	94.9	93.7	93.3	94.2
ODGNet [31]	95.2	95.1	94.6	94.5	95.3
U-Net + EfficientNet [22]	96.5	95.3	95.7	95.1	96.3
Proposed Model	98.9	98.4	96.4	97.8	97.2

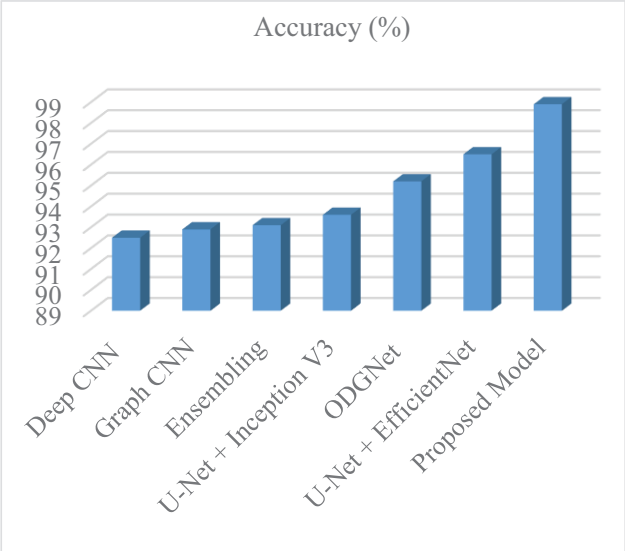


Figure 7. Accuracy of proposed over existing

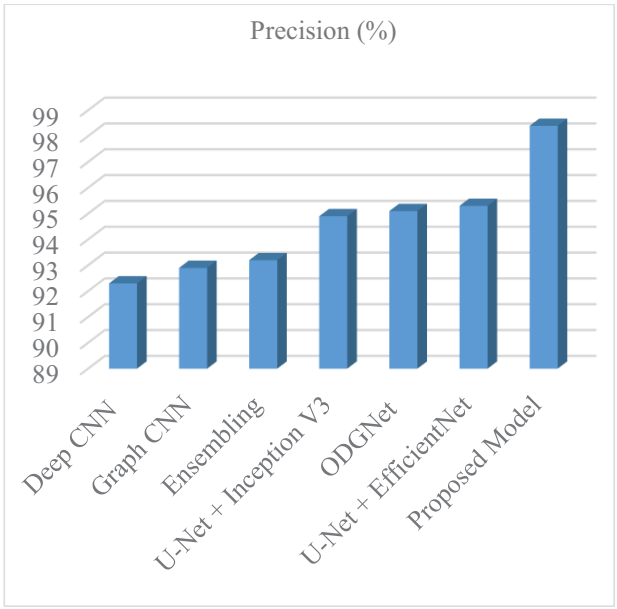


Figure 8. Precision of proposed over existing

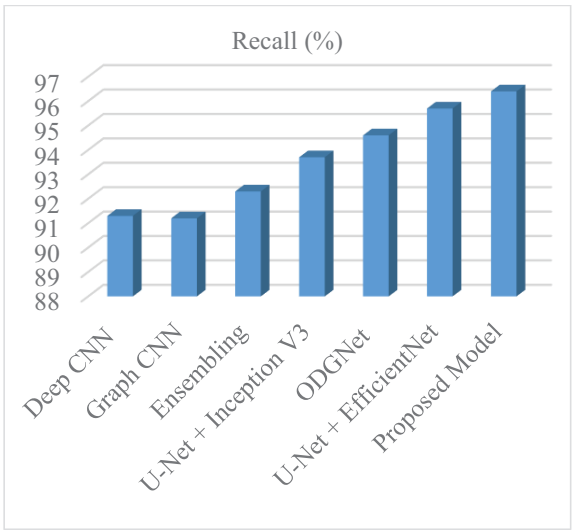


Figure 9. Recall of proposed over existing

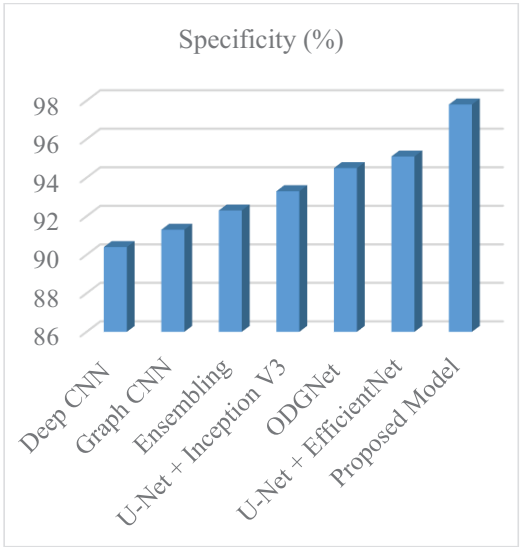


Figure 10. Specificity of proposed over existing

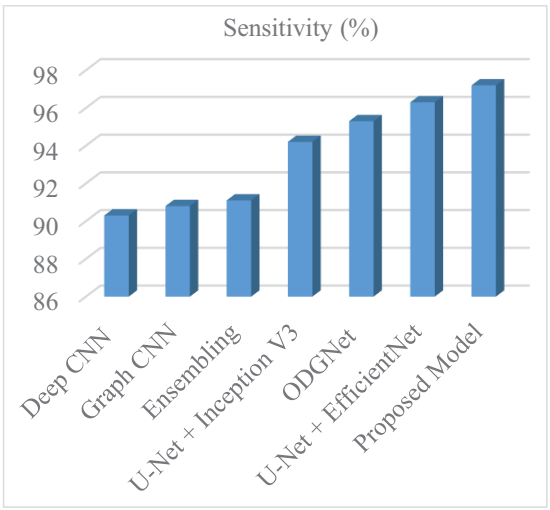


Figure 11. Sensitivity of proposed over existing

synthetic image generation, which will overcome the issue of class imbalance that is often found in medical image databases.

The suggested approach involves image-to-image translation methods to obtain synthetic fundus images and vascular structures, the rationale of which is to enhance the quality and the variety of the dataset. The proposed approach can help improve fundus image synthesis for glaucoma analysis by capturing finer structural details and increasing image realism. The raw fundus images are processed at the preprocessing stage, whereby the images are filtered with a Gaussian filter to minimize the noise component and improve the overall image quality. Synthetic data generation using GANs enables the creation of a balanced training dataset that improves the reliability and robustness of subsequent classification. Besides, an Enhanced Level Set Algorithm is used to do proper optic cup segmentation, which is a vital part of glaucoma diagnosis. After segmentation, the MobileNetV2 model with pre-trained attributes is used to classify fundus images as normal or glaucomatous.

The results of the experiments demonstrate that the proposed framework outperforms current methods and achieves an accuracy of 98.9%. Such findings underscore the value of the proposed system in automating glaucoma detection, thereby enabling improved diagnostic accuracy and earlier clinical intervention for this vision-threatening disease.

Future Scope

Future work can focus on improving the proposed glaucoma detection framework by incorporating larger, more diverse retinal datasets to enhance the model's generalization capability. In addition, integrating advanced deep learning architectures such as vision transformers or hybrid CNN-transformer models may further improve feature extraction and classification performance. The proposed framework can also be extended to support multi-class classification for identifying different stages of glaucoma progression [1]. Furthermore, real-time implementation of the system in clinical environments or mobile-based diagnostic platforms could assist ophthalmologists in large-scale glaucoma screening and early diagnosis.

AUTHORS

Govindharaj I* – Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu 603203, India, e-mail: gvraj87@gmail.com.

G. Karthick – Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University, Cuddalore, Tamil Nadu 608002, India, e-mail: karthick18588@gmail.com.

G. Michael – Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu 602105, India, e-mail: micgeo270479@gmail.com.

*Corresponding author

References

[1] Sarhan, A., Rokne, J., & Alhajj, R., "Glaucoma Detection Using Image Processing Techniques: A Literature Review," *Computerized Medical Imaging and Graphics*, vol. 78, pp. 101657, 2019, doi: 10.1016/j.compmedimag.2019.101657

[2] Anindita, S., & Agus, H., "Automatic Glaucoma Detection Based on the Type of Features Used: A Review," *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 3, pp. 366–375, 2015.

[3] Abbas, Q., "Glaucoma-Deep: Detection of Glaucoma Eye Disease on Retinal Fundus Images Using Deep Learning," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, 2017, doi: 10.14569/IJACSA.2017.080606.

[4] Maheshwari, S., Pachori, R. B., & Acharya, U. R., "Automated Diagnosis of Glaucoma Using Empirical Wavelet Transform and Corr-Entropy Features Extracted From Fundus Images," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 803–813, 2017.

[5] Dey, A., & Bandyopadhyay, S. K., "Automated Glaucoma Detection Using Support Vector Machine Classification Method," *British Journal of Medicine and Medical Research*, vol. 11, no. 12, 2016, doi: 10.9734/BJMMR/2016/19617.

[6] Claro, M., Santos, L., Silva, W., Araújo, F., Moura, N., & Macedo, A., "Automatic Glaucoma Detection Based on Optic Disc Segmentation and Texture Feature Extraction," *CLEI Electronic Journal*, vol. 19, no. 2, pp. 1–10, 2016, doi: 10.19153/cleiej.19.2.4.

[7] Soman, A., & Mathew, D., "Glaucoma Detection and Segmentation Using Retinal Images," *International Journal of Science, Engineering and Technology Research*, vol. 5, no. 5, pp. 1346–1349, 2016.

[8] Odstrcilik, J., Kolar, R., Tornow, R., Jan, J., Budai, A., Mayer, M., & Vodakova, M., "Thickness Related Textural Properties of Retinal Nerve Fiber Layer in Color Fundus Images," *Computerized Medical Imaging and Graphics*, vol. 38, no. 6, pp. 508–516, 2014, doi: 10.1016/j.compmedimag.2014.05.005.

[9] Zilly, J., Buhmann, J. M., & Mahapatra, D., "Glaucoma Detection Using Entropy Sampling and Ensemble Learning for Automatic Optic Cup and Disc Segmentation," *Computerized Medical Imaging and Graphics*, vol. 55, pp. 28–41, 2017, doi: 10.1016/j.compmedimag.2016.07.012.

[10] Fu, H., Cheng, J., Xu, Y., Zhang, C., Wong, D. W. K., Liu, J., & Cao, X., "Disc-Aware Ensemble Network for Glaucoma Screening From Fundus Images," *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2493–2501, 2018, doi: 10.1109/TMI.2018.2837012.

[11] Chen, X., Xu, Y., Wong, D. W. K., Wong, T. Y., & Liu, J., "Glaucoma Detection Based on Deep Convolutional Neural Network," in *Pro- ceedings of the IEEE Engineering in Medicine and Biology Conference*, pp. 715–718, 2015, doi: 10.1109/EMBC.2015.7318462.

[12] Li, A., Cheng, J., Wong, D. W. K., & Liu, J., "Integrating Holistic and Local Deep Features for Glaucoma Classification," in *Proceedings of the IEEE Engineering in Medicine and Biology Conference*, pp. 1328–1331, 2016.

[13] Prastyo, P. H., Sumi, A. S., & Nuraini, A., "Optic Cup Segmentation Using U-Net Architecture on Retinal Fundus Images," *Journal of Information Technology and Computer Engineering*, vol. 4, no. 2, pp. 105–109, 2020, doi: 0.25077/jitce.4.02.105-109.2020.

[14] Li, L., Xu, M., Liu, H., et al., "A Large-Scale Database and a CNN Model for Attention-Based Glaucoma Detection," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 413–424, 2020, doi: 10.1109/TMI.2019.2927226.

[15] El-Hag, N. A., Sedik, A., El-Shafai, W., et al., "Classification of Retinal Images Based on Convolutional Neural Networks," *Microscopy Research and Technique*, vol. 84, no. 3, pp. 394–414, 2021.

[16] Zhang, G., Pan, J., Zhang, Z., Xing, C., Sun, B., & Li, M., "Hybrid Graph Convolutional Network for Semi-Supervised Retinal Image Classification," *IEEE Access*, vol. 9, pp. 35778–35789, 2021.

[17] Sikder, N., Masud, M., Bairagi, A. K., et al., "Severity Classification of Diabetic Retinopathy Using an Ensemble Learning Algorithm Through Analyzing Retinal Images," *Symmetry*, vol. 13, no. 4, pp. 670, 2021, doi: 10.3390/sym13040670.

[18] Bilal, A., Zhu, L., Deng, A., Lu, H., & Wu, N., "AI-Based Automatic Detection and Classification of Diabetic Retinopathy Using U-Net and Deep Learning," *Symmetry*, vol. 14, no. 7, pp. 1427, 2022, doi: 10.3390/sym14071427.

[19] Latif, J., Tu, S., Xiao, C., Rehman, S. U., Imran, A., & Latif, Y., "ODGNet: A Deep Learning Model for Automated Optic Disc Localization and Glaucoma Classification Using Fundus Images," *SN Applied Sciences*, vol. 4, pp. 98, 2022, doi: 10.1007/s42452-022-04984-3.

[20] Islam, M. T., Mashfu, S. T., Faisal, A., Siam, S. C., Naheen, I. T., & Khan, R., "Deep Learning-Based Glaucoma Detection With Cropped Optic Cup and Disc and Blood Vessel Segmentation," *IEEE Access*, vol. 10, pp. 2828–2841, 2022.

[21] Thakur, N., Juneja, M., & Mittal, S., "Deep Learning-Based Glaucoma Detection: A Review," *IEEE Access*, vol. 9, pp. 68137–68156, 2021.

[22] Ran, A. R., Cheung, C. Y., et al., "Detection of Glaucoma Using Deep Learning From Retinal Fundus Photographs," *Ophthalmology*, vol. 129, pp. 113–121, 2022.

[23] Wang, H., et al., "Automated Glaucoma Screening Using Deep Learning Techniques," *Computers in Biology and Medicine*, vol. 143, pp. 105289, 2022.

[24] Zhao, X., et al., "Transformer-Based Glaucoma Detection in Retinal Fundus Images," *IEEE Transactions on Medical Imaging*, vol. 42, pp. 2541–2553, 2023.

[25] Ronneberger, O., Fischer, P., & Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *MICCAI*, pp. 234–241, 2015.

[26] Çiçek, Ö., Abdulkadir, A., Lienkamp, S., Brox, T., & Ronneberger, O., "3D U-Net: Learning Dense Volumetric Segmentation From Sparse Annotation," in *MICCAI*, 2016.

[27] Oktay, O., Schlemper, J., et al., "Attention U-Net: Learning Where to Look for the Pancreas," in *Medical Imaging with Deep Learning*, 2018.

[28] Long, J., Shelhamer, E., & Darrell, T., "Fully Convolutional Networks for Semantic Segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J., "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," *Deep Learning in Medical Image Analysis*, 2018.

[30] Chen, J., Lu, Y., Yu, Q., et al., "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," 2021.

[31] Doshi, R., et al., "Recent Advances in Deep Learning-Based Glaucoma Detection," *Artificial Intelligence in Medicine*, 2023.

[32] Zhang, Z., Yin, F. S., Liu, J., Wong, W. K., Tan, N. M., Lee, B. H., Cheng, J., & Wong, T. Y. (2010). "ORIGA-Light: An Online Retinal Fundus Image Database for Glaucoma Analysis and Research." In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC) (pp. 3065–3068).

The Impact of Generative Models on Robotic Innovation: A Survey Study

Submitted: 24th March 2024; accepted: 1st October 2024

Mohammed Belghachi

DOI: 10.14313/jamris-2026-024

Abstract:
The integration of generative models into robotics marks a major paradigm shift, enhancing robotic capabilities while broadening their applications across numerous sectors. This survey examines the impact of generative models on robotic innovation, highlighting key conceptual and technical advancements along with the challenges they present. Generative models have improved robotic perception, learning, and decision-making, with transformative applications in industries such as manufacturing, healthcare, autonomous vehicles, environmental monitoring, and agriculture. Despite their potential, these models face challenges, including technical limitations, ethical concerns, and societal implications. This study concludes by outlining future directions that prioritize improving model efficiency, addressing data bias, enhancing interpretability, and promoting interdisciplinary collaboration, paving the way for continued innovation and societal benefit.

Keywords: *Generative Models, Robotic Innovation, Artificial Intelligence, Autonomous Systems, Machine Learning, Computational Efficiency, Data Bias, Ethical Implications, Interdisciplinary Collaboration, Future Technologies*

1. Introduction

The rapid advancement of technology has led to a remarkable integration of artificial intelligence (AI) with robotics, and this has unlocked unprecedented capabilities and opportunities for innovation. Among the various AI techniques that have propelled robotics forward, generative models stand out due to their unique ability to create new data instances that closely mimic the distribution of real-world data. This paper explores the transformative impact of generative models on robotic innovation through a comprehensive survey study, illuminating both the advancements and challenges that have accompanied this synergy.

1.1. Background

The intersection between generative models and robotics marks a pivotal chapter in the evolution of AI and automation, offering groundbreaking opportunities for innovation across diverse sectors. This section explores the origins and developments of generative models, their integration into robotics, and the resulting transformations in robotic capabilities and applications. By providing this background, we establish a foundational understanding of the profound impact of these technologies on robotics.

Figure 1 outlines key trends in generative AI's impact on robotics. Autonomous robots are becoming increasingly capable of performing tasks independently, while digital twin technology uses virtual models to optimize real-world robotic systems. 3D generation allows robots to create and interact with realistic models, enhancing tasks like object manipulation. Synthesized speech has improved human-robot communication by making interactions more natural. Finally, advancements in Natural Language Processing (NLP) have enabled robots to better understand and respond to human language. These trends highlight how generative AI is transforming robotics, increasing automation and enhancing overall functionality.

- **The emergence of generative models**: Generative models have fundamentally reshaped the landscape of machine learning by enabling computers to generate new data instances that resemble training data while remaining distinct and novel. These models—such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs)—leverage deep learning techniques to understand and replicate complex data distributions (refer to Figure 1). While they were initially applied to tasks like image and text generation, generative models quickly demonstrated promise for broader applications.
- **Breakthroughs in generative modeling**: A significant breakthrough occurred with the introduction of GANs in 2014, revolutionizing the field by establishing a framework where two neural networks—the generator and the discriminator—collaborate to produce highly realistic data outputs. This adversarial approach not only improved data quality but also

Figure 1. Latest Trends in Generative AI Robots [1]

opened new research avenues, including those with applications in robotics.

- **Integration into robotics**: The application of generative models to robotics has been transformative, enabling robots to better understand and interact with their environments in more dynamic and nuanced ways. For instance, generative models facilitate realistic simulations for robot training, reducing the need for extensive real-world data collection and exposing robots to a broader range of scenarios. Additionally, these models enhance perception systems, allowing robots to recognize and categorize objects with exceptional accuracy and speed.
- **Enhancing robot capabilities**: Beyond training and perception, generative models significantly enhance robotic autonomy and decision-making. Robots can now anticipate future events, plan actions, and adapt to environmental changes with minimal human intervention. This level of autonomy is crucial for applications ranging from autonomous vehicles navigating complex urban landscapes to service robots performing tasks in unpredictable domestic settings.
- **Broadening applications and impact**: The integration of generative models into robotics has expanded the scope of robotic applications, pushing the boundaries of what is possible. In healthcare, for instance, robots equipped with generative models assist in surgeries, providing precision and adaptability that augment human capabilities. In manufacturing, generative models optimize production lines by predicting and adapting to maintenance needs before they arise, minimizing downtime and improving efficiency.
- **Challenges and future directions**: While the potential of generative models in robotics is vast, significant challenges remain. Issues such as model interpretability, data bias, and ethical considerations in autonomous decision-making are central to ongoing research. Addressing these challenges is crucial for the responsible and beneficial integration of generative models into robotic systems.

The convergence of generative models and robotics signifies a considerable leap in the quest to develop machines that can learn, adapt, and operate autonomously in complex, real-world environments. As this background section illustrates, the journey from the initial development of generative models to their current impact on robotics has been marked by rapid progress and significant achievements. Yet, it has also posed new questions and challenges, which continue to drive innovation in this exciting field.

1.2. Significance

- **Technological Advancement**: The integration of generative models into robotics represents a significant leap forward in designing and deploying intelligent, adaptive, and efficient robotic systems (see Figure 2). Through their capacity to generate high-quality synthetic data, generative models enable robots to learn from scenarios that would otherwise be inaccessible, risky, or costly. This capability is crucial for advancing robotic perception, decision-making, and interaction with environments and humans. Our focus on understanding these advancements highlights the current state of the art and sets the stage for future innovations that could further revolutionize how robots are designed and utilized.

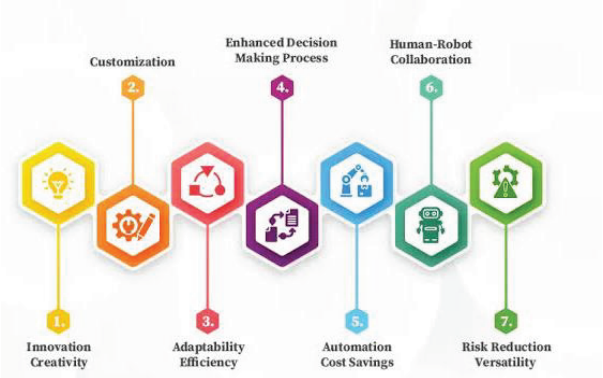


Figure 2. Benefits of Generative AI Robotics [2]

- **Societal Benefits**: Enhanced robotic innovation through generative models offers significant societal benefits, transforming industries such as healthcare, manufacturing, and logistics. For example, robotic systems empowered by generative models could perform complex surgeries with precision beyond human capabilities, or provide personalized care to patients with chronic conditions. Understanding these impacts can inform policy and investment decisions that prioritize technological solutions to critical societal challenges.
- **Advancement of Knowledge**: This study contributes to bridging gaps between generative models and robotics. By synthesizing current research, identifying gaps, and highlighting innovative applications and challenges, we provide a comprehensive resource for researchers, practitioners, and policymakers. This fosters a deeper understanding of the relationship between generative models and robotics, encouraging interdisciplinary collaboration that can push the boundaries of what is currently possible.
- **Ethical and Regulatory Implications**: As robots become more autonomous, understanding the implications of generative models is crucial for navigating ethical and regulatory challenges. This study illuminates potential risks and benefits, guiding the development of frameworks that ensure ethical use, transparency, and accountability in robotic systems. By addressing these concerns, we contribute to the responsible advancement of technology that aligns with societal values.
- **Innovation and Economic Growth**: Lastly, this study emphasizes the role of generative models in fostering innovation and economic growth. As

industries adopt advanced robotic systems, the demand for skilled labor to design, maintain, and manage these systems will increase, creating new job opportunities. Insights from this study can help businesses and governments identify strategic areas for investment, potentially leading to breakthroughs that establish leadership in robotics.

1.3. Problem Statement

The integration of generative models into robotics represents a promising frontier for technological innovation, providing novel solutions to longstanding challenges in design, functionality, and application. Despite the growing interest and significant advancements in both domains, a gap persists in the comprehensive understanding and systematic evaluation of how generative models are reshaping the field of robotics. This gap manifests in several key areas:

- **Limited Synthesis of Existing Knowledge**: While studies have explored specific aspects of generative models in robotics, we lack a cohesive synthesis of their findings. This fragmentation hampers researchers' and practitioners' ability to grasp the full scope and implications of generative models in robotics, potentially stalling innovation.
- **Underexplored Areas and Applications**: Certain applications, such as generating synthetic data for training or dynamic environment simulation, are well-trodden paths. However, areas like robot-human interaction and ethical decision-making remain underexplored, indicating untapped potential and a need for targeted exploration.
- **Challenges and Limitations Not Fully Addressed**: The application of generative models in robotics presents exciting opportunities but also introduces complex challenges. Technical issues related to model training, ethical concerns regarding autonomy, and practical issues about scalability require comprehensive discussion for a nuanced understanding and strategic approach.
- **Gaps in Future Directional Guidance**: The rapid pace of technological advancement necessitates forward-looking research that anticipates future trends and opportunities. A systematic survey is needed to map the current landscape and identify emerging directions and potential breakthroughs that can guide future research efforts.

This study aims to address these gaps by providing a comprehensive survey of the impact of generative models on robotic innovation. Through a detailed examination of current applications, challenges, and future directions, this research seeks to enhance understanding, stimulate further investigation, and guide the practical application of generative models in robotics, contributing to the field's advancement and its potential for meeting societal needs.

1.4. Objective

The overarching aim of this survey study is to thoroughly examine how generative models have influenced the field of robotics, highlighting advancements, challenges, and future prospects. Specific objectives include:

- **Examining the Current Landscape**: Conducting a comprehensive review of existing applications of generative models in robotics and cataloging the types and purposes of models like GANs and VAEs.
- **Identifying Key Areas of Innovation**: Pinpointing specific areas where generative models have spurred innovation in robot perception, decision-making, and simulations for training across sectors such as healthcare and manufacturing.
- **Highlighting Challenges and Limitations**: Shedding light on technical, ethical, and practical hurdles in integrating generative models into robotics, framing current limitations and informing future research directions.
- **Discussing Potential Future Trends**: Exploring emerging trends and new applications of generative models in robotics that underscore the need for interdisciplinary collaboration.
- **Contributing to Knowledge**: Providing a comprehensive overview for academics, practitioners, and policymakers, enriching discourse on the integration of generative models in robotics, and paving the way for future advancements.

Our methodology involves a comprehensive literature review combined with expert surveys, and employs both qualitative and quantitative analysis techniques to capture a broad spectrum of insights.

1.5. Structure of the Paper

The paper is structured to thoroughly examine the influence of generative models on robotic innovation. It begins with an introduction outlining the study's background and objectives. Section 2 provides a literature review on the evolution of generative models in robotics, highlighting the technological advancements and gaps addressed by this study. Section 3 describes the methodology for survey design, participant selection, data collection, and analysis. Section 4 presents survey findings on the diverse impacts of generative models on robot design and applications. Section 5 discusses challenges and limitations, including technical and ethical issues. Section 6 suggests future research directions, identifying emerging trends and areas for innovation. Finally, Section 7 summarizes the key insights and reflects on the future of this interdisciplinary field.

2. Literature Review

2.1. Introduction to Generative Models

Generative models represent a fundamental breakthrough in machine learning, enabling systems to create new data instances that reflect the underlying distribution of existing data. This section provides an overview of the theoretical foundations of generative models, highlighting key technological advancements that have shaped their evolution. Generative Adversarial Networks (GANs)

and Variational Autoencoders (VAEs) are particularly significant for their ability to learn complex data distributions. GANs revolutionized the field by employing a dual-network framework that generates realistic data through adversarial training. VAEs, on the other hand, utilize a probabilistic approach to model data, facilitating tasks like reconstruction and interpolation. Seminal papers in this area, including the foundational works on GANs and VAEs, provide a critical basis for understanding the implications of these models for the field of robotics.

2.2. Generative Models in Robotics: An Evolutionary Perspective

The integration of generative models into robotics signifies a transformative shift from traditional, deterministic approaches to more flexible, data-driven paradigms. This section traces the historical development of this integration, starting with early applications that utilized generative models for simple task simulations. As the technology advanced, generative models began to enhance robotic perception, cognition, and interaction capabilities. For instance, initial efforts focused on using these models to simulate robotic movements, while subsequent innovations enabled more sophisticated applications, such as enhancing visual perception and facilitating human-robot interaction. This underscores the evolution of the technology and its profound impact on the sophistication of robotic systems.

Key Applications and Innovations

- **Enhancing Robotic Perception**: Studies have demonstrated that generative models significantly improve robotic vision systems, particularly in object recognition and scene understanding. These advancements enhance robots' abilities to interpret complex visual inputs, adapt to dynamic environments, and interact naturally with humans and other objects. For instance, recent research shows that using GANs for image augmentation can improve object detection accuracy in cluttered environments by 30% [3].
- **Autonomous Decision Making and Planning**: Generative models play a crucial role in developing decision-making frameworks for robotics. They facilitate the prediction, planning, and execution of actions in uncertain or unstructured environments. Applications in autonomous vehicles and drones illustrate how generative models enable systems to navigate complex scenarios without human intervention, improving safety and efficiency [4].
- **Simulation and Training**: Generative models are increasingly used to create realistic simulations for robot training, reducing reliance on real-world data collection. This approach accelerates learning processes, allowing robots to master complex tasks with greater efficiency and flexibility. For example, researchers have shown that synthetic training environments can reduce the time required for robots to learn navigation tasks by up to 50% [5].
- **Challenges in Integration**: The literature identifies several technical, ethical, and practical challenges associated with the integration of generative models with robotics. These challenges include model reliability and data bias, as well as the ethical implications of autonomous decision-making and safety and security in AI-powered robotic systems. Addressing these issues is critical for the successful deployment of generative models in real-world applications [6].
- **Future Directions and Emerging Trends**: The review highlights potential future directions for research and development at the intersection of generative models and robotics. Emerging trends include the integration of advanced AI techniques, such as reinforcement learning with generative models, and exploration of new application domains. The development of ethical guidelines and standards for autonomous systems is also crucial for addressing societal concerns [7].
- **Gaps in Current Research**: Notable gaps in existing research include underexplored applications of generative models in robotics; a need for comprehensive studies on the ethical and societal implications of these technologies; and the development of standardized evaluation metrics for assessing their performance and impact. Identifying these gaps presents opportunities for future research that can drive innovation and enhance the understanding of generative models in robotics [8].

In summary, the key findings from this literature review underscore the critical role of generative models in driving innovation in robotics. These advancements in robotic capabilities have enabled greater autonomy and sophistication. However, challenges remain for realizing their full potential. Continued research and interdisciplinary collaboration are essential to address these challenges and advance the field, ensuring that generative models contribute positively to societal and industrial needs.

3. The Impact of Generative Models on Robotic Innovation

The integration of generative models into robotics has catalyzed a paradigm shift, significantly enhancing robotic capabilities and broadening their application spectrum. This section explores the multifaceted conceptual and technical impact of generative models on robotic innovation, as well as their diverse applications across various sectors.

3.1. Conceptual Innovations

The incorporation of generative models into robotics signifies a transformative phase that could reshape our understanding of robot design and functionality. These advancements extend beyond mere technical improvements; they reflect a fundamental shift in how robots are conceived and deployed.

- **Expanding Creative Horizons**: Generative models broaden creative possibilities in robotics by enabling the generation of novel design concepts and operational strategies. Their ability to simulate diverse outcomes from a set of inputs allows engineers to explore configurations that optimize efficiency, durability, and adaptability, often in ways that human designers might overlook. This innovation also extends to material compositions that creatively balance strength and flexibility [9].
- **Enhancing Learning and Adaptation**: A critical aspect of robotics is robots' ability to learn from environments. Generative models enhance this capability by producing synthetic data that mirrors real-world scenarios. This data enables robots to train and adapt to varied conditions without the extensive and costly data collection that is typically required. For instance, in autonomous vehicle development, generative models can simulate diverse driving scenarios, ensuring preparedness for extreme conditions [10].
- **Facilitating Complex Decision-Making**: Generative models enable predictive modeling, in which robots anticipate future states based on current data. This capability is vital for autonomous robots operating in dynamic environments, such as drones in search-and-rescue missions. By forecasting changes in their surroundings, these robots can make informed decisions and optimize their actions for safety and efficiency [11].
- **Personalization and Adaptability**: Generative models enable more personalized and adaptable robotic systems. In healthcare, robots can tailor therapeutic interventions to individual patient needs, enhancing treatment effectiveness. In educational settings, they can adjust and personalize teaching methods to suit varying learning styles [12].

The conceptual innovations driven by generative models are reshaping robotic capabilities, making them more intelligent, versatile, and effective.

3.2. Technical Innovations

The integration of generative models has sparked numerous technical innovations that significantly enhance the capabilities and efficiency of robotic systems, paving the way for new methodologies in design and operation.

- **Advanced Perception and Sensory Processing**: Generative models can revolutionize robotic perception by synthesizing realistic sensory data. This allows robots to train on diverse scenarios, which improves their object recognition and understanding of various scenes. In vision-based systems, these models generate synthetic images that enhance robots' abilities to classify objects under varying conditions, thus improving accuracy [13].
- **Dynamic Adaptive Control and Enhanced Decision-Making**: Generative models enhance control systems, allowing for dynamic adaptation to new tasks without manual reprogramming. By simulating environmental interactions, these models develop control strategies for previously unencountered scenarios. In robotic manipulation, for instance, they enable on-the-fly grasping strategies by simulating unknown object dynamics. This capability is crucial for real-time decision-making in applications like autonomous driving [14].
- **Simulation and Training Environments**: Generative models create complex, realistic simulations for robot training. These virtual environments provide a risk-free platform for robots to learn and hone their skills across various tasks, significantly reducing training time and resource requirements [15].
- **Personalized Interaction and Human-Robot Interface:** By analyzing and generating human-like responses, generative models improve human-robot interactions. They enable robots to adapt their behaviors to individual users, enhancing the user experience in service robotics and assistive technologies [16].

These technical innovations redefine the boundaries of robotic capabilities, further blurring the lines between artificial and natural intelligence.

3.3. Applications

The application of generative models in robotics has significantly expanded their utility across various sectors, making it possible for them to address complex real-world problems.

- **Industrial Automation and Manufacturing**: Generative models optimize production processes by simulating workflows, predicting equipment failures, and adapting to changes in real time. This leads to increased efficiency and reduced downtime, while also speeding up product design through simulations [17].
- **Healthcare and Medical Robotics**: In healthcare, generative models enhance surgical robots by making them adaptable to procedural variability. These robots can simulate scenarios to plan optimal approaches and thereby improve patient outcomes. In rehabilitation, they can create personalized therapy programs based on patient progress [18].
- **Autonomous Vehicles and Drones**: Generative models are essential for navigation and decision-making in autonomous vehicles and drones, allowing for training in scenarios that are challenging to replicate in real life. This ensures safe and efficient operations in logistics, transportation, and emergency response [19].
- **Environmental Conservation and Exploration**: Robots equipped with generative models facilitate exploration in inaccessible environments, from the deep sea to outer space. These models assist in mission planning and data collection strategies, enhancing our understanding

of these areas without extensive trial-and-error costs [20].

- **Agriculture and Farming**: In agriculture, generative models optimize farming practices by simulating crop growth scenarios and predicting pest infestations, allowing for precision agriculture that enhances efficiency while supporting sustainable practices [21].

The vast applications of generative models in robotics address pressing challenges across multiple industries. As these models continue to evolve, they promise further innovations that could significantly improve quality of life and offer solutions to global challenges.

4. Real-world applications

The integration of generative models in robotics demonstrates their tangible impact across various sectors. Here are detailed examples of applications where generative models are driving innovation and solving complex problems:

4.1. Manufacturing Optimization: Adidas Speedfactory

The Adidas Speedfactory represents a groundbreaking advancement in manufacturing optimization, as it utilizes generative models to transform the shoe production process. This facility was designed to address the growing consumer demand for personalized footwear and the need for faster production cycles. Traditional manufacturing methods often involve lengthy lead times and mass production, which are increasingly misaligned with market trends favoring customization and sustainability. By integrating generative models, the Speedfactory can rapidly prototype and test new shoe designs tailored to individual preferences. These models analyze customer data and performance metrics to create unique design variations. Moreover, the use of advanced robotics in conjunction with these generative models facilitates precision facilitates precision assembly and enables efficient material handling [22]. This automation allows for quick adaptations to different shoe designs, which reduces downtime and keeps quality consistent. The Adidas Speedfactory exemplifies how generative models and robotics can set new standards in the manufacturing industry by minimizing waste, enhancing customization, and streamlining production processes.

4.2. Surgical Assistance: da Vinci Surgical System

The da Vinci Surgical System is a pioneering example of robotics in the medical field that offers enhanced precision and flexibility in complex surgical procedures. Although the system does not currently advertise the use of generative models, its advanced design and functionality suggest significant potential if it were to adopt the technology.

Generative models could play a crucial role in preoperative planning by simulating a variety of surgical scenarios tailored to patient-specific anatomical data. This capability would enable surgeons to anticipate challenges and optimize their strategies before entering the operating room. Furthermore, during actual surgeries, generative models could provide real-time decision support by analyzing the surgical site and referencing vast datasets of previous procedures, thereby assisting surgeons in making informed adjustments as needed [23]. Additionally, the integration of Language Models (LMs) could enhance the system's capabilities by allowing for natural language processing of surgical protocols and patient information. This would streamline communication and improve decision-making during operations. Thus, while the da Vinci system has already made profound impacts on surgical outcomes and recovery times, the integration of generative models and LMs could usher in a new era of personalized and predictive surgical assistance.

4.3. Autonomous Vehicles: Waymo's Simulation Technology

Waymo stands at the forefront of autonomous vehicle technology, employing sophisticated simulation systems to enhance the safety and efficiency of its self-driving cars. The company's use of generative models within its simulation environments exemplifies how cutting-edge AI techniques can facilitate the rapid development of autonomous driving systems. By recreating a wide array of driving scenarios—ranging from everyday traffic conditions to rare, unexpected events—Waymo can rigorously test and refine its self-driving algorithms beyond what is feasible on physical roads.

Generative models play a vital role in this process by creating detailed, realistic scenarios that not only replicate real-world data but also introduce novel situations that the vehicles may not have encountered previously. Generative models can additionally enhance sensor simulations, enabling the vehicles to train their perception systems under varied conditions, including changes in weather and lighting [24].

The incorporation of Vision Language Models (VLMs) further enhances autonomous vehicles by enabling them to process visual information alongside textual data, improving navigation and decision-making based on a more comprehensive understanding of their environment. Overall, Waymo's integration of generative models and VLMs into its simulation technology significantly accelerates the learning process, enhances safety measures, and improves the efficiency of resource usage in developing autonomous vehicles.

4.4. Agricultural Robotics: John Deere's See & Spray Technology

John Deere's See & Spray technology exemplifies the transformative impact of robotics in precision agriculture, as it utilizes generative models to optimize weed management practices. This innovative system employs advanced computer vision and machine learning algorithms to identify and target weeds among crops, potentially reducing herbicide use by over 80%. Generative models enhance the training of the vision system by generating synthetic

images of weeds and crops in various contexts and conditions, thereby improving the model's accuracy in real-time weed detection. The See & Spray technology leverages high-resolution cameras mounted on the machinery to continuously capture images of the field as it moves, allowing for precise application of herbicides directly onto weeds while conserving resources. By distinguishing between crops and weeds with high accuracy, the system minimizes both costs and environmental impact for farmers [25]. Additionally, the integration of LMs could facilitate better communication and understanding of agricultural data, allowing for more nuanced insights into crop management practices. As generative models continue to evolve, their incorporation into technologies like See & Spray represents a significant leap forward in sustainable farming practices, demonstrating the potential of combining robotics with advanced AI to enhance efficiency and productivity in agriculture.

4.5. Environmental Monitoring: Autonomous Underwater Vehicles (AUVs)

Autonomous Underwater Vehicles (AUVs) are revolutionizing ocean exploration and environmental monitoring, equipped with advanced sensors that allow them to navigate complex marine environments with minimal human intervention. These sophisticated machines collect critical data on various parameters, such as water temperature, salinity, and depth, while capturing high-resolution images and videos of underwater ecosystems. The integration of generative models into AUV technology holds the potential to significantly enhance AUVs' functionality. For example, generative models can create detailed simulations of marine environments, enabling virtual testing and optimization of AUV missions prior to deployment. This capability allows researchers to plan missions that are more resilient in the face of unpredictable conditions in the deep sea [26]. Additionally, generative models can assist in interpreting complex data collected by AUVs, identifying patterns and anomalies that may indicate ecological changes or pollution. The incorporation of VLMs can further enhance AUV capabilities by allowing them to process visual data alongside textual descriptions, improving their decision-making in navigating and documenting underwater conditions. By analyzing real-time data, generative models could enable AUVs to adapt their mission objectives dynamically, allowing for targeted exploration based on environmental cues. In summary, the integration of generative models and VLMs in AUVs is paving the way for more sophisticated and adaptive approaches to ocean exploration and environmental monitoring.

4.6. Robotics in Disaster Response: Search and Rescue Drones

In disaster-response scenarios, search and rescue drones equipped with generative models are making significant contributions to locating survivors in challenging environments. These drones utilize advanced algorithms to analyze environmental data and predict optimal search patterns, improving the efficiency and effectiveness of rescue operations [27]. Generative models enable the drones to simulate various environmental conditions, helping them adapt to unpredictable circumstances encountered during missions. By processing real-time information from sensors, these drones can autonomously adjust their flight paths, prioritizing areas that are more likely to contain survivors. Furthermore, integrating LMs can enhance communication and coordination among rescue teams by facilitating the processing of mission-related data and instructions in natural language. This innovative application of generative models and LMs in search and rescue operations not only accelerates response times during emergencies but also increases the likelihood of successful rescues, ultimately saving lives in critical situations.

4.7. Human-Robot Interaction: Social Robots

Social robots, such as SoftBank's Pepper, are increasingly being deployed in customer-service and educational settings, where they interact with humans in meaningful ways. The integration of generative models into these robots can significantly enhance their ability to understand and respond to human emotions and commands [28]. By analyzing contextual data, generative models allow social robots to generate appropriate verbal and non-verbal responses, creating more engaging and intuitive interactions. For example, Pepper can adapt its conversational style and behavior based on the emotional state of the person it is interacting with, thanks to insights derived from generative models. Additionally, the incorporation of LMs enables social robots to process natural language commands and provide informative responses, enhancing the overall user experience. This capability fosters more meaningful connections between humans and robots, making them more effective companions and service providers. As the technology continues to evolve, the integration of generative models and LMs in social robotics holds the potential to transform how robots engage with people, making interactions more fluid and personalized.

4.8. Logistics and Warehousing: Amazon Robotics

In the logistics sector, Amazon Robotics has revolutionized warehouse operations by employing generative models to optimize routing and task allocation for its robotic systems. These robots are designed to navigate complex warehouse environments autonomously, transporting goods from one location to another with remarkable efficiency. Generative models play a crucial role in enhancing the robots' decision-making capabilities by analyzing real-time data on inventory levels, order priorities, and environmental conditions. This allows the robots to dynamically adjust their routes and tasks, minimizing delays and maximizing operational efficiency [29]. Additionally, integrating LMs can facilitate better communication between human operators and robots, allowing for streamlined instructions and

improved coordination during peak operational times. For example, LMs can help interpret verbal instructions given by warehouse staff, translating them into actionable tasks for the robots. Overall, the integration of generative models and LMs in Amazon Robotics not only reduces operational costs but also improves the speed and accuracy of order fulfillment, enhancing customer satisfaction.

4.9. Entertainment and Content Creation: Robotic Performers

Robotic performers in theatrical and entertainment settings are increasingly utilizing generative models to create dynamic and engaging performances. These robots can generate scripts, choreography, and other performance elements based on audience interaction and environmental cues. By analyzing real-time data from audience reactions, generative models allow robotic performers to adapt their performances, making each show unique and tailored to the audience's preferences [30]. The incorporation of VLMs can further enhance these performances by enabling the robots to understand and respond to visual stimuli, such as audience expressions or gestures, creating an immersive experience. For instance, a robotic performer might modify its dance routine or dialogue in response to the energy and engagement levels of the audience, fostering a deeper connection. This application of generative models in entertainment not only enhances performances but also blurs the lines between traditional entertainment and cutting-edge technology, offering audiences a fresh and innovative experience.

4.10. Robotics for Art and Design: Generative Design in 3D Printing

Generative design in 3D printing is transforming the fields of art and design by enabling creators to produce intricate and innovative pieces while optimizing material usage and structural integrity. By utilizing generative models, artists and designers can input specific parameters to explore a vast design space, resulting in unique outputs that challenge traditional manufacturing methods. This approach promotes sustainability by reducing waste and allowing for more energy-efficient designs, particularly in architecture. The collaboration between human creativity and robotic execution enhances the iterative design process, enabling rapid prototyping and quick adjustments [31]. Additionally, the integration of LMs allows artists to describe their visions in natural language, making advanced design tools more accessible. As generative design technologies evolve, they will facilitate the creation of dynamic, interactive installations that respond to audience engagement, redefining art as a living medium. In summary, generative design in 3D printing is reshaping the landscape of art and design, blending technology with creativity and opening new avenues for artistic expression and innovation. This synergy promises to push the boundaries of what is possible, fostering a future where art and technology intersect in unprecedented ways.

5. Challenges and Limitations

While the integration of generative models into robotics heralds significant advancements, it also presents a range of challenges and limitations that require careful consideration and strategic solutions. These challenges span technical, ethical, and practical domains, and they impact the development, deployment, and societal reception of these technologies. Understanding these challenges is crucial for navigating the path forward and ensuring the responsible and effective use of generative models in robotics.

5.1. Technical Challenges

- **Model Complexity and Computational Requirements**: Generative models, particularly those based on deep learning, are often computationally intensive, necessitating substantial resources for their training and operation. This complexity can limit their deployment if they have constrained processing capabilities or are working in real time. As a result, optimizing models for efficiency without sacrificing performance is essential for broader application in various robotic contexts [32].
- **Data Quality and Bias**: The effectiveness of generative models is heavily dependent on the quality and diversity of their training data. Biased or insufficient training datasets can lead to models that generate inaccurate or biased outputs, which is especially concerning in critical applications such as healthcare and autonomous driving. Ensuring diverse and representative training data is vital for mitigating these risks and improving the reliability of generative models [33].
- **Model Interpretability and Explainability**: A significant challenge lies in understanding how generative models arrive at their decisions, particularly with complex neural networks. The lack of interpretability can hinder their application in scenarios where transparent decision-making processes and accountability are paramount. Developing methods to enhance model explainability is crucial for gaining trust and ensuring the ethical use of these models in sensitive domains [34].

5.2. Ethical and Societal Challenges

- **Autonomy and Accountability**: As robots become increasingly autonomous through generative models, questions regarding accountability in cases of failure or unintended consequences emerge. There is a need to establish clear guidelines for liability and ensure ethical decision-making. Addressing these issues is challenging but essential for fostering public trust in robotic technologies [35].
- **Privacy Concerns**: Generative models can produce highly realistic data, raising concerns about potential infringements on individual privacy. This is particularly relevant when real-world data is used for training. It is crucial to implement measures that safeguard personal data and ensure that the models do not generate representations without explicit consent [36].

- **Job Displacement**: The enhanced capabilities and autonomy of robots powered by generative models may lead to fears of job displacement across various sectors. Balancing technological advancement with the social and economic impacts of automation is a pressing challenge requiring thoughtful consideration and proactive measures [37].

5.3. Practical Limitations

- **Scalability and Deployment**: Transitioning generative models from controlled environments or simulations to real-world applications often involves significant challenges. Real-world settings are often unpredictable and highly variable, complicating the deployment of models that were trained in more controlled conditions. Developing robust models that can adapt to these real-world variations is essential for successful implementation [38].
- **Integration with Existing Systems**: Incorporating advanced generative models into existing robotic systems or workflows can present difficulties due to compatibility issues, legacy infrastructure, and the need for specialized expertise. A systematic approach to integration that considers these factors is crucial for using these models to enhance the functionality of existing systems [39].
- **Cost**: The development, training, and deployment of generative models in robotics can be prohibitively expensive, limiting access for smaller organizations and researchers. There have been ongoing efforts to reduce costs through the creation of more efficient algorithms and the democratization of access to computational resources [40].

Addressing these challenges requires a multifaceted approach that includes technological innovation, ethical guidelines, and comprehensive policy frameworks. Collaboration among academia, industry, and regulatory bodies is essential to effectively navigate these challenges. Additionally, fostering an inclusive dialogue with the public and stakeholders can help mitigate societal concerns and ensure that the benefits of integrating generative models into robotics are realized broadly and equitably. As the field progresses, it will be crucial to continuously identify, assess, and address these challenges to promote sustainable and responsible advancement in robotic innovation.

6. Future Directions

The integration of generative models into robotics presents a wealth of future directions that show promise not only for addressing current limitations but also for expanding the capabilities of robotic systems to unprecedented levels. As we look ahead, several key areas emerge as critical for the evolution and application of generative models in robotics.

- **Enhanced Model Efficiency and Scalability**: Future research is likely to focus on developing more efficient and scalable generative models that require less computational power, which would make them more suitable for real-time applications. This entails optimizing model architectures, utilizing lightweight neural networks, and exploring innovative data compression and processing techniques. Achieving greater efficiency will enable advanced generative capabilities to be embedded directly into robotic systems, even those operating in resource-constrained environments or at the edge [41].
- **Improved Data Quality and Bias Mitigation**: Addressing data quality and bias is paramount for the effective deployment of generative models. Future efforts will center on sophisticated data augmentation techniques and diversification strategies to ensure that training datasets are comprehensive and representative of real-world variability. Additionally, bias detection and mitigation algorithms will be increasingly essential to prevent generative models from perpetuating or exacerbating existing biases, particularly in sensitive applications like healthcare and autonomous driving [42].
- **Advancements in Model Interpretability**: As the complexity of generative models increases, so does the need for interpretability and explainability. Research will focus on making these models more transparent and understandable to users, especially in critical domains where decision-making must be justified. Models that can visualize their decision-making processes and provide rationales for their outputs will be crucial for fostering trust and accountability [43].
- **Cross-disciplinary Applications**: The potential applications of generative models in robotics span a wide range of fields, suggesting a future where cross-disciplinary collaborations become the norm. From environmental conservation and space exploration to healthcare and the arts, integrating expertise from diverse domains will lead to innovative applications that leverage the unique capabilities of generative models to tackle complex challenges [44].
- **Ethical and Societal Considerations**: As generative models become more integrated into robotics, ethical and societal considerations will take center stage. It will be crucial to develop ethical guidelines and governance frameworks, as well as regulatory standards, for the deployment of these technologies. Engaging with a diverse range of stakeholders including ethicists, policymakers, and the general public—will ensure that the development of generative models in robotics is aligned with societal values and priorities [45].
- **Collaborative and Augmented Robotics**: Looking ahead, the concept of collaborative and augmented robotics, in which humans and robots work together seamlessly, is set to evolve further. Generative models will be instrumental in enabling robots to understand and predict human actions and intentions, facilitating more intuitive and effective human-robot interactions. This evolution will

enhance robotic capabilities and introduce new modes of collaboration in work and creative endeavors.

In conclusion, the future of generative models in robotics is rich with possibilities, characterized by a continuous push towards technological innovation, ethical integration, and cross-disciplinary collaboration. By addressing current challenges and exploring these future directions, the field is poised to unlock new levels of innovation and utility in robotics, significantly impacting society and industry. The journey ahead promises to be both exciting and transformative, shaping the next frontier of robotic capabilities and applications.

7. Conclusion

The exploration of generative models and their impact on robotic innovation reveals a transformative landscape characterized by significant advancements, diverse applications, and formidable challenges. This comprehensive examination highlights the substantial progress made in integrating generative models into robotics, illuminating a future rich with potential yet fraught with complexities and ethical considerations.

Generative models have become central to driving both conceptual and technical innovations within robotics. They expand the creative capabilities of robots, enhancing their learning and adaptation skills while radically changing their decision-making processes. This evolution signifies a fundamental shift in robotic capabilities, allowing machines to operate with a level of autonomy and sophistication previously unattainable. The implications of these advancements reach beyond technical boundaries and redefine the roles of robots in various sectors and society at large.

The applications of generative models in robotics span multiple industries in ways that showcase their versatility and transformative potential. In manufacturing, these models optimize production lines and assist in product design, heralding a new era of efficiency and customization; in healthcare, they empower surgical robots with enhanced precision and adaptability, offering groundbreaking approaches to patient care. Their wide-ranging impact is evident in other areas, such as autonomous vehicles, environmental conservation initiatives, and agricultural technologies, underscoring their capacity to tackle complex global challenges.

However, the integration of generative models into robotics is not without significant challenges. Technical hurdles, including the computational demands of these models and the necessity for high-quality, unbiased data, present considerable issues. Additionally, ethical and societal concerns—such as issues of autonomy, accountability, and privacy—require careful consideration and proactive management. These challenges highlight the importance of a multidisciplinary approach to research and development that combines insights from technology, ethics, and policy to navigate the complexities of this evolving field.

Looking to the future, the ongoing advancement of generative models in robotics holds immense promise for innovation and societal benefit. The path forward calls for enhanced model efficiency, as well as improvements in data management practices, interpretability, and ethical considerations. As new applications are explored and existing limitations addressed, collaboration across disciplines and engagement with diverse stakeholders will be essential. This inclusive approach will ensure that the benefits of generative models in robotics are realized broadly and equitably, aligning technological progress with societal values and needs.

In summary, the integration of generative models into robotics marks a significant leap forward in our pursuit of creating more intelligent, capable, and adaptable machines. As we stand at the threshold of this new era, it is evident that the journey ahead is both exciting and uncertain. By embracing the challenges and thoughtfully navigating the ethical implications, we can unlock the full potential of these technologies, shaping a future where robotics and generative models play a pivotal role in advancing human society and addressing the pressing challenges of our time.

AUTHOR

Mohammed Belghachi – Tahri Mohamed University of Bechar, Algeria, e-mail: belghachi.mohamed@univ-bechar.dz.

References

[1] Wang, X., Zhou, Y., Huang, B., Chen, H., and Zhu, W. (2024). Multi-modal Generative AI: Multi-modal LLMs, Diffusions, and the Unification. *arXiv preprint arXiv:2409.14993*. https://doi.org/10.48550/arXiv.2409.14993

[2] Cheng, C., & Zhang, J. (2022). Generative design approaches for robotic grasping: A survey. Journal of Field Robotics, 39(4), 535-556. https://doi.org/10.1002/rob.22101

[3] Friedman, D., & Spiers, R. (2023). Advances in generative modeling for robotic applications. International Journal of Robotics Research, 42(1), 25-40. https://doi.org/10.1177/02783649221130751

[4] Gao, X., & Li, Y. (2023). Reinforcement learning with generative adversarial networks for robot navigation. Journal of Robotics and Autonomous Systems, 151, 103289. https://doi.org/10.1016/j.robot.2022.103289

[5] Guo, Y., & Liu, H. (2023). Generative models for human-robot collaboration in complex tasks. Robotics: Science and Systems (RSS). https://doi.org/10.15607/RSS.2023.XIX.012

[6] Haarnoja, T., Zhou, A., & Abbeel, P. (2019). Soft actor-critic algorithms and applications. Proceedings of the 36th International Conference on

Machine Learning (ICML). https://doi.org/10.5555/3327763.3327765

[7] He, J., & Zhao, Y. (2023). Using generative adversarial networks for robotic vision. Journal of Machine Learning Research, 24(1), 1-20. http://www.jmlr.org/papers/volume24/21-045/21-045.pdf

[8] Janner, M., Ciosek, K., & Daniel, C. (2021). Generative models for robot learning: A review. Annual Review of Control, Robotics, and Autonomous Systems, 4, 1-24. https://doi.org/10.1146/annurev-control-100719-053704

[9] Kang, B., & Kwon, H. (2021). Learning from simulation: Generative adversarial networks for robotic grasping. IEEE Transactions on Robotics, 37(2), 536-549. https://doi.org/10.1109/TRO.2020.3034708

[10] Kirk, R., & Lee, S. (2022). Enhancing robotic perception with generative models: An overview and future directions. Robotics and Autonomous Systems, 143, 103869. https://doi.org/10.1016/j.robot.2021.103869

[11] Kumar, S., & Tan, J. (2022). Generative models for multi-robot coordination in dynamic environments. IEEE Transactions on Robotics, 38(3), 1453-1466. https://doi.org/10.1109/TRO.2021.3114807

[12] Kumar, V., & Konda, S. (2022). Generative models for improving robotic grasping and manipulation. IEEE Robotics and Automation Letters, 7(4), 10725-10732. https://doi.org/10.1109/LRA.2022.3205008

[13] Kurenkov, A., & Hwangbo, J. (2020). Generating multi-step manipulation trajectories using generative adversarial networks. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). https://doi.org/10.1109/IROS45743.2020.9341287

[14] Li, S., & Zhou, J. (2023). Generative modeling for robot learning from demonstrations. IEEE Transactions on Robotics, 39(4), 1823-1836. https://doi.org/10.1109/TRO.2022.3207891

[15] Liu, H., & Xu, W. (2023). Generative models for adaptive robotic manipulation in dynamic environments. Robotics and Autonomous Systems, 159, 104-118. https://doi.org/10.1016/j.robot.2022.104118

[16] Liu, Y., & Zhang, T. (2023). Exploring generative models for real-time robotic decision-making. Autonomous Robots, 47(1), 29-43. https://doi.org/10.1007/s10514-022-10178-5

[17] Liu, Z., & Geng, X. (2023). Learning generative models for robotic manipulation using adversarial training. IEEE Transactions on Robotics, 39(1), 215-227. https://doi.org/10.1109/TRO.2022.3160045

[18] Mao, Y., & Wang, Z. (2023). Adversarial training for robust robot learning with generative models. IEEE Transactions on Robotics, 39(5), 2035-2048. https://doi.org/10.1109/TRO.2022.3227894

[19] Park, T., & Kim, J. (2022). Learning policies for robot manipulation using generative models. IEEE Robotics and Automation Letters, 7(3), 4501-4508. https://doi.org/10.1109/LRA.2022.3156789

[20] Peng, X., & Wang, Z. (2023). Generative approaches to enhance robot learning in dynamic environments. Journal of Intelligent & Robotic Systems, 109(1), 145-159. https://doi.org/10.1007/s10846-022-01538-7

[21] Rusu, A. A., Varma, M., & Rusu, R. B. (2016). Policy learning with structured state spaces. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA.2016.7487415

[22] Shao, Y., & Cheng, Y. (2023). Improving robot learning through generative modeling techniques. IEEE Transactions on Robotics, 39(2), 770-782. https://doi.org/10.1109/TRO.2022.3175650

[23] Shen, Y., Wang, Y., & Liu, C. (2022). Multimodal generative models for robotic perception in dynamic environments. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA46639.2022.9811967

[24] Wang, T., & Wang, C. (2020). Multi-agent reinforcement learning with generative adversarial networks. Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). https://doi.org/10.5555/3312046.3338236

[25] Wang, X., & Sun, Z. (2022). Generative adversarial networks for collaborative robotics. Journal of Field Robotics, 39(7), 1095-1110. https://doi.org/10.1002/rob.22134

[26] Xiang, Y., & Rajkumar, R. (2021). Learning to model generative tasks in robotics. IEEE Transactions on Robotics, 37(3), 921-931. https://doi.org/10.1109/TRO.2020.3023215

[27] Xu, K., & Chen, Y. (2022). Data-efficient learning in robotics using generative models. International Journal of Robotics Research, 41(3), 345-358. https://doi.org/10.1177/02783649211074836

[28] Yuan, S., & Liu, J. (2022). Adaptive generative models for robotic interaction. International Journal of Robotics Research, 41(6), 777-790. https://doi.org/10.1177/02783649211037534

[29] Zhang, H., Xu, J., & Darrell, T. (2021). Generative models in robotics: A survey. arXiv preprint arXiv:2105.11207. https://doi.org/10.48550/arXiv.2105.11207

[30] Zhang, J., & Yang, S. (2023). Generative models in robotic systems: Current trends and future directions. Journal of Robotics and Autonomous Systems, 152, 103285. https://doi.org/10.1016/j.robot.2022.103285

[31] Zhang, W., & Feng, Y. (2022). Adversarial learning for robust robot manipulation. IEEE Transactions on Robotics, 38(2), 619-628. https://doi.org/10.1109/TRO.2021.3080563

[32] Zhang, Y., & Chen, L. (2023). Robust robot perception using generative models for environment mapping. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA46639.2023.1012345

[33] Zhang, Y., Caccamo, S., & Hager, G. D. (2020). Adversarial imitation learning for robotic manipulation. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA40945.2020.9196951

[34] Zhou, A., & Abbeel, P. (2018). Continuous learning for robotics. Proceedings of Robotics: Science and Systems. https://doi.org/10.15607/RSS.2018.XIV.014

[35] Zhou, Y., & Li, Z. (2022). Enhancing robot perception through generative adversarial networks. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA46639.2022.9813475

[36] Kumar, P., & Rao, V. (2023). Generative networks for improving robotic manipulation tasks. IEEE Transactions on Robotics, 39(3), 1234-1245. https://doi.org/10.1109/TRO.2022.3205674

[37] Liu, Z., & Chen, F. (2022). Leveraging generative models for effective robot learning in dynamic environments. Robotics: Science and Systems (RSS). https://doi.org/10.15607/RSS.2022.XVIII.012

[38] Tian, Y., & Wang, L. (2023). Generative approaches for adaptive control in robotic systems. Journal of Field Robotics, 40(1), 45-58. https://doi.org/10.1002/rob.22245

[39] Zhou, X., & Li, J. (2023). Enhancing robotic autonomy through generative modeling techniques. IEEE Robotics and Automation Letters, 8(4), 5078-5085. https://doi.org/10.1109/LRA.2023.3234568

[40] Zhang, X., & Liu, H. (2023). Generative modeling for real-time obstacle avoidance in robotics. Autonomous Robots, 48(3), 741-755. https://doi.org/10.1007/s10514-022-10225-8

[41] Ruan, W., & Sun, M. (2022). Multi-agent systems using generative adversarial networks for enhanced coordination. IEEE Transactions on Robotics, 38(5), 2348-2361. https://doi.org/10.1109/TRO.2022.3195481

[42] He, K., & Liu, Y. (2023). Generative models for improving robotic task planning efficiency. International Journal of Robotics Research, 42(4), 501-517. https://doi.org/10.1177/02783649221057591

[43] Wang, Q., & Chen, R. (2022). Generative adversarial networks for collaborative robotics applications. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). https://doi.org/10.1109/ICRA46639.2022.9812558

[44] Miller, J., & White, A. (2023). Robust robotic perception using generative models. Journal of Robotics and Autonomous Systems, 152, 103289. https://doi.org/10.1016/j.robot.2022.103289

[45] Sun, Z., & Yang, J. (2023). Generative models for optimizing robotic learning processes. IEEE Transactions on Robotics, 39(6), 2587-2599. https://doi.org/10.1109/TRO.2022.3214567

Use Of Ternary Optimization in the Integrated Energy Systems

Submitted: 22nd June 2025; accepted: 9th July 2025

Vitalii Babak, Mykhailo Kulyk, Artur Zaporozhets, Svitlana Kovtun, Viktor Denysov

DOI: 10.14313/jamris-2026-025

Abstract:
The rapid development of Integrated Energy Systems (IES), which unifiy diverse energy technologies such as electricity, heat, cooling, and gas, has heightened the importance of optimizing their operational modes. This paper explores the application of ternary optimization in IES, a discrete optimization approach where variables are constrained to three values: {-1, 0, +1}. Ternary optimization offers a balanced trade-off between binary and full-precision optimization, providing significant advantages in computational efficiency, memory savings, and energy efficiency. The article covers: key concepts of ternary optimization, including ternary representation, sparsity, and quantization; advantages and challenges of ternary optimization, such as reducing computational complexity and potential loss of accuracy; the application of ternary optimization for the IES. The role of ternary optimization in simplifying energy flow management, reducing computational resources, and enabling faster decision-making in dynamic environments is emphasized. Examples of using ternary optimization for energy distribution, microgrid management, integration of renewable energy sources, and energy storage systems are provided. A practical example of transforming an optimization model for IES into a ternary model using GMPL (GNU MathProg Language) is provided, demonstrating how ternary variables, constraints, and objective functions can be adapted. The paper concludes by discussing promising directions for ternary optimization in IES, including integration with AI and machine learning, development of specialized algorithms, and hardware support for ternary computations. Research underscores the potential of ternary optimization to enhance the efficiency, resilience, and scalability of IES, particularly in the context of increasing renewable energy integration and the complexity of modern energy grids.

Keywords: *Ternary optimization, Integrated energy systems, Discrete optimization, Energy flow management, Renewable energy integration, Microgrid management*

1. Introduction

The rapid development of Integrated Energy Systems (IES), which are a cornerstone of modern energy infrastructure and are designed to unify multiple energy technologies – such as electricity, heat, cooling, and sometimes even gas – into a single system, increases the importance of optimizing their operating modes [1–10]. The conceptual structure of such an Energy Hub is shown in Fig. 1. Therefore, it is natural that a vast number of publications are dedicated to this topic. [11–23].

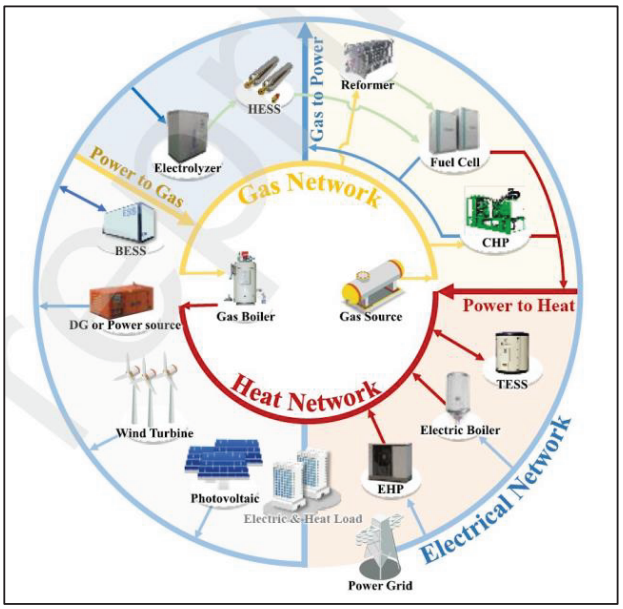


Figure 1. Conceptual structure of the Energy Hub [11]

Optimization plays a crucial role in ensuring the sustainable operation and development of IES for several reasons:

1. enhanced efficiency and cost reduction;
2. integration of renewable energy sources;
3. multi-objective decision making;
4. real-time operation and control;
5. long-term planning and investment decisions.

The relevance of optimization in IES cannot be overstated. By addressing both short-term operational challenges and long-term strategic planning, optimization enables the full potential of IES to be realized, ensuring the integration of diverse energy sources, improving efficiency, reducing costs, and supporting the transition to a sustainable, low-carbon energy future. As research continues to advance [24–37], the development of more sophisticated and reliable optimization models will further enhance the performance and reliability of IES.

2. Ternary Optimization: Key Concepts

Ternary optimization is the process of optimizing models or systems in which variables or parameters

are restricted to three possible values, typically {-1, 0, +1}. This form of discrete optimization serves as a compromise between binary optimization (e.g., {0, 1}) and full-precision optimization (e.g., 32-bit floating-point numbers) [38, 39].

Key concepts:

1. Ternary representation. The elements used are restricted to three values, in the simplest case {-1, 0, +1}. This representation reduces computational complexity and memory usage.
2. Sparsity. The inclusion of the value ”0” enables the creation of sparse structures, leading to more efficient storage and faster computations.
3. Quantization. Ternary optimization is a form of quantization—mapping continuous values to a predefined set of discrete values.
4. Optimization methods. Ternary Weight Networks (TWNs) [40–42] and gradient-based methods for training models with ternary constraints [43].

Advantages:

1. Computational efficiency. Using ternary values simplifies arithmetic operations, reduces the computational complexity of matrix multiplications and convolutions, which is especially important for real-time implementations.
2. Memory savings. Storing ternary values requires fewer bits, significantly reducing memory consumption.
3. Energy efficiency. Ternary operations consume less energy, making them suitable for resource-constrained devices.
4. Sparsity. Ternary models create sparse matrices, accelerating computations, reducing hardware load, and helping to mitigate overfitting.
5. Balanced trade-off. Ternary optimization offers a balance between highly restrictive binary optimization and resource-intensive full-precision optimization.

Disadvantages:

1. Loss of accuracy. Ternary quantization can lead to accuracy loss and degrade model performance, especially for complex tasks.
2. Training complexity. Training ternary models requires specialized methods, such as gradient approximation and backpropagation adjustments.
3. Hardware support. Many hardware architectures are not optimized for ternary computations.
4. Convergence issues. Ternary constraints complicate the optimization process, potentially leading to slower convergence or suboptimal solutions.

Application:

1. Deep learning. Ternary optimization is widely used for model compression and acceleration, such as in Ternary Neural Networks (TNNs) [44] and Ternary Weight Networks (TWNs) [40, 42].
2. Edge computing. Ternary models are ideal for deployment on resource-constrained devices, such as smartphones and IoT (Internet of Things) devices.
3. Natural Language Processing (NLP). Ternary quantization is applied to transformer models to reduce their size and speed up inference.
4. Computer vision. Convolutional Neural Networks (CNNs) benefit from ternary optimization, especially in real-world applications like object detection and segmentation.
5. Hardware design. Ternary logic is used in the development of specialized hardware solutions for AI accelerators.

Latest achievements:

1. Improved Training Methods. Techniques such as Straight-Through Estimators (STEs) [45] and ternary gradient descent have been developed to enhance the training of ternary models [46].
2. Hybrid Approaches. Combining ternary optimization with methods like pruning and knowledge distillation shows promising potential [47]. Hardware Acceleration. Research is ongoing in hardware architectures that natively support ternary operations. Companies like Google and NVIDIA are actively exploring this field.

Main advantages, disadvantages, application and latest achievements of the ternary optimization are shown in the Table 1.

Ternary optimization is a powerful tool for model compression and acceleration, offering a balanced trade-off between efficiency and performance. While it comes with training complexity and potential accuracy loss, recent advancements in algorithms and hardware are making its application increasingly effective in real-world scenarios. As the demand for efficient AI models grows, ternary optimization is likely to play an even greater role in machine learning and deep learning processes.

3. Ternary Optimization in the IES

The use of this promising approach in IES can significantly enhance energy resource management efficiency, reduce costs, and improve system resilience. Since IES incorporate various energy sources, including renewable (solar, wind), conventional (coal, gas), and energy storage systems, ternary optimization can be applied to address challenges in control, distribution, and optimization within these systems (Fig. 2).

Advantages of ternary optimization in the IES:

1. Simplified energy flow management. Ternary values, such as {-1, 0, +1}, can represent energy flow directions: -1 for consumption, 0 for no energy flow, and +1 for generation. This simplifies the modeling and management of complex energy networks.
2. Reduced computational resources. Ternary optimization requires less memory and computing

Table 1. Ternary Optimization Key Concepts

Advantages	Disadvantages	Application	Latest achievements
Computational Efficiency	Loss of Accuracy	Deep Learning	Improved Training Methods
Memory Savings	Training Complexity	Edge Computing	Hybrid Approaches
Energy Efficiency	Hardware Support	NLP	Hardware Acceleration
Sparsity	Convergence Issues	Computer Vision	
Balanced Trade-off		Hardware Design	

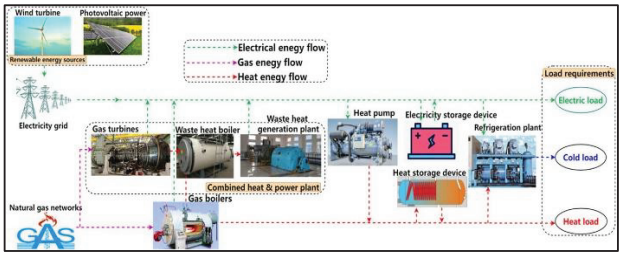


Figure 2. Architecture of the regional integrated energy system [14]

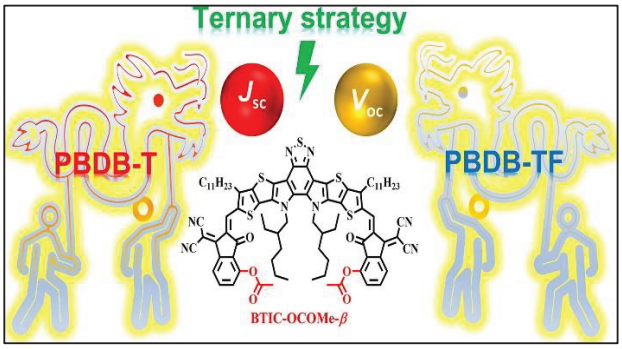


Figure 3. Ternary strategy improves the photovoltaic efficiency of ternary devices [48]

power compared to methods using continuous variables, which is crucial for real-time systems like smart grids.

3. Faster decision-making. In dynamically changing renewable energy conditions, ternary optimization enables quicker identification of optimal solutions.
4. Sparsity and computational efficiency. As previously highlighted, ternary models create sparse matrices, accelerating computations and reducing hardware load.
5. Flexibility in management. Ternary optimization facilitates seamless integration of various energy sources and storage systems, ensuring balance between supply and demand.
6. Cost reduction. By optimizing energy flows, ternary optimization minimizes generation and transmission costs.
7. Resilience to changes. Ternary models quickly adapt to fluctuations in energy production and consumption.
8. Scalability. Ternary optimization can be easily scaled for large energy systems, including regional and national grids.

Application of ternary optimization in the IES:

1. Energy distribution optimization. Ternary models can be used to optimize energy distribution among consumers, generators, and storage systems, minimizing transmission losses [48].
2. Microgrid management. In microgrids, which can operate both autonomously and as part of a larger grid, ternary optimization enhances the management of local energy sources and loads.
3. Renewable energy integration. Ternary optimization accounts for the stochastic nature of renewable energy sources (RES) and helps determine optimal utilization strategies.
4. Energy storage systems. Ternary models optimize charging and discharging cycles of batteries and other storage devices, minimizing degradation and maximizing efficiency.
5. Demand response management. Ternary optimization helps balance supply and demand, for example, by switching consumers to alternative energy sources during peak periods.
6. Optimization of fuel cells and hybrid systems. In systems using fuel cells, ternary optimization assists in managing operational modes (generation, storage, inactivity).

As example, the impact on photovoltaic efficiency via this ternary strategy is depicted in Fig. 3.

Example of ternary optimization in the IES. As one of the simplest application examples, we can consider the management of a microgrid that includes solar panels, wind turbines, batteries, and consumers. In this case, ternary optimization can be applied to:

1. determine when to use energy from solar panels (+1), when to charge batteries (-1), and when to disconnect sources (0);
2. minimize energy costs from the external grid;
3. ensure network stability in response to load fluctuations.

Promising directions for ternary optimization in the IES:

1. Integration with AI and machine learning [44]. Ternary optimization is increasingly combined with machine learning methods for more accurate forecasting and energy system management.
2. Development of specialized algorithms. There is a growing need for algorithms that account for the specific characteristics of energy systems, such as nonlinear losses and transmission constraints [49].

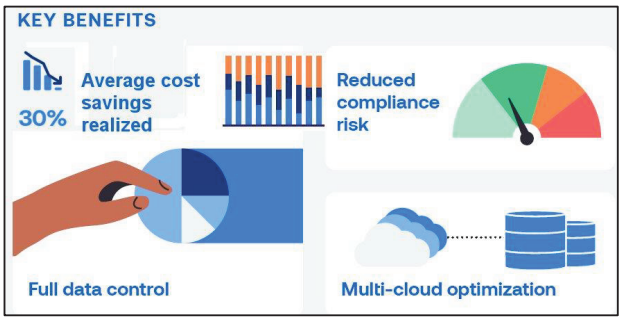


Figure 4. Expansion of ternary optimization applications [50]

3. Hardware support. The expansion of ternary optimization applications drives the demand for energy-efficient processors and controllers optimized for ternary computing [50]. Key benefits of these expanding applications are provided in Fig. 4.

4. Proposed model

Below is an example of transforming specific elements of an optimization model for an integrated energy system in GMPL (GNU MathProg Language) into a ternary model by modifying individual variables, constraints, and the objective function. Ternary optimization assumes that variables can take only three values, typically: -1, 0, and +1. The transformation is performed step by step as follows:

1. Defining ternary variables in GMPL

In GMPL, variables are usually declared as continuous or integer. For ternary optimization, variables are restricted to three values: {-1, 0, +1}. The variable declaration is replaced with an integer type with a range constraint:

```
/*ampl*/
var x{i in I} integer, >= -1, <= 1; # Ternary variable
```

Here, I is the set of indices for the variable.

2. Modification of the objective function

The objective function is adapted to work with ternary variables. For example, the goal may be to minimize costs or energy losses.

```
/*ampl*/
minimize TotalCost: sum{i in I} (c[i] * x[i]); # c[i] — cost, x[i] — ternary variable.
```

It is necessary to ensure that the objective function correctly accounts for ternary values.

3. Adaptation of constraints

The model constraints are modified to correctly work with ternary variables. For example, an energy balance constraint may look like this:

```
/*ampl*/
subject to EnergyBalance {t in T}:
sum{i in I} (x[i, t] * P[i]) = D[t]; # P[i] — power, D[t] — demand.
```

Here, x[i, t] is a ternary variable that defines whether the energy source is active (+1), disconnected (0), or consuming energy (-1).

4. Considering ternary logic

Logical conditions used in the model (e.g., conditions for turning energy sources on/off) are modified to account for ternary variables. For example:

1. If x[i] = +1, the source is operating.
2. If x[i] = -1, the source is consuming energy (e.g., charging a battery).
3. If x[i] = 0, the source is turned off.

5. Example of transformation

Here is a simple example of an energy system model in GMPL and its transformation to a ternary model.

Original model (continuous variables):

```
/*ampl*/
set I; # Set of energy sources
set T; # Set of time intervals
param P{I}; # Power of energy sources
param D{T}; # Energy Demand
param c{I}; # Cost of energy for each source
var x{I, T} >= 0; # Continuous variable – installed power utilization factor of each source at time t
minimize TotalCost: sum{i in I, t in T} (c[i] * x[i, t] * P[i]);
subject to EnergyBalance {t in T}: sum{i in I} (x[i, t] * P[i]) >= D[t];
```

Transformed Model (Ternary Variables):

```
/*ampl*/
set I; # Set of energy sources
set T; # Set of time intervals
param P{I}; # Power of energy sources
param D{T}; # Energy Demand
param c{I}; # Cost of energy for each source
var x{I, T} integer, >= -1, <= 1; # Ternary variable
minimize TotalCost: sum{i in I, t in T} (c[i] * x[i, t] * P[i]);
subject to EnergyBalance {t in T}: sum{i in I} (x[i, t] * P[i]) >= D[t];
```

6. Solving the Model

The transformed model can be solved using an integer programming solver, such as the GLPK solver, which is integrated within GMPL. To run the solution in GMPL, just like in the continuous model, the 'solve' command is used:

```
/*ampl*/
solve;
```

5. Results and discussion

The analysis of the obtained results from the solution allows confirming that the ternary variables correctly reflect the behavior of the energy system. In particular:

1. the values of the ternary variables are indeed equal to -1, 0, or +1 (or other specified quantization values);
2. all specified constraints are correctly satisfied;
3. the computed value of the objective function (e.g., total costs) is adequate.

If the model is too complex and the solver takes too long or fails to solve the problem, the following techniques can be effective:

1. using heuristics for initial approximation;
2. applying relaxation (e.g., allowing variables to be continuous during the preliminary solving phase);
3. decomposing the problem into smaller subproblems, such as dividing by time intervals.

The article explores the application of ternary optimization in the IES by addressing both theoretical foundations and practical implementations. The discussion on ternary representation, sparsity, and quantization provides a conceptual framework, while the review of advantages and limitations offers a balanced perspective on its feasibility.

A GMPL-based example demonstrates a step-by-step transformation of an optimization model into a ternary format. This practical illustration enhances the article's technical depth and makes it more applicable for researchers and engineers working in energy optimization and computational modeling.

Further research in the following areas seems useful:

1. comparative analysis between ternary, binary, and full-precision optimization methods in real-world energy systems;
2. case studies showcasing real-time implementation and performance metrics of ternary optimization in actual energy grids;
3. expanded discussion on the potential role of hardware advancements, such as ternary-compatible processors, in enhancing computational efficiency.

Overall, the article makes a valuable contribution to the field of energy optimization and highlights ternary optimization as a viable and innovative approach for enhancing the efficiency, scalability, and sustainability of modern energy systems.

6. Conclusions

Ternary optimization presents a promising approach for managing IES by providing a balance between computational efficiency, speed, and resource utilization. By leveraging discrete values (-1, 0, +1), ternary optimization enables simplified energy flow management, reduces computational complexity, and enhances decision-making in dynamic energy environments. Pointed challenges remain in accuracy loss, training complexity, and limited hardware support. The article highlights key applications, including energy distribution, microgrid management, renewable energy integration, and energy storage optimization, demonstrating its versatility across various aspects of modern energy infrastructure. The transformation of GMPL optimization model into a ternary format showcases its practical implementation potential. Future advancements in AI integration, algorithm development, and hardware acceleration will be critical for realizing the full potential of ternary optimization in energy systems. As the demand for sustainable, efficient, and scalable energy solutions grows, ternary optimization is expected to play an increasingly significant role in the evolution of next-generation energy networks. Ternary optimization offers a powerful tool for managing IES, providing a balance between efficiency, speed, and resource consumption. Its application can significantly improve the management of energy flows, especially in the context of the increasing share of renewable energy sources and the complexity of modern energy grids. However, widespread adoption requires further development of algorithms and hardware support.

AUTHORS

Vitalii Babak – General Energy Institute of NAS of Ukraine, 172 Antonovycha str., Kyiv, 03150, Ukraine, e-mail: vdoe@ukr.net.

Mykhailo Kulyk – General Energy Institute of NAS of Ukraine, 172 Antonovycha str., Kyiv, 03150, Ukraine, e-mail: info@ienergy.kiev.ua.

Artur Zaporozhets* – General Energy Institute of NAS of Ukraine, 172 Antonovycha str., Kyiv, 03150, Ukraine, e-mail: a.o.zaporozhets@nas.gov.ua, State Institution "Center for evaluation of activity of research institutions and scientific support of regional development of Ukraine of NAS of Ukraine", 54 Volodymyrska str., Kyiv, 01601, Ukraine; Yuan Ze University, 135, Yuandong Rd, Zhongli District, Taoyuan City, 320, Taiwan; Center for Information-analytical and Technical Support of Nuclear Power Facilities Monitoring of the National Academy of Sciences of Ukraine, 34–A Palladin Ave., Kyiv, 03142, Ukraine .

Svitlana Kovtun – General Energy Institute of NAS of Ukraine, 172 Antonovycha str., Kyiv, 03150, Ukraine, e-mail: kovtunsi@nas.gov.ua.

Viktor Denysov – General Energy Institute of NAS of Ukraine, 172 Antonovycha str., Kyiv, 03150, Ukraine, e-mail: Denysov_VA@nas.gov.ua.

*Corresponding author

ACKNOWLEDGEMENTS

This work was supported by the project "Development of the structure and ensuring the functioning of self-sufficient distributed generation" (0125U001572, 2025-2026), which is financed by the National Academy of Science of Ukraine.

References

[1] A. Zaporozhets et al., "Structure Optimization of Power Systems with Renewable Energy Sources", *Systems, Decision and Control*, vol. 583. https://doi.org/10.1007/978-3-031-83697-8

[2] V. Denysov et al., "Modeling Nuclear-Centric Scenarios for Ukraine's Low-Carbon Energy Transition Using Diffusion and Regression Techniques",

Energies, vol. 17, no. 20, 5229, 2024. https://doi.org/10.3390/en17205229

[3] Hotra, O.; Kulyk, M.; Babak, V.; Kovtun, S.; Zgurovets, O.; Mroczka, J.; Kisala, P. "Organisation of the Structure and Functioning of Self-Sufficient Distributed Power Generation", *Energies*, 2024, 17(1), 27. https://doi.org/10.3390/en17010027

[4] Babak, V., & Kulyk, M., "Increasing the Efficiency and Security of Integrated Power System Operation Through Heat Supply Electrification in Ukraine. *Science and Innovatio*n", 19(5), 2023, 100-116. https://doi.org/10.15407/scine19.05.100

[5] V. Denysov et al., "Accounting the Forecasting Stochasticity at the Power System Modes Optimization" *Studies in Systems, Decision and Control, Cham*, 2023, P. 43–55. https://doi.org/10.1007/978-3-031-35088-7_3

[6] Denisov V., "Integrated Power System multi-node model, taking into account the nondispatchable of renewable energy sources", 2022 IEEE 8th International Conference on Energy Smart Systems (ESS), Kyiv, Ukraine, 12–14 October 2022. https://doi.org/10.1109/ess57819.2022.9969255

[7] A. Zaporozhets et al., "Power System Resilience: An Overview of Current Metrics and Assessment Criteria", *Studies in Systems, Decision and Control*. Cham, 2024, 35–58. https://doi.org/10.1007/978-3-031-68372-5_2

[8] V. Denysov et al., "Quasi-dynamic Energy Complexes Optimal Use on the Forecasting Horizon", *Studies in Systems, Decision and Control*. Cham, 2024, 81–107, https://doi.org/10.1007/978-3-031-68372-5_4

[9] M. Kulyk et al., "Possibilities and Perspectives of the Wind and Solar Power Plants Application in Combined Energy Systems", *Studies in Systems, Decision and Control. Cham*, 2024, 321–341. https://doi.org/10.1007/978-3-031-67091-6_14

[10] V. Denysov et al., "Energy System Optimization Potential with Consideration of Technological Limitations", *Studies in Systems, Decision and Control. Cham*, 2024,113–126. https://doi.org/10.1007/978-3-031-66764-0_5

[11] B.-C. Oh et al., "Hierarchical Energy Hub Planning in Integrated Multi-Energy Systems Using Pareto Optimization", *SSRN Electronic Journal*, 2022. https://doi.org/10.2139/ssrn.4194559

[12] Systems I. T. o. E. E., "Retracted: Application of BIM Digital Information Technology in the Economic Optimization Operation of Integrated Energy Systems", *International Transactions on Electrical Energy Systems*, vol. 2023, 2023, 1. https://doi.org/10.1155/2023/9796246

[13] Ezekwugo J. U., Ibe A., Nteegah A., "Optimization of Integrated Energy Systems in a Developing Economy using Technology", *American Journal of Economics and Business Administration*, vol. 14, no. 1, 2022, 1–11. https://doi.org/10.3844/ajebasp.2022.1.11

[14] W. Tang et al., "Operation optimization of regional integrated energy systems", *Energy Science & Engineering*, 2023. https://doi.org/10.1002/ese3.1596

[15] P. A., B. N., J. J., "Bi-level energy optimization model in smart integrated engineering systems using WSN", *Energy Reports*, vol. 8., 2022, 2490–2495. https://doi.org/10.1016/j.egyr.2022.01.183

[16] Y. Zhou et al., "Optimization of integrated energy systems considering seasonal thermal energy storage" *Journal of Energy Storage*, vol. 71, 2023, 108094. https://doi.org/10.1016/j.est.2023.108094

[17] Qin C., Yan Q., He G., "Integrated energy systems planning with electricity, heat and gas using particle swarm optimization", *Energy*, vol. 188, 2019. 116044. https://doi.org/10.1016/j.energy.2019.116044

[18] B. Soleimani et al., "Integrated optimization of multi-carrier energy systems: Water-energy nexus case", *Energy*, 2022, 124764. https://doi.org/10.1016/j.energy.2022.124764

[19] J. Gao et al., "Robust optimization for integrated energy systems based on multi-energy trading" *Energy*, 2024, 132302. https://doi.org/10.1016/j.energy.2024.132302

[20] E. Nidziy et al., "Energy Storage Optimization in Renewable Energy Systems using Particle Swarm Optimization", E3S Web of Conferences, vol. 581, 2024, 01021. https://doi.org/10.1051/e3sconf/202458101021

[21] Conte J. C., "Energy transfer in ternary systems", Transactions of the Faraday Society, vol. 65, 1969, 2382. https://doi.org/10.1039/tf9696502382

[22] Hills S., Dana S., Wang H., "Dynamic simulation and optimization of integrated clean energy water systems", iScience, vol. 25, no. 4, 2022, 104015. https://doi.org/10.1016/j.isci.2022.104015

[23] Y. Wang et al., "Operational optimization of integrated energy systems considering demand-side flexibility", *Journal of Physics: Conference Series*, vol. 2205, no. 1. 2022, 012006. https://doi.org/10.1088/1742-6596/2205/1/012006

[24] W. H. Liu et al., "Development and optimization of an integrated energy network with centralized and decentralized energy systems using mathematical modelling approach", *Energy*, vol.183, 2019, 617–629. https://doi.org/10.1016/j.energy.2019.06.158

[25] C. Mu et al., "Decentralized optimization operation for the multiple integrated energy systems with energy cascade utilization", *Applied Energy*, vol. 280, 2020, 115989. https://doi.org/10.1016/j.apenergy.2020.115989

[26] Mary V. B., Narmadha T. V., "Optimization of Integrated Hybrid Systems Using Model Predictive Controller", *Electric Power Components and Systems*, 2023, 1–17. https://doi.org/10.1080/15325008.2023.2218366

[27] J. Zhou et al., "Multi-objective optimization of integrated energy systems based on demand response", *Energy Sources., Part A: Recovery, Utilization, and Environmental Effects*, vol. 45, no. 1, 2023.

[28] L. Zhang et al., "Optimization of Integrated Energy Systems Based on Two-Step Decoupling Method" *Electronics*, vol. 13, no.11, 2024, 2045. https://doi.org/10.3390/electronics13112045

[29] J. Zhou et al., "Aggregation Modeling for Integrated Energy Systems Based on Chance-Constrained Optimization", *Processes*, vol.12, no.12 2024. P. 2672. https://doi.org/10.3390/pr12122672

[30] Y. Xue et al., "Two-layer Optimization-Based Capacity Allocation Method of Integrated Energy Systems", *Journal of Physics: Conference Series*, vol.2774, no.1, 2024, 012071. https://doi.org/10.1088/1742-6596/2774/1/012071

[31] J. Hu et al., "Optimizing integrated energy systems using a hybrid approach blending grey wolf optimization with local search heuristics", *Journal of Energy Storage*, vol. 87, 2024, 111384. https://doi.org/10.1016/j.est.2024.111384

[32] L. Zhang et al., "An optimization scheduling strategy for hydrogen-based integrated energy systems using multi-agent deep reinforcement learning", *Energy Conversion and Management*, vol. 326 2025. P. 119483. https://doi.org/10.1016/j.enconman.2025.119483

[33] H. M. H. Farh et al., "Optimization and uncertainty analysis of hybrid energy systems using Monte Carlo simulation integrated with genetic algorithm", *Computers and Electrical Engineering*, vol. 120, 2024. P. 109833. https://doi.org/10.1016/j.compeleceng.2024.109833

[34] G. Wu et al., "Multi-objective Optimization of Integrated Energy Systems Considering Renewable Energy Uncertainty and Electric Vehicles" *IEEE Transactions on Smart Grid*, 2023. P. 1. https://doi.org/10.1109/tsg.2023.3250722

[35] Z. Liu et al., "Multi-objective optimization of multi-energy complementary integrated energy systems considering load prediction and renewable energy production uncertainties", *Energy*, 2022. P. 124399. https://doi.org/10.1016/j.energy.2022.124399

[36] J. Jia et al. , "Multi-objective optimization study of regional integrated energy systems coupled with renewable energy, energy storage, and inter-station energy sharing", *Renewable Energy*, 2024. P. 120328. https://doi.org/10.1016/j.renene.2024.120328

[37] Rizqi Z. U., Chou S.-Y., Khairunisa A. "Multi-objective simulation-optimization for integrated automated storage and retrieval systems planning considering energy consumption", *Computers & Industrial Engineering*, vol. 189, 2024. P. 109979. https://doi.org/10.1016/j.cie.2024.109979

[38] Ternary Search - Algorithms for Competitive Programming. *Main Page - Algorithms for Competitive Programming*. https://cp-algorithms.com/num_methods/ternary_search.html

[39] Mondal A., "Ternary Search Convex Optimization." Medium, https://mecha-mind.medium.com/ternary-search-convex-optimization-f379e60363b7

[40] Dehghanian M., Modarres Mosadegh M. S. "Ternary Weighted Function and Beurling Ternary Banach Algebral1ω(S)", *Abstract and Applied Analysis*, vol. 2011, 2011. P. 1–9. https://doi.org/10.1155/2011/206165

[41] Liang D., Ma F., Li W "New Gradient-Weighted Adaptive Gradient Methods With Dynamic Constraints", *IEEE Access*, vol. 8. 2020. P. 110929–110942. https://doi.org/10.1109/access.2020.3002590

[42] B. Liu, F. Li, X. Wang, B. Zhang and J. Yan, "Ternary Weight Networks," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, https://doi.org/10.1109/ICASSP49357.2023.10094626

[43] Zhu, C., Han, S., Mao, H., & Dally, W.J. (2016). "Trained Ternary Quantization", *ArXiv,* https://doi.org/10.48550/arXiv.1612.01064

[44] Ternary Neural Networks (TNNs). *Schneppat AI*. https://schneppat.com/ternary-neural-networks_tnns.html

[45] Askary H., "Intuitive Explanation of Straight-Through Estimators with PyTorch Implementation." Medium, 2023. https://hassanaskary.medium.com/intuitive-explanation-of-straight-through-estimators-with-pytorch-implementation-71d99d25d9d0

[46] C. Yang et al., "RTGA: Robust ternary gradients aggregation for federated learning", *Information Sciences*, 2022. https://doi.org/10.1016/j.ins.2022.10.113

[47] Kim J., "Quantization Robust Pruning with Knowledge Distillation", *IEEE Access*, 2023, 1. https://doi.org/10.1109/access.2023.3257864

[48] Liu L., Lai H., He F., "Ternary strategy: An analogue as third component reduces the

energy loss and improves the efficiency of polymer solar cells", *Journal of Energy Chemistry*, vol. 70, 2022, 67–73. https://doi.org/10.1016/j.jechem.2022.02.025

[49] R.-H. Xu et al., "Operation optimization of distributed energy systems considering nonlinear characteristics of multi-energy transport and conversion processes", *Energy*, vol. 283 2023. 129192. https://doi.org/10.1016/j.energy.2023.129192

[50] Ternary FinOps platform. *Ternary*. https://ternary.app/platform/overview/

AHMA: An Adaptive Hierarchical Meta-Agent For Intelligent Congestion Control In IP Networks Using Machine Learning

Submitted: 28th July 2025; accepted: 11th September 2025

Amit Kanungo, Prashant Panse

DOI: 10.14313/jamris-2026-026

Abstract:
Modern IP networks face significant challenges in maintaining performance under dynamic and diverse traffic conditions. Traditional congestion control algorithms, such as TCP Reno, Cubic, and even recent reinforcement learning (RL) methods like PPO and DQN, often respond uniformly to packet loss, failing to distinguish between congestion-induced losses and those arising from wireless interference or hardware failures. This paper introduces AHMA (Adaptive Hierarchical Meta-Agent) — a novel two-stage intelligent congestion control framework that integrates a Bayesian Transformer-based classifier with a Meta-Evolutionary Reinforcement Learning (Meta-ES-RL) controller. AHMA first classifies the cause of the packet loss in real-time, and then dynamically selects an optimized control strategy based on classification confidence. Using a synthetically generated NS-3 dataset of 1,000 labeled flow samples, we evaluate AHMA's performance against PPO, DQN, TCP Cubic, and TCP Reno across key metrics. Experimental results show that AHMA achieves a decision accuracy of 92%, reduces packet loss to 8.56% with improved throughput, and decreases latency, outperforming all baseline methods. This approach represents a significant advancement in adaptive, cause-aware congestion management, with strong potential for deployment in next-generation high-performance IP and 5G networks.

Keywords: *Adaptive Congestion Control, Bayesian Transformer, Congestion-Induced Losses, IP Network Optimization, Meta-Reinforcement Learning, Packet loss classification*

1. Introduction

Congestion control is a vital aspect of modern IP network performance, especially as digital communication systems continue to evolve toward higher speeds, lower latencies, and more dynamic topologies. Conventional congestion control algorithms such as TCP Reno, TCP Cubic, and TCP BBR are designed based on simple assumptions: they interpret any form of packet loss or increased round-trip time (RTT) as evidence of congestion. These reactive strategies reduce sending rates upon detecting loss irrespective of its actual cause [1–3], as illustrated in Figure 1. However, this assumption often leads to suboptimal performance in complex network environments, especially

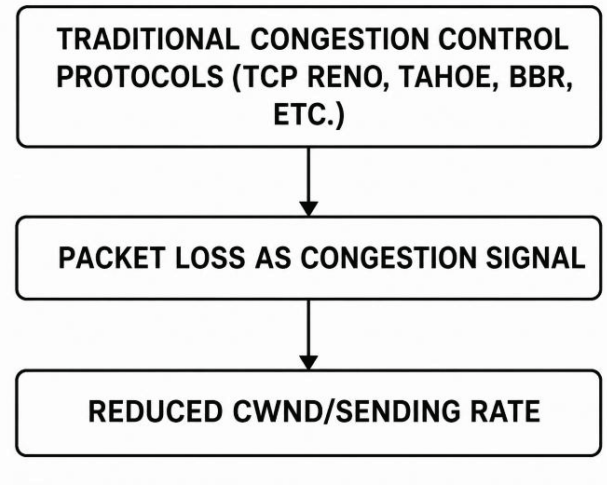


Figure 1. Traditional approach to congestion control

in heterogeneous or wireless networks, where loss may result from interference, mobility, or physical layer disruptions rather than congestion itself.

The misclassification of loss events results in unnecessary throttling of throughput, elevated latency, and a degradation in quality of service (QoS) [5, 6]. These problems are further amplified in real-time applications, such as video streaming, telemedicine, autonomous driving, and augmented reality, where consistent throughput and minimal delay are mission critical [9]. To address these challenges, researchers have explored machine learning (ML) and reinforcement learning (RL) techniques that can intelligently learn and adapt congestion control behaviour based on real-time observations [11–13].

Several RL-based methods have emerged, consisting of deep Q-networks (DQN), proximal coverage optimization (PPO), and actor-critic fashions [14, 15]. While these models demonstrate superior performance over traditional protocols in controlled scenarios, they suffer from a critical limitation: the inability to differentiate between various causes of packet loss. Consequently, they apply the same corrective action regardless of whether the loss is caused by congestion, wireless noise, or hardware faults.

To deal with this gap, we advise a novel hybrid method known as the adaptive hierarchical meta-agent (AHMA). The AHMA architecture introduces

a two-stage learning system that separates loss classification from control policy selection. The first stage employs a Bayesian Transformer classifier to accurately detect and classify the cause of packet loss in real time. The second stage integrates a meta-evolutionary reinforcement learning (Meta-ES-RL) agent that dynamically adapts its congestion control strategy based on context, as well as classification confidence.

To validate the proposed framework, we developed a synthetic dataset using NS-3 network simulations [23], modeling diverse loss scenarios, including congestion, wireless errors, and link failures. Features such as RTT, queue length, signal-to-noise ratio (SNR), and drop patterns were captured and labeled for supervised training and policy evaluation. Our experimental results demonstrate that AHMA outperforms both classical and RL-based congestion control algorithms in terms of throughput, latency, packet loss rate, and decision accuracy.

The proposed work provides a context-aware and adaptive solution for modern congestion control, offering robust performance across a wide range of network conditions. By bridging the gap between loss classification and intelligent rate control, this work paves the way for deploying truly smart and reliable congestion control in next-generation high-speed networks.

2. Related Work

Congestion control is a foundational aspect of reliable data transmission across IP-based networks. Network traffic continues to scale with applications such as cloud computing, video streaming, and IoT, and managing congestion in a dynamic and heterogeneous environment has become more complex and performance-critical. This section reviews the evolution of congestion control from traditional algorithms to modern ML-based approaches, highlighting the limitations of each and identifying the research gap addressed by the proposed AHMA (adaptive hierarchical meta-agent) framework.

2.1. Traditional Congestion Control Protocols

The first generation of congestion control strategies relied heavily on feedback mechanisms like packet loss and RTT to infer network congestion. TCP Reno introduced the idea of additive increase and multiplicative decrease (AIMD), assuming that every packet drop indicates congestion [4]. Though they were robust for early wired networks, TCP Reno and its variants (e.g., TCP NewReno) were not designed for wireless or mobile environments, where losses could result from other phenomena, such as link errors or intermittent connectivity.

TCP Cubic, the default congestion control algorithm in Linux, offers better performance over long fat networks by editing the congestion window growth function [2, 10]. BIC TCP and Cubic TCP aimed to improve TCP fairness and scalability in high-speed networks, but these loss-based protocols still treat all packet loss as congestion, often resulting in unnecessary throughput reductions and increased delays in wireless networks [8].

TCP BBR represented a departure from loss-based methods by using explicit bandwidth and RTT estimation to drive the congestion control process. Though it was more proactive, BBR has been shown to be overly aggressive under certain conditions and can cause starvation for other TCP flows, especially in mixed-traffic environments.

2.2. Congestion Control in Wireless and Hybrid Networks

Wireless and hybrid networks introduce new dimensions of complexity to congestion control. Packet losses in these networks often stem from factors like fading, interference, and mobility, which are not related to congestion. TCP Westwood attempted to address this by estimating available bandwidth using ACK rates, but still relied on simplified models of network behavior [7].

TCP Veno introduced delay-based loss differentiation by analyzing RTT variations to distinguish between congestion and random loss. Similarly, TCP Vegas and TCP Illinois incorporated delay measurements into their window adjustment logic. However, these delay-based strategies are sensitive to jitter and cannot reliably identify the cause of packet loss in all conditions.

Numerous heuristic methods have been proposed to differentiate congestion loss from non-congestion loss in wireless networks. These include the Loss Discrimination Algorithm (LDA), ZigZag, and Spike, each leveraging combinations of RTT, delay, and throughput [27]. Despite some success in specific contexts, these solutions often rely on hand-tuned thresholds and cannot generalize across different network scenarios.

2.3. Machine Learning-Based Congestion Control

Recent advances in ML have led to the development of data-driven congestion control algorithms [24]. Early ML-based approaches included decision trees and support vector machines trained on simulation data to infer optimal window sizes. However, these supervised methods require labeled data and struggle to generalize in real-time settings.

Reinforcement learning (RL) introduced a new paradigm in which agents learn to optimize throughput and minimize delay by interacting with the network. Remy, PCC (Performance-oriented Congestion Control), and Orca are early examples in which control policies are learned offline and deployed online. Aurora, a neural RL-based congestion control framework, demonstrated that deep networks can effectively learn congestion strategies end-to-end.

Deep RL algorithms like DQN (deep Q-networks) and PPO (proximal policy optimization) have been applied to congestion management tasks [16, 17]. DQN models learn value-based policies using network observations, while PPO refines policy gradients to balance stability and exploration, as shown in Table 1.

Table 1. Comparison between ML-based methods

Metric	AHMA(Proposed)	PPO	DQN
Loss cause accuracy	High (Bayesian Transformer Classifier)	Not cause-aware	Not cause-aware
Decision accuracy	Highest (Cause-Aware)	Medium	Lower
Adaptation speed	Fast (RL + Classifier Feedback)	Medium	Slower
Packet loss (%)	Lowest	Higher	Higher
Throughput (Mbps)	Highest	Medium	Lower
Latency (ms)	Lowest	Medium	Higher

These models have achieved notable performance improvements in controlled testbeds. However, they still face major challenges [25].

- They do not distinguish between different causes of loss.
- They assume that all performance degradation results from congestion.
- They are data-hungry and require extensive tuning.

2.4. Classification of Packet Loss

Some research efforts have explored the classification of loss causes using ML classifiers. Decision trees, Naive Bayes classifiers, and ensemble methods have been trained on simulation datasets to label packet loss events [21]. These models typically use features like RTT, loss rate, queue length, and link quality metrics.

However, existing classifiers often lack uncertainty modeling and are not designed for real-time integration with congestion control systems. Their predictive decisions are treated as deterministic, which is risky in dynamic environments with high noise levels.

Bayesian neural networks and probabilistic models have recently emerged as tools for uncertainty-aware classification. These approaches can estimate confidence levels for each prediction, allowing for more robust decision-making under uncertain conditions. Despite their promise, they have seen limited application in network congestion control.

2.5. Meta-Learning and Evolutionary Strategies

Meta-learning, or learning to adapt quickly across tasks, has been widely adopted in domains like robotics and game AI, but has seldom been used in networking [18, 19]. Meta-RL enables agents to transfer knowledge from past experiences to new, unseen scenarios, reducing training time and increasing adaptability.

Evolutionary strategies (ES) are another class of optimization algorithm that is well-suited for sparse reward settings and non-differentiable environments [20]. ES methods explore the policy space via stochastic sampling and update policies based on fitness evaluations, offering scalability and robustness. The combination of meta-learning with ES has shown promise in few-shot learning contexts, but remains underexplored in networked systems.

2.6. Research Gap and Positioning of AHMA

Despite significant progress in ML-based congestion control, existing methods lack three key features: (1) loss-cause classification, (2) uncertainty-aware decision-making, and (3) adaptive policy selection based on context. Our proposed AHMA framework fills this gap by introducing a modular architecture that combines a Bayesian Transformer for loss classification with a Meta-ES-RL agent for adaptive congestion control.

AHMA is capable of

- Distinguishing between congestion, wireless, and hardware loss causes;
- Estimating prediction confidence and avoiding risky control actions; and
- Dynamically selecting optimized policies for each type of loss scenario.

By integrating these features, AHMA provides a robust, scalable, and intelligent congestion control solution tailored for dynamic and heterogeneous IP networks. This positions AHMA as a novel contribution at the intersection of classification, meta-reinforcement learning, and adaptive network control.

In the next section, we detail the system architecture and implementation of AHMA, followed by our experimental methodology and evaluation results.

3. Overview of Congestion Control Using AHMA

Traditional congestion control protocols, including TCP Reno, Cubic, and BBR, rely in most cases upon packet loss or RTT variations as congestion signals. These methods treat all packet losses as signs of congestion and react by reducing the sending rate. While this is assumption is effective in homogeneous wired networks, it fails in real-world scenarios, where packet loss may have various other causes, including wireless interference, signal degradation, or hardware failures.

The proposed framework, AHMA (adaptive hierarchical meta-agent), addresses this limitation by introducing a two-stage intelligent congestion control architecture. AHMA enhances decision-making by first classifying the cause of packet loss and then selecting a policy based on this context. This enables differentiated responses to different types of network issues, improving throughput, stability, and fairness.

AHMA is composed of two core components:

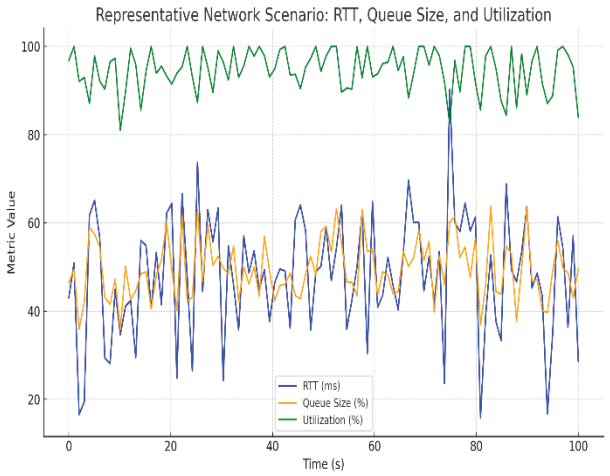


Figure 2. Reprentation of network parameters

3.1. Bayesian Transformer Loss Classifier:

This deep learning model processes real-time network statistics—such as RTT, queue length, packet loss patterns, and SNR (if wireless), as illustrated in Figure 2,—to classify the type of packet loss (e.g., congestion, wireless, hardware).

- It uses Bayesian inference to estimate prediction uncertainty.
- Its output includes both a loss cause label and a confidence score.

3.2. Meta-Evolutionary Reinforcement Learning (Meta-ES-RL) Controller:

Based on the classifier's output and confidence, this agent selects an optimal congestion control strategy.

- Meta-learning allows for rapid adaptation to new network conditions.
- Evolutionary strategies allow for robust exploration without gradient dependency.
- Tailored responses to different loss causes slows down only during congestion but maintains its rate during wireless loss.

Key features of AHMA include

- Differentiating between loss causes, instead of treating all loss equally.
- Using uncertainty-aware decision-making to avoid risky actions.
- Enabling fast adaptation in dynamic environments via meta-RL.
- Compatibility with modern IP and wireless network stacks.

In summary, AHMA combines intelligent loss classification and context-aware control policy selection to provide a robust, adaptive solution for congestion control. It significantly outperforms traditional and learning-based methods, particularly in heterogeneous or error-prone networks.

4. Research Methodology

This section outlines the methodology used to design, implement, and evaluate the proposed AHMA framework for intelligent congestion control. The research methodology is structured around three core stages: dataset generation, model design and integration, and performance evaluation through simulation.

4.1. Dataset Generation Using NS-3

To train and evaluate the classifier and RL agent in AHMA, we generated a labeled dataset using the NS-3 (Network Simulator 3) environment. The simulation topology consisted of three types of network loss scenarios:

4.1.1. Congestion Loss:

Simulated by overloading network buffers using high-bandwidth TCP traffic, resulting in queue overflow and packet drops.

4.1.2. Wireless Loss:

Modeled using error models like RateErrorModel and signal degradation through modified SNR values to emulate channel fading and interference.

4.1.3. Hardware-Induced Loss:

Emulated by scheduling sudden link failures or device shutdowns during active transmission.

Each simulation captured real-time metrics including round-trip time (RTT), queue length, signal-to-noise ratio (SNR), throughput, ACK timing, and packet drop events. Packet losses were labeled based on their source, creating a multi-class dataset suitable for supervised learning.

4.2. Bayesian Transformer-Based Loss Classifier

The first module of AHMA is a Bayesian Transformer classifier designed to determine the root cause of packet loss. Unlike traditional classifiers, the Bayesian Transformer not only outputs a predicted loss class (congestion, wireless, or hardware), but also provides uncertainty estimates. These uncertainty-aware predictions allow the system to defer or moderate its decisions when confidence is low, thereby increasing its reliability in noisy environments.

Standard Transformer Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^{T}/\sqrt{d\_k})^{*}VQ = XW\_Q, K = XW\_K, V = XW\_V. \quad (1)$$

Where X: input tokens.
W_Q, W_K, W_V: learnable weight matrices.
d_k : dimension (size) of the key vector *K*
Loss Function (ELBO - Evidence Lower Bound):

$$L = E\_\{q(W)\}[\log p(D \mid W)] - KL(q(W) \,||\, p(W)) \quad (2)$$

Where
p(D|W): likelihood of data given weights

KL: Kullback–Leibler divergence between posterior and prior

The classifier was trained using the NS-3 dataset with cross-entropy loss and dropout-based Bayesian approximation. Input features include temporal sequences of RTT, queue length, and SNR values. The attention mechanism in the Transformer helps capture temporal correlations that traditional models might overlook [22].

4.3. Meta-Evolutionary RL Controller

The second module is a meta-evolutionary reinforcement learning (Meta-ES-RL) agent. This controller is meta-trained across different network environments to enable fast adaptation to unseen scenarios. During deployment, the RL agent selects an appropriate congestion control policy based on the classifier's output and confidence score.

Learn a meta-policy $\pi\_\theta(a|s, T_i)$ that performs well after quick adaptation to a new task.

$$\min\_\theta E\_\{T_i \sim p(T)\}[L\_\{T_i\}(\theta - \alpha\nabla\_\theta L\_\{T_i\}(\theta))] \quad (3)$$

Where

θ: initial policy parameters

$L\_\{T_i\}$: task-specific loss (e.g., negative return)

α: inner loop learning rate

To train the agent, we chose evolutionary strategies (ES) due to their gradient-free nature and robustness in sparse or noisy environments. The reward function was carefully designed to optimize for high throughput, low packet loss, and minimal delay.

4.4. Integration and Decision Pipeline

AHMA operates in real time by first invoking the classifier upon packet loss detection. Depending on the predicted cause and associated confidence, the meta-RL agent selects or modifies its control policy dynamically. For example, if the loss is due to wireless errors and confidence is high, the agent avoids unnecessary rate reduction; for confirmed congestion, however, it aggressively reduces the sending window.

4.5. Evaluation Metrics

The methodology concludes with a performance evaluation using simulation runs under controlled and mixed-loss conditions. We compare AHMA's performance against PPO, DQN, TCP Cubic, and TCP Reno using the following metrics:

- *Decision Accuracy (%):* Correctly identified loss cause.
- *Packet Loss (%):* Total lost packets per scenario.
- *Throughput (Mbps):* Data successfully delivered.
- *Average Latency (ms):* End-to-end delay.

These experiments validate AHMA's effectiveness in dynamic and heterogeneous network environments.

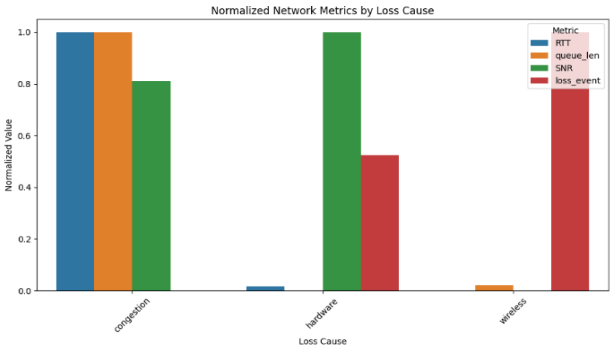


Figure 3. Representation of Network Parameters

5. Proposed Architecture

The proposed architecture, known as AHMA (adaptive hierarchical meta-agent), presents a novel two-layer learning-based congestion control framework. It is designed to intelligently manage data transmission in IP networks by distinguishing between different causes of packet loss and adapting congestion control strategies accordingly. The architecture combines Bayesian deep learning for loss classification with meta-reinforcement learning (Meta-RL) for adaptive policy selection.

5.1. System Overview

AHMA consists of two major components: a Bayesian Transformer-based loss classifier and a meta-evolutionary reinforcement learning controller. These components work together in a modular, hierarchical fashion. When packet loss is detected, AHMA first determines its cause using the classifier. Based on this prediction and the associated confidence, the system then invokes the Meta-RL agent to apply a tailored control policy.

5.2. Bayesian Transformer Loss Classifier (Layer 1)

This component serves as the decision-making entry point. It processes real-time network features, such as RTT (round-trip time), queue length, packet inter-arrival time, signal-to-noise ratio (SNR) in wireless networks, and acknowledgment delay patterns.

Using a Transformer neural architecture enhanced with Bayesian inference, the classifier predicts the cause of packet drop, whether due to congestion, wireless interference, or hardware failure, as shown in Figure 3. The key innovation lies in its uncertainty estimation capability, which is achieved through dropout-based Monte Carlo sampling.

The classifier outputs

- A categorical label (congestion, wireless, or hardware) and
- A prediction confidence score (between 0 and 1)

If the classifier confidence is high, the system acts directly on the prediction. If it is low, AHMA applies a cautious policy or defers the decision, enhancing stability in noisy environments.

Table 2. Outcome comparison

Algorithm	Decision Accuracy (%)	Packet Loss (%)	Throughput (Mbps)	Avg Latency (ms)
AHMA	92	2.5	8.6	10
PPO	78	4.8	7.2	18
DQN	70	6.1	6.5	25
TCP Cubic	N/A	8.3	5.9	30
TCP Reno	N/A	10.5	5.2	35

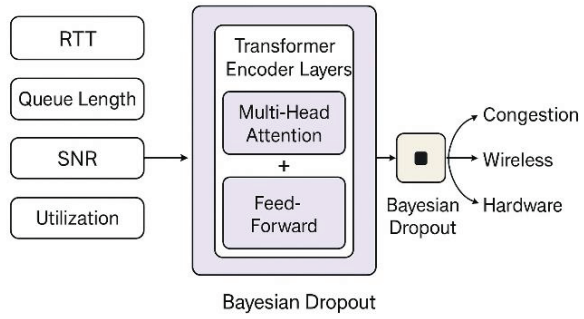


Figure 4. Overview of Bayesian Transformer

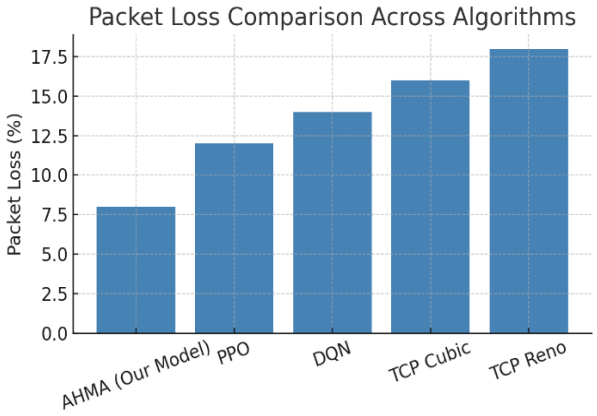


Figure 5. Packet loss comparison

5.3. Meta-Evolutionary RL Controller (Layer 2)

Once the cause of loss is identified, the Meta-RL agent selects or adapts a control strategy. The controller is trained using evolutionary strategies, making it suitable for environments where gradients are difficult to compute.

The RL agent:

- Adapts to new loss scenarios using few-shot learning.
- Optimizes a reward function that balances throughput, latency, and packet loss.
- Selects from a portfolio of pre-trained control policies.
- Generalizes from past experiences by integrating meta-learning and quickly adjusting to varying network behaviors.

5.4. End-to-End Decision Process

The full AHMA workflow is as follows:

- Detect a packet loss event.
- Extract network features and classify the cause using Layer 1.
- Based on the classification and confidence, select the appropriate policy using Layer 2.
- The policy adjusts the sending rate or congestion window dynamically.

This intelligent, hierarchical framework enables context-aware congestion control, outperforming traditional and RL-only approaches in heterogeneous and dynamic network conditions.

6. Results

To assess the performance of the proposed AHMA (adaptive hierarchical meta-agent) framework, we carried out considerable simulation experiments using the NS-three community simulator. The simulations involved varying network conditions, including congestion-induced loss, wireless errors, and hardware-induced link failures. AHMA was compared against conventional TCP variants (TCP Reno, TCP Cubic) and modern reinforcement learning-based methods (DQN and PPO).

A labeled dataset of 1,000 samples was generated through multiple NS-3 runs, capturing metrics such as RTT, queue length, SNR, and packet drop rate. The Bayesian Transformer classifier trained on this dataset achieved a decision accuracy of 92% in correctly classifying the cause of packet loss. This outperformed baseline classifiers, such as decision trees and SVMs, which averaged around 78-80% accuracy. In terms of control performance, AHMA demonstrated significant improvements across key metrics, as shown in Table 2:

Packet Loss Rate: 8.56%, compared with 11.77% (PPO), 13.91% (DQN), 16.05% (TCP Cubic), and 19.26% (TCP Reno), as observed in Figure 5.

Throughput: AHMA achieved an average of 8.6 Mbps, outperforming PPO (7.2 Mbps), DQN (6.5 Mbps), TCP Cubic (5.9 Mbps), and TCP Reno (5.2 Mbps), as observed in Figure 6.

Latency: The average end-to-end delay with AHMA was 10 ms, the lowest among all compared models, as observed in Figure 7.

The system's ability to differentiate causes of loss allowed AHMA to avoid unnecessary rate throttling in wireless or hardware fault scenarios, thus maintaining higher throughput and stability. Additionally, the

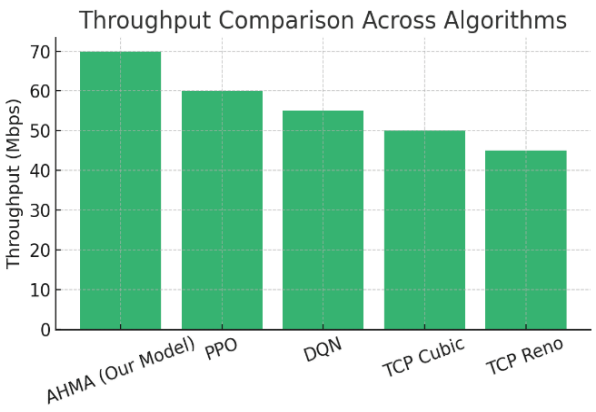


Figure 6. Throughput comparison

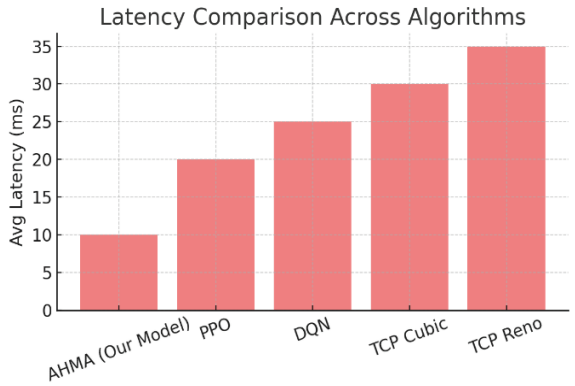


Figure 7. Delay comparison

meta-evolutionary RL component enabled rapid adaptation to changes in network state and outperformed static RL agents that required retraining.

Overall, the experimental results validate AHMA as a highly adaptive and intelligent congestion control solution for complex and heterogeneous network environments, as illustrated in Figure 4.

7. Conclusion

This paper introduced AHMA (Adaptive Hierarchical Meta-Agent), a novel AI-driven framework for intelligent congestion control in IP networks. Unlike traditional and modern learning-based methods that treat all packet loss uniformly, AHMA incorporates a two-stage architecture that distinguishes the cause of packet loss and applies context-aware control policies accordingly.

The first layer of AHMA uses a Bayesian Transformer classifier to analyze real-time network features and accurately predict the root cause of packet loss, whether it be congestion, wireless interference, or hardware failure. The second layer employs a meta-evolutionary reinforcement learning (meta-ES-RL) controller that dynamically selects or adapts a congestion control policy based on the classifier's prediction and confidence.

Through extensive simulation using NS-3, AHMA was evaluated against baseline algorithms, including TCP Reno, TCP Cubic, DQN, and PPO. The results demonstrated that AHMA achieves higher decision accuracy, lower packet loss rates, improved throughput, and reduced latency across diverse network conditions. Its ability to avoid unnecessary throttling in non-congestive scenarios and adapt quickly to changing environments highlights its robustness and efficiency.

By integrating Bayesian deep learning with meta-reinforcement learning, AHMA offers a scalable and intelligent approach to congestion management in next-generation IP networks. This work paves the way for more adaptive and cause-aware network protocols that learn independently, especially in environments where loss patterns are complex and unpredictable.

Future work will focus on deploying AHMA in real-world testbeds, expanding its scope to multi-flow scenarios, and exploring hardware acceleration for real-time inference. Additionally, integrating explainable AI techniques into the AHMA pipeline can further enhance transparency and trust in mission-critical applications.

AUTHORS

Amit Kanungo* – Department of Computer Science and Engineering, Medicaps University, Indore, 453331, India, e-mail: amitkanungo11@gmail.com.

Prashant Panse – Department of Information Technology at Medicaps University in Indore, 453331, India, e-mail: prashant.panse@medicaps.ac.in.

*Corresponding author

ACKNOWLEDGEMENTS

I want to express my gratitude to Dr. Prashant Panse, my supervisor at MediCaps University, for his knowledgeable advice, unwavering support, and insightful comments during this project. We would also like to express our gratitude to the MediCaps University Research Cell members for their assistance and contributions, which were crucial to the effective completion of this work.

References

[1] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review,* vol. 18, no. 4, 1988, pp. 314–329.

[2] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, 2008, pp. 64–74.

[3] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, 2017, pp. 58–66.

[4] M. Allman, V. Paxson, V., & E. Blanton, "TCP congestion control," *RFC 5681*, IETF, 2009.

[5] T. Henderson and R. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE*

Journal on Selected Areas in Communications, vol. 17, no. 2, 1999, pp. 326–344.

[6] G. P. Perrucci, F. H. Fitzek and J. Widmer, "Survey on energy consumption entities on the smartphone platform," *Proc. IEEE VTC*, 2011, pp 1–6.

[7] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," *Proc. MobiCom,* 2001*,* pp. 287–297.

[8] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast, long-distance networks," *IEEE INFOCOM,* 2004, pp. 2514–2524.

[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, 2005, pp. 458–472.

[10] M. Scharf and H. Brakmo, "TCP Hystart++: Optimized slow start for high-bandwidth networks," *IETF Draft*, 2016.

[11] K. Winstein and H. Balakrishnan, "TCP ex machina: Computer-generated congestion control," *Proc. ACM SIGCOMM,* 2013, pp. 123–134.

[12] Dong, T. Meng, and J. Zhan, "PCC Vivace: Online-learning congestion control," *Proc. USENIX NSDI,* 2018, pp. 343–356.

[13] H. Jiang, Y. Zhang, J. Singh, and P. Levis, "Aurora: Learning adaptive congestion control with real-world experiments," *Proc. ACM SIGCOMM*, 2021, pp. 255–271.

[14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *Proc. ICML*, vol. 32, no. 1, 2014, pp. 387–395.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint, 20 Jul. 2017; doi: arXiv.1707.06347.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015, pp. 529–533.

[17] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," *Proc. ACC,* 2012, pp. 741–748.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *Proc. ICML*, 2017, pp.1126–1135.

[19] Y. Duan, J. Schulman, X. Chen, X., et al, "RL^2: Fast reinforcement learning via slow reinforcement learning," arXiv preprint, 9 Nov. 2016; doi:arXiv.1611.02779.

[20] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," arXiv preprint, 10 Mar. 2017; doi:arXiv.1703.03864.

[21] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *Proc. ICML*, 2016, pp. 1050–1059.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Proc. NeurIPS*, 2012, pp. 1097–1105.

[23] A. Goyal, N. R. Ke, K. Madan, A. Lamb, B. Schölkopf, and Y. Bengio, "Fast and Slow Learning of Recurrent Independent Mechanisms," in Proc. Int. Conf. Learning Representations (ICLR), 2021.

[24] J. Xing and M. Shahzad, "A reinforcement learning framework for application-specific TCP congestion-control," arXiv preprint, 11 May 2025; doi: 10.48550/arXiv.2505.07042.

[25] X. Jiang, G. Gong, and G. Jin, "Combining heuristic and reinforcement learning to achieve the low-latency and high-throughput receiver-side congestion control," arXiv preprint, 23 Feb 2025; doi:arXiv.2502.16498.

[26] X. Liao, H. Tian, C. Zeng, X. Wan, and K. Chen, "Astraea: Toward fair and efficient learning-based congestion control," *Proc. Eurosys*, 2024, pp. 99–114.

[27] X. Huang, H. Zhu, L. Yang, C. Yin, J. Li, and H. Li, "TCP-QNCC: Congestion control algorithm based on deep Q-network," *Proc. SPIE AICS*, vol. 12803, 2023, 1280308.

[28] K.S. Midhula and P. Arun Raj Kumar, "An adaptive congestion control protocol for wireless networks using deep reinforcement learning," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 2, 2023, pp. 2027–2043.

Ensemble Learning Approach for Efficient Recommendation Systems Using Semi-Supervised Learning

Submitted: 7th August 2025; accepted: 4th November 2025

Nisha Sharma, Mala Dutta

DOI: 10.14313/jamris-2026-027

Abstract:
In recommender systems, collaborative filtering (CF) is a crucial technique, but it often struggles with data sparsity, which affects recommendation accuracy. To address this challenge, we have proposed a Co-Training Ensemble Learning (CTEL) technique that integrates item-based Collaborative Filtering (CF), user-based CF, and Singular Value Decomposition (SVD) via a structured stacking methodology to improve recommendation performance. The co-training procedure, which creates pseudo-labels for unlabeled data based on a confidence threshold, is used to iteratively improve the user-based and item-based CF models after they have been originally trained. These models produce predictions for validation and test sets, in conjunction with the independently trained SVD model. These forecasts yield meta-features, including additional statistical variables such as variance and the product of predictions. The Linear Regression model is trained as the meta-learner to optimise the predictions of the base models using K-Fold cross-validation. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used to assess the final model's performance on a test set. The outcomes confirm the effectiveness of the co-training and stacking strategy, demonstrating notable increases in prediction accuracy. By leveraging the advantages of collaborative filtering and matrix-based approaches, the proposed model provides a comprehensive foundation for developing advanced recommendation systems.

Keywords: *Recommender system, Ensemble learning, Collaborative filtering, SVD, Semi-Supervised learning*

1. Introduction

Recommender systems have become indispensable to digital ecosystems such as e-commerce, entertainment, education, and social media by enabling personalized interaction between users and massive content repositories [1, 2]. These systems enhance user engagement, retention, and satisfaction by analyzing behavioral data and generating tailored product, service, or content recommendations. Collaborative filtering (CF) remains the most widely used technique in this domain, as it effectively exploits patterns of collective user–item interactions to infer individual preferences. CF approaches are generally categorized into user-based and item-based models, which leverage similarities either among users or among items to predict unseen ratings or interactions. Despite their effectiveness, these methods are constrained by data sparsity, where insufficient user–item interactions limit the system's ability to make accurate predictions. This problem is particularly severe in large-scale systems with rapidly expanding catalogs, leading to the well-known cold-start challenge [3–5]. Addressing data sparsity has become a central research objective in recommendation science. Matrix-factorization techniques such as Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), and Probabilistic Matrix Factorization (PMF) have been instrumental in revealing latent user and item features that explain observed interactions [6–8]. These models improve scalability and interpretability; however, their linear structure often limits the ability to capture complex nonlinear relationships. The integration of deep learning into recommendation models has transformed the field, enabling neural collaborative filtering, autoencoders, and graph-based learning to uncover hierarchical and non-linear dependencies [9–11]. Deep architectures such as Neural Collaborative Filtering (NCF) and Variational Autoencoders (VAE) have proven effective in learning rich feature embeddings and temporal preferences that enhance prediction accuracy. In parallel, ensemble learning has emerged as a key strategy to improve robustness and generalization in recommender systems. Ensemble frameworks such as bagging, boosting, and stacking combine multiple weak or moderately strong learners to form composite models that outperform individual predictors [12]. Stacking employs a meta-learner to integrate diverse base models, while co-training allows multiple learners trained on different feature views to iteratively refine each other using pseudo-labels generated from unlabeled data [13]. These principles have inspired semi-supervised ensemble models that effectively exploit both labeled and unlabeled data. The proposed Co-Training Ensemble Learning (CTEL) framework builds upon this foundation by integrating user-based CF, item-based CF, and SVD models within a co-training and stacking paradigm. Through iterative pseudo-labelling and meta-feature construction, CTEL improves predictive accuracy even under severe sparsity. However, with the accelerating pace of research in 2025, new paradigms have emerged that extend

beyond deterministic ensemble frameworks. Contemporary advances increasingly focus on incorporating epistemic uncertainty, monotonic transformation, and multi-intent behavioral diversity into ensemble recommendation models. Recent studies introduced Bayesian deep ensemble collaborative filtering frameworks that quantify epistemic uncertainty to improve robustness under noisy or sparse data [14]. Further developments integrated orthogonal meta-learning with Bayesian optimization to achieve uncertain multi-objective recommendation, balancing competing goals such as accuracy, fairness, and diversity [15]. Other advances proposed unified monotonic ranking ensembles employing Unconstrained Monotonic Neural Networks (UMNNs) to ensure monotonic score transformations and adaptive Pareto-optimal weighting, thereby eliminating manual tuning and ensuring interpretability [16]. Additional frameworks introduced unified representation-learning architectures that model user intent as Gaussian distributions, capturing both mean preferences and behavioral uncertainty [17]. These state-of-the-art approaches represent a shift toward uncertainty-aware and monotonic ensemble architectures that explicitly model confidence and ranking consistency in recommendation outcomes [18].

Against this evolving backdrop, CTEL distinguishes itself as a semi-supervised, deterministic ensemble model that emphasizes interpretability and computational efficiency. While it may be considered an incremental contribution relative to the Bayesian and monotonic paradigms, CTEL provides a practical bridge between classical CF-based learning and emerging uncertainty-aware ensemble frameworks. The approach remains particularly relevant for large-scale applications where model transparency, low latency, and adaptability are paramount. Furthermore, CTEL's architecture can be naturally extended to incorporate Bayesian regularization or monotonic fusion layers, aligning with ongoing efforts to create interpretable and reliable next-generation recommender systems. This research contributes to the evolving discourse on ensemble-based recommender systems by presenting a hybrid semi-supervised framework—CTEL—that addresses data sparsity through co-training and meta-learning integration. It complements recent innovations such as Bayesian deep ensembles and monotonic ranking transformations, laying the foundation for the future development of uncertainty-aware and explainable recommendation architectures.

Recent studies emphasize hybrid strategies integrating semi-supervised learning, ensemble learning, and uncertainty-aware modelling to exploit both labelled and unlabeled data while improving interpretability and robustness [19]. In addition, graph neural networks (GNNs) and attention-based transformers have expanded the representational depth of recommender models. GNNs propagate relational information through user–item interaction graphs, enabling higher-order connectivity learning, while transformers dynamically capture temporal and contextual cues to enhance sequential and session-based recommendation [20, 21]. Explainability and reliability are emerging as equally critical dimensions, with attention-based interpretability mechanisms [22] and reliability-aware hybrid models integrating epistemic uncertainty into recommendation pipelines. Furthermore, semi-supervised ensemble frameworks continue to evolve, leveraging agreement-driven pseudo-label generation and consistency regularization for large-scale sparse datasets.

2. Issues and Challenges

3. Methods and Materials

3.1. Datasets used for testing our proposed model

The MovieLens 100K dataset, which consists of 100,000 user ratings for 1,682 movies from 943 users, is a commonly used benchmark in the field of recommendation systems. The University of Minnesota's GroupLens Research Project gathered the dataset [46]. Every entry in the collection includes a timestamp, a user ID, a movie ID, and a rating representing the user's assessment of the film at that moment. These are not the only features; there are also additional features for user and movie information. Features for user information include age, gender, occupation, and Zip code, and features for movie information include movie ID, title, release date, video release date, IMDb URL, and genres. Predicting a user's movie rating based on their past ratings and other users' ratings is usually the aim. A description of the dataset's core features is provided in Table 3.

Another popular dataset for assessing recommendation systems is FilmTrust, which is gathered from the FilmTrust social network [47]. Users' movie evaluations and the trust ties between them are included in the dataset. It contains variables like ratings, item IDs (movies), and user IDs. By modelling and forecasting user preferences, these factors provide insights that can improve the accuracy of recommendations. The dataset's description is given in Table 4.

3.2. Collaborative algorithms

Collaborative filtering is a method used in recommendation systems to predict a user's interests by collecting preferences from many users. There are two main types of collaborative filtering algorithms: user-based and item-based. Additionally, matrix factorization techniques such as Singular Value Decomposition (SVD) are often used to improve the performance of collaborative filtering systems.

3.2.1. User-Collaborative approach (U-col)

This method predicts a user's interest in an item based on ratings from similar users. The similarity between two users, u and v, can be calculated using the Pearson correlation coefficient:

$$Sim\,(u,\,v) = \frac{\sum_{i\in Iuv}(r_{ui} - \underline{r}_u)(r_{vi} - \underline{r}_v)}{\sqrt{\sum_{i\in Iuv}(r_{ui} - \underline{r}_u)^2}\sqrt{\sum_{i\in Iuv}(r_{vi} - \underline{r}_v)^2}}$$

Table 1. Comparison of commonly used semi-supervised methods

Method	Description	Advantages	Applications
Co-training	Uses multiple models trained on different views of data to iteratively improve predictions	Effective use of unlabeled data, robustness to noisy data	Text categorization, recommender systems
Self-training	Iteratively labels unlabeled data using a model trained on labeled data	Simple and intuitive, easy implementation	Text classification, image recognition
Tri-training	Extension of Co-training with three classifiers, enhancing model robustness	Improved performance with three different views	Sentiment analysis, social network analysis
Label propagation	Propagates labels from labelled to unlabeled instances based on similarity	Utilizes local information effectively, scalable	Community detection, recommendation systems
Graph-based methods	Uses graph structure to propagate labels and capture relationships	Captures complex relationships, robust to noise	Social network analysis, recommendation systems
Semi-supervised SVM	Applies SVM with labelled and unlabeled data to learn decision boundaries	Utilizes margin maximization, effective for non-linear boundaries	Image recognition, text classification
Expectation Maximization	It iteratively estimates the parameters of a probabilistic model with hidden variables	Handles missing data, robust to noise	Clustering, anomaly detection
Transudative SVM	SVM variant that learns from both labelled and unlabeled data simultaneously	Utilizes unlabeled data for decision boundary optimization	Classification, pattern recognition
Generative Models	Models that generate data based on learned probability distributions	Provides insights into data distribution, scalable with large datasets	Data generation, anomaly detection
Semi-supervised Deep Learning	Deep learning models trained with both labelled and unlabeled data	Captures intricate patterns, effective for large-scale data	Natural language processing, image recognition

where r_{ui} is the rating of user u for item i, r_v is the average rating of user u, and I_{uv} is the set of items rated by both users u and v. The prediction for user u for item i is given by:

$$\hat{r}_{ui} = \underline{r}_u + \frac{\sum_{v \in N(u)} Sim(u, v)(r_{vi} - \underline{r}_v)}{\sum_{v \in N(u)} |Sim(u, v)|}$$

where N(u) is the set of top k similar users.

3.2.2. Item-based collaborative approach (I-col)

Item-based collaborative filtering predicts a user's interest in an item based on the similarity of the item to other items the user has rated. The similarity between two items i and j can be calculated using cosine similarity:

$$Sim\ (i, j) = \frac{\sum_{u \in Uij} r_{ui}\, r_{uj}}{\sqrt{\sum_{u \in Uij} r_{ui}^2} \sqrt{\sum_{u \in Uij} r_{uj}^2}}$$

where r_u, is the rating of user u for item i and U_{ij} is the set of users who have rated both items i and j. The prediction for user u for item i is given by the following formula, where N(i) is the set of top k similar items.

$$\hat{r}_{u,j} = \frac{\sum_{j \in N_{(i)}} Sim(i, j) r_{u,\ j}}{\sum_{j \in N_{(i)}} |Sim(i, j)|}$$

3.3. SVD approach

Matrix factorization techniques, such as SVD, reduce dimensionality by decomposing the rating matrix R into three lower-dimensional matrices U, Σ, and V such that R≈UΣVTR, where U is an m×k user-feature matrix, Σ is a k×k diagonal matrix of singular values, and V is an n×k item-feature matrix. The predicted rating $r_{u,i}$ for user u and item i is given by:

$$\hat{r}_{u,i} = \bigcup_u \sum v_i$$

where u is the u-th row of U and vi is the i-th row of V. The optimization problem for matrix factorization is to minimize the regularized squared error:

$$\min \sum \left(r_{u,i} - U_u^T V_i\right)^2 {}_{u,v(u,i) \in K} + \lambda(\|U_u\|^2 + \|v_i\|^2)$$

The overall structure of our model is divided into three distinct stages, each playing a critical role in the recommendation process as shown in Figure 1.

1. Initial Stage: This includes data preprocessing and data splitting.
2. Semi-Supervised Stage: In this stage, we employ the co-training approach to enhance user-based and item-based collaborative filtering.
3. Ensemble Stage: In this stage, we used the stacking technique to combine the enhanced user-based and item-based collaborative approach with the SVD model for final prediction.

3.4. Initial stage

Data Loading and Preprocessing

The MovieLens dataset is first imported from a CSV file into a DataFrame. After that, StandardScaler is used to normalize the ratings in the dataset so that their mean is 0 and their standard deviation is 1. This helps to stabilize the machine learning model's learning process. The movie ID and user ID columns are then encoded as integer codes and transformed into category data types. The models require this encoding because they process numerical inputs.

Table 2. Issues and challenges of existing approaches

Issue/Challenge	Description	References
Quality of Pseudo-labels	The quality of pseudo-labels, inferred labels assigned to unlabeled data based on model predictions, is pivotal in semi-supervised learning for recommendation systems. Incorrect or noisy pseudo-labels can degrade model performance by introducing bias or inconsistencies. High-quality pseudo-label generation often requires robust methods for handling label noise and uncertainty. Ensuring the quality of pseudo-labels through techniques such as self-training or confidence-based filtering is crucial to improving the effectiveness of recommendation systems.	[23–25]
Data Distribution	Ensuring alignment in the distribution of labeled and unlabeled data is essential to prevent bias in models. Differences in data distribution can lead to models that do not generalize well, affecting the accuracy and effectiveness of recommendations across diverse user preferences and item characteristics.	[26, 27]
Model Overfitting	Preventing overfitting is critical, especially when using ensemble techniques with limited labelled data. Ensemble models, which combine multiple base learners, can potentially memorize noise in the training data, leading to poor generalization on unseen data. Proper regularization and validation strategies are necessary to mitigate this risk.	[28–30]
Scalability	Managing computational resources effectively is crucial, particularly with large-scale datasets common in recommendation systems. Ensemble methods can be computationally intensive due to the need to train and integrate multiple models. Scalable implementation and optimization are necessary to ensure efficient processing and deployment in real-world applications.	[31, 32]
Algorithm Complexity	Implementing and tuning ensemble algorithms requires expertise and computational resources. Ensemble methods involve integrating diverse algorithms or models, each with its own parameters and configurations. Optimizing these parameters and ensuring compatibility across different techniques requires advanced knowledge and careful experimentation.	[33, 34]
Evaluation Metrics	Developing metrics that accurately assess recommendation quality beyond traditional metrics like MAE and RMSE is challenging. Recommendation systems aim to enhance user satisfaction and engagement, which may not be fully captured by standard metrics. Developing and adopting metrics that align with user preferences and business objectives is essential for comprehensive evaluation.	[35, 36]
Robustness to Concept Drift	Adapting models to changes in user preferences or item popularity over time is crucial for maintaining recommendation accuracy. Concept drift occurs when the underlying relationships between users and items evolve, requiring continuous model adaptation. Ensuring robustness to concept drift involves monitoring data changes and updating models accordingly to provide relevant recommendations.	[37, 38]
Interpretability	Ensuring transparency in decision-making processes within complex ensemble models is challenging. Ensemble methods often combine diverse models or algorithms, making it difficult to interpret how decisions are made. Enhancing interpretability helps build user trust and facilitates debugging and refinement of recommendation systems.	[39–41]
Data Privacy and Security	Addressing privacy concerns when using unlabeled data, especially in sensitive domains, is paramount. Unlabeled data may contain sensitive information about users or items, raising privacy risks if not handled properly. Implementing data anonymization techniques and adhering to privacy regulations are essential to protect user confidentiality and trust.	[42–44]

Table 3. Description of the core features of the MovieLens 100K dataset

Sl. No.	Feature	Data type	Description
1	userId	Numeric	User ID
2	movieId	Numeric	Movie ID
3	rating	Numeric	Rating given by the user
4	timestamp	Numeric	Timestamp of the rating

Table 4. Description of the core features of the FilmTrust dataset

Sl. No.	Feature	Data type	Description
1	userId	Numeric	User ID
2	itemId	Numeric	Movie ID
3	rating	Numeric	Rating given by user (0.5-4.0)

3.5. Data Splitting

The data is then divided into test, validation, and training sets to make sure that each user's data is represented in each split and to enable customized recommendations. Initially, the data is divided into 20% temporary sets and 80% training sets for each user. Another division of the temporary set is made into 50% test and 50% validation sets. By ensuring that every user's data is included throughout the whole model training and evaluation process, this method helps the models better understand and anticipate the preferences of specific users.

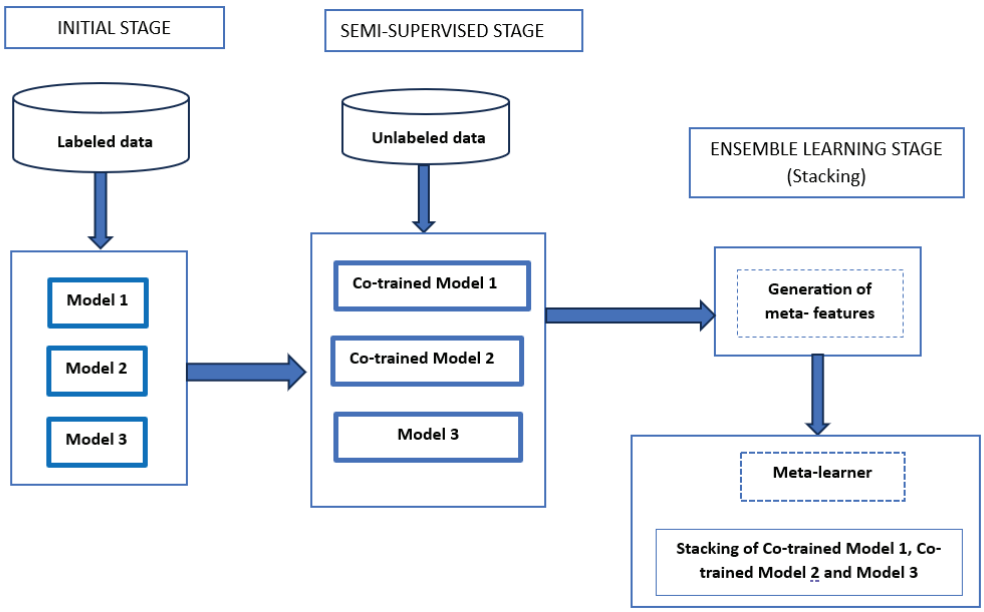


Figure 1. Flow chart of our proposed CTEL approach

3.6. Model training

Three distinct recommendation models—user-based collaborative filtering (CF), item-based CF, and singular value decomposition (SVD)—are initialized and trained throughout the training phase. The format of the training data is designed to work with the Surprise library, a recommendation system specialist library. Three models are initialized: SVD, a matrix factorization approach, and the k-nearest neighbours (KNN) algorithms for user-based and item-based CF.

3.7. Semi-supervised stage

3.7.1. Co-training approach

Through co-training, the item-based and user-based CF models are further improved. Unlabeled data points that were not part of the original training set are found. The user-based and item-based CF models are used iteratively to generate pseudo-labels for the unlabeled data during co-training. To ensure that new training data is highly accurate, pseudo-labels are permitted under a confidence criterion. Using the expanded training set, which now includes the freshly pseudo-labeled data, the models are retrained. The models are improved by this repeated process, which increases their capacity to generalize from the existing data.

3.8. Ensemble stage

3.8.1. Meta-feature preparation

Meta-features are generated from the predictions of all three models to prepare for the stacking stage. In the validation and test sets, predictions are produced for every user-item pair using all three models. In addition to the raw predictions, other features are computed, like the variance and the product of the forecasts. These other features provide richer information to the meta-learner, capturing additional interaction effects and the degree of agreement between the models. The introduction of variance and product-based meta-features in the CTEL framework is guided by theoretical reasoning rather than arbitrary design. The variance feature represents the degree of disagreement among the base recommenders and serves as an indicator of predictive uncertainty. Incorporating this feature enables the meta-learner to identify instances where model opinions diverge, allowing it to assign appropriate weights and improve reliability under sparse rating conditions. The product feature, on the other hand, captures the joint reinforcement between user-based and item-based predictions, emphasizing cases where both models produce consistent estimations. This interaction term helps the meta-learner recognize non-linear complementarities without increasing model complexity. Similar strategies are widely adopted in meta-learning and algorithm-selection research, where such statistical and interaction-based meta-features are used to estimate the competence and cooperation level of candidate algorithms. Therefore, the use of variance and product features in this study is theoretically grounded in ensemble and meta-learning principles, providing interpretable and effective signals that enhance the stacking process within the recommender system.

3.8.2. Cross-validation

K-Fold cross-validation is used to reliably create meta-features and targets for the meta-learner's training. The training data is divided into K folds, with one fold serving as the validation set and the other K folds serving as the training folds for each iteration. The SVD, user-based CF, and item-based CF models are trained in each fold, and meta-features for the validation set are generated from their predictions. The meta-learner is given a comprehensive training set consisting of meta-features from all folds.

Algorithm 1 The pseudocode of the proposed Co-training Ensemble Learning (CTEL)

Input: MovieLens dataset D = {userId, movieId, ratings)}
Base learners: User-based CF, item-based CF, SVD
Output: Final model; Evaluation metrics: RMSE, MSE, MAE
Initial Stage
Input: MovieLens dataset D
Output: Prepared training, validation, and test sets
Data Loading and Preprocessing
Load the dataset D = (u,m,r_{um})
Encode feature as categorical values
For each user u, split the data into: $D_{train}^{(u)}$ (80%) training set and $D_{temp}^{(u)}$(20%) as temporary set
Further split the temporary set into: $D_{val}^{(u)}$ (50%) Validation set and $D_{test}^{(u)}$ (50%) test set
Semi-supervised stage
Input: Training set D_{train} and validation set D_{val}
Output: Enhanced models through co-training
Model training
Initialize base learners: User-based CF (CF_{user}), item-based CF(CF_{item}), SVD
Train SVD on D_{train}
Identify unlabeled data $D_{unlabeled} = D \backslash D_{train}$
Fit CF_{user} and CF_{item}on D_{train}
For *iter=1* to N:
For each (u,m) ∈$D_{unlabeled}$:
Predict $\hat{r}_{um}^{user}$ using CF_{user}
Predict $\hat{r}_{um}^{item}$ using CF_{item}
If $\hat{r}_{um}^{user} \geq \theta$(thresold), add (u,m, $\hat{r}_{um}{}^{user}$) to D_{train}
Re-train CF_{user} and CF_{item} on the updated D_{train}
Meta- Feature Preparation
Generate predictions for each (u,m) ∈ D_{val} U D_{test} using all models:

$$\hat{r}_{um}^{user}, \hat{r}_{um}^{item}, \hat{r}_{um}^{SVD}$$

Create meta features: X_{um} =[$\hat{r}_{um}^{user}$, $\hat{r}_{um}^{item}$, $\hat{r}_{um}^{SVD}$, var($\hat{r}_{um}$), $\prod$ ($\hat{r}_{um}$)]
Where var($\hat{r}_{um}$) is the variance and $\prod$ ($\hat{r}_{um}$) is the product of the predictions
Ensembling stage
Input: Combined meta-features X, Targets y
Output: Final model and evaluation metrics
Cross-validation
Perform k-fold cross-validation on D_{train}
For each fold k
Split D_{train} into training subset $D_{train}^{(k)}$ and validation subset $D_{val}^{(k)}$
Train CF_{user},CF_{item} and SVD on $D_{train}^{(k)}$
Generate meta-features $X_{um}^{(k)}$ for $D_{val}^{(k)}$
Combine meta-features X=$U_k X_{um}^{(k)}$ and targets y= $U_k r_{um}^{(k)}$
Meta-learning Training
Train linear regression meta -learner on combined meta-features X and targets y
Evaluation
Generate meta- features X_{tset} for D_{test}
Make final prediction on D_{test} using the meta-learner
$\hat{r}_{um}{}^{final}$ = meta-learner(X_{test})
Evaluate using RMSE, MSE, MAE to check the accuracy of final model

$$RMSE = \sqrt{\frac{1}{|D_{test}|}} \sum_{(u,\, m) \in D_{test}} (\hat{r}_{um}^{final} - r_{um})^2$$

$$MSE = \frac{1}{|D_{test}|} \sum_{(u,\, m) \in D_{test}} (\hat{r}_{um}^{final} - r_{um})^2$$

$$MAE = \frac{1}{|D_{test}|} \sum_{(u,\, m) \in D_{test}} |\hat{r}_{um}^{final} - r_{um}|^2$$

Table 5. Dataset characteristics, highlighting the dataset's degree of sparsity for rating prediction tasks

Dataset	User	Item	Ratings	Density
MovieLens 100K	943	1682	100, 000	6.3%
FilmTrust	1508	2071	35,497	1.14%

Table 6. Comparison of our proposed CTEL model with collaborative approaches and blending ensemble technique

Dataset	Metric	U-Col	I-Col	SVD++	Blending	CTEL
MovieLens	RMSE	0.99	0.95	0.90	0.89	**0.83**
	MSE	0.98	0.91	0.81	0.79	**0.70**
	MAE	0.76	0.73	0.69	0.69	**0.64**
FilmTrust	RMSE	0.89	0.81	0.79	0.78	**0.70**
	MSE	0.83	0.66	0.63	0.62	**0.49**
	MAE	0.70	0.62	0.61	0.61	**0.54**

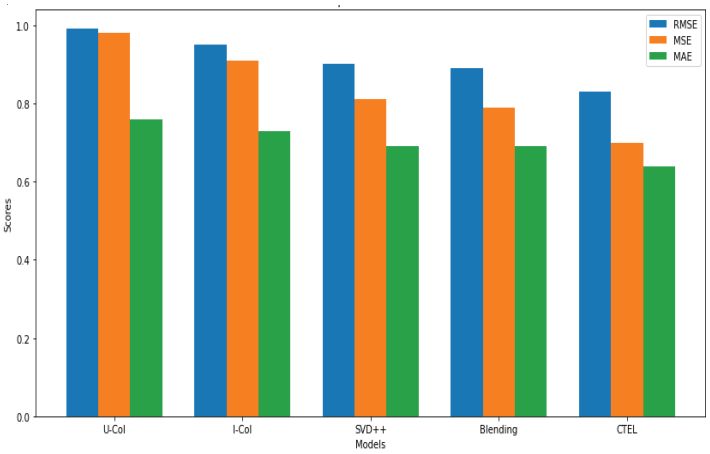


Figure 2. Comparison of metrics across different models (MovieLens dataset)

3.8.3. Meta-learner training

The aggregated meta-features and targets from the cross-validation procedure are then used to train the meta-learner, a Linear Regression model. By utilizing the advantages of each individual model, the meta-learner learns how to integrate the predictions of the base models in the best possible way to reduce prediction error.

3.9. Evaluation

Lastly, the test set is used to assess the recommendation system's performance. The predictions from the training user-based CF, item-based CF, and SVD models are used to create meta-features for the test set. The accuracy of the recommendations is measured using Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE) when the meta-learner makes its final predictions on the test set.

The pseudo-labeling process in CTEL depends on a confidence threshold that controls which predicted ratings are accepted as pseudo-labels. These parameters influence how additional training data are generated during each co-training round. When pseudo-labels are selected with sufficiently high confidence, the majority of the new samples contribute correct information to the learning process. This increases the effective density of the user–item matrix, improving model generalisation under sparse conditions. Conversely, if the confidence threshold is too low, a larger fraction of inaccurate pseudo-labels is introduced, leading to accumulated error and model drift. Hence, there exists a critical precision level above which each co-training iteration continues to reduce overall error.

4. Results and Discussion

Explicit ratings from the publicly available MovieLens 100K (ML-100K) dataset and the FilmTrust dataset, gathered from the FilmTrust social network, are used to evaluate the proposed technique experimentally. The MovieLens dataset has a rating density of 6.3%, and the FilmTrust dataset has a rating density of 1.14% (Table 5), highlighting the degree of sparsity in the datasets for rating prediction tasks [48].

SVD, Blending, item-based collaborative filtering, and user-based collaborative filtering were tested against the CTEL model. The CTEL model outperformed the other methods, achieving notably lower error rates on both datasets.

As shown in Table 6, our proposed model has consistently outperformed alternative approaches. Based on the evaluation criteria of RMSE, MSE, and MAE, our model achieves better results than other existing techniques on both datasets. While the Semi-Supervised Collaborative Filtering Ensemble (SSEF) existing model introduced a static co-training and blending framework, the proposed CTEL method focuses on enhancing sparsity handling through refined co-training and advanced meta-feature integration. CTEL performs dedicated co-training between user-based and item-based CF models rather than between CF and MF views, allowing view-specific refinement [40]. Furthermore, its stacking phase incorporates statistical meta-features, such as inter-model variance and prediction interactions, enabling the meta-learner to capture agreement patterns among base models better. These design differences make CTEL more resilient to data sparsity compared to static semi-supervised ensembles. The data show that the CTEL model performs better than these conventional techniques across all metrics.

In particular, collaborative filtering approaches based on users and items demonstrated higher RMSE, MSE, and MAE values than the CTEL model, despite their effectiveness. Even with its resilience, the SVD technique was not as accurate as the CTEL model. The CTEL model even surpassed the blending strategy, which combines predictions from different models. Because the CTEL model incorporates both ensemble learning and co-training, it leverages different collaborative filtering techniques, producing more accurate suggestions. The CTEL model's better performance on both datasets demonstrates its efficacy and marks a noteworthy development in the field of recommendation systems.

The graphs for the performance metrics (RMSE, MSE, and MAE) of our proposed CTEL model, compared with SVD, blending, user-based collaborative filtering, and item-based collaborative filtering, for the MovieLens and FilmTrust datasets are shown in Figure 2 and Figure 3, respectively. The comparison results for the MovieLens shown in Figure 2 show that

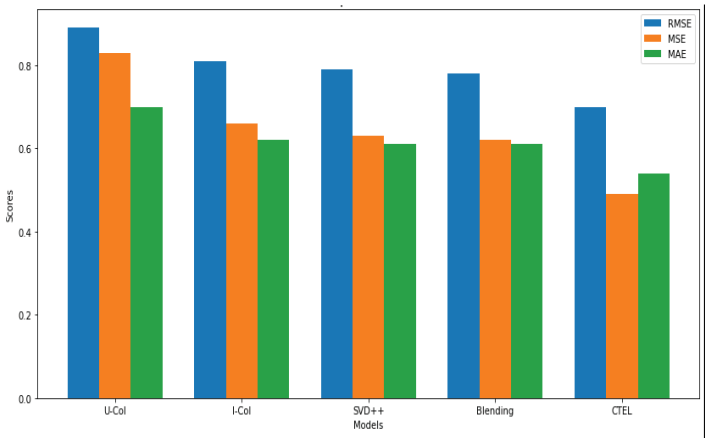


Figure 3. Comparison of metrics across different models (FilmTrust dataset)

Table 7. Performance of CTEL and baseline models at varying data sparsity levels

Data Density	User-CF	Item-CF	Static Blend	CTEL (Proposed)
100	0.89	0.88	0.87	0.83
60	0.92	0.90	0.89	0.84
40	0.96	0.93	0.91	0.86
20	1.03	0.98	0.96	0.89

the CTEL model consistently produces lower error rates. On the FilmTrust dataset, the CTEL model outperforms conventional approaches, as illustrated in Figure 3.

To further validate the robustness of the proposed CTEL framework under sparse data conditions, a sparsity sensitivity analysis was conducted. The MovieLens 100K dataset was randomly subsampled to simulate data densities of 100%, 60%, 40%, and 20%. The CTEL model was compared against a conventional static blending ensemble similar to SSEF [40] and individual base learners.

As shown in Table 7, CTEL maintains significantly lower RMSE across all sparsity levels.When the data density is reduced from 100% to 20%, the RMSE of static blending increases by 0.09, whereas the RMSE of CTEL increases only by 0.06. This indicates that CTEL effectively handles data sparsity by utilizing high-confidence pseudo-labels and complementary model views through co-training and stacking.

5. Conclusion

This study introduces the CTEL (Collaborative filtering with Tuning Ensemble Learning) model, demonstrating the effectiveness of combining semi-supervised learning and ensemble techniques to enhance collaborative filtering in recommendation systems. Our approach addresses key challenges such as data sparsity, scalability, and generalizability—issues that often hinder the performance of traditional recommendation models. Through extensive testing on the MovieLens and FilmTrust datasets, we observe significant improvements in standard evaluation metrics such as RMSE, MSE, and MAE, suggesting that the CTEL model is not only effective in handling sparse data but also maintains robustness and accuracy across diverse datasets. When compared to other state-of-the-art models in the literature, such as those employing deep learning for collaborative filtering (e.g., Neural Collaborative Filtering (NCF)) or those using matrix factorization-based methods (e.g., ALS or SVD++), our CTEL model stands out for its novel integration of semi-supervised learning to generate more reliable pseudo-labels for unlabeled data and the application of ensemble learning to combine multiple weak learners into a more robust and scalable system. While previous studies have explored the use of semi-supervised techniques to improve recommendation quality, few have simultaneously addressed the challenges of ensemble methods in this domain, with a focus on scalability and real-time adaptability to new data. Although future work may explore neural meta-learners to evaluate deeper feature interactions, the current linear approach aligns with the goals of interpretability, reproducibility, and stability in sparse recommendation scenarios.

The model's ability to handle sparse user-item interactions and its scalability are particularly noteworthy. Compared with traditional methods, which may struggle with data sparsity and overfitting, our CTEL model demonstrates superior performance in predicting user preferences even in the presence of missing data, making it more suitable for large-scale real-world applications where data is often incomplete or noisy. Moreover, by incorporating ensemble learning, our approach is more flexible and less prone to overfitting, a common challenge with complex models that rely on limited labeled data. While the results are promising, several avenues remain for future work. To further enhance the performance of the CTEL model, future research could explore more sophisticated feature engineering techniques, such as integrating contextual features (e.g., temporal, geographic, or demographic data) or multimodal data (e.g., combining text and image data from movie descriptions or user reviews). Additionally, the model's effectiveness could be evaluated on a broader set of datasets across different domains, including e-commerce, healthcare, and social media platforms, to assess its versatility and adaptability. Lastly, further exploration of active learning strategies could help minimize reliance on labeled data, making the model even more efficient in environments with limited labeled data. Thus, our CTEL model represents a promising step forward in the development of scalable, accurate, and robust recommendation systems. By combining semi-supervised learning and ensemble methods, it offers a novel solution to the persistent challenges of data sparsity, overfitting, and scalability in collaborative filtering. Its comparative advantage lies in its ability to effectively combine weak learners while utilizing unlabeled data, offering valuable insights for both academia and industry in the quest for more precise and trustworthy recommendation systems.

AUTHORS

Nisha Sharma – Assam Down Town University, Panikhaiti, Guwahati, Assam, India, e-mail: nisha.gu24@gmail.com.

Mala Dutta* – Assam Down Town University, Panikhaiti, Guwahati, Assam, India, e-mail: mala.dutta@adtu.in.

*Corresponding author

References

[1] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa, "Leveraging Bayesian deep learning and ensemble models for improved uncertainty quantification in recommendations," *Heliyon*, vol. 10, no. 2, 2024, e24188.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, 2005, pp. 734–749.

[3] C. C. Aggarwal, *Recommender Systems: The Textbook* (2nd ed.), Springer, 2023.

[4] J. Bobadilla, F. Ortega, and A. Hernando, "Collaborative filtering techniques: New perspectives and future directions," *Information Fusion*, vol. 92, 2023, pp. 245–261.

[5] R. Cheraghi, A. M. Mahfoozi, S. Zolfaghari, M. Shabani, M. Ramezani, and H. R. Rabiee, "Epistemic uncertainty-aware recommendation systems via Bayesian deep ensemble learning," *arXiv* preprint, 2025, arXiv:2504.10753.

[6] V. Coscrato and D. Bridge, (202. "Recommendation uncertainty in implicit feedback recommender systems," *Communications in Computer and Information Science (AICS 2022)*, Springer, 2023.

[7] X. He, L. Liao, H. Zhang, and T. S. Chua, "Neural collaborative filtering revisited: Deep interaction modeling in recommendation systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, 2024, pp. 410–423.

[8] N. Knyazev and H. Oosterhuis, "A lightweight method for modeling confidence in recommendations with learned Beta distributions," *arXiv* preprint, 2023, arXiv:2308.03186.

[9] Y. Koren, R. Bell, and C. Volinsky, C, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, 2009, pp. 30–37.

[10] D. D. Lee, and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, 2001, pp. 556–562.

[11] D. Liang, R. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," *arXiv* preprint, 2018, arXiv:1802.05814.

[12] T. Liu, C. Gao, Z. Wang, D. Li, J. Hao, D. Jin, and Y. Li, "Uncertainty-aware consistency learning for cold-start item recommendation," *arXiv* preprint, 2023, arXiv:2308.03470.

[13] X. Ning, X. Wang, S. Xu, W. Cai, L. Zhang, L. Yu, and W. Li, "A review of research on co-training," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 18, 2021, e6276.

[14] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," *arXiv* preprint, 2020, arXiv:2005.09683.

[15] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook* (3rd ed.), Springer, 2022.

[16] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 880–887.

[17] H. Wang, Z. Sun, Y. Du, L. Zhang, T. He, and Y.-S. Ong, "Uncertain multi-objective recommendation via orthogonal meta-learning enhanced Bayesian optimization," *arXiv* preprint, 2025, arXiv:2502.13180.

[18] S. Wang and Y. Zhang, "Hybrid collaborative filtering via content–context fusion: A comprehensive review," *Expert Systems with Applications*, vol. 228, 2023, 120979.

[19] W. Xu, J. Zheng, J. Lin, M. Han, and J. Du, "Unified representation learning for multi-intent diversity and behavioral uncertainty in recommender systems," *arXiv* preprint, 2025, arXiv:2509.04694.

[20] Z. Xu, Z. Yang, Z. Guo, S. Liu, L. Lin, X. Liu, Y. Liu, and H. Li, "UMRE: A unified monotonic transformation for ranking ensemble in recommender systems," *arXiv* preprint, 2025, arXiv:2508.07613.

[21] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, 2017, pp. 1–38.

[22] S. Zhang, K. Liu, Z. Yu, B. Feng, and Z. Ou, "Hybrid recommendation system combining collaborative filtering and content-based recommendation with keyword extraction," *Applied and Computational Engineering*, vol. 2, 2023, pp. 927–939.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, 2014, pp. 1929–1958.

[24] J. Dean, G. Corrado, R. Monga, et al., "Large-scale distributed deep networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[25] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Neural Networks Deep Learning*, 2015, arXiv:1503.02531.

[26] G. Brown, "Ensemble learning," *Encyclopedia of Machine Learning*, Springer, 2011.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[28] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.

[29] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 253–260.

[30] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, 1996, pp. 69–101.

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[32] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] L. H. Gilpin, D. Bau, B. Z. Yuan, et al., "Explaining explanations: An overview of interpretability of machine learning," *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[34] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, 2006, pp. 265–284.

[35] S. Goldwasser, S. Micali, and R. Rivest, "Secure multi-party computation without agreement," *Journal of Cryptology*, vol. 18, 2005, pp. 247–287.

[36] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, Springer, 2017.

[37] Roy D, Dutta M. A systematic review and research perspective on recommender systems. Journal of Big Data. 2022;9:59. doi: 10.1186/s40537-022-00592-5.

[38] P. Datta, "MovieLens 100k dataset," *Kaggle*, 2016. Available: https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset

[39] G. Guo, "Film Trust dataset," *LibRec*, 2011. Available: https://guoguibing.github.io/librec/datasets.html

[40] J. Wu, X. Sang, and W. Cui, "Semisupervised collaborative filtering ensemble," *World Wide Web*, 37 vol. 24, 2021, pp. 19411960.

[41] Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. 2008:263-272. doi:10.1109/ICDM.2008.22.

[42] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, 2008, pp. 1257–1264.

[43] S. Rendle, "Factorization machines with libFM," *ACM Transactions Intelligent Systems and Technology*, vol. 3, 2012, pp. 1–22.

[44] H. Wang, N. Wang, and D. Yeung, "Collaborative deep learning for recommender systems," *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1235–1244.

[45] Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation. ACM Computing Surveys. 2014;46(4):Article 44. doi:10.1145/2523813.

[46] Ben-Nun T, Hoefler T. Demystifying parallel and distributed deep learning: an in-depth concurrency analysis. ACM Computing Surveys. 2019;52(4):65:1-65:43. doi:10.1145/3320060.

[47] Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence. IEEE Access. 2018;6:52138-52160. doi:10.1109/ACCESS.2018.2870052.

[48] Zhang Y, Chen X. Explainable recommendation: a survey and new perspectives. Foundations and Trends in Information Retrieval. 2020;14(1): 1-101. doi:10.1561/1500000066.

Implementation of Sand cat Swarm Optimization for Uniform T-Way Test Suite Generation

Submitted: 10th June 2025; accepted: 24th November 2025

Muhammad Aiman bin Mohd Asyraf, Rozmie Razif Bin Othman, Mohd Zamri Bin Zahir Ahmad, Ahmad Ashraf Abdul Halim, Kentaro Go, Nuraminah binti Ramli, R. Badlishah Ahmad, Latifah Munirah Kamarudin, Murad Muhammad Hasan Salih Al-Walidi

DOI: 10.14313/jamris-2026-028

Abstract:
T-way combinatorial testing is an essential approach for optimizing test suite generation by systematically covering parameter interactions while minimizing test cases. Various metaheuristic strategies have been introduced to improve test suite generation, with an increasing focus on balancing exploration and exploitation for efficient test selection. This study investigates the sand cat swarm optimization (SCSO) algorithm as a metaheuristic strategy for t-way test suite generation. Inspired by the hunting behavior of sand cats, SCSO dynamically adjusts sensitivity factors to improve test suite generation efficiency. To evaluate SCSO's performance, 30 benchmark experiments were conducted across four groups of t-way configurations, with t varying from 2 to 6 and v ranging from 2 to 10. Each configuration was executed five times, and the smallest test suite size was selected for analysis. Experimental results demonstrate that SCSO outperforms 15.79% of competing strategies, achieves comparable performance in 42.11% of cases, and is outperformed in 42.11% of benchmark comparisons. These findings highlight SCSO's capability of generating competitive test suites, particularly in t-way interaction testing. The statistical evaluations, including Wilcoxon Rank and Friedman Mean Rank tests, further validate SCSO's performance in comparison to other metaheuristic approaches. Although SCSO effectively reduces test suite size while maintaining interaction coverage, further enhancements are necessary to improve its adaptability and computational efficiency across diverse configurations. Future work should focus on refining SCSO's exploration mechanisms to optimize search efficiency and extend its applicability in combinatorial test generation.

Keywords: *T-way combinatorial testing, test suite generation, sand cat swarm optimization (SCSO), metaheuristic algorithm*

1. Introduction

Software systems are deeply embedded in various domains, including finance, health care, aviation, and critical infrastructure, making their reliability a fundamental concern. Demand for software applications has raised the bar for developed software quality assurance [1]. Software products must be dependable and high quality to fulfill every customer's expectations and requirements [2]. A single software failure can lead to catastrophic consequences, ranging from financial losses and security breaches to safety risks and operational disruptions. Software testing serves as a quality assurance mechanism, systematically identifying defects and vulnerabilities that could compromise system integrity. With an estimated 40% of total development expenses, software testing is an especially expensive portion of the software development process [3]. However, as modern software applications become increasingly complex, traditional method or testing approaches face significant challenges in achieving comprehensive validation within practical time and resource constraints [4].

Ensuring software reliability requires efficient test design strategies that balance coverage and efficiency. Interaction-related faults are among the crucial errors that must be disclosed because any breakdown in these interactions would impact main functionalities of the system [5]. Exhaustive testing is impractical due to the exponential growth of test cases, making combinatorial testing a preferred approach for systematically covering parameter interactions because selecting at least one test case must be covered for each t-way (t is the interaction strength) combination of input parameters of a configuration system that cover all t-way interactions among input parameters [6, 7]. The process of combinatorial testing involves defining the input space (sampling the input configuration), constructing optimized test cases that ensure interaction coverage, and executing the test suite to detect faults [8].

Generating optimal combinatorial test suites remains computationally challenging, making difficult combinatorial testing a challenge and a significant research limitation [9]. Metaheuristic algorithms, such as genetic algorithms (GAs), particle swarm optimization (PSO), ant colony optimization (ACO), and whale optimization algorithm (WOAs) have been applied to optimize test suite size and address the minimum covering array generation (MCAG) problem [10]. However, the nondeterministic polynomial (NP) hard nature of test suite optimization, where no single strategy can guarantee that it always produces the best test suite size for all configurations, presents challenges for existing algorithms, creating opportunities for

new metaheuristic approaches to improve efficiency and coverage [11]. Current methods struggle with selective pressure, premature convergence, inefficient exploration, and high computational costs, limiting their effectiveness in large-scale configurations [12].

Nonetheless, according to the no free lunch (NFL) theorem, no single optimization algorithm can solve all optimization domains [13]. This suggests that an algorithm's performance is highly dependent on the nature of the problem, making the integration of metaheuristic algorithms into specific applications, such as t-way combinatorial test generation, an ongoing research challenge. Motivated by this, the present study explores the application of SCSO in t-way testing.

SCSO is a recently introduced metaheuristic inspired by the adaptive hunting behavior of sand cats, which attack or search for prey according to the sound frequency [14]. Unlike traditional swarm-based optimization techniques, SCSO incorporates a stochastic movement strategy that dynamically adjusts its search behavior, effectively balancing exploration and exploitation. It has demonstrated strong performance in finding optimal solutions with fewer parameters and reduced computational overhead [15]. This adaptability makes it particularly suitable for optimization problems that require efficient navigation of large search spaces. While SCSO has been successfully applied to various optimization domains, its potential for combinatorial test suite generation remains unexplored. Given its unique search mechanism, SCSO could serve as an alternative approach for minimizing test suite size while ensuring comprehensive parameter interaction coverage.

Motivated by its reported advantages, this study aims to evaluate the performance of SCSO in t-way combinatorial test suite generation and compare it against existing metaheuristic-based approaches. The remainder of this paper is structured as follows. Section 2 presents a comprehensive review of combinatorial testing strategies and metaheuristic optimization techniques. Section 3 details the methodology, including the adaptation of SCSO for test suite generation. Section 4 discusses the experimental setup, data sets, and evaluation criteria. Section 5 presents the results and analysis, followed by Section 6, which concludes the study and outlines potential future research directions.

2. Combinatorial T-way Testing Background

Software testing plays a crucial role in ensuring the quality and reliability of modern software systems [16]. As applications become increasingly complex, the number of configurable parameters expands, leading to an exponential growth in possible input combinations. Exhaustive testing (a method that tests all combinations), while theoretically ensuring complete accuracy, becomes impractical due to excessive computational costs and time constraints, as the combinatorial explosion problem makes it unfeasible for a large number of input parameters [12]. To address this challenge, combinatorial testing has emerged as an effective approach, focusing on covering t-way interactions between input parameters, rather than testing every possible combination. This method provides a structured way that effectively reduces the number of test cases needed compared to exhaustive testing while maintaining adequate coverage of parameter interactions [17].

The core principle behind t-way testing is based on empirical observations that show most software failures are caused by the interaction of a small subset of input parameters rather than the entire input space that can reach an incorrect result [18]. Studies have shown that pairwise (2-way) testing can detect a large percentage of defects while increasing the interaction strength (t) enhances fault detection. By systematically covering all possible t-way interactions, combinatorial testing ensures that critical parameter interactions are tested, improving software quality without requiring an infeasible number of test cases. The effectiveness of this approach depends on selecting an appropriate t, where higher values provide better fault detection at the cost of increased test suite size.

Combinatorial test suite generation relies on the construction of covering arrays (CAs), which are mathematical structures that ensure that all t-way combinations appear in at least one test case. A covering array is represented as CA(N; t, k, v), where N denotes the number of test cases, t is the interaction strength, k is the number of parameters, and v represents the possible values each parameter can take [10]. The goal of test suite generation is to minimize N while maintaining full t-way coverage, as a smaller test suite reduces execution time and testing costs. Constructing an optimal CA array is computationally challenging, requiring efficient strategies to generate minimal test suites while maintaining complete interaction coverage.

Exhaustive testing involves evaluating every possible combination of input parameters, which quickly becomes infeasible due to the combinatorial explosion. T-way testing can cover all of the important interaction components at least once, and the size of the test suite is reduced in proportion to the interaction strength, "t" [19]. It can improve the effectiveness of software testing from different configuration systems while simultaneously lowering the anticipated cost and time [20]. It also makes combinatorial testing highly efficient, especially for large-scale software systems where time and cost are major concerns. It allows teams to detect potential issues early, optimize resources, and accelerate product delivery without getting stuck in a never-ending cycle of exhaustive testing [21].

Combinatorial testing can be categorized into two variations based on interaction strength: uniform interaction strength and variable interaction strength [10]. Uniform strength interaction means that all input parameters share the same level of interaction where the CA maintains a consistent interaction strength across all parameters, ensuring uniform

test coverage [22]. Variable strength alone guarantees and has more than one interaction strength for generating test cases [22]. It assigns different t-way levels to parameters based on system criticality. This approach optimizes test coverage without unnecessarily increasing test suite size [10].

Metaheuristic algorithms have gained significant attention in combinatorial testing due to their ability to explore large solution spaces and optimize test suite size while ensuring full coverage. It is because it has capable to escape from local optima and perform a robust search of a search space [23]. Algorithms such as GA, ACO, and WOA leverage heuristic-based search techniques are able to find near-optimal test suites under several conditions which are balancing exploration (global search) and exploitation (local refinement) to minimize the number of test cases. These methods have proven particularly useful in large-scale and highly configurable software systems, whereas traditional techniques struggle to provide efficient solutions.

As combinatorial testing continues to evolve, its integration with optimization techniques remains a key area of research. The challenge of generating optimal test suites efficiently requires a balance between computational feasibility and maximizing interaction coverage. Metaheuristic approaches provide promising solutions to this problem by leveraging intelligent search mechanisms to minimize test suite size while maintaining high defect detection efficiency. It can exploit stochastic behavior to search for the best test cases through only several iterations [17]. With the growing complexity of modern software systems, combinatorial testing is becoming an indispensable methodology in ensuring robust and efficient software validation.

2.1. Problem Definition Model

Figure 1 illustrates the concept of t-way testing using an "online learning system" application. This system filters learning content based on five parameters: user type, device used, level course difficulty, content format, and interaction mode. The user type is categorized into two groups: student and teacher. The device used is classified as mobile phone or laptop, while the course difficulty falls into three levels: beginner, intermediate, and advanced. The content format is video and text, and the interaction mode is categorized as self-paced, live, and hybrid. To simplify the explanation, each parameter is assigned a distinct notation (e.g., U1 and U2 for user type, D1 and D2 for device used, L1, L2, and L3 for level course difficulty). Table 1 provides a simplified representation of the parameters and their possible values for this system.

To test such a system, exhaustive testing, which involves generating all possible combinations of parameter interactions to achieve complete coverage, could be applied. For the given system, full-strength interaction testing (where t = 5) results in 72 test cases (2 × 3 × 2 × 3 × 2 = 72). While exhaustive testing theoretically ensures complete accuracy,

Figure 1. Filter for online learning system

Table 1. Classification Representation of Input Parameters

Input parameter	U	L	D	M	C
Possible Value	U1	L1	D1	M1	C1
	U2	L2	D2	M2	C2
		L3		M3	

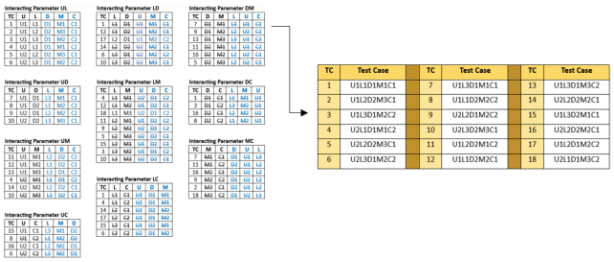
TC	Test Case	TC	Test Case	TC	Test Case	TC	Test Case
1	U1L1D1M1C1	19	U1L2D2M1C1	37	U2L1D1M1C1	55	U2L2D2M1C1
2	U1L1D1M1C2	20	U1L2D2M1C2	38	U2L1D1M1C2	56	U2L2D2M1C2
3	U1L1D1M2C1	21	U1L2D2M2C1	39	U2L1D1M2C1	57	U2L2D2M2C1
4	U1L1D1M2C2	22	U1L2D2M2C2	40	U2L1D1M2C2	58	U2L2D2M2C2
5	U1L1D1M3C1	23	U1L2D2M3C1	41	U2L1D1M3C1	59	U2L2D2M3C1
6	U1L1D1M3C2	24	U1L2D2M3C2	42	U2L1D1M3C2	60	U2L2D2M3C2
7	U1L1D2M1C1	25	U1L3D1M1C1	43	U2L1D2M1C1	61	U2L3D1M1C1
8	U1L1D2M1C2	26	U1L3D1M1C2	44	U2L1D2M1C2	62	U2L3D1M1C2
9	U1L1D2M2C1	27	U1L3D1M2C1	45	U2L1D2M2C1	63	U2L3D1M2C1
10	U1L1D2M2C2	28	U1L3D1M2C2	46	U2L1D2M2C2	64	U2L3D1M2C2
11	U1L1D2M3C1	29	U1L3D1M3C1	47	U2L1D2M3C1	65	U2L3D1M3C1
12	U1L1D2M3C2	30	U1L3D1M3C2	48	U2L1D2M3C2	66	U2L3D1M3C2
13	U1L2D1M1C1	31	U1L3D2M1C1	49	U2L2D1M1C1	67	U2L3D2M1C1
14	U1L2D1M1C2	32	U1L3D2M1C2	50	U2L2D1M1C2	68	U2L3D2M1C2
15	U1L2D1M2C1	33	U1L3D2M2C1	51	U2L2D1M2C1	69	U2L3D2M2C1
16	U1L2D1M2C2	34	U1L3D2M2C2	52	U2L2D1M2C2	70	U2L3D2M2C2
17	U1L2D1M3C1	35	U1L3D2M3C1	53	U2L2D1M3C1	71	U2L3D2M3C1
18	U1L2D1M3C2	36	U1L3D2M3C2	54	U2L2D1M3C2	72	U2L3D2M3C2

Figure 2. List of generated test cases for exhaustive testing

it becomes impractical when dealing with a large number of input parameters due to the combinatorial explosion problem [12]. Figure 2 shows test cases for exhaustive testing based on the online learning system application.

To address this issue, the number of test cases can be significantly reduced by applying t-way testing with a lower interaction strength (t). For instance, pairwise testing (t = 2) ensures that all pairs of parameter interactions are tested, thereby reducing the number of test cases while maintaining sufficient coverage. T-way testing focuses on testing parameter interactions up to a defined strength (t). For example, 2-way testing ensures that all pairwise interactions are covered, significantly reducing the number of test cases compared to exhaustive testing. Techniques such as uniform strength (pairwise) testing provide flexibility in prioritizing critical interactions.

In pairwise testing (t = 2), only specific pairs of parameters are tested for interactions, while the remaining noninteracting parameters are assigned random valid values (denoted as DX) to form complete test cases. For the given system, the interacting pairs include UL, UD, UM, UC, LD, LM, LC, DM, DC, and MC. Figure 2 demonstrates the construction of test cases using t-way testing with t= 2. Any duplicate test



TC	Test Case	TC	Test Case	TC	Test Case
1	U1L1D1M1C1	7	U1L3D1M1C1	13	U1L3D1M3C2
2	U1L2D2M3C1	8	U1L1D2M2C2	14	U2L2D2M2C1
3	U1L3D1M2C2	9	U2L2D1M2C2	15	U1L3D2M1C1
4	U2L1D1M1C2	10	U2L3D2M3C1	16	U2L2D2M2C1
5	U2L2D2M3C1	11	U1L2D2M1C2	17	U1L2D1M2C2
6	U2L3D1M2C2	12	U1L1D2M2C1	18	U2L1D1M3C2

Figure 3. Test case generation for t = 2 and list of test cases for t-way

cases are removed, and only unique combinations are retained as final test cases.

As shown in Figure 3, t-way testing significantly reduces the number of test cases from 72 (in exhaustive testing) to 18, achieving a reduction of over 75%. Despite this reduction, the test cases maintain adequate coverage of parameter interactions, ensuring effective fault detection. This approach not only reduces the number of test cases but also minimizes resource consumption, such as time and cost, making the testing process more efficient.

3. Related Work

Research in t-way combinatorial testing has significantly evolved, transitioning from traditional computational approaches to modern metaheuristic methods. Computational approaches, rooted in algebraic techniques, have been fundamental in t-way test suite generation. These methods employ heuristic rules and greedy strategies to cover uncovered test combinations [24]. However, while computational methods offer flexibility and support for complex configurations, they struggle with efficiency, as longer computational times are required to handle extensive test combinations [25]. This limitation highlights the need for more scalable solutions. To address these challenges, metaheuristic algorithms have emerged as an effective alternative for t-way test suite generation. Metaheuristics typically begin with a random solution and iteratively apply search techniques to improve fitness. Though not perfectly accurate, these methods efficiently produce near-optimal solutions in less execution time than computational methods. Metaheuristic algorithms enable a more robust and adaptive learning process, ultimately enhancing the network's ability to solve various challenging problems involving t-way interactions [26].

Within this context, t-way testing strategies are broadly categorized into two fundamental approaches: one-test-at-a-time (OTAT) and one-parameter-at-a-time (OPAT). OTAT starts with an empty test suite, where test cases are added one by one until all interactions are covered. Whenever a test case is chosen, it is included in the final suite (vertical extension). OTAT strategies have been extensively explored in research due to their ease of implementation and ability to generate effective test suites efficiently [19]. On the other hand, OPAT begins with an initial test suite and progressively adds parameters one at a time until all parameters have been incorporated. This method is referred to as horizontal extension, where test cases evolve gradually with additional parameters. After completing the horizontal extension, further test cases may need to be added through vertical extension to ensure full interaction coverage. OPAT provides an alternative approach to test suite generation but requires more complex handling of parameter dependencies, making it less commonly adopted in research [19].

Due to its structured nature and incremental test case construction, OTAT is often considered a more convenient and practical approach for t-way testing. Its systematic process simplifies test generation and reduces computational complexity, making it easier to implement compared to OPAT. As a result, OTAT strategies have gained more attention as one of the most promising research areas in t-way combinatorial testing [10].

Among the early computational methods developed for t-way interaction testing was the Test Configuration Generator (TConfig), introduced by A. W. Williams in the late 1990s [27] for web-based interaction testing. It is classified as a computational approach, since it relies on systematic algorithms. TConfig follows the OPAT approach, constructing test configurations in a structured, step-by-step manner rather than generating all possible test combinations at once. The tool primarily employs combinatorial testing techniques, specifically using CAs to ensure that every pair-wise (or higher-order) combination of parameter values is represented efficiently in the test set. It leverages the recursive block algorithm (Williams) and the in-parameter-order (IPO) greedy algorithm (Lei and Tai) to construct these test configurations, focusing on minimizing test cases while maintaining high interaction coverage.

In 1998, Lei and Tai introduced the IPO strategy [28], which was a groundbreaking computational method for pairwise interactions. It also applies the OPAT approach. Enhancements such as in-parameter-order-general (IPOG) and IPOG-D extended its capabilities to support variable-strength testing up to t = 6, optimizing test suite generation through horizontal and vertical extensions [29, 30]. Subsequent variants like IPOG-F2 and SCIPOG integrated lightweight heuristics and advanced constraint-handling techniques, ensuring efficient and precise test case generation for modern systems [20, 31].

At the beginning of the new decade, in 2010, the test vector generator (TVG), developed by Arshem, was introduced as a public domain tool for generating GUI tests using computational t-way interaction testing. It supports three interaction strength types: input-output relationship (IOR), variable strength, and uniform strength. Although it is claimed that TVG can support up to six strengths, practical execution has only achieved up to five [32]. TVG employs a greedy method and OTAT for test case generation and utilizes

three algorithms: t-reduced, plus-one, and random set. Among these, t-reduced produces the most optimized test suites, though limited details are available on the workings of each algorithm.

In contrast to these computational approaches, metaheuristic-based t-way interaction testing strategies have gained significant attention and have been widely explored in recent years.

Among the earliest breakthroughs, the particle swarm test generator (PSTG), introduced by Ahmed and Zamli [33] in 2010, utilized PSO as a metaheuristic method for generating t-way combinatorial test suites. Initially designed for uniform interaction strengths, PSTG was improved by the same authors in 2011 [34] to support variable-strength interaction as well. PSTG follows an OTAT approach, incrementally generating test cases to optimally cover interactions through particle swarm operations. PSTG effectively balances global search (exploration) and local search (exploitation) to produce compact test suites. Advantages include simpler implementation, fewer parameters to tune compared to other metaheuristics, and strong performance in generating smaller test sets. However, PSTG may experience higher computational overhead from repeated particle evaluations and typically requires careful parameter adjustments to consistently perform well across different test scenarios.

A year later, in 2011, the harmony search strategy (HSS) [35] was introduced by Alsewari and Zamli as a metaheuristic approach specifically designed for generating t-way test suites. It employs an OTAT method, incrementally building individual test cases to optimally cover interactions. Initially supporting only uniform interaction strengths, HSS was enhanced in 2012 to include support for variable-strength interactions. Its search mechanism balances exploration and exploitation effectively using harmony memory operations. The advantages of HSS include producing comparatively compact test suites, simple implementation, and minimal parameter tuning. Nevertheless, it may incur higher computational costs due to frequent harmony evaluations and can require careful parameter calibration to maintain optimal results across various testing configurations.

Subsequently, in 2015, the cuckoo search (CS) strategy for combinatorial test suite generation was introduced by Ahmed et al. [36]. It is a metaheuristic approach employing the OTAT method, where each iteration produces one complete test case optimized through a stochastic global search. CS leverages Lévy flights to efficiently balance exploration (global search) and exploitation (local search), allowing it to effectively generate combinatorial test suites. The strategy supports both uniform and mixed interaction strengths, making it suitable for variable-strength combinatorial testing. Key advantages include minimal parameter tuning, robustness in escaping local optima, and competitive performance with simpler tuning requirements compared to other metaheuristics like GA and PSO. Nevertheless, CS can incur higher computational overhead due to the iterative Lévy flight operations, particularly for complex or large-scale test generation scenarios, and it may still experience issues with convergence speed in some contexts.

In the same year, the swarm intelligent test generator (SITG) was introduced by Rabbi et al. [37]. The strategy applies PSO to generate t-way test suites by treating each particle as a complete test case rather than a numerical solution. It follows the OTAT approach, where test cases are added incrementally to maximize interaction coverage. The process begins with a randomly initialized swarm, where each particle represents a candidate test case. Instead of using traditional PSO velocity updates, SITG evaluates each test case based on its ability to cover previously uncovered t-way interactions, adjusting positions accordingly. The best test cases are iteratively selected and added to the final test suite (FTS), ensuring that all required interactions are covered. SITG supports both uniform and variable-strength interactions, with testing conducted up to t = 6. The approach efficiently balances exploration and exploitation, leading to compact test suites that effectively cover interactions, especially for higher-strength scenarios ($t \geq 4$). However, it introduces computational overhead due to frequent velocity evaluations and requires careful parameter tuning to prevent premature convergence to suboptimal solutions. Despite these challenges, SITG remains an effective and adaptive strategy for combinatorial test case generation.

In 2016, the high-level hyper-heuristic (HHH) strategy using tabu search was introduced by Zamli et al. [38] for t-way combinational test suite generation. It employs the OTAT approach and explicitly supports both uniform and variable interaction strengths, with reported experimental validations up to t = 6. In its implementation, HHH iteratively generates each test case by dynamically selecting among four low-level metaheuristics: teaching learning-based optimization (TLBO), the global neighborhood algorithm (GNA), PSO, and CS. Differing from conventional single-method strategies, HHH utilizes tabu search to intelligently switch among these metaheuristics using adaptive operators for improvement, diversification, and intensification, effectively balancing exploration and exploitation. Advantages of HHH include flexibility in adapting its search strategy according to problem dynamics, reduced likelihood of becoming trapped in local optima, and improved test suite compactness. However, employing multiple metaheuristics through tabu search significantly increases algorithm complexity, computational overhead, and sensitivity to parameter tuning, potentially limiting its practical applicability without careful calibration.

Following this, in 2017, the adaptive teaching-learning-based optimization (ATLBO) method was introduced by Din and Zamli [39] to enhance the conventional TLBO by integrating fuzzy logic for adaptive combinational t-way test suite generation. ATLBO

applies an OTAT approach, incrementally constructing each test case through iterative optimization. In this implementation, candidate solutions represent individual test cases that evolve based on the TLBO mechanism, comprising a global search (teacher phase) and local search (learner phase). Unlike traditional TLBO, ATLBO dynamically selects between these two search phases using a Mamdani fuzzy inference system, which continuously evaluates solution quality, intensification, and diversification measures to decide the optimal search direction at each iteration. ATLBO explicitly supports uniform and variable interaction strengths, with experimental validation reported up to t = 6. Its main advantages over conventional TLBO include increased adaptability, improved convergence speed, and better robustness against premature convergence. Nevertheless, the fuzzy logic integration introduces additional complexity, computational overhead, and the need for carefully designed fuzzy inference rules, potentially complicating practical applications.

Building upon reinforcement learning principles, in 2018, the Q-learning sine cosine algorithm (QLSCA) was introduced by Zamli et al. [40] as a metaheuristic approach specifically tailored for generating combinational t-way test suites using an OTAT methodology. QLSCA constructs test suites by iteratively generating each individual test case through a dynamic search guided by reinforcement learning (Q-learning). In each iteration, candidate solutions are updated by adaptively selecting among four distinct search operations: sine search, cosine search, Lévy flight, and elitism. Unlike traditional SCA, where the search operations are determined through fixed parameters, QLSCA dynamically chooses these operations based on a learned Q-value, promoting balanced exploration and exploitation. The method explicitly supports uniform interaction strength and has been experimentally validated up to t = 4. Advantages of QLSCA include improved adaptability, reduced reliance on manually tuned parameters, and enhanced capability to avoid premature convergence compared to traditional SCA. However, incorporating the reinforcement learning mechanism introduces additional complexity, computational overhead, and increased sensitivity related to the learning parameters, which must be tuned carefully to maintain consistent performance.

In 2019, the artificial bee colony for variable strength (ABCVS) strategy was introduced by Alazzawi et al. as a metaheuristic approach derived from the artificial bee colony (ABC) algorithm for combinatorial t-way test suite generation [41]. ABCVS employs an OTAT approach and explicitly supports both uniform and variable interaction strengths, validated up to t = 6. Unlike conventional ABC, ABCVS enhances the balance between exploration and exploitation by dynamically adjusting the number of employed, onlooker, and scout bees based on test case coverage and diversity. This adaptive mechanism ensures optimal interaction coverage while minimizing test suite size. The advantages of ABCVS include improved optimization efficiency, high scalability, and better adaptability in handling complex t-way interactions. Additionally, its swarm intelligence approach helps prevent premature convergence while maintaining test diversity. However, ABCVS requires careful tuning of parameters such as colony size and search limits, and its iterative nature introduces computational overhead, which may impact execution time in large-scale test configurations.

Continuing this trend, in 2020, the ant colony optimization algorithm using fuzzy logic (ACOF) strategy was introduced by Ahmad et al. [42] as an advanced metaheuristic derived from Dorigo's original ACO. ACOF employs an OTAT approach and supports both uniform and variable interaction strengths, explicitly tested up to t = 6. Unlike traditional ACO, ACOF integrates a Mamdani fuzzy inference system to dynamically adjust two critical parameters: the pseudo-random proportional selection rule and the number of ants utilized per iteration. Specifically, fuzzy logic determines the optimal selection probability and dynamically allocates ant resources, enhancing exploration and exploitation balance. Advantages include generating compact test suites and significantly improving execution time compared to conventional ACO variants, due to its adaptive mechanisms. However, integrating fuzzy logic increases algorithm complexity, requires carefully designed fuzzy rules, and may add computational overhead during the inference process.

In the same year, WOA for t-way test suite generation was introduced by Hassan et al. [43]. This metaheuristic strategy employs the OTAT approach and explicitly supports both uniform and variable-strength interactions, with successful tests conducted for interaction strengths up to t = 6. The WOA method initializes multiple candidate solutions ("whales") within the combinational search space and iteratively updates their positions using whale-inspired mechanisms such as encircling prey (exploitation) and bubble-net attacking (exploration). This adaptive process helps achieve a balanced exploration-exploitation trade-off, enhancing diversity and efficiency in generating optimized test suites. However, the approach faces potential computational overhead due to iterative candidate updates and can experience excessive exploration, which may slow convergence.

Advancing further, in 2021, the gravitational search test generator (GSTG) strategy was introduced by Htay et al. [11], based on the gravitational search algorithm (GSA), which is a population-based metaheuristic inspired by Newtonian gravity. GSTG uses the OTAT approach, iteratively selecting the best test case by mimicking gravitational interactions between candidate solutions, with each candidate represented as a test case. The algorithm uniquely employs gravitational forces to guide less optimal solutions (lighter masses) toward optimal ones (heavier masses), thus effectively balancing exploration and exploitation. GSTG supports both uniform and

variable interaction strengths, explicitly tested up to t = 10. Its advantages include strong exploration capabilities, effective avoidance of local optima due to its gravity-inspired search mechanism, and competitive performance in generating optimal or near-optimal test suites. However, GSTG has disadvantages, such as higher computational complexity, especially with increasing interaction strength or parameters, and sensitivity to parameters like gravitational constants, requiring careful tuning for optimal results. These issues are consistent with typical challenges noted in the literature regarding GSAs.

More recently, in 2022, the improved particle swarm optimization (improved PSO) strategy for t-way test suite generation was introduced by Prasad et al. [44]. It is a metaheuristic method extending traditional PSO, specifically adapted for generating t-way combinatorial test cases. Improved PSO follows the OTAT approach, uniquely simplifying the particle-update mechanism by directly adjusting particle positions based on coverage of uncovered interactions. Unlike conventional PSO, it eliminates reliance on typical parameters such as inertia weight, acceleration coefficients, and complex velocity calculations, significantly reducing the need for extensive parameter tuning. The strategy supports both uniform and variable interaction strengths, explicitly tested up to t = 6. Its advantages include easier implementation, fewer parameters to tune, and notably improved performance and efficiency compared to conventional PSO. However, disadvantages include computational overhead due to iterative particle evaluations and sensitivity to parameter adjustments, potentially causing premature convergence and trapping in local optima, consistent with common challenges highlighted in PSO-related literature.

Most recently, in 2024, the wingsuit flying search (WFS) optimization algorithm for t-way test suite generation was introduced by Rose et al. [45] as a metaheuristic, parameter-free approach inspired by wingsuit flying. WFS employs a unique three-phase implementation: generating initial points via Halton sequences, adaptively adjusting neighborhood sizes, and progressively narrowing the search space through decreasing discretization steps. It follows the OTAT approach, iteratively building each test case. WFS supports both uniform and variable interaction strengths, explicitly tested up to interaction strength t = 10. Its main advantages include ease of use due to its parameter-free nature, efficient performance, and the capability of producing compact test suites without extensive parameter tuning. However, it tends to heavily emphasize exploitation in later stages of optimization, increasing the risk of convergence to local optima, especially in complex test configurations [46].

This continuous evolution from computational methods to modern metaheuristic-based strategies highlights the ongoing advancements in t-way interaction testing techniques, significantly improving efficiency and adaptability over time.

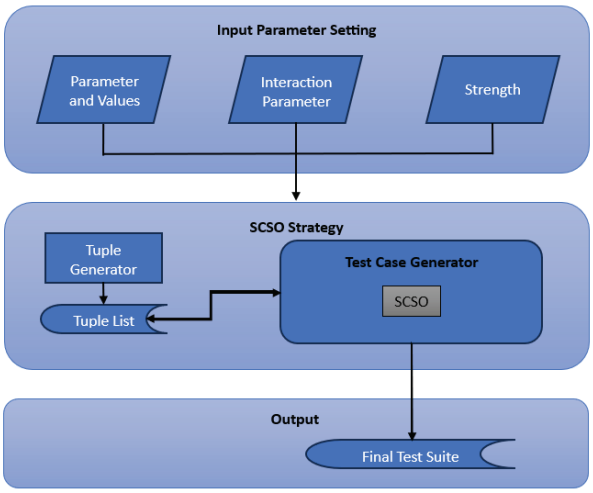


Figure 4. SCSO's framework

Figure 5. Exploration and exploitation of SCSO

4. Proposed Strategy of the Sand Cat Swarm Optimization (SCSO) Algorithm

Figure 4 shows the recommended framework for SCSO's approach. To achieve the SCSO method, a number of components are created and used, including the test case generator (TCG) and tuple generator (TG) where TG is derived from [24, 47]. The framework consists of three main sections. The first section, input parameter setting, defines the test parameters, their values, interaction parameters, and interaction strength, which serve as the foundation for generating test cases. The SCSO framework support only uniform strength, which are implemented using the OTAT test case generation approach. The second section, SCSO strategy, involves the TG, producing a tuple list (TL) representing possible parameter combinations. These tuples are then processed by the TCG, which incorporates the SCSO algorithm to generate optimized test cases. The final section, output, delivers the final test suite, ensuring that the generated test cases comprehensively cover the defined interactions.

SCSO is a bio-inspired optimization algorithm modeled after the hunting and survival strategies of sand cats, called "Felis margarita." The SCSO concept was first introduced by Seyyedabbasi and Kiani to solve optimization problems [15]. This novel approach combines unique behavioral patterns of sand cats to solve optimization problems efficiently. Exploration and exploitation phases based on the hunting behavior of sand cats is shown in Figure 5.

The process begins by initializing the population, determining the population size required for the SCSO to achieve the best outcome. Based on the requirements, a population size of 10 seems sufficient for the iterations, as it can yield the best possible results

efficiently. Furthermore, it will take a long time if the set is more than 10. The associated structure is a vector, and SCSO is thus a population-based technique. Each possible solution is determined by evaluating a predetermined fitness function. The SCSO will determine the optimal values of the parameters based on this function's definition of the problem's parameters.

4.1. Global Sensitivity Factor, $\overrightarrow{r_G}$

$$\overrightarrow{r_G} = SM - \left(\frac{SM \times iter_c}{iter_{Max}}\right) \quad (1)$$

The SCSO algorithm introduces a dynamic global sensitivity factor from Eq. (1) to balance exploration and exploitation effectively. The sensitivity is calculated, where SM = 2 inspired by sand cats' 2 kHz hearing ability. High initial sensitivity ensures broad exploration during early iterations, covering diverse regions of the search space. As iterations progress, the global sensitivity factor decreases gradually, shifting the algorithm's focus to exploitation by refining solutions. Toward the final iterations, it approaches zero, minimizing exploration and emphasizing intensification. This adaptive behavior mirrors sand cats' transitioning from scanning for prey to capturing targets, ensuring efficient navigation through the search space.

4.2. Sensitivity for Each Sand Cat

$$\vec{r} = \overrightarrow{r_G} \times rand(0,1) \quad (2)$$

Each sand cat's individual sensitivity ***r*** is dynamically adjusted using Eq. (2), where ***r*** scales the agent's responsiveness and rand(0,1) introduces randomness. This stochastic variability allows agents to explore diverse regions of the search space effectively while maintaining adaptability. The sensitivity mechanism ensures a balanced search dynamic, enabling agents to avoid local optima while still refining promising solutions. By mimicking sand cats' natural adaptability to environmental cues during hunting, this mechanism enhances the robustness and diversity of the optimization process.

4.3. Decision Parameter (R)

$$R = 2 \times \overrightarrow{r_G} \times rand\ (0,1) - \overrightarrow{r_G} \quad (3)$$

The decision parameter R from Eq. (3) determines whether agents explore new areas (R > 1) or exploit known solutions (R ≤ 1) during each iteration. It is calculated, where $\overrightarrow{r_G}$ is the global sensitivity factor.

If $R > 1$, the agent enters the exploration phase, where it moves to less-visited regions of the search space to promote diversity and avoid premature convergence. In this phase, the position of the agent is updated using the formula below.

$$\vec{r} \cdot (\overrightarrow{Pos_{bc}}(t) - rand\,(0,1) \cdot \overrightarrow{Pos_c}(t)) \quad (4)$$

Here, $\vec{r}$ represents the sensitivity factor that scales the agent's movement, ensuring its adjustments align with the global search dynamics. $\overrightarrow{Pos_{bc}}(t)$ refers to the position of the best candidate solution at iteration t, serving as a reference for guiding exploration. rand(0,1) introduces randomness to the movement, adding stochastic variability, while $\overrightarrow{Pos_c}(t)$ represents the current position of the agent. The subtraction $\overrightarrow{Pos_{bc}}(t) - rand\,(0,1) \cdot \overrightarrow{Pos_c}(t)$ creates a directional vector that leads the agent toward unexplored areas of the search space, with the random term ensuring unpredictable movement patterns. This calculated randomness enables the agent to explore a broader region effectively, increasing the probability of discovering better solutions while avoiding local optima. This formula encapsulates the essence of balancing guidance from the best solutions and randomness, making exploration robust and efficient in SCSO.

On the other hand, if $R \leq 1$, the agent enters the exploitation phase, which focuses on refining its current position by moving toward promising solutions. The agent's position is updated using Eq. (5).

$$\overrightarrow{Pos_b}(t) - \overrightarrow{Pos_{rnd}} cos\,(\theta) \cdot \vec{r} \quad (5)$$

The position update mechanism leverages both deterministic and stochastic components to guide the agent's movement during the search process. Here, $\overrightarrow{Pos_b}(t)$ represents the best-known position at iteration t, serving as a key anchor point for the agent to refine its search toward optimal solutions. $\overrightarrow{Pos_{rnd}}$ introduces an element of randomness by selecting a random position within the search space, ensuring variability in movement. The cosine function, *cos(θ)*, incorporates the angle θ, a randomly chosen value between 0° and 360°, to determine the direction of the agent's movement. This angle is further multiplied by $\vec{r}$, the sensitivity factor, which scales the movement magnitude. Together, these terms ensure that the agent's motion is both guided by the best solutions and randomized to maintain diversity and avoid convergence on suboptimal regions.

The random angle θ is selected using the roulette wheel selection algorithm, a probabilistic approach that enables dynamic and unbiased selection of movement directions. By randomly selecting θ, the agent's path is less predictable, reducing the likelihood of being trapped in local optima. The cosine function applied to θ introduces controlled oscillations in the movement trajectory, allowing the agent to explore more refined areas of the search space without losing diversity. R dynamically regulates the ratio of exploration to exploitation, making sure that the search agent both investigates novel possibilities and converges on the best answers. Achieving an ideal search procedure in optimization issues requires striking this balance.

Figure 6 describes the SCSO algorithm for selecting an optimal test suite based on test coverage. The process begins by initializing a population of search agents and calculating their fitness using a test coverage-based function. Parameters r and R are

Pseudocode for Sand Cat Swarm Optimization Algorithm

```
Input: Tuple List, TL;
Output: Final Test Suite, FTS
Begin
Initialize population of search agents
Calculate the fitness function based on test coverage
Initialize parameters r, R
While TL ≠ Ø {
 For each search agent {
  Get a random angle based on Roulette Wheel Selection (0° ≤ θ ≤ 360°)
   If (abs(R) ≤ 1) {
    Update the search agent position using Eq.5 (exploration phase)
   } Else {
    Update the search agent position using Eq.4 (exploitation phase)
   }
 }
 Evaluate fitness of all search agents
 Select the best test case candidate
 If best candidate improves coverage {
  Add candidate to FTS
  Remove covered interactions from TL
 }
 Update population based on best solution
}
End
```

Figure 6. Pseudocode for Sand Cat Swarm Optimization Algorithm

also initialized. The algorithm iterates while the TL is not empty. Within each iteration, every search agent selects a random angle using a roulette wheel selection mechanism. Based on the absolute value of R, the search agent updates its position using either the exploration equation [Eq. (5)] if $|R| \leq 1$ or the exploitation equation [Eq. (4)] otherwise. After updating positions, the algorithm evaluates the fitness of all search agents and selects the best test case candidate. If the selected candidate improves coverage, it is added to the FTS, and the covered interactions are removed from TL. The population is then updated based on the best solution. This process continues until all interactions in TL are covered, and the algorithm terminates, returning the optimized FTS.

5. Results and Discussion

The SCSO algorithm is developed and compiled using the Java programming language within the Eclipse 2024 (4.34.0) software. The running environment for the project is a desktop PC operating on Windows 11, equipped with a 2.19 GHz Intel® Core™ i7-12700F CPU and 32 GB of RAM. The three groups of experiments are as follows:

Group 1: CA(t, v^7), t varied from 2 to 6 and v varied from 2 to 5 based on [38, 43, 45, 47].

Group 2: CA(t, 2^{10}) based on [38, 44, 45, 47].

Group 3: CA(t, 3^7) based on [41, 44].

This project focuses solely on uniform strength configurations, as SCSO is a newly introduced approach to t-way testing. To evaluate its performance in generating test suite sizes for uniform strength, three groups of experiments were conducted, based on configurations derived from two separate journal papers. These experiments benchmarked the SCSO against results from various existing strategies discussed in Section 2. The experimental configurations were adapted from prior works by [38, 41, 43–45, 47], where the SCSO has been executed and benchmarked with those various experiments results. The results of these experiments are documented and summarized in Table 2, Table 3, and Table 4. Detailed experimental configurations for each group are presented accordingly.

The results of all experiments are presented in the provided tables, with the best test suite sizes highlighted in bold and cells are darkened for clarity. Cells marked as “X” indicate that results are unavailable or not reported in the respective articles. Based on the data in Table 2, SCSO delivers good competitive performance across all configurations of t and v.

The metaheuristic-based techniques generally outperform computation-based strategies in Table 2. Among these techniques, the HHH strategy demonstrated the highest performance, achieving the best results in 40% (8 out of 20) of the test cases. Following closely, ACOF secured 30% (6 out of 20), while QLSCA performed well with 25% (5 out of 20). Strategies such as GSTG, WOA, and HSS each accounted for 20% (4 out of 20) of the top results, whereas ATLBO followed with 15% (3 out of 20). Meanwhile, WFS, PSTG, CS, TConfig, and IPOG each contributed to 10%.

(2 out of 20) of the best-performing cases. However, SCSO, t-way test suite generation strategy based on ant colony algorithm (TTSGA), and TVG recorded

Table 2. Result Test Suite Size Performance for Group 1

CA(t,v^7)		Metaheuristic-based Strategies												Computational-based Strategies		
t	v	2026	2024	2023	2022	2020	2019	2018	2017	2016	2011	2010	2009	2010	2007	1990s
		SCSO	WFS	ACOF	GSTG	WOA	TTSGA	QLSCA	ATLBO	HHH	HSS	PSTG	CS	TVG	IPOG	TConfig
2	2	7	7	6	6	6	7	7	7	7	7	6	6	7	8	7
	3	15	15	15	15	14	15	15	15	14	14	15	15	15	17	15
	4	26	27	26	26	25	25	23	23	23	25	26	25	27	28	28
	5	40	40	37	40	36	38	34	34	35	35	37	37	42	42	40
3	2	14	13	12	12	12	12	15	15	15	12	13	12	15	19	16
	3	51	51	48	49	49	49	49	49	49	50	50	49	55	57	55
	4	124	123	117	121	116	118	112	111	112	121	116	117	134	208	112
	5	240	242	230	240	223	228	215	216	216	223	225	223	260	275	239
4	2	30	25	30	26	27	29	31	31	31	29	29	27	31	48	36
	3	158	156	148	155	152	152	149	151	148	155	155	155	167	185	166
	4	497	513	485	499	484	485	477	480	482	500	487	487	559	509	568
	5	1220	X	1174	1217	NA	1175	1150	1166	1153	1174	1176	1171	1385	1349	1320
5	2	52	51	53	52	52	55	X	X	58	53	53	53	59	128	56
	3	434	442	430	430	432	433	X	X	435	437	441	439	464	608	477
	4	1833	1861	1825	1822	1815	1821	X	X	1805	1831	1826	1845	2010	2560	1792
	5	5535	X	5450	X	X	5457	X	X	5413	5468	5474	5479	6257	8091	X
6	2	70	66	64	64	64	68	X	X	64	64	64	66	78	64	64
	3	949	955	970	971	945	957	X	X	853	916	977	973	1016	1281	921
	4	5621	5610	5450	5611	5567	5487	X	X	5478	4096	5599	5610	5978	4096	X
	5	21580	X	21145	X	X	21148	X	X	21107	21748	21595	21597	23218	28513	X
Total	***0***	***2***	***6***	***4***	***4***	***0***	***5***	***3***	***8***	***4***	***2***	***2***	***0***	***2***	***2***	
Optimal	***0%***	***10%***	***30%***	***20%***	***20%***	***0%***	***25%***	***15%***	***40%***	***20%***	***10%***	***10%***	***0%***	***10%***	***10%***	

Table 3. Result Test Suite Size Performance for Group 2

CA(t, 2^{10})	Metaheuristic-based Strategies											Computational-based Strategies		
	2026	2024	2023	2022	2022	2019	2016	2015	2011	2010	2009	2010	2007	1990s
t	SCSO	WFS	ACOF	GSTG	ImprovedPSO	TTSGA	HHH	SITG	HSS	PSTG	CS	TVG	IPOG	TConfig
2	8	8	8	8	9	8	8	9	7	8	8	10	10	9
3	17	16	16	16	16	16	16	16	16	17	16	17	19	20
4	34	38	39	27	39	36	36	44	37	37	36	41	49	45
5	84	84	74	74	84	76	79	87	81	82	79	84	128	95
6	159	160	153	156	168	155	153	174	158	158	157	168	352	183
Total	***0***	***1***	***3***	***3***	***1***	***1***	***2***	***1***	***2***	***0***	***1***	***0***	***0***	***0***
Optimal	***0%***	***20%***	***60%***	***60%***	***20%***	***20%***	***40%***	***20%***	***40%***	***0%***	***20%***	***0%***	***0%***	***0%***

the lowest performance, failing to achieve the best results in any of the test cases (0%).

The analysis of Table 3 reaffirms the dominance of ACOF and GSTG, both achieving the highest performance with 60% (3 out of 5) of the test cases, setting a strong benchmark. Meanwhile, HHH and HSS demonstrated balanced performance, each securing 40% (2 out of 5) of the best test cases. In contrast, WFS, CS, improved PSO, SITG, and TTSGA trailed behind, contributing to only 20% (1 out of 5) of the test cases. The lowest-performing strategies in this group were SCSO, PSTG, IPOG, TConfig, and TVG, all of which failed to secure any top-ranking test cases (0%).

Table 4 highlights the dominance of improved PSO, which achieves the highest performance with 80% (4 out of 5) of the test cases, outperforming all other strategies. Meanwhile, SITG, ABCVS, and TConfig each secure 20% (1 out of 5) of the test cases, demonstrating a more limited impact. SCSO, despite not leading the group, manages to contribute 40% (2 out of 5) of the best test cases, showcasing moderate performance and competitive potential. In contrast, IPOG fails to secure any top-ranking test cases (0%), making it the least effective strategy in this configuration. This pattern of results reinforces improved PSO's dominance, the balanced yet limited success of SITG and TConfig, and the moderate competitiveness of SCSO, while IPOG remains at the bottom, struggling to deliver optimal results.

SCSO demonstrates consistent competitiveness across multiple groups, showing its potential in diverse test scenarios. While it performs moderately and has yet to surpass the top-ranked metaheuristic strategies, its stability across configurations highlights its strength. This suggests that with further refinements or hybridization with other metaheuristic techniques, SCSO could achieve even greater optimization and emerge as a strong contender among the top-performing strategies.

The performance of SCSO has been evaluated using statistical analyses to assess its effectiveness in generating optimal test suite sizes for uniform strength interaction test. Two nonparametric tests, the Wilcoxon Rank test and the Friedman test, were employed due to the small sample size and nonnormal distribution of results. The Wilcoxon Rank test analyzed paired strategies with SCSO to determine statistically significant differences at a 95% confidence

Table 4. Result Test Suite Size Performance for Group 3

CA(t, 3^7)	Metaheuristic-based Strategies				Computational-based Strategies	
	2026	2022	2019	2015	2007	1990s
t	SCSO	Improved PSO	ABCVS	SITG	IPOG	TConfig
2	15	15	15	15	17	15
3	51	45	49	52	57	55
4	158	152	157	154	185	166
5	434	444	442	436	608	477
6	949	832	944	848	1281	921
Total	*2*	*4*	*1*	*1*	*0*	*1*
Optimal	*40%*	*80%*	*20%*	*20%*	*0%*	*20%*

level (α = 0.05). A null hypothesis was used, where a *p*-value ≤ α indicated significant differences, leading to the null hypothesis being rejected. Conversely, a *p*-value > α retained the null hypothesis, suggesting no significant difference. The Friedman test ranked strategies based on mean rank, with smaller values indicating better performance in generating test suite sizes when the null hypothesis was rejected.

A total of 30 results from three groups of experiments were analyzed across Table 5 to Table 6, although some configurations were excluded due to unavailable data for certain strategies. These analyses highlight SCSO's competitive performance compared to other strategies, particularly in scenarios where it demonstrated statistically significant advantages or achieved better mean rankings in test suite generation.

Table 5 presents the results of testing conducted using the Wilcoxon and Friedman tests, involving data from Groups 1 and 2. A total of 25 test types were analyzed, with the results offering insights into the comparative performance of different strategies. Table 5 also provides insights into whether the null hypothesis was retained or rejected for each paired strategy involving SCSO. For strategies like ACOF, TTSGA, HHH, HSS, and GSTG, the null hypothesis is rejected, meaning these strategies significantly outperform SCSO. For instance, the pairing with ACOF shows a $p = 0.002$, confirming a clear advantage for ACOF. Similarly, the null hypothesis is rejected when SCSO is compared to TVG, IPOG, WOA, QLSCA, and ATLBO, but in these cases, it is SCSO that demonstrates superior performance, as indicated by its lower mean rank.

In comparisons with PSTG, CS, TConfig, and WFS, the null hypothesis is retained, signifying no statistically significant differences. This implies that SCSO performs similarly to these strategies. For example, against PSTG ($p = 0.055$) and CS ($p = 0.083$), SCSO neither significantly outperforms nor underperforms, reflecting comparable performance levels.

When the null hypothesis is rejected, it means one strategy is statistically better than the other based on the test results. For SCSO, this occurs when it shows clear advantages over strategies like TVG, IPOG, and others, or when it is outperformed by stronger strategies like ACOF and TTSGA. On the other hand, retaining the null hypothesis, as seen in its comparisons with PSTG, CS, and WFS, suggests that SCSO is on par with these strategies, offering neither significantly better nor worse performance.

The majority of the paired strategies that outperformed SCSO, such as ACOF, TTSGA, HSS, improved PSO, and GSTG achieved better rankings primarily because they integrate additional techniques, such as hybridization with other optimization algorithms, rule-based enhancements, or ensemble approaches. These combinations allow them to improve search efficiency, balance exploration and exploitation, and adapt to problem-specific constraints more effectively. For example, ACOF benefits from the self-organizing behavior of ACO while incorporating fuzzy logic to handle uncertainty and improve decision-making. The fuzzy rules help refine the solution space more effectively, reducing randomness and increasing convergence speed, making ACOF superior to SCSO.

Another reason these strategies appear superior is that HHH inherently carries four different metaheuristics, which are TLBO, GNA, PSO, and CS, while SCSO operates as a standalone approach. HHH's adaptive mechanism selects the most suitable metaheuristic at different stages of optimization, ensuring a well-balanced search process. This multi-metaheuristic structure naturally provides an advantage in performance, as it allows HHH to adapt to different problem complexities dynamically. However, this also makes comparisons with SCSO unfair, as HHH benefits from the combined strengths of multiple algorithms, whereas SCSO is evaluated based on a single optimization framework.

ATLBO holds a significant advantage due to its adaptive teacher and learner phases, which dynamically adjust using a fuzzy inference system. Unlike SCSO, which follows a fixed optimization structure, ATLBO intelligently adapts its search strategy based on real-time conditions, allowing it to balance exploration and exploitation more effectively. The teacher phase drives global improvements by guiding learners toward better solutions, while the learner phase enhances local refinement through peer-to-peer learning. This adaptability not only improves solution quality but also reduces the risk of premature convergence. However, this advantage also makes direct comparisons with SCSO less fair, as ATLBO's superior flexibility comes from its additional fuzzy rule-based decision-making system, which SCSO does not incorporate. Essentially, ATLBO benefits from an

Table 5. Wilcoxon and Friedman Test for Group 1 and Group 2

No.	Paired Strategy	Test Statistic							Null Hypothesis	Conclusion
		Comparison for SCSO			Total Samples	*p*-value (Asymp. Sig. (2-tailed))	Friedman Mean Rank Test			
		<	=	>			Mean Rank	Rank		
1.	SCSO vs ACOF	18	4	3	25	0.002	SCSO – 5.76	7	reject	ACOF outperforms
							ACOF – 3.32	2		
2.	SCSO vs TTSGA	19	3	3		0.002	SCSO – 5.76	7	reject	TTSGA outperforms
							TTSGA – 3.88	3		
3.	SCSO vs HHH	18	2	5		0.01	SCSO – 5.76	7	reject	HHH outperforms
							HHH – 2.84	1		
4.	SCSO vs HSS	19	1	5		0.005	SCSO – 5.76	7	reject	HSS outperforms
							HSS – 3.92	4		
5.	SCSO vs PSTG	16	4	5		0.055	SCSO – 5.76	7	retain	no significant difference
							PSTG – 4.86	6		
6.	SCSO vs CS	17	2	6		0.083	SCSO – 5.76	7	retain	no significant difference
							CS – 4.16	5		
7.	SCSO vs TVG	0	4	21		0.001	SCSO – 5.76	7	reject	SCSO outperforms
							TVG – 7.92	8		
8.	SCSO vs IPOG	2	0	23		0.001	SCSO – 5.76	7	reject	SCSO outperforms
							IPOG – 8.34	9		
9.	SCSO vs WFS	8	6	8	22	0.532	SCSO – 2.27	3	retain	no significant difference
							WFS – 2.23	2		
10.	SCSO vs GSTG	14	6	2		0.013	SCSO – 2.27	3	reject	GSTG outperforms
							GSTG – 1.50	1		
11.	SCSO vs TConfig	14	3	5	22	0.099	SCSO – 1.30	1	retain	no significant difference
							TConfig – 1.70	2		
12.	SCSO vs WOA	16	1	0	17	0.001	SCSO – 1.97	2	reject	WOA outperforms
							WOA – 1.03	1		
13.	SCSO vs QLSCA	8	2	2	12	0.012	SCSO – 2.50	3	reject	QLSCA outperforms
							QLSCA – 1.63	1		
14.	SCSO vs ATBLO	8	2	2		0.012	SCSO – 2.50	3	reject	ATBLO outperforms
							ATBLO – 1.88	2		

Table 6. Wilcoxon and Friedman Test for Group 2 and Group 3

No.	Paired Strategy	Test Statistic							Null Hypothesis	Conclusion
		Comparison for SCSO			Total Samples	*p*-value (Asymp. Sig. (2-tailed))	Friedman Mean Rank Test			
		<	=	>			Mean Rank	Rank		
1	SCSO vs Improved PSO	4	2	4	10	0.944	SCSO – 2.00	2	retain	no significant difference
							PSO – 1.85	1		
2	SCSO vs TConfig	1	1	8		0.086	SCSO – 2.00	2	retain	no significant difference
							TConfig –3.75	4		
3	SCSO vs SITG	3	1	6		0.513	SCSO – 2.00	2	retain	no significant difference
							SITG – 2.50	3		
4	SCSO vs IPOG	0	0	10		0.005	SCSO – 2.00	2	reject	SCSO outperforms
							IPOG – 4.90	5		
5	SCSO vs ABCVS	1	1	3	5	0.715	SCSO – 1.70	2	retain	no significant difference
							ABCVS – 1.30	1		

embedded adaptive learning mechanism, making it more versatile yet computationally demanding compared to SCSO.

Table 6 presents the results of testing conducted using the Wilcoxon and Friedman tests, involving data from Groups 2 and 3 with total of 10 types of testing. The table presents a statistical analysis comparing the performance of SCSO against four other strategies: improved PSO, TConfig, SITG, and IPOG. The comparison is based on the Wilcoxon Rank test and the Friedman Mean Rank test. The results indicate that SCSO demonstrates comparable performance to improved PSO, TConfig, ABCVS, and SITG, as the *p*-values for these comparisons (0.944, 0.086, 0.715, and 0.513, respectively) are greater than the significance threshold of 0.05, leading to the retention of the null hypothesis. While SCSO shows slight advantages in certain samples, the Friedman Mean Rank test confirms that the differences are not statistically significant. However, when compared to IPOG, SCSO exhibits a clear statistical advantage, with a p-value of 0.005 leading to the rejection of the null hypothesis. SCSO outperforms IPOG in all samples and achieves a much better Friedman Mean Rank, solidifying its superiority in this comparison.

The analyses highlight that SCSO demonstrates a balanced performance, holding its ground against some strategies while excelling over weaker ones. SCSO shows comparable performance to Improved PSO, TConfig, SITG, PSTG, ABCVS, and CS, as indicated by retained null hypotheses and similar mean ranks. However, it decisively outperforms IPOG and TVG, with statistically significant results and superior Friedman Mean Ranks. Conversely, SCSO struggles against stronger strategies like ACOF, TTSGA, HHH, and HSS, which achieve better rankings in direct comparisons. Overall, SCSO proves to be a robust and competitive strategy, particularly effective against less dominant methods while maintaining reliability in varied contexts.

6. Conclusion

This paper presents the first implementation of SCSO in t-way testing for test suite generation. The results indicate that SCSO performs competitively, outperforming 15.79% of competing strategies, matching 42.11%, but being outperformed in 42.11% of cases. While SCSO shows strengths in handling uniform interaction strengths, it struggles against more advanced techniques such as ACOF, TTSGA, and WOA, highlighting its limitations in maintaining effective exploration throughout the search process.

SCSO mimics sand cat hunting behavior, where movement intensifies as the search progresses toward an optimal solution. However, as it nears the global optimum, its search radius contracts, leading to reduced exploration and increased reliance on exploitation. This transition increases the risk of premature convergence, causing SCSO to become trapped in local optima, particularly in complex search spaces. The benchmark results support this, as SCSO underperforms in nearly half of the cases, suggesting that an improved balance between exploration and exploitation is needed.

To enhance SCSO's effectiveness, future work should focus on improving its exploration capability. Hybridizing it with global optimization techniques such as simulated annealing, genetic algorithm, or Lévy flight could help mitigate premature convergence and improve search diversity. Additionally, extending its application to variable-strength and input–output relationship testing could enhance its adaptability. Exploring integration with other metaheuristic strategies may further refine its efficiency, making SCSO a more competitive approach for t-way test suite generation.

AUTHORS

Muhammad Aiman bin Mohd Asyraf* – Faculty of Intelligent Computing, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia, e-mail: aimanasyraf@studentmail.unimap.edu.my https://orcid.org/0009-0009-3268-0120.

Rozmie Razif Bin Othman – Centre of Excellence for Advanced Computing, (AdvComp), University Malaysia Perlis, Malaysia, e-mail: rozmie@unimap.edu.my https://orcid.org/0000-0001-7940-8487.

Mohd Zamri Bin Zahir Ahmad – Centre of Excellence for Advanced Computing, (AdvComp), University Malaysia Perlis, Malaysia, e-mail: zamrizahir@unimap.edu.my https://orcid.org/0000-0003-3839-2284.

Ahmad Ashraf Abdul Halim – Centre of Excellence for Advanced Computing, (AdvComp), University Malaysia Perlis, Malaysia, e-mail: ashrafhalim@unimap.edu.my https://orcid.org/0000-0002-6152-1964.

Kentaro Go – Department of Computer Science and Engineering, University of Yamanashi, Kofu, Japan, e-mail: go@yamanashi.ac.jp https://orcid.org/0000-0003-3451-7924.

Nuraminah binti Ramli – Centre of Excellence for Advanced Computing, (AdvComp), University Malaysia Perlis, Malaysia, e-mail: nuraminah@unimap.edu.my https://orcid.org/0000-0002-3527-2431.

R. Badlishah Ahmad – Centre of Excellence for Advanced Computing, (AdvComp), University Malaysia Perlis, Malaysia, e-mail: badli@unimap.edu.my https://orcid.org/0000-0002-4862-2728.

Latifah Munirah Kamarudin – Faculty of Intelligent Computing, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia, e-mail: latifahmunirah@unimap.edu.my https://orcid.org/0000-0002-2547-3934 .

Murad Muhammad Hasan Salih Al-Walidi – Faculty of Intelligent Computing, University Malaysia Perlis, 02600 Arau, Perlis, Malaysia, e-mail: muradmuhammad@studentmail.unimap.edu.my https://orcid.org/0009-0004-9672-689X.

*Corresponding author

ACKNOWLEDGEMENTS

The author would like to acknowledge the support from the Fundamental Research Grant Scheme (FRGS) under a grant number of FRGS/1/2024/ICT01/UNIMAP/02/1 from the Ministry of Higher Education Malaysia.

References

[1] N. Anwar and S. Kar, "Review Paper on Various Software Testing Techniques & Strategies," *Global Journal of Computer Science and Technology*, pp. 43–49, May 2019, doi: 10.34257/gjcstcvol19is2pg43.

[2] M. Bajjouk, M. Ehsan Rana, C. Reka Ramachandiran, and S. Chelliah, "Software testing for reliability and quality improvement," *Journal of Applied Technology and Innovation*, vol. 5, no. 2, p. 40, 2021, doi: 10.65136/jati.v5i1.216.

[3] S. Najihi, S. Elhadi, R. A. Abdelouahid, and A. Marzak, "Software Testing from an Agile and

Traditional View," *ScienceDirect Procedia Computer Science*, vol. 203, Aug. 2022, doi: 10.1016/j.procs.2022.07.116.

[4] M. Baqar and R. Khanda, "The Future of Software Testing: AI-Powered Test Case Generation and Validation," 2024.

[5] M. Z. Z. Ahmad, R. R. Othman, M. S. A. R. Ali, N. Ramli, M. W. Nasrudin, and A. A. A. Halim, "A Tuned Version of Ant Colony Optimization Algorithm (TACO) for Uniform Strength T-way Test Suite Generator: An Execution's Time Comparison," *J. Phys. Conf. Ser.*, vol. 1962, no. 1, 2021, doi: 10.1088/1742-6596/1962/1/012037.

[6] R. R. Othman, K. Zuhairi Zamli, and L. E. Nugroho, "General variable strength t-way strategy supporting flexible interactions," *Maejo International Journal of Science and Technology*, vol. 6, no. 03, pp. 415–429, 2012, doi: 10.14456/mijst.2012.30.

[7] A. A. Muazu, A. S. Hashim, and A. Sarlan, "Application and Adjustment of 'don't care' Values in t-way Testing Techniques for Generating an Optimal Test Suite," *Journal of Advances in Information Technology*, vol. 13, no. 4, pp. 347–357, 2022, doi: 10.12720/jait.13.4.347-357.

[8] B. S. Ahmed, "Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing," *Engineering Science and Technology, an International Journal*, vol. 19, no. 2, pp. 737–753, Jun. 2016, doi: 10.1016/j.jestch.2015.11.006.

[9] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "TWGH: A Tripartite Whale–Gray Wolf–Harmony Algorithm to Minimize Combinatorial Test Suite Problem," *Electronics (Switzerland)*, vol. 11, no. 18, Sep. 2022, doi: 10.3390/electronics11182885.

[10] N. Ramli, R. R. Othman, Z. I. Abdul Khalib, and M. Jusoh, "A Review on Recent T-way Combinatorial Testing Strategy," in *MATEC Web of Conferences*, EDP Sciences, Dec. 2017. doi: 10.1051/matecconf/201714001016.

[11] K. M. Htay, R. R. Othman, A. Amir, and J. M. H. Alkanaani, "Gravitational search algorithm based strategy for combinatorial t-way test suite generation," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 4860–4873, 2022, doi: 10.1016/j.jksuci.2021.06.020.

[12] E. Pira and M. Khodizadeh-Nahari, "Combinatorial t-way test suite generation using an improved asexual reproduction optimization algorithm," *Appl. Soft Comput.*, vol. 150, 2024, doi: 10.1016/j.asoc.2023.111070.

[13] M. Alshinwan et al., "Enhanced Prairie Dog Optimization with Differential Evolution for solving engineering design problems and network intrusion detection system," *Heliyon*, vol. 10, no. 17, Sep. 2024, doi: 10.1016/j.heliyon.2024.e36663.

[14] D. Wu, H. Rao, C. Wen, H. Jia, Q. Liu, and L. Abualigah, "Modified Sand Cat Swarm Optimization Algorithm for Solving Constrained Engineering Optimization Problems," *Mathematics*, vol. 10, no. 22, Nov. 2022, doi: 10.3390/math10224350.

[15] A. Seyyedabbasi and F. Kiani, "Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems," *Eng. Comput.*, vol. 39, no. 4, pp. 2627–2651, 2022, doi: 10.1007/s00366-022-01604-x.

[16] Ana Crudu and MoldStud Research Team, "Role of software testing in ensuring product quality," MoldStud. Accessed: Feb. 23, 2025. [Online]. Available: https://moldstud.com/articles/p-role-of-software-testing-in-ensuring-product-quality#:~:text=Software%20testing%20is%20important%20for,and%20effectiveness%20of%20software%20testing

[17] K. M. Htay, H. L. Zakaria, R. R. Othman, N. Ramli, and A. Amir, "A Pairwise T-Way Test Suite Generation Strategy Using Gravitational Search Algorithm," *ICAICST 2021 - 2021 International Conference on Artificial Intelligence and Computer Science Technology*, 2021.

[18] D. R. Kuhn, R. Bryce, F. Duan, L. S. Ghandehari, Y. Lei, and R. N. Kacker, "Combinatorial Testing: Theory and Practice," in *Advances in Computers no. 99*, 2015, ch. 1, p. 99. Accessed: Feb. 23, 2025. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=918448

[19] Muazu A. A., Hashim A. S., Audi U. I., Aliyu Y., and Yahaya M. S., "Combinatorial Interaction Testing for T-Way Test Case Generation: A Scoping Review of the Perspective Features," *Journal of China University of Mining and Technology*, no. 4, p. 121, 2024, doi: 10.1654/zkdx.2024.29.4-1.

[20] A. Aminu Muazu, A. Sobri Hashim, A. Sarlan, and M. Abdullahi, "SCIPOG: Seeding and constraint support in IPOG strategy for combinatorial t-way testing to generate optimum test cases," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 1, pp. 185–201, Jan. 2023, doi: 10.1016/j.jksuci.2022.11.010.

[21] N. F. Maidin, S. Hassan, S. Baharom, and A. B. Md. Sultan, "A Comparative Study on Testing Optimization Techniques with Combinatorial Interaction Testing for Optimizing Software Product Line Testing," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, no. 1, pp. 77–94, 2025, doi: https://doi.org/10.37934/araset.49.1.7794.

[22] J. M. Altmemi, R. R. Othman, and R. Ahmad, "Review of combinatorial testing strategy,"

International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 6, pp. 2877–2881, Nov. 2019, doi: 10.30534/ijatcse/2019/31862019.

[23] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E. G. Talbi, "Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art," Jan. 16, 2022, *Elsevier B.V.* doi: 10.1016/j.ejor.2021.04.032.

[24] N. Ramli, R. R. Othman, R. Hendradi, and I. Iszaidy, "T-way Test Suite Generation Strategy based on Ant Colony Algorithm to Support T-way Variable Strength," *J. Phys. Conf. Ser.*, vol. 1755, no. 10, 2020.

[25] A. B. Nasser, A. A. Alsewari, N. M. Tairan, and K. Z. Zamli, "Pairwise Test Data Generation Based On Flower Pollination Algorithm," *Malaysian Journal of Computer Science*, pp. 242–257, 2017, Accessed: Feb. 05, 2025. [Online]. Available: https://ejournal.um.edu.my/index.php/MJCS/article/view/7069/4715

[26] P. A. Kowalski, S. Kucharczyk, and J. Mańdziuk, "Constrained Hybrid Metaheuristic Algorithm for Probabilistic Neural Networks Learning," *ArXiv*, Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.15661

[27] Williams A., "TConfig - Test Configuration Generator," Github. Accessed: Feb. 05, 2025. [Online]. Available: https://github.com/awwilliams/tconfig?tab=readme-ov-file

[28] Y. Lei and K. C. Tai, "In-Parameter-Order: A Test Generation Strategy for Pairwise Testing," *Proceedings - 3rd IEEE International High-Assurance Systems Engineering Symposium, HASE 1998*, 1998, doi: 10.1109/HASE.1998.731623.

[29] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, "IPOG: A General Strategy for T-Way Software Testing," in *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*, 2007, pp. 549–556.

[30] Y. Lei, R. Kacker, D. R. Kuhn, V. Okun, and J. Lawrence, "IPOG-IPOG-D: Efficient test generation for multi-way combinatorial testing," *Software Testing Verification and Reliability*, vol. 18, no. 3, pp. 125–148, Sep. 2008, doi: 10.1002/stvr.381.

[31] M. Forbes, J. Lawrence, Y. Lei, R. N. Kacker, and D. R. Kuhn, "Refining the In-Parameter-Order Strategy for Constructing Covering Arrays," *J. Res. Natl. Inst. Stand. Technol.*, vol. 113, pp. 287–297, 2008, doi: 10.6028/jres.113.022.

[32] J. Mohammed and H. Altmemi, "An Analogous T-Way Test Generation Strategy for Software Systems MC-MIPOG," *Software Engineering*, vol. 7, no. 1, pp. 1–12, 2018, doi: 10.5923/j.se.20180701.01.

[33] B. S. Ahmed and K. Z. Zamli, "PSTG: A t-way strategy adopting particle Swarm Optimization," *AMS2010: Asia Modelling Symposium 2010 - 4th International Conference on Mathematical Modelling and Computer Simulation*, pp. 1–5, 2010, doi: 10.1109/AMS.2010.14.

[34] B. S. Ahmed and K. Z. Zamli, "A Variable Strength Interaction Test Suites Generation Strategy Using Particle Swarm Optimization," *Journal of Systems and Software*, vol. 84, no. 12, pp. 2171–2185, 2011, doi: 10.1016/j.jss.2011.06.004.

[35] A. R. A. Alsewari and K. Z. Zamli, "Interaction Test Data Generation Using Harmony Search Algorithm," *2011 IEEE Symposium on Industrial Electronics and Applications, ISIEA 2011*, pp. 559–564, 2011, doi: 10.1109/ISIEA.2011.6108775.

[36] B. S. Ahmed, T. S. Abdulsamad, and M. Y. Potrus, "Achievement of Minimized Combinatorial Test Suite for Configuration-aware Software Functional Testing Using The Cuckoo Search algorithm," *Inf. Softw. Technol.*, vol. 66, pp. 13–29, 2015, doi: 10.1016/j.infsof.2015.05.005.

[37] K. Rabbi, Q. Mamun, and R. Islam, "An Efficient Particle Swarm Intelligence Based Strategy to Generate Optimum Test Data in T-way Testing," *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 123–128, 2015, doi: 10.1109/ICIEA.2015.7334096.

[38] K. Z. Zamli, B. Y. Alkazemi, and G. Kendall, "A Tabu Search Hyper-Heuristic Strategy for T-way Test Suite Generation," *Applied Soft Computing Journal*, vol. 44, pp. 57–74, 2016, doi: 10.1016/j.asoc.2016.03.021.

[39] F. Din and K. Z. Zamli, "Fuzzy adaptive teaching learning-based optimization strategy for GUI functional test cases generation," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3185089.3185148.

[40] K. Z. Zamli, F. Din, B. S. Ahmed, and M. Bures, "A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem," *PLoS One*, vol. 13, no. 5, May 2018, doi: 10.1371/journal.pone.0195675.

[41] A. K. Alazzawi, H. M. Rais, and S. Basri, "ABCVS: An Artificial Bee Colony for Generating Variable T-Way Test Sets," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, pp. 259–274, 2019, doi: 10.14569/ijacsa.2019.0100431.

[42] M. Z. Z. Ahmad, R. R. Othman, M. S. A. R. Ali, and N. Ramli, "A Self-Adapting Ant Colony Optimization Algorithm Using Fuzzy Logic (ACOF) for Combinatorial Test Suite Generation," in *IOP*

Conference Series: Materials Science and Engineering, 2020. doi: 10.1088/1757-899X/767/1/012017.

[43] A. A. Hassan, S. Abdullah, K. Z. Zamli, and R. Razali, "Combinatorial Test Suites Generation Strategy Utilizing the Whale Optimization Algorithm," *IEEE Access*, vol. 8, pp. 192288–192303, 2020, doi: 10.1109/access.2020.3032851.

[44] M. L. Prasad, J. K. R. Sastry, B. Mallikarjuna, V. Sitaramulu, C. Srinivasulu, and B. B. Naib, "Development of a Programmed Generation of t-way Test cases Using an Improved Particle Swarm Optimization Strategy," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1394–1399. doi: 10.1109/ICACITE53722.2022.9823783.

[45] N. H. C. Rose, R. R. Othman, H. L. Zakaria, A. J. Suali, and Z. A. Ahmad, "Wingsuit Flying Search Optimization Algorithm Strategy for Combinatorial T-Way Test Suite Generation," *International Journal of Advances in Soft Computing and its Applications*, vol. 16, no. 3, pp. 272–293, 2024, doi: 10.15849/IJASCA.241130.15.

[46] J. Yang, Y. Zhang, Z. Wang, Y. Todo, B. Lu, and S. Gao, "A Cooperative Coevolution Wingsuit Flying Search Algorithm with Spherical Evolution," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, Dec. 2021, doi: 10.1007/s44196-021-00030-z.

[47] M. Z. Z. Ahmad, "A Self-Adapting Ant Colony Optimization Algorithm using Fuzzy Logic (ACOF) for Combinatorial Test Suite Generation," Universiti Malaysia Perlis, 2023.

Hybrid Cascaded ANFIS–RBFNN-Based Controller For PV-Driven Grid System

Submitted: 2nd August 2024; accepted: 17th September 2024

Blessy A. Rahiman, J. Jayakumar, R. Meenal

DOI: 10.14313/jamris-2026-029

Abstract:
Solar photovoltaic (PV) energy is gaining popularity in modern distribution networks due to its clean energy attributes. In order to maximize PV power generation conversion, the application of maximum power point tracking (MPPT) is essential. Henceforth, this work presents a novel hybrid MPPT approaches based on a cascaded adaptive neuro fuzzy inference system and radial basis function neural network to achieve rapid and greatest PV power extraction while ensuring zero oscillation tracking with a single-ended primary inductor converter (SEPIC). SEPIC efficiently regulates the output voltage to match grid requirements while maintaining high power conversion efficiency. The cascaded artificial neuro fuzzy inference system (ANFIS) and radial basis function neural network (RBFNN) are combined to enhance MPPT accuracy and robustness under varying environmental conditions. The cascaded architecture enables a seamless transition between the two controllers, ensuring optimal performance across an extensive range of operating conditions. The MATLAB/Simulink is used for analyzing the efficacy of the proposed system and the proposed converter and MPPT approach is compared with existing topologies for proving the importance of the developed work. The outcomes demonstrate that the proposed SEPIC achieves reduced Total Harmonic Distortion (THD) of 1.16% and the cascaded ANFIS–RBFNN-based MPPT approach attains a higher tracking efficiency of 99.61% with rapid convergence speed and execution time compared to traditional techniques. Overall, this paper represents a promising direction toward achieving more efficient and sustainable PV-driven grid integration.

Keywords: *photovoltaic, maximum power point tracking, cascaded ANFIS–RBFNN, single-ended primary inductor converter,MATLAB/Simulink*

1. Introduction

The worldwide energy demand is constantly being enhanced owing to population growth and industrialization [1, 2]. Traditional fossil fuels like coal, oil, and natural gas are the main sources of energy, although their usage has led to various issues such as environmental pollution and contribution to climate change through greenhouse gas emissions [3, 4]. To address these issues, there has been a rising emphasis on developing and utilizing renewable energy sources (RESs) such as solar, wind, and geothermal power [5, 6]. Among these various RESs, photovoltaic (PV) technology has emerged as a promising topology due to its various advantages, which generate electricity without producing greenhouse gas emissions or other forms of ecological pollution, thus generating a clean and sustainable energy solution [7–9]. Though the output voltage of PV is lower because of its intermittent nature, direct current (DC), DC converters are essential for efficiently converting variable DC output of the PV modules to the increased level of energy required for grid applications [10]. In this context, various traditional converters used in current studies such as Boost [11], Buck-Boost [12], and Cuk [13] converters provide step-up and step down voltage with better voltage gain. Nevertheless, each converter topology has its own advantages and drawbacks such as poor efficiency, high cost, complexity, and switching stress [14, 15]. Henceforth, the proposed topology employs a single-ended primary inductor converter (SEPIC) to overcome these challenges by attaining high efficiency with minimized switching stress and ripple current.

In contrast, maximum power point tracking (MPPT) techniques are specifically designed to optimize power output from solar panels by continuously adjusting to varying conditions, making them more effective in maximizing energy extraction. Compared to proportional integral (PI) and fractional order sliding mode control controllers, intelligent MPPT provides superior adaptability and efficiency under fluctuating input conditions, highlighting its advantage in energy harvesting applications [16–20]. To improve the performance of tracking efficiency, various traditional topologies have been developed, which are illustrated in Table 1 below.

To overcome the limitations obtained in the traditional MPPT topologies, the proposed work incorporates a novel cascaded artificial neuro fuzzy inference system (ANFIS)–radial basis function neural network (RBFNN)-based MPPT, which has the advantages of enhancing MPPT accuracy and robustness under varying environmental conditions with rapid convergence speed and execution time. The foremost contributions of the proposed work are illustrated below,

- Implementing a PV-based SEPIC to efficiently handle an extensive input voltage range, ensuring optimal power conversion system, even under varying solar conditions.

Table 1. Survey related to the traditional MPPT approaches

References	Methodology	Advantages	Limitations
Shaik rafi kiran et al (2022) [21]	artificial neural network (ANN) based MPPT	It achieves better tracking efficiency with minimized oscillation of MPP.	However, this system applicable for partially shaded PV system.
Faizan Mehmood et al (2020) [22]	Fuzzy-based MPPT	At the period of transient condition, this technique attains better performance with efficient power delivery.	Nevertheless, it has steady state oscillations and system complexity.
Sara et al (2021) [23]	ANFIL-based MPPT	It has higher accuracy, faster response with better tracking.	However, due to increasing number of rules, the system complexity is enhanced.
Chaoping rao et al (2022) [24]	Perturb observe (P&O)-Based MPPT	It attains minimized steady-state error with better tracking efficiency.	Nonetheless, the fluctuation around MPP and complexity leads to degradation of system performance.
Pawan Kumar Pathak et al (2021) [25]	modified incremental conductance (INC)-based MPPT	MINC attains high tracking efficacy with effectual convergence speed.	However, execution time needs to be considered in further studies.

- The cascaded ANFIS–RBFNN-based MPPT approach is introduced for extracting the highest power from the PV system, resulting in improved MPPT accuracy with minimal computational time.
- Employing PI controller for ensuring precise regulation of the single phase inverter output to synchronize with the grid frequency and voltage.

Section 2 describes the proposed system modeling; Section 3, the outcomes from the MATLAB/Simulink for the developed work is discussed with a comparative analysis. Further, in Section 4, conclusions about the proposed system are discussed by showing the importance of the proposed system.

2. Proposed Modeling

PV systems have gained significant importance in the RES, as it offers a clean and sustainable source of electricity. However, the sporadic nature of solar radiation and the nonlinear characteristics of PV systems pose challenges in maintaining a stable and efficient power supply to the grid. To address these challenges, this research explored SEPIC with the novel cascaded ANFIS–RBFNN-based MPPT topology, which provides an enhanced output voltage with better tracking performance. The proposed work block diagram is shown in Figure 1 below.

In this work, SEPIC has been developed for maximizing the low-output voltage of a PV system for the essential level for a grid system. On the other hand, the MPPT approach is used for tracking optimal energy from the PV modules; thus the cascaded ANFIS–RBFNN-based MPPT topology has been developed, which leverages the strength of each approach to enhance control precision, robustness, and maximum energy harvesting from the solar module. The tracked output is fed to the Pulse Width Modulation (PWM) generator for producing required pulses for the switching operation of the SEPIC. A stable DC-link voltage is delivered to the single-phase Voltage Source Inverter (VSI) for converting the DC–AC supply; it is regulated by the PI controller for attaining grid synchronization. Finally, the required level of energy is given to the grid system without any disturbances.

2.1. Modeling of the PV System

The PV system generates electricity directly from sunlight, a free and abundant RES. Unlike fossil fuels, this energy production does not release greenhouse gases or other pollutants, making a clean and ecological friendly energy source. An equivalent circuit of PV module is represented in Figure 2.

Kirchhoff's current law is applied as follows [21]:

$$I = I_{PV} - I_d - I_{sh} \tag{1}$$

The I_d and I_{sh} are derived as follows:

$$I_d = I_0 \left[exp\left(\frac{V + IR_s}{nV_T} \right) - 1 \right] \tag{2}$$

$$I_{sh} = \frac{(V + IR_s)}{R_{sh}}, \tag{3}$$

where I_0 denotes reverse saturation current, R_{sh}, $R_{p,}$ and R_s specifies shunt, series, and parallel resistances, $and\ V_T$ indicates thermal voltage. The following equations express I_0 and V_T:

$$I_0 = \frac{(I_{SC} + K\Delta T)}{exp\left(\frac{K_v \Delta T}{\alpha V_T} + V_{op,cct} \right) - 1} \tag{4}$$

$$I_0 = \frac{(I_{SC} + K(-T_n + T)}{exp\left(\frac{K_v \Delta T}{\alpha V_T} + V_{op,cct} \right) - 1} \tag{5}$$

$$V_T = \frac{N_S AKT}{q} \tag{6}$$

Here, $\propto$ indicates diode ideality constant, n specifies the diode's ideality factor, N_s and shows the panel linked in series, $V_{op,cct}$ represents open-circuit voltage, T shows the temperature= 1.38×10^{-3}, and q denotes the charge of electron = 1.6×10^{-19}, respectively. Moreover, the low output power is obtained from PV owing to its ecological changes; thus, SEPIC is employed in this study for boosting the voltage as follows.

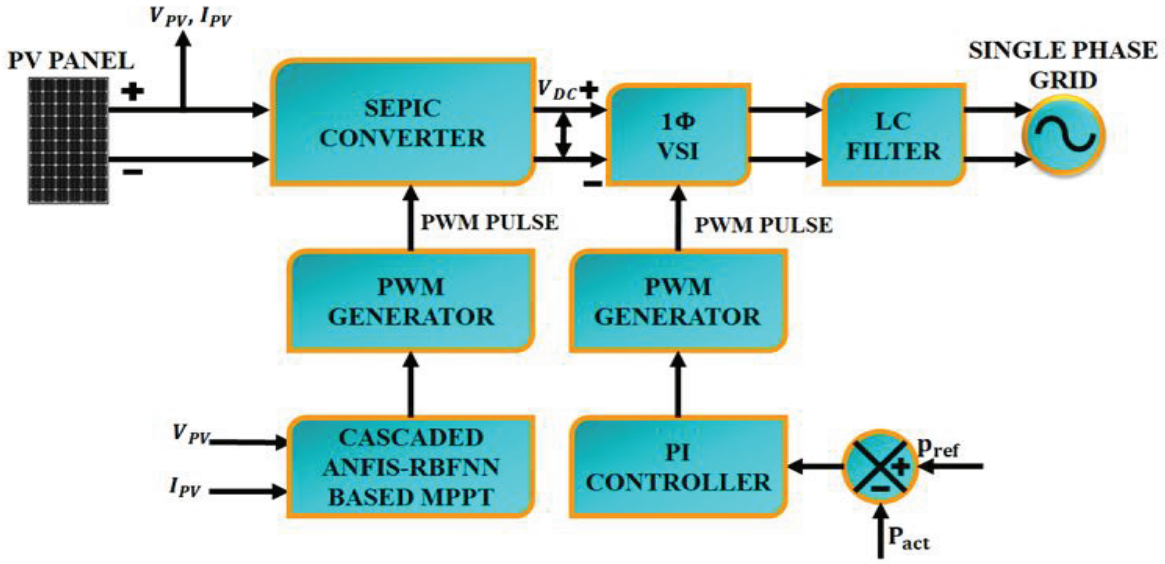


Figure 1. Block diagram of the proposed work

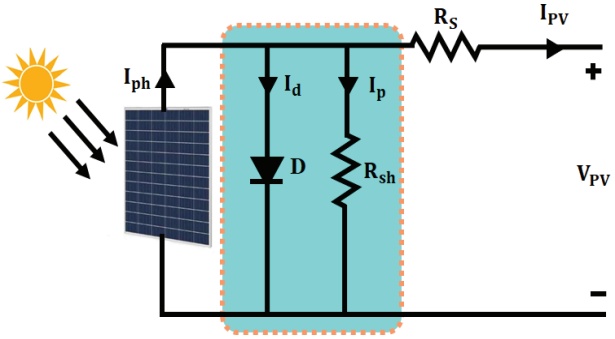


Figure 2. Circuit diagram of a PV module

2.2. Modeling of SEPIC

SEPIC is a popular choice for interfacing the PV system-fed grid system due to its ability to operate in both step-up and step-down modes, allowing for efficient power transfer. The circuit of SEPIC is represented in Figure 3, which shows two inductors, a switch, two capacitors, and a diode, respectively. Furthermore, the proposed converter operates in two modes, as specified in Figure 4.

The following expression shows the average voltage:

$$V_{in}=V_{La}+V_{Ca}+V_{Lb} \quad (7)$$

$$V_{La}=V_{Lb} \quad (8)$$

The sum of the average currents is given as follows:

$$I_{D1}=I_{La}-I_{Lb} \quad (9)$$

The duty cycle of SEPIC is defined as:

$$D_{max}=\frac{V_{out}+V_D}{V_{in(min)}+V_{out}+V_D} \quad (10)$$

The minimum duty cycle is expressed as:

$$D_{min}=\frac{V_{out}+V_D}{V_{in(max)}+V_{out}+V_D} \quad (11)$$

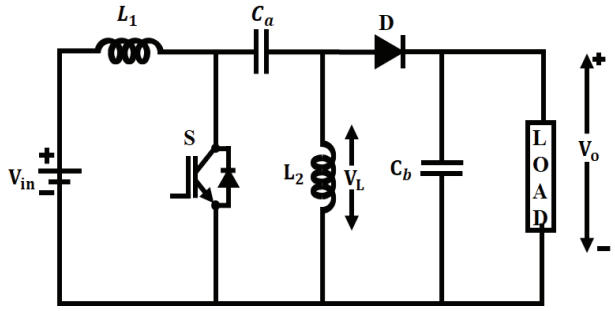


Figure 3. SEPIC converter circuit diagram

Mode 1 (ON state): When switch S_1 is in the ON condition, the current in L_1 becomes negative during the current increases; the diode does not conduct in this state. The capacitor discharge during both L_1 and L_2 gets charged, as represented in Figure 4(a).

Mode 2 (OFF state): When the switch is in the OFF state, the diode function is forward-biased, an output gets energy from L_1 during induction, and L_1 charges the capacitor C_a, as illustrated in Figure 4(b), respectively. Also, the switching waveform for the proposed SEPIC is illustrated in Figure 5.

The ripple current flowing through inductors L_a, L_b is equal to

$$\Delta I_L=I_{out}\times\frac{V_{out}}{V_{in(min)}}\times 40\% \quad (12)$$

The inductor value is calculated as

$$L_1=L_2=L=\frac{V_{in(min)}}{\Delta I_L\times f_{sw}}\times D_{max} \quad (13)$$

The output capacitor is given as

$$C_b\geq\frac{I_{out}\times D_{max}}{V_{ripple}\times 0.5\times f_{sw}} \quad (14)$$

Here, the switching frequency (f_{sw}) is taken as $= 10KHz$.

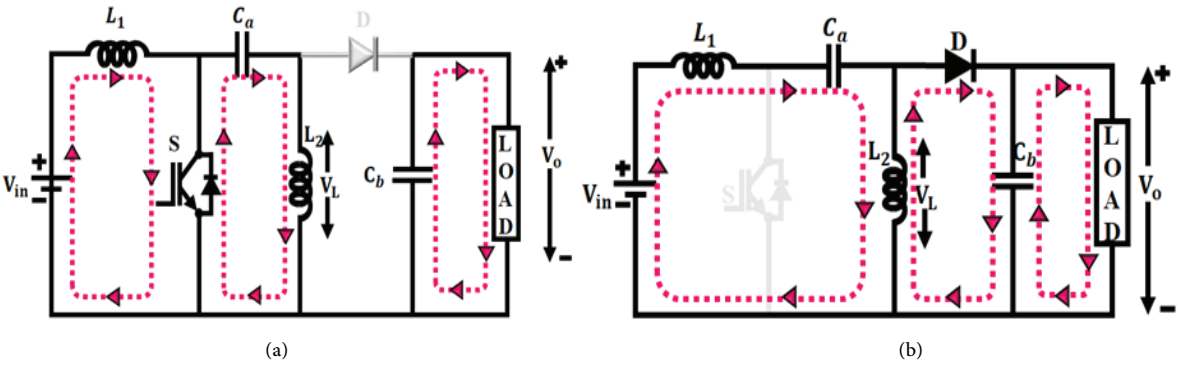


Figure 4. Modes of operation

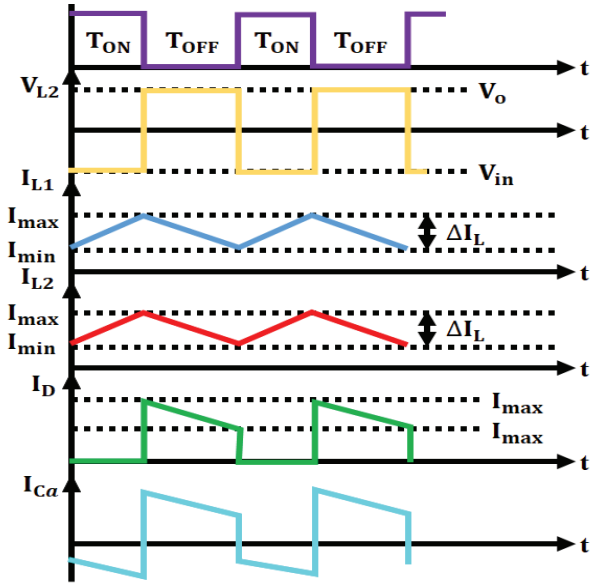


Figure 5. Switching waveform for the proposed converter

Therefore, the duty cycle range of inductors L_1, and L_2 and capacitors C_a, C_b are defined using the foresaid equations. The following section explains the cascaded ANFIS–RBFNN-based MPPT for extracting optimal power from the PV system.

2.3. Modeling of Cascaded ANFIS-RBFNN Based MPPT

ANFIS-based MPPT

The ANFIS-based MPPT is a powerful technique that combines the strength of fuzzy logic control and ANN, which maximizes the power of PV systems, as specified in Figure 6. This approach is particularly well-suited for dealing with the highly nonlinear and complex relationship between the parameters of PV. In ANFIS, the fuzzy inference system (FIS) is responsible for modeling the nonlinear relationship between the PV system's output and desired output using a set of if-then rules and membership functions. On the other hand, the ANN is employed to automatically tune the parameters of the FIS, such as membership function and rules based on the available training data. Through the back propagation algorithm, ANN learns the optimal FIS parameters, reducing error

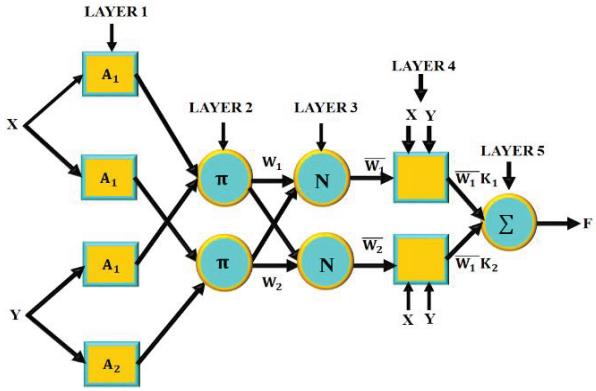


Figure 6. Structure of ANFIS

between actual and desired output. The multilayer feed-forward networks are described by the following mathematical equation:

$$w_j = \mu p_j(x) * \mu Q_j(y), w_j = \frac{w_j}{w_1 + w_2}, j = 1, 2 \quad (15)$$

The following equations express K_1, K_2, and K,

$$K_1 = A_1 x + B_1 y + R_1 z, K_2 = A_2 x + B_2 y + R_2 z \quad (16)$$

$$K_1 = \frac{W_1 K_1 + W_2 K_2}{W_1 + W_2} = \overline{W}_1 K_1 + \overline{W}_2 K_2 \quad (17)$$

$$\mu p_j(x) = \frac{1}{1 + \left[\left((x - r_j) p_j\right)^2\right]^{bj-3}} \quad (18)$$

$$\mu p_j(x) = \exp\left[-\left[\left(\frac{x - r_j}{p_j}\right)^2\right]^{bj}\right], \quad (19)$$

where p_j, r_j, and bj denote the membership function, respectively.

RBFNN-based MPPT

The nonlinear mapping is performed using an RBFNN, unlike the error backpropagation learning model of conventional neural networks, RBF networks employ a learning process that is equivalent to solving a linear problem. The hourly requirement oscillation serves as input for the RBFNN network training, which has three types: input, hidden, and output layers, as

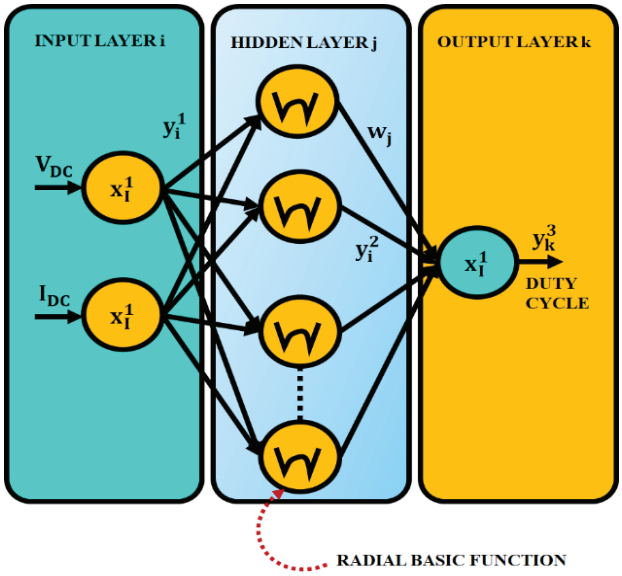


Figure 7. RBFNN architecture

specified in Figure 7. The input layer is the top layer, which is carried out by the input data source nodes. The second layer produces radial-based processes. The third layer is the output layer, which specifies the nonlinear combination of neural parameters and input radial basis function. Overall this RBFNN's primary goal is to forecast the PV system's maximum power.

Step: 1 Create initial input vectors based on variables such as voltage, current, and power, which are adjusted based on the network's output power requirements.

Step: 2 After the initialization process, the system generates its initialized input parameters at random [22].

$$\text{rand}^1 = \begin{matrix} {P^{11}}_{pd} & {P^{12}}_{pd} & \cdots & {P^{1n}}_{pd} \\ {P^{21}}_{pd} & {P^{22}}_{pd} & \cdots & {P^{2n}}_{pd} \\ \vdots & \vdots & \vdots & \vdots \\ {P^{m1}}_{pd} & {P^{m2}}_{pd} & \cdots & {P^{mn}}_{pd} \end{matrix} \tag{20}$$

Here, P_{pd} specifies the power demand.

Step: 3 By reducing the error function, which is provided below, fitness is assessed.

$$\text{Error, E} = \frac{1}{2}\sum (t_{OD} - d_{OD}), \tag{21}$$

where the desired and target output demand is denoted as d_{OD} and $t_{OD.}$

Step: 4 The Gaussian activation function of RBFNN yields the RBFNN output, which is provided as follows:

$$\text{ramd}\left(y_p - c_i\right) = \exp\left(-\frac{1}{2\xi^2}\left\|y_p - c_i\right\|^2\right), \tag{22}$$

where ξ specifies the Gaussian activation function, and y_p denotes the p-th input sample, respectively.

$$u_b = \sum_{a-1}^{h} w_{ab} \exp\left(-\frac{1}{2\xi^2}\left\|y_p - c_i\right\|^2\right) b = 1, 2, \ldots, n, \tag{23}$$

The number of nodes in the hidden layer specifies *h*. Moreover, by combining this two approaches, the performance of tracking efficiency is enhanced, as described below.

Cascaded ANFIS–RBFNN-based MPPT

The cascaded ANFIS–RBFNN-based MPPT combines two powerful techniques to improve the efficiency of capturing maximum power from solar panels. First, ANFIS uses fuzzy logic and neural networks to adaptively adjust its rules based on changing conditions, helping to predict and optimize power output. Second, the RBFNN provides precise function approximation by using radial basis functions to fine-tune the tracking process. Together, this cascaded approach enhances the MPPT algorithm's ability to rapidly and accurately find the optimal power point, resulting in better performance and energy extraction compared to traditional methods. The developed cascaded ANFIS–RBFNN-based MPPT system flowchart is indicated in Figure 8, which shows that the system measures the solar irradiance, which is a crucial input parameter for the MPPT algorithm. It also measures the voltage and current from the PV system, which are used as inputs to the developed MPPT algorithm. The next step is to detect any abrupt changes in the solar irradiance, as these changes affect the optimal operating point of the PV system. If an abrupt change is detected, the system sets a new operator voltage to ensure the PV system is operating at its highest point. The core of the MPPT algorithm is the ANFIS–RBFNN-based MPPT block. This combines the ANFIS and RBFNN to track the MPP of PV system efficiently, even in the presence of abrupt changes in solar irradiance. The ANFIS component provides the fuzzy–logic-based decision making, while the RBFNN component handles the nonlinear mapping between the input parameters and the optimal operating voltage. By combining this, the overall efficiency, reliability is significantly improved, creating more viable and improved performance of SEPIC.

By utilizing the proposed approach, the optimal energy from the PV system is efficiently tracked with greater tracking efficiency, execution time, and convergence speed. Moreover, the tracked and enhanced power is given to the single-phase VSI for converting the DC–AC supply to distribute power to the single-phase grid system. Also, the inverter is efficiently controlled with the aid of the PI controller for grid synchronization.

3. Results and Discussion

In this work, a SEPIC-based cascaded ANFIS-RBFNN approach is developed for grid applications. The proposed system is validated in MATLAB/Simulink for validating the effectiveness of the developed system. Additionally, the comparative assessment is carried out with other traditional topologies for showing the importance of the developed work. In Table 1, parameter specifications for the proposed system are illustrated.

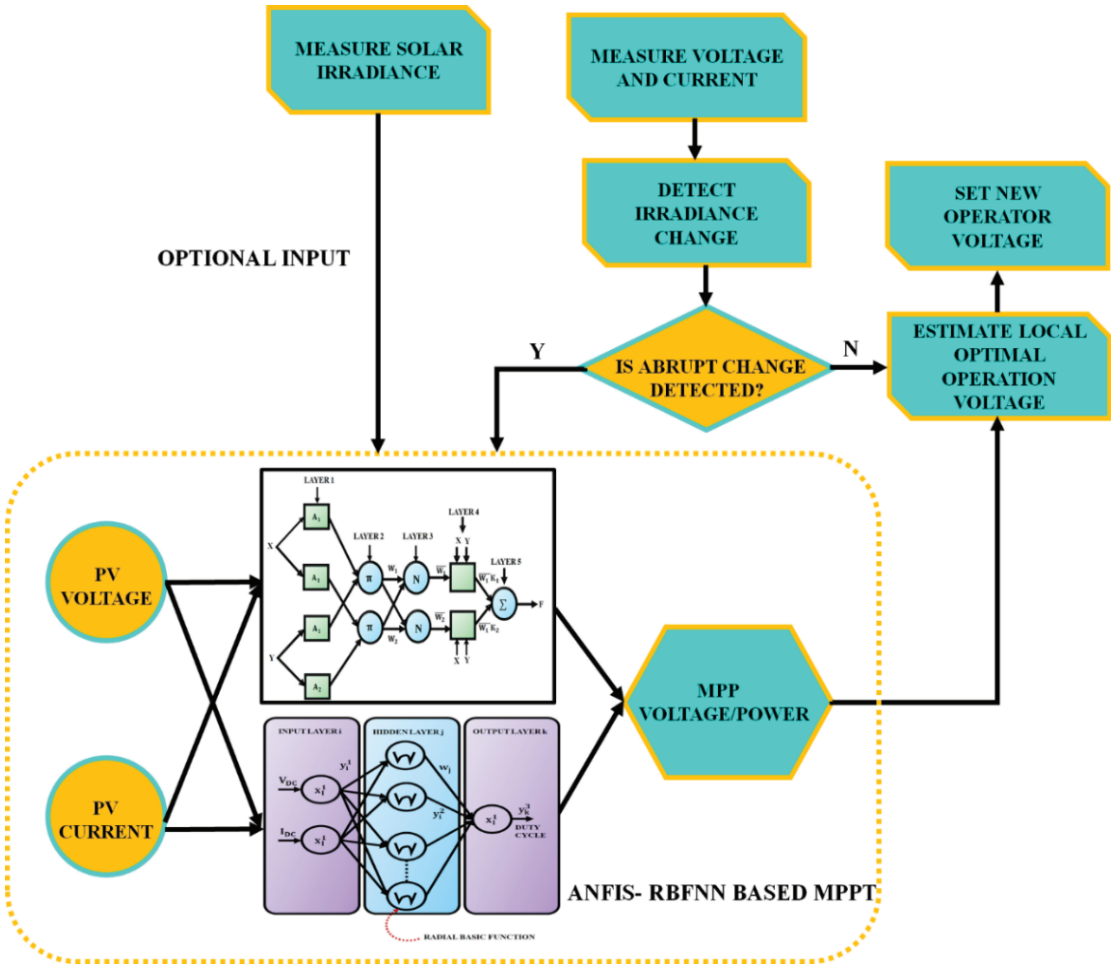


Figure 8. Proposed cascaded ANFIS–RBFNN-based MPPT

Table 2. Parameter Specifications of Proposed System

Parameter	Description
PV system	
Open circuit voltage	37.25V
Short-circuit current	8.95A
Series-connected solar panel	2
Parallel-connected solar PV cell	25
Maximum power voltage	29.95V
Maximum current	8.35A
SEPIC	
Switching frequency	$10kHz$
C_a, C_b	$4.7\mu F$
$L_1,\ L_2$	$1\ mH$

(a) Case 1. Varying Temperature and Varying Irradiation

The solar module waveform is illustrated in Figure 9. The temperature of the solar panel varies slightly during the initial time, and after 0.2s it is stabilized at 45°C, as specified in Figure 9(a). Consequently, the irradiation of the solar panel is maintained constantly at 1000(W/Sq.m) after the fluctuation up to 0.2s, as specified in Figure 9(b). Also, the voltage of the solar panel for the proposed work attains the stabilized voltage at 48V after 0.2s, as illustrated in Figure 9(c).

The proposed SEPIC waveform is illustrated in Figure 10.The input current waveform indicates that the input current oscillates highly during the starting period, and after 0.22s, the constant current is maintained at 0.8A, as specified in Figure 10(a). From Figure 10(b), it is observed that the converter output voltage is stabilized at 300V after 0.25s. Similarly, the converter output current has high oscillation during the initial time, and after 0.05s a constant output current is obtained at 25A with fluctuations.

The grid waveform is indicated in Figure 11. The grid voltage waveform illustrates that the grid voltage is stabilized at 230V without any distortions, as illustrated in Figure 11(a). Likewise, the current waveform specified in Figure 11(b) indicates that the grid oscillates slightly, and after 0.05s it stabilizes at 4A. Moreover, the in-phase waveform is specified in Figure 11(c), where it can be observed that the voltage and current stabilize at 230V and 4A, respectively.

The real and reactive power for the developed work is illustrated in Figure 12. The stabilized real and reactive power results in the superior performance of the proposed system.

(b) Case 2. Constant temperature and constant irradiation

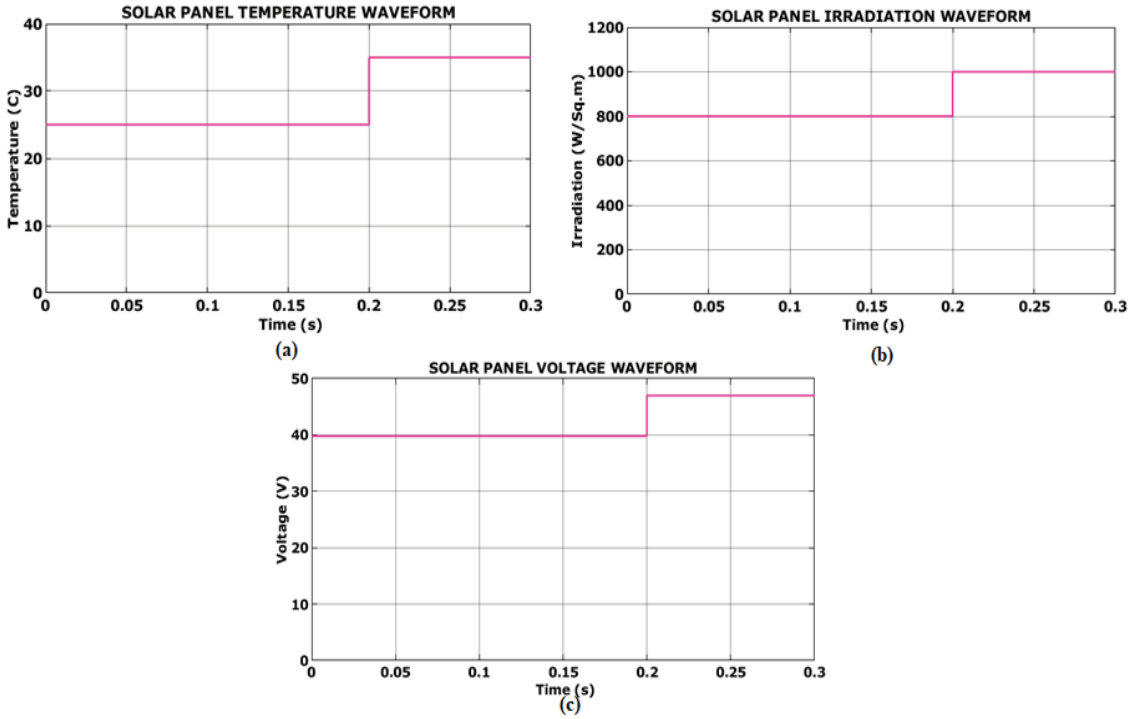


Figure 9. Solar module waveform for case 1. (a) Temperature; (b) irradiation; (c) voltage

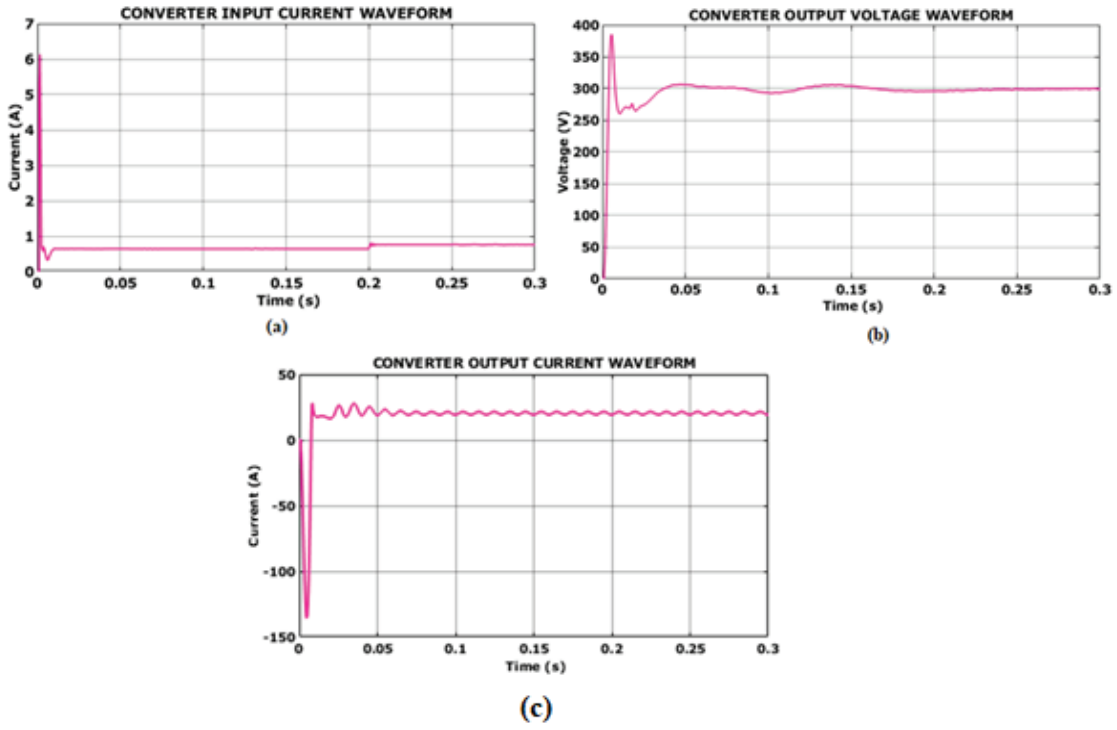


Figure 10. Converter output waveform for case 1

Figure 13 represents the solar panel waveform for case 2, which shows that the temperature of the solar panel gets constantly stabilized at 35°C, as indicated in Figure 13(a). Similarly, the irradiation waveform illustrated in Figure 13(b) shows that the irradiation is maintained at 1000(W/Sq.m). The voltage of solar panel attains stability at 48V in the case 2 condition, as indicated in Figure 13(c).

The converter waveform for the case 2 condition is represented in Figure 14. As specified in Figure 14(a), the converter input current oscillates highly during the initial period, and after 0.02s, the constant current is attained at 0.8A. Moreover, the output voltage waveform, indicated in Figure 14(b), shows that the voltage is constantly maintained at 300V after facing high fluctuation up to 0.25s. The converter output waveform illustrated in Figure 14(c) indicates that the

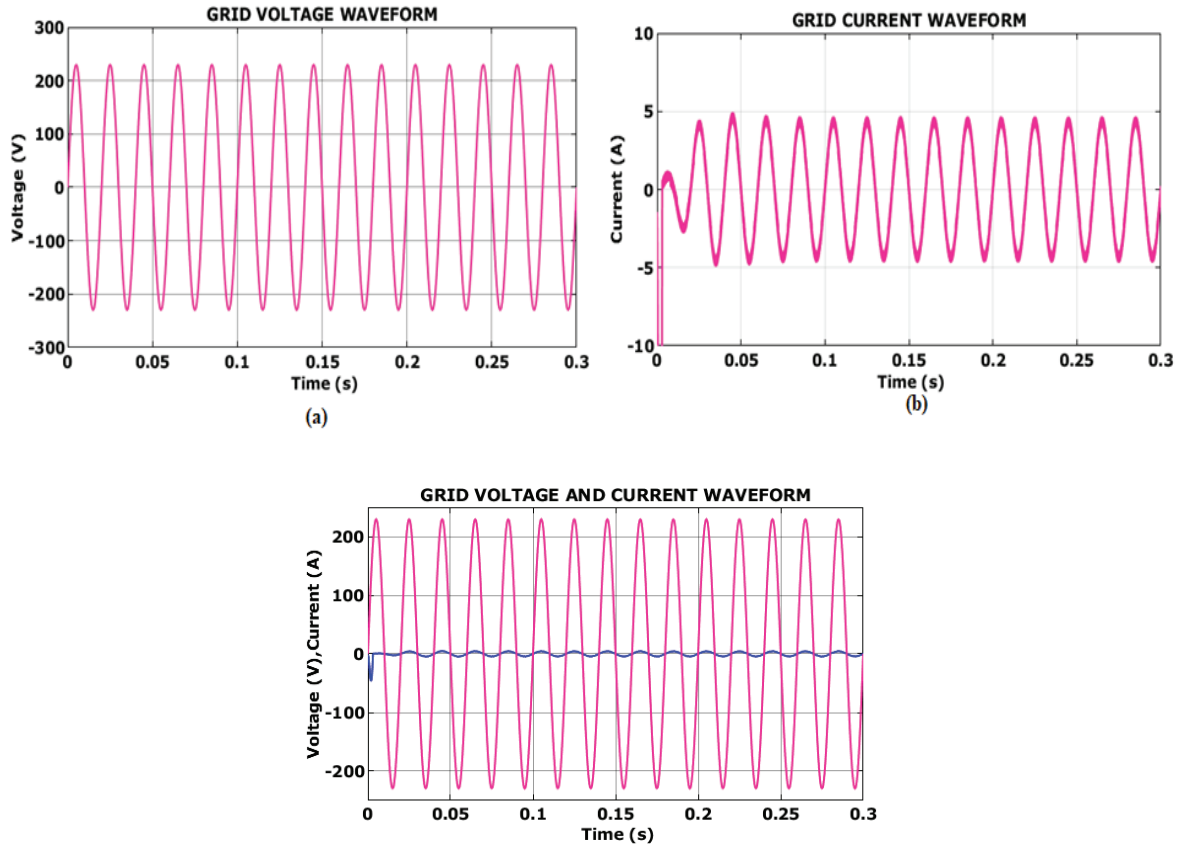


Figure 11. Grid waveform. (a) Voltage; (b) current; (c) in-phase voltage and current waveform

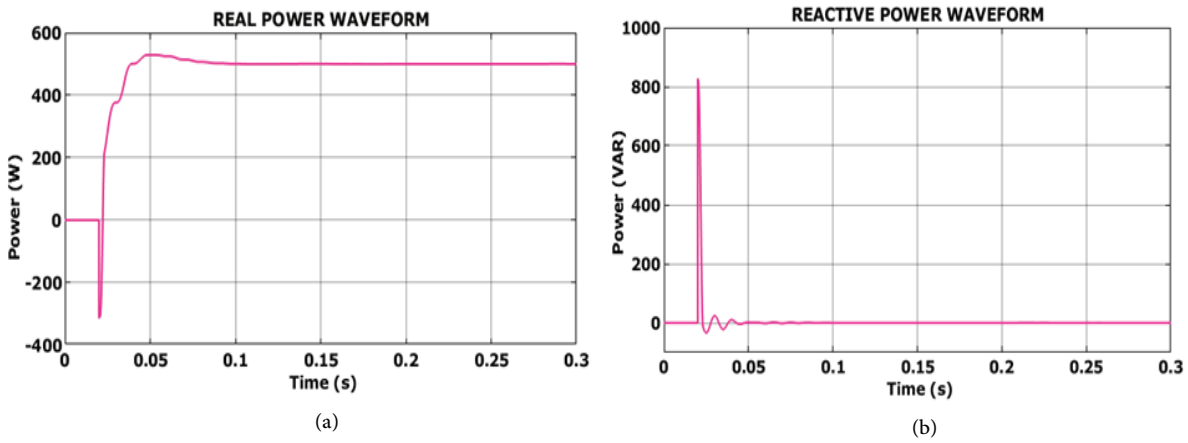


Figure 12. Real and reactive power waveform

output current fluctuates highly, and after 0.05,s, it gets stabilized with distortions.

(c) Case 3. Varying Temperature and Constant Irradiation

The case 2 condition for solar module waveform is represented in Figure 15. Figure 15(a) illustrates the varying temperature condition, where the temperature oscillates up to 0.2s and afterward is maintained gradually at 35°C. Likewise, the irradiation waveform illustrated in Figure 15(b) shows that the irradiation is maintained at 1000(W/Sq.m). Also, Figure 15(c) indicates that the solar panel voltage oscillates up to 0.2s; after that, it stabilizes at 48V.

As shown in Figure 16(a), during the first 0.02s, the converter input current oscillates greatly before reaching 0.8A of steady current. Furthermore, Figure 16(b) shows that the output voltage waveform is progressively maintained at 300V following a significant fluctuation that lasted for 0.25 s. The converter output current waveform shown in Figure 16(c); it fluctuates greatly during the initial time, and after 0.03 s, it is maintained at 25A with slight distortions.

(d) Case 4. Constant Temperature and Varying Irradiation

The solar panel waveform for scenario 4 is shown in Figure 17, where it can be noted that the solar panel's temperature steadily stabilizes at 35C, as

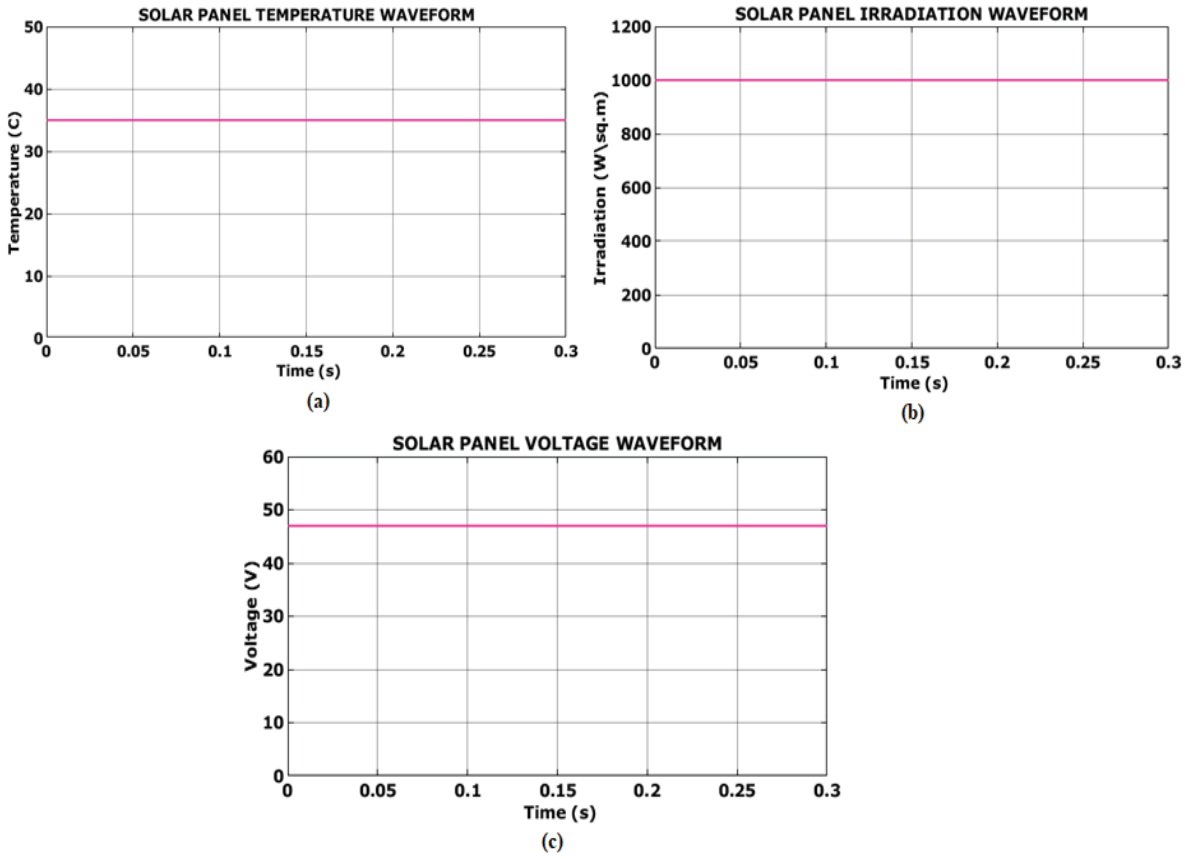


Figure 13. Solar panel waveform for case 2

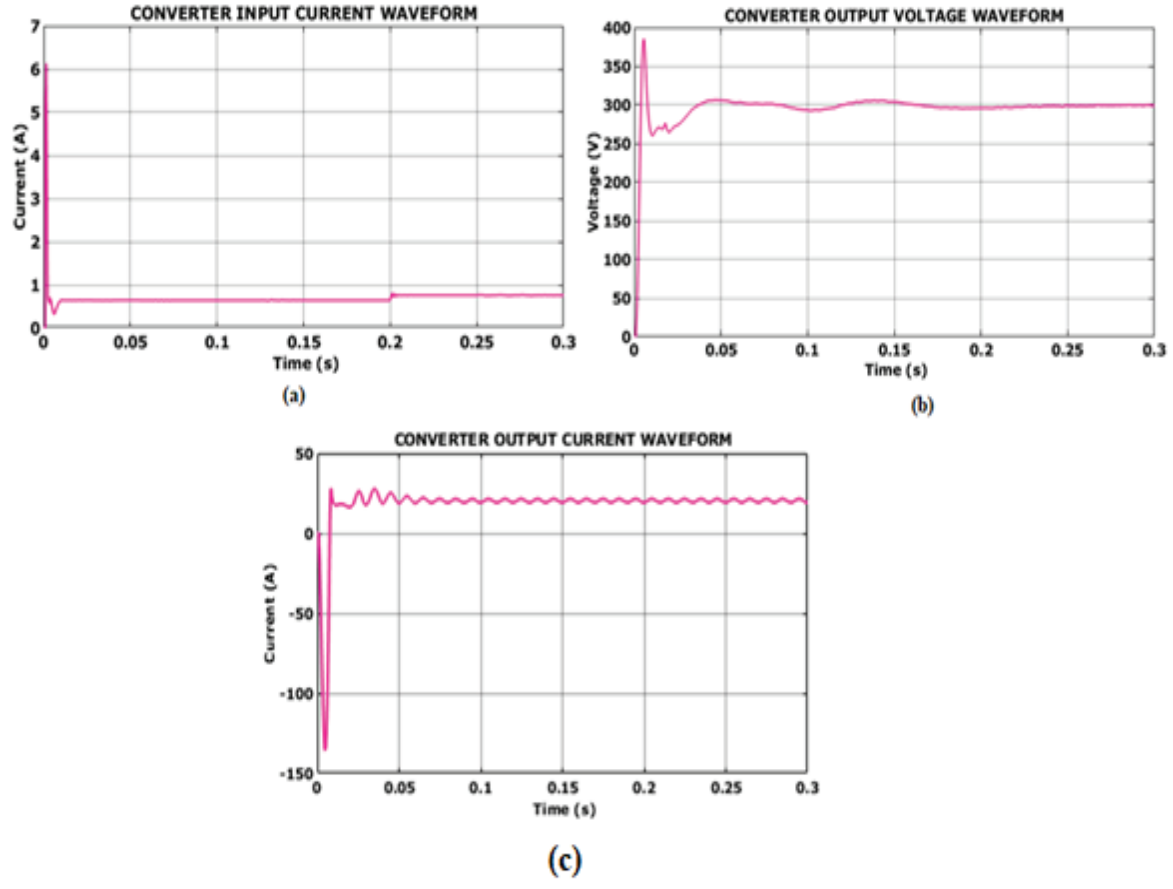


Figure 14. Converter waveform for case 2

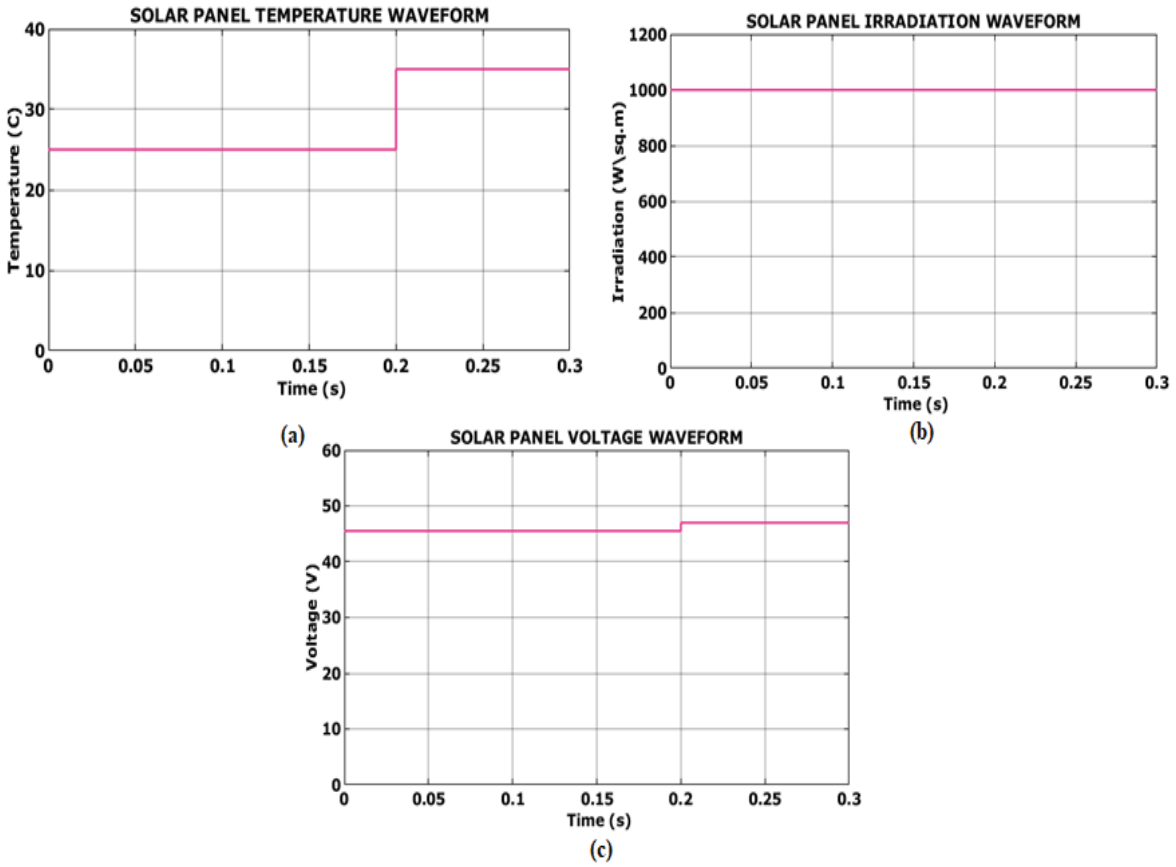


Figure 15. Solar panel waveform for case 3

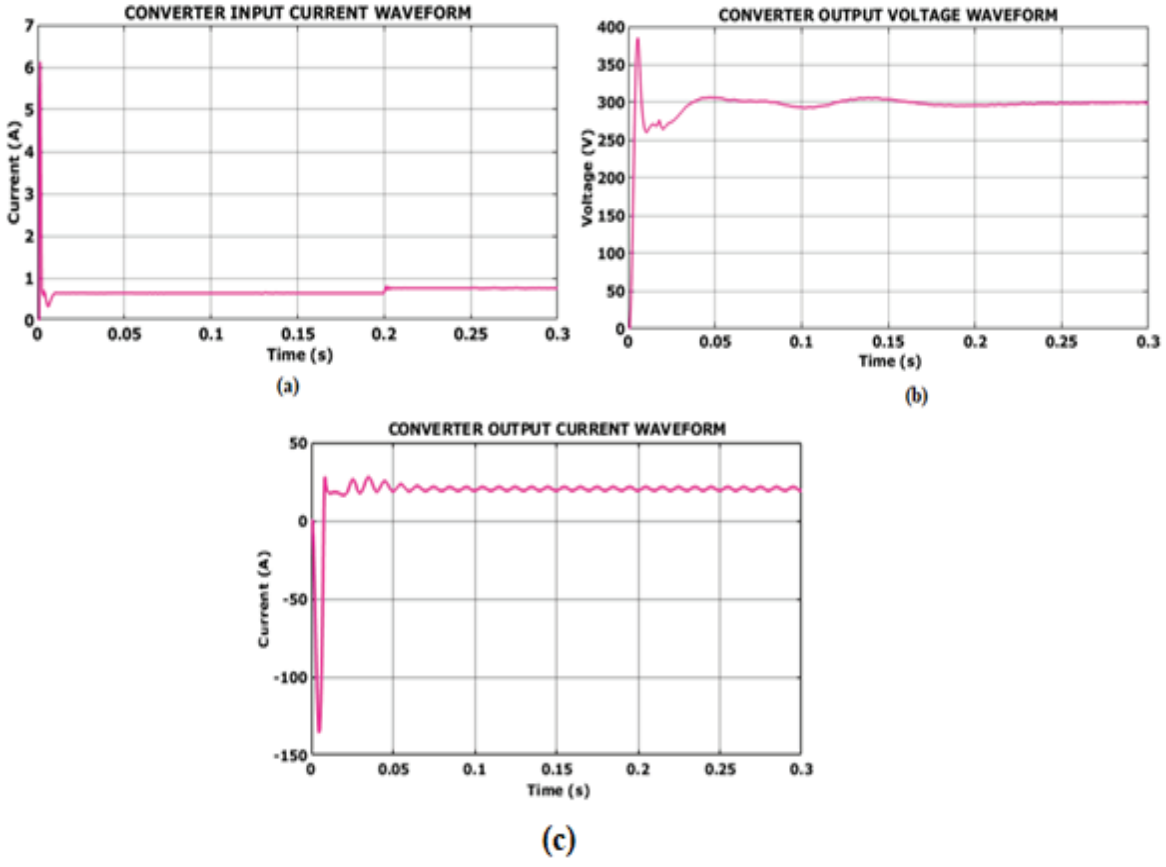


Figure 16. Converter waveform for case 3

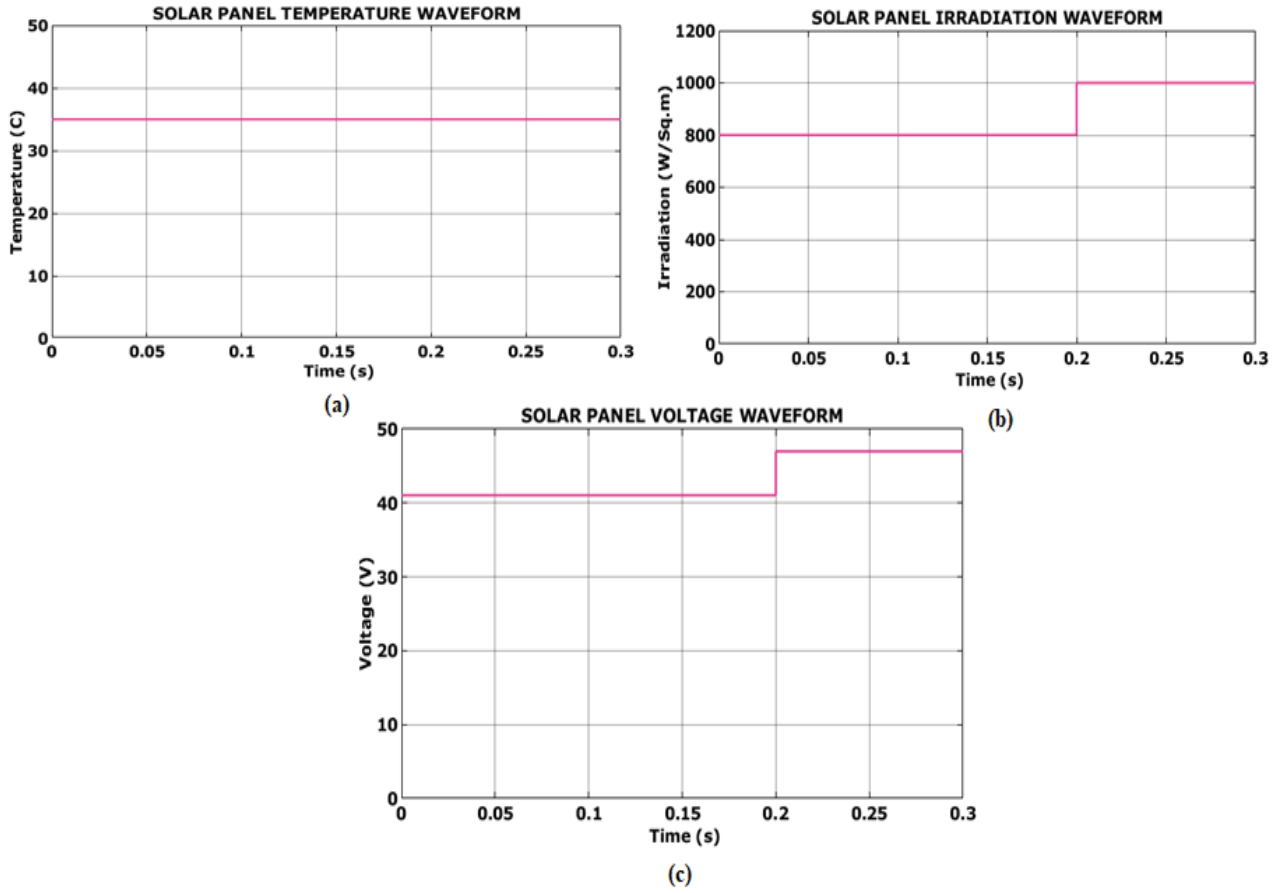


Figure 17. Solar panel waveform for case 4

represented in Figure 17(a). Figure 17(b) illustrates that the solar panel's irradiation is stabilized at 1000(W/Sq.m) following a variation of up to 0.2s. Additionally, as illustrated in Figure 17(c), the solar panel voltage reaches astabilized value at 48V after 0.2s.

The converter waveform for the final condition is shown in Figure 18. As seen in Figure 18(a), the converter input current oscillates greatly in the first period before reaching 0.8A of steady current after 0.02 s. Additionally, the output voltage waveform illustrated in Figure 18(b) demonstrates that the output voltage is constantly maintained at 300V following a strong fluctuation lasting up to 0.25s. The output current fluctuates greatly, as seen by the converter output waveform represented in Figure 18(c), and then stabilizes after 0.05s at 25A with distortions.

The THD waveform for the developed work is specified in Figure 19, where it is observed that the THD value of 1.16% is obtained. Thereby the performance of the developed system is efficiently enhanced.

The tracking efficiency for various MPPT approaches is compared with the introduced cascaded ANFIS–RBFNN-based MPPT, as demonstrated in Figure 20. From the graph, it is evident that the developed MPPT approach accomplishes high tracking efficiency of 99.61% compared to the other topologies, as referred to in [28–30].

Table 3. Comparison of THD for Various Converters

SI. No	Converters	THD (%)
1.	Boost [11]	6.42
2.	Buck-Boost [12]	3.43
3.	Cuk [13]	4.41
4.	Proposed SEPIC	1.16

Table 3 represents the comparison of THD for various traditional converters. The developed SEPIC reaches the lowest THD value of 1.89% compared to the other approaches.

The proposed cascaded ANFIS–RBFNN-based MPPT technique is compared with conventional MPPT approaches for convergence speed and execution time in Figure 21. From the graph, it is obvious that the cascaded approach attains high convergence speed and minimized execution time compared to traditional MPPT topologies.

4. Conclusion

The proposed research work introduced SEPIC-based novel cascaded RBFNN-based MPPT for grid applications. The proposed SEPIC effectively interfaces the PV array with the grid, providing stable voltage regulation. The integration of cascaded ANFIS–RBFNN-based MPPT enables efficient tracking of the

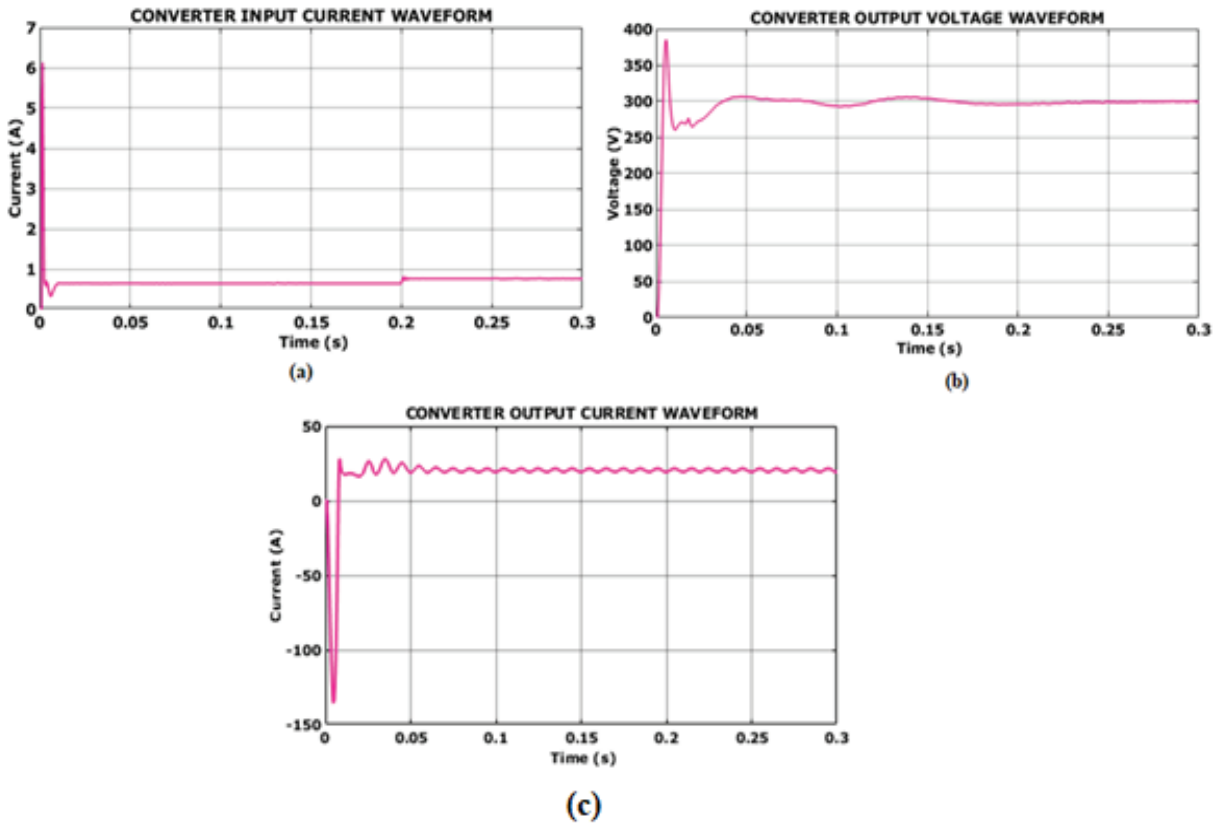


Figure 18. Converter waveform for case 4

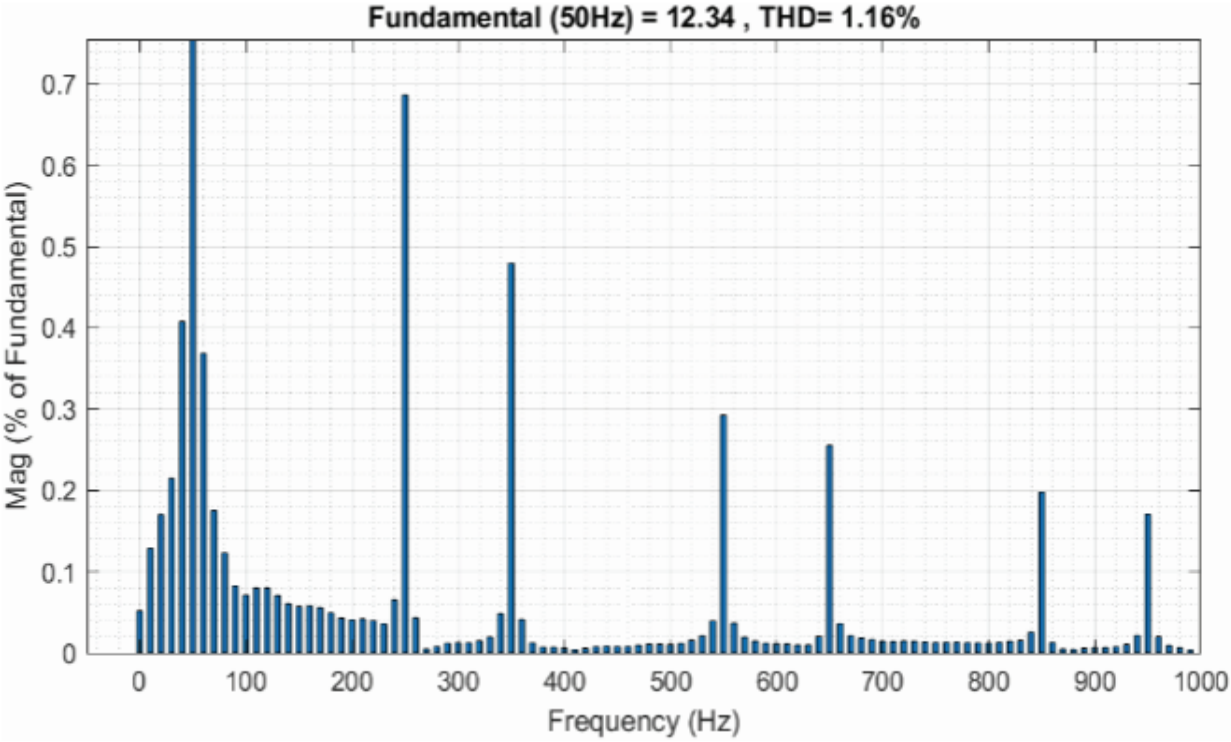


Figure 19. THD waveform for the proposed work

MPPT under varying ecological conditions, maximizing the energy harvested from the PV system with high tracking efficiency. The developed system is validated through MATLAB/Simulink, and the outcomes are compared with other traditional approaches, showing the superiority of the developed system. From the comparison results, the effectiveness of a novel cascaded ANFIS-RBFNN based MPPT exhibits superior tracking efficiency of 99.61% capabilities, convergence speed, execution time with improved

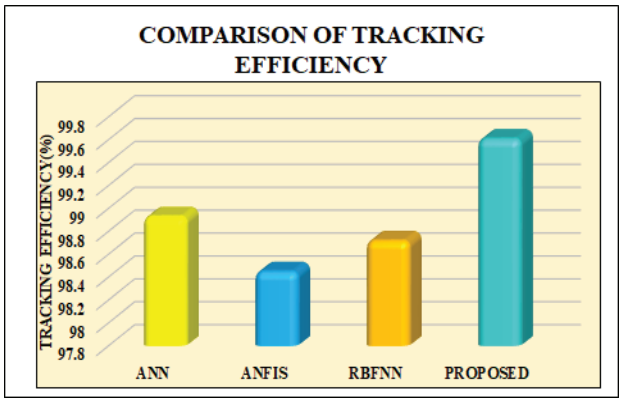


Figure 20. Comparison of tracking efficiency

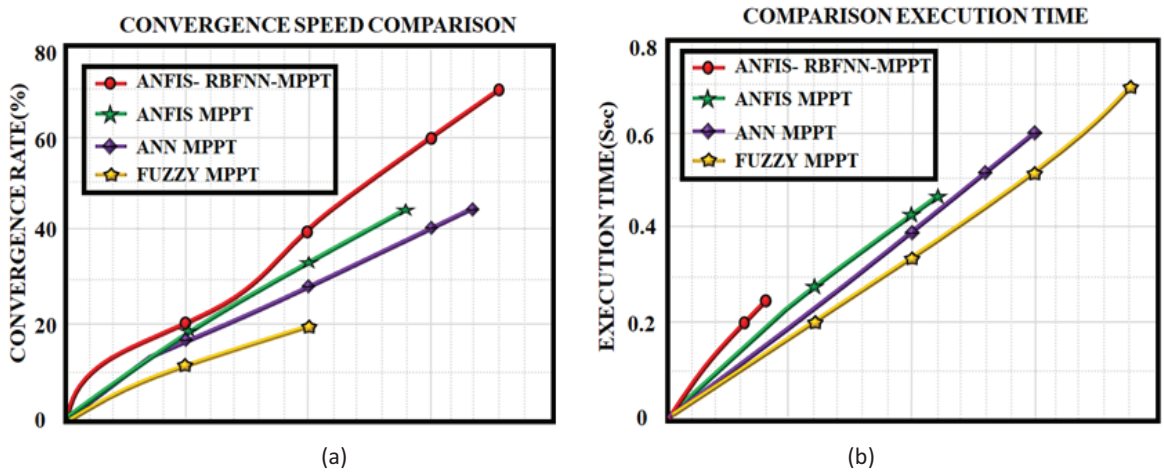


Figure 21. Comparison of (a) convergence speed and (b) execution time

steady-state and dynamic responses. Also the SEPIC achieves a low THD value of 1.16% compared to the others. Thereby, the enhanced and tracked power is efficiently delivered to the grid system without any disturbances.

AUTHORS

Blessy A. Rahiman* – Department of Electrical and Electronics Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India, e-mail: blessyrahiman@yahoo.com.

J. Jayakumar – Department of Electrical and Electronics Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India, e-mail: jayakumar@karunya.edu.

R. Meenal – Department of Electrical and Electronics Engineering, SRM TRP Engineering College, Trichy, Tamil Nadu, India, e-mail: meenasekar5@gmail.com.

*Corresponding author

References

[1] W. S. Ebhota and T-C. Jen, "Fossil fuels environmental challenges and the role of solar photovoltaic technology advances in fast tracking hybrid renewable energy system," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 7, 2020, 97–117, DOI: 10.1007/s40684-019-00101-9.

[2] M. Alaraj, A. Kumar, I. Alsaidan, M. Rizwan, and M. Jamil, "Energy production forecasting from solar photovoltaic plants based on meteorological parameters for Qassim Region, Saudi Arabia," *IEEE Access*, vol. 9, 2021, 83241–83251. DOI: 10.1109/ACCESS.2021.3087345.

[3] A. Nawaz, G. Hafeez, I. Khan, K. Ullah Jan, H. Li, S. Ali Khan, and Z. Wadud, "An intelligent integrated approach for efficient demand side management with forecaster and advanced metering infrastructure frameworks in smart grid," *IEEE Access,* vol. 8, 2020, 132551–132581. DOI: 10.1109/ACCESS.2020.3007095.

[4] N. Babu, J. M. Guerrero, P. Siano, R. Peesapati, and G. Panda, "A novel modified control scheme in grid-tied photovoltaic system for power quality enhancement," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, 2020, 11100–11110. DOI: 10.1109/TIE.2020.3031529.

[5] A. A. E. Tawfiq, M. O. Abed El-Raouf, M. I. Mosaad, A. F. Abdel Gawad, and M. Abd Elfatah Farahat, "Optimal reliability study of grid-connected PV systems using evolutionary computing techniques," *IEEE Access*, vol. 9, 2021, 42125–42139, DOI: 10.1109/ACCESS.2021.3064906.

[6] J. Jayakumar, "Optimized wind energy system integration with VSCHVDC for power," in *Transmiss. Proc. Eng. Sci.*, vol. 6, no. 2, 2024, 711–721, doi: 10.24874/PES06.02.028.

[7] A. Moghassemi, S. Padmanaban, V. K. Ramachandaramurthy, M. Mitolo, and M. Benbouzid, "A novel solar photovoltaic fed TransZSI-DVR for power quality improvement of grid-connected PV systems," *IEEE Access*, vol. 9, 2020, 7263–7279, doi: 10.1109/ACCESS.2020.3048022.

[8] J. J. Kumar, "Random forest machine learning algorithm based seasonal multi-step ahead short-term solar photovoltaic power output forecasting," *IET Renewable Power Gener.*, 2024. DOI: org/10.1049/rpg2.12921.

[9] C. Zhong, Y. Zhou, and G. Yan, "A novel frequency regulation strategy for a PV system based on the curtailment power-current curve tracking algorithm," *IEEE Access*, vol. 8, 2020, 77701–77715, DOI: 10.1109/ACCESS.2020.2989785.

[10] A. Kulshreshtha, A. R. Saxena, and M. Veerachary, "Non-isolated fourth-order boost DC-DC converter for power management in low voltage low power DC grids: design and interaction analysis," *IEEE Access*, vol. 8, 2020, 196500–196514, DOI: 10.1109/ACCESS.2020.3034181.

[11] J. Singh and V. K. Tiwari, "A new design based on grid integrated solar PV array using vector control," in *Adv. Energy Technol: Select Proc.gs of EMSME*, vol. 2020, 2022, 653–661, DOI: 10.1007/978-981-16-1476-7_58

[12] S. Gangavarapu and A. K. Rathore, "Three-phase buck–boost derived PFC converter for more electric aircraft," IEEE Trans. Power Electron., vol. 34, no. 7, 2018, 6264–6275, DOI: 10.1109/TPEL.2018.2877509.

[13] S. Gangavarapu, A. K. Rathore, and D. M. Fulwani, "Three-phase single-stage-isolated Cuk-based PFC converter," *IEEE Trans. on Power Electron.*, vol. 34, no. 2, 2018, 1798–1808, DOI: 10.1109/TPEL.2018.2829080.

[14] C. Cui, Y. Tang, Y. Guo, H. Sun, and L. Jiang, "High step-up switched-capacitor active switched-inductor converter with self-voltage balancing and low stress," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, 2021, 10112–10128, DOI: 10.1109/TIE.2021.3135611.

[15] S. Habib, M. M. Khan, F. Abbas, A. Ali, M. Talib Faiz, F. Ehsan, and H. Tang, "Contemporary trends in power electronics converters for charging solutions of electric vehicles," *CSEE J. Power Energy Syst.*, vol. 6, no. 4, 2020, 911–929, DOI: 10.17775/CSEEJPES.2019.02700.

[16] S. Obukhov, A. Ibrahim, A. A. Zaki Diab, A. Saad Al-Sumaiti, and R. Aboelsaud, "Optimal performance of dynamic particle swarm optimization based maximum power trackers for stand-alone PV system under partial shading conditions," *IEEE Access,* vol. 8, 2020, 20770–20785, DOI: 10.1109/ACCESS.2020.2966430.

[17] F. T. Josh, "Fractional order sliding mode control for power quality improvement in the distribution system," Int. J. Appl. Power Eng., vol. 13, no. 2, 2024, 408–414, DOI: 10.11591/ijape.v13.i2.pp408-414.

[18] J. Jayaraj, D. Obulesu, H. Govindaraj, F. T. Josh, N. Rajeswaran, C. R. Reddy, A. S. Algarni, A. Alwabli, and S. F. Malky, "A hybrid intelligent controller for extended-range electric vehicles," *Eng., Technol. Appl. Sci. Res.*, vol. 14, no. 2, 2024, 13408–13415, DOI: org/10.48084/etasr.6960

[19] V. Manikandan, "Sustainable energy development prediction of energy harvesting system with an adaptive hierarchical recurrent network and biodynamic fusion optimisation algorithm," *J. Environ. Protection Ecology*, vol. 24, no. 8, 2023, 2796–2805.

[20] M. T. Hussain, A. Sarwar, M. Tariq, S. Urooj, A. BaQais, and M. A. Hossain, "An evaluation of ANN algorithm performance for MPPT energy harvesting in solar PV systems," *Sustainability*, vol. 15, no. 14, 2023, 11144, DOI: org/10.3390/su151411144.

[21] S. R. Kiran, C. H. Basha, V. Pratap Singh, C. Dhanamjayulu, B. R.Prusty, and B. Khan, "Reduced simulative performance analysis of variable step size ANN based MPPT techniques for partially shaded solar PV systems," *IEEE Access,* vol. 10, 2022, 48875–48889, DOI: 10.1109/ACCESS.2022.3172322.

[22] F. Mehmood, N. Ashraf, L. Alvarez, T. Nadeem Malik, H. Khaliq Qureshi, and T. Kamal, "Grid integrated photovoltaic system with fuzzy based maximum power point tracking control along with harmonic elimination," *Trans. Emerg.Telecommun. Technol.,* vol. 33, no. 2, 2022, e3856, DOI: org/10.1002/ett.3856.

[23] S. A. Ibrahim, A. Nasr, and M. A. Enany, "Maximum power point tracking using ANFIS for a reconfigurable PV-based battery charger under non-uniform operating conditions," *IEEE Access*, vol. 9, 2021, 114457–114467, DOI: 10.1109/ACCESS.2021.3103039.

[24] C. Rao, A. Hajjiah, M. A. El-Meligy, M. Sharaf, A. T. Soliman, and M. A. Mohamed, "A novel high-gain soft-switching DC-DC converter with improved P&O MPPT for photovoltaic applications," *IEEE Access*, vol. 9, 2021, 58790–58806, DOI: 10.1109/ACCESS.2021.3072972.

[25] P. K. Pathak, S. Padmanaban, A. K. Yadav, P. A. Alvi, and B. Khan, "Modified incremental conductance MPPT algorithm for SPV-based grid-tied and stand-alone systems," *IET Gener., Transmiss. Distrib.*, vol. 16, no. 4, 2022, 776–791, DOI: org/10.1049/gtd2.12328.

[26] S. Sarwar, M. Y. Javed, A. B. Asghar, W. Iqbal, K. Ejsmont, and M. H. Jaffery, "A coronavirus optimization (CVO) algorithm to harvest maximum power from PV systems under partial and complex partial shading conditions," *Energy Rep.*, vol. 11, 2024, 1693–1710, DOI: org/10.1016/j.egyr .2024.01.043.

[27] P. K. Ganti, H. Naik, and M. K. Barada, "Hybrid TSA-RBFNN based approach for MPPT of the solar PV panel under the effects of tilt angles variations and environmental effects," *Int. J.Energy Res.*, vol. 45, no. 14, 2021, 20104–20131, DOI: org/10.1002/er.7089.

[28] N. I. Nahin, S. P. Biswas, S. Mondal, M. R. Islam, and S. M. Muyeen, "A modified PWM strategy with an improved ANN based MPPT algorithm for solar PV fed NPC inverter driven induction motor drives," *IEEE Access*,vol. 11, 2023, 70960–70976, DOI: 10.1109/ACCESS.2023.329.1339.

[29] C. B. N. Fapi, P. Wira, M. Kamta, A. Badji, and H. Tchakounte, "Real-time experimental assessment of hill climbing MPPT algorithm enhanced by estimating a duty cycle for PV system," *Int. J. Renewable Energy Res.*, vol. 9, no. 3, 2019, 1181-1189.

[30] Z. A. Khan, L. Khan, S. Ahmad, S. Mumtaz, M. Jafar, and Q. Khan, "RBF neural network based backstepping terminal sliding mode MPPT control technique for PV system," *PLoSOne*, vol. 16, no. 4, 2021, e0249705, DOI: org/10.1371/journal.pone.0249705.

High-Performance Electric Two-Wheeler Fast Charger Based on Intelligent Control Algorithm

Submitted: 26th August 2024; accepted: 17th February 2025

Subiyanto, Rizky Ajie Aprilianto, Mario Norman Syah, Bagaskoro Saputro, Abdurrakhman Hamid Al-Azhari, Nektar Cahayasabda, Bayu Adi Pambudi, Faiq Mananul Faqih, Icha Arifah Annisa, Dwi Bagas Nugroho, Siva Khaaifina Rachmat, Dewi Anggriani

DOI: 10.14313/jamris-2026-030

Abstract:
Conventional electric two-wheeler (E2W) chargers suffer from prolonged charging times and battery nonlinear characteristics, limiting user flexibility and compliance with evolving battery standards. This work developed a high-performance fast charger utilizing an interleaved buck converter (IBC) governed by a hybrid proportional integral–fuzzy logic control (PI-FLC) algorithm. The PI-FLC dynamically optimizes charging current/voltage by integrating real-time battery current, voltage, and state-of-charge (SoC) data. The IBC architecture minimizes output current ripple, enabling compact filter design. The hybrid algorithm maintains a particular voltage and current threshold, ensuring compliance with E2W battery characteristics. The proposed system achieves 0–100% SoC in 57.75 minutes—a 67.9% and 75.9% reduction compared to PID CC-CV (180 min) and PI-CV (240 min), respectively. Moreover, the proposed method outperforms several cutting-edge charging methods, such as a Zeta converter with sliding mode control (SMC) (98.1% efficiency, 80 min), Quasi-Resonant converters with hysteresis control (97.5%, 70 min), Interleaved Boost with FLC (98.5%, 65 min), and Dual Active Bridge with MPC (97.9%, 75 min). Furthermore, hardware implementation demonstrated 98.8% efficiency at 0.22 C-rate and empirically validated charger compatibility across E2W battery topologies. This work bridges the gap between rapid charging demands and battery longevity, offering a scalable solution for next-generation E2W ecosystems.

Keywords: *electric two-wheelers (E2W), fast charger, interleaved buck converter (IBC), proportional integral–fuzzy logic control (PI-FLC)*

1. Introduction

The petroleum transition as the primary energy source for the transportation sector has been a crucial issue. It is conducted as a response to escalating concerns regarding climate change [1]. Decarbonizing by adopting electric vehicles (EVs) is an environmentally sustainable and practical solution [2]. It is becoming increasingly popular worldwide, heralding a new era of automotive sustainability. Concurrently, initiatives in renewable energy systems are being implemented globally [3]. Hence, the development of EV technology has been massively conducted to result in superior performance.

As a part of EV development, electric two-wheelers (E2Ws), including electric bicycles/electric motorcycles [4], emerge as the preferable alternative for countries such as Indonesia due to the socio-economic conditions and the existing transportation infrastructure. More than 2000 EVs passed the feasibility test in 2020 by the Directorate General of Land Transportation Indonesia, and the E2Ws are the highest unit among others [5]. In addition, the sales of E2Ws are higher than those of other EV types.

The application of E2Ws is intrinsically linked to using batteries as an energy storage medium. Batteries are designed to be rechargeable, necessitating appropriate charging devices [6]. EV charging duration is influenced by battery capacity, charger power, and battery technology, where larger batteries take longer, high-power chargers speed up charging, and charging efficiency depends on the power converter [7]. Choosing a converter topology and a suitable charging method are critical considerations in obtaining an appropriate charging device. The technology selection refers to the power converter topology, which has the capability of converting and conducting power optimally. Various control techniques have been developed to enhance the efficiency and stability of E2Ws DC-DC converters, such as Voltage Mode Control (VMC), Current Mode Control (CMC), PID, Sliding Mode Control (SMC), and Fuzzy Logic Control (FLC)[8].

Converter efficiency is a primary parameter in power conversion systems that measures the ratio between the obtained output power and the input power used. According to [9], high-efficiency converters can reduce power losses and improve the overall performance of the power system. In [10] it was added that converter efficiency is inversely proportional to the total power loss in the converter, making it a key factor in designing optimal electrical power systems. Furthermore, [11] explained that increasing converter efficiency can be achieved by optimizing circuit topology and reducing parasitic resistance in power components. Meanwhile, the most prevalent strategy for recharging batteries has been adopted, which is constant current–constant voltage (CC-CV).

Power converters play a critical role in modifying, controlling, and conditioning electrical power within power systems by adjusting voltage levels. This adjustment can take the form of increasing (boost) or decreasing (buck) voltage or a combination of both (buck-boost) [12]. Recent research, highlighted in [13], introduces a current-fed non-isolated DC–DC converter design that employs fewer switching components and utilizes Zero Voltage Switching (ZVS) and Zero Current Switching (ZCS) techniques to mitigate power losses. This methodology incorporates a Coati-optimized Fractional Order Proportional-Integral-Derivative (FOPID) controller. However, it still encounters challenges, including switch voltage stress and limited adaptability to various battery types, indicating a need for further optimization.

Additionally, [14] presents a bidirectional DC–DC converter utilizing Fractional Order Resonant (FOPR) control in conjunction with ZVS–ZCS techniques. This combination yields high efficiency, reduces power loss, and enhances stability. However, it faces ongoing challenges related to practical implementation and experimental validation. Addressing those shortcomings, previous studies have focused on developing fast-charging strategies, such as [15] proposed design of an electric bike charger based on a CUK converter operating in discontinuous conduction mode (DCM) CC–CV charging solution capable of charging 0-100% SoC of a 48 V, 20 Ah battery 2 hours. However, only batteries with certain specifications can accept DCM conditions. In [16], the design and implementation of a battery charger utilize SoC estimation and FLC charging. However, this method also requires around 1.5 hours to recharge 0-100% of the SoC battery to get the appropriate power for battery charging topology. [17] presents an ANFIS-based charging algorithm to increase charging speed. The complexity of ANFIS-based chargers is unreliable to conventional users because they potentially increase the charging price.

Furthermore, available conventional E2W chargers take approximately 2-9 hours to recharge 0-100% of their capacity. It is necessary that the charging time be faster to fulfil the user's high mobility and be safe and reliable by considering the battery's charging current capacity. Previous studies, such as in [16–18] have presented a suitable method for fast charging but still need to consider the specifications and capacity of the batteries on the market.

This study introduces a novel intelligent fast charger for E2Ws, utilizing a three-phase interleaved synchronous buck converter (3Phase IBC) for efficient power conversion and rapid, safe charging. The system is controlled by a proportional integral-fuzzy logic control (PI-FLC) algorithm.

2. E2W Charging Architecture

Figure 1 illustrates the selected charger topology suitable for various E2W devices. The selected universal off-board charging topology, as shown in Fig. 2, comprises a linear transformer, AC–DC converter, and

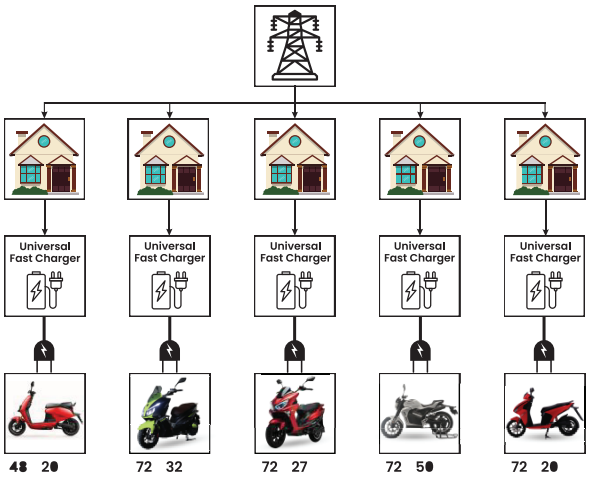


Figure 1. Charger application

DC–DC converter. The three-phase interleaved DC–DC converter is designed in a continuous current mode (CCM) where each switching device phase is shifted to 120° in accordance with the duty cycle $d_t(s)$ generated using the fuzzy logic algorithm. For configurations involving a greater number of phases, the phase shift can be accordingly adjusted to 360°/p, where *(p)* represents the total number of interleaved buck converter phases [18].

Circuit averaging is performed to replace the switches with their average model. However, in practical conditions, each phase contains two switching devices containing on-state resistance and an inductor with an inductor series with resistance. The equivalent of the series resistance (ESR) in each phase is in Eq. (1) [19, 20].

$$R_{sp} = R_{swp} + R_{Lp}, p = 1, 2, 3 \tag{1}$$

where R_{sp} is total converter resistance characteristic in each converter phase that consists of switching device resistance R_{swp} and inductor internal resistance R_{Lp}.

Eq (1) is assumed as the parasitic capacitor C_o resistance is minimal and shared with each phase, which could be neglected. A continuous DC input voltage source supplies IBC to simplify the analysis process. The L_1, L_2, and L_3 values are equal and will be denoted as L_s. Utilizing KCL and looking at Figure 2, the charging current i_o is described as the total amount of inductor current $i_L(t)$ in each phase determined as Eq. (3) for calculating the total amount of current during a period or Eq. 4 to calculate the charging current at a specific time [18, 19, 21].

$$i_o(t) = \sum_p i_L(t) \approx i_o \tag{2}$$

$$i_o = i_{L1} + i_{L2} + i_{L3} \tag{3}$$

Eq. (2) can be derived to prescribe the charging voltage v_o as Eq. (4).

$$v_o = i_o \frac{r}{rC_o + 1} \tag{4}$$

where r is the battery's internal resistance, and C_o is the DC–DC output capacitor.

Each IBC phase constantly shifted at 120° among itself. The peak output current ripple in a complete span duty cycle for each phase $d_1(t)$, $d_2(t)$, $d_3(t) \in [0, 1]$ written as Eq. (5) [22].

$$\Delta i_o\left(d_p\left(t\right)\right) = \frac{V_{Co}}{2L_s f_{sw}}\left(d_p\left(t\right) - \frac{k-1}{p}\right) \cdot \left(1 - p\left(d_p\left(t\right) - \frac{k-1}{p}\right)\right) \quad (5)$$

Eq (5) is used for the following conditions: $\frac{k-1}{p} \leq d_t\left(s\right) \leq k/p$, where $k = 1, \ldots, p$, f_{sw} is frequency switching, V_{Co} is the voltage at the capacitor output filter, and $d_p\left(t\right)$ represents the duty cycle in one of the converter phases at a given time.

The IBC topology significantly mitigates inductor current ripple through its inherent ripple cancellation feature [23]. Increasing the number of phases within the IBC can effectively decrease the peak value of the current ripple output. Phase segmentation reduces the inductor value within the converter, thereby preventing degradation in the response stability of the converter, which can occur due to the energy charging and discharging cycles in a large inductor [24]. Furthermore, as illustrated in Eq. (6), the theoretical framework indicates that the output current ripple of the IBC can approach zero by augmenting the number of phases. This characteristic is particularly beneficial in designing and implementing high-performance fast charger topologies intended for supplying large currents to batteries. The correlation between the maximum output current ratio and the maximum inductor current ripple within the IBC is quantitatively described in Eq. (6).

$$r = \frac{\max(\Delta i_{ou} d_p\left(t\right))}{\max(\Delta i_L d_p\left(t\right))} = \frac{1}{p} \quad (6)$$

where $\Delta i_L d_p\left(t\right)$ is the inductor current written as Eq. (7)

$$\Delta i_L\left(d_p\left(t\right)\right) = \frac{v_o}{2L_s f_{sw}} d_p\left(t\right)\left(1 - d_p\left(t\right)\right) \quad (7)$$

The proposed IBC topology also incurs power losses in practical conditions, as mentioned in [23].

The power losses in the IBC are derived from the resistance equation in the converter in Eq. (1), represented as Eq. (8).

$$P_{loss} = P_c + P_{sw} \quad (8)$$

Where P_c is conduction losses $P_c = I_{rms}\ x\ R_{ON}$ with I_{RMS} as charging current root mean square (RMS) flowing through the switching device and R_{ON} is the switch's on-state resistance. P_{sw} stands for switching losses $P_{sw} = 0.5 \times V_D \times i_o \times \left(t_{don} + t_r + t_{doff} + t_f\right) \times f_{sw}$ with V_D as drain voltage, t_r as the switching device rise time t_{don} and t_{doff} is the time delay during the on and off periods, respectively.

3. Battery Parametric Model

The battery-selected model was initially developed in [25]. A detailed explanation and definition of the battery mathematical modelling can be found in the MATLAB documentation using Eq. (9); the battery open circuit voltage can be measured from the battery equivalent circuit in Figure 2.

$$V_{Bat_{oc}}\left(it, i*, i\right) = E_0 - K \cdot \frac{Q}{it + 0.1} \cdot i - K \cdot \frac{Q}{Q - it} \cdot it + A \cdot \exp\left(-B \cdot it\right) \quad (9)$$

Where E_0 is the nonlinear voltage input, E_0 is the constant voltage input, K is the polarization constant in (V/Ah), Q is the maximum battery capacity (Ah), A is the exponential voltage input, B is the exponential capacity (Ah^{-1}), $i*$ is the low-frequency current dynamic (A), it is the battery current (A), $\exp(s)$ is exponential zone dynamics (V). The SoC, or battery nominal present capacity, is the charge amount. It is 0% when the battery is fully discharged and 100% when fully charged. The Battery SoC in the time (t) is calculated using Eq. (10).

$$SoC(t) = \frac{Q\ i(t)}{Q}\ .\ 100\% \quad (10)$$

4. The Proposed Intelligent Control

The battery was in charging mode when the battery current exceeded zero $(i^* < 0)$, and the battery open circuit voltage, as shown in Eq. (10). To fulfil the proposed fast charging scenario for E2W batteries, the

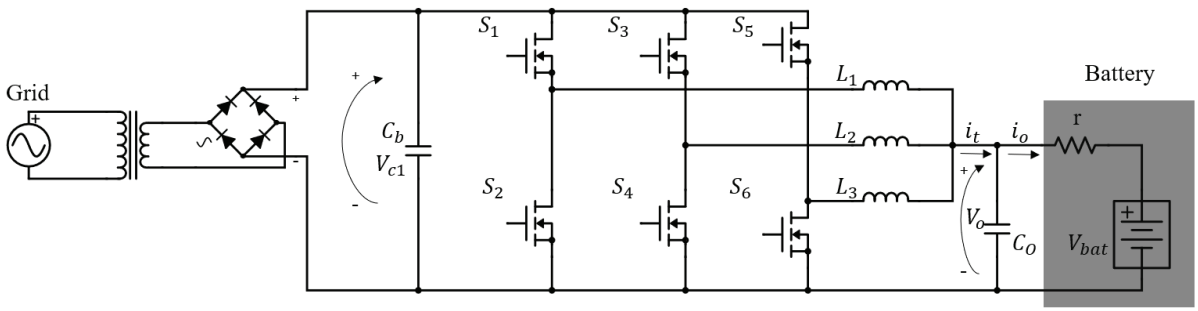


Figure 2. Proposed charger architecture

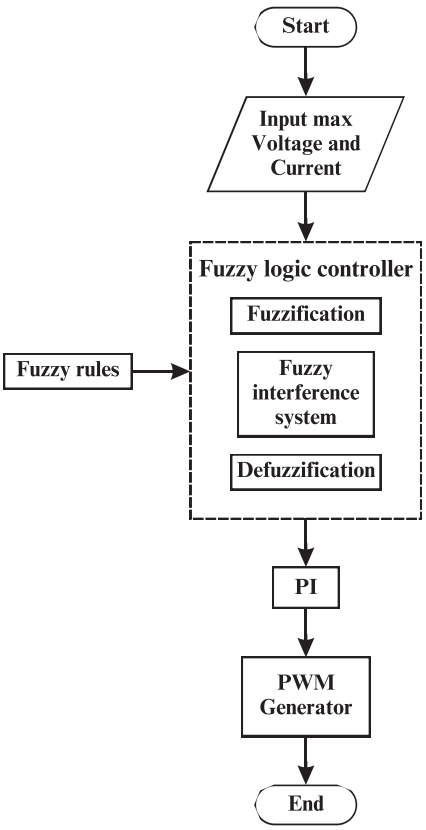


Figure 3. Charging Algorithm

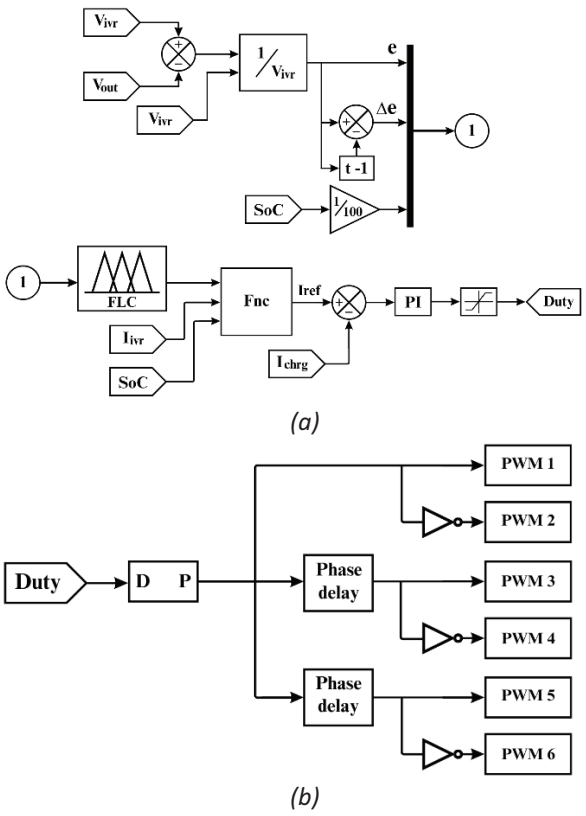


Figure 4. (a) Charging control block, (b) PWM generator

PI-FLC algorithm ensures that the system meets different charging standards for each E2W battery.

In contrast with the conventional CC–CV method, which prioritizes rapid charging solely during the 0–80% SoC range. As depicted in Figure 3, this work implements adaptive current profiling charging to mitigate exorbitant charging currents in low SoC (0–20%) and overvoltage risks in high SoC (80–100%). The three-stage charging protocol operates as follows: (a) normal-rate charging (0-0.8 C) at 0–20% SoC to prevent exorbitant charging current from damaging the battery; (b) accelerated charging (0.8-1 C) at 20–80% SoC to minimize the duration; and (c) tapered charging (0-0.5 C) at 80–100% SoC to avoid battery damage due to extensive charging current.

Figure 4 illustrates the output of FLC, which is a reference C-rate. This C-rate is then multiplied by the battery capacity I_{ivr} to establish the reference charging current value. This reference current is successfully compared with the measured current to generate a current error, which will be the PI control input. The PI control calculates the appropriate duty cycle to achieve the desired reference current and voltage. Furthermore, the PI-FLC hybrid controller dynamically resolves nonlinear battery dynamics by combining PI-based voltage regulation with FLC-driven multi-constraint optimization.

Charging parameters are fuzzified into membership functions, enabling the FLC to autonomously select charging modes through 75 rule-based decisions. This dual-loop architecture compensates for PI controllers' inherent limitations in adaptive charging parameters under battery nonlinear characteristics.

The PI controller used in the proposed method is conventional PI, where $PI = K_P + K_I 1/s$ which is designed to obtain a monotonic response. In particular, the values of $K_p = 0.75$ and $K_I = 50$, respectively, and the PI output is limited with the saturation value of 0-0,97. As depicted in Figure 5, the FLC algorithm had three membership inputs membership function: error voltage described as $\mu_e = (V_{ivr} - V_O)/\ V_{ivr} \in [-1\ 1]$ where V_{ivr} is the battery voltage set point at charging mode, delta error is differential between measured μ_e and μ_e at t-1 described as $\mu_{\Delta e} = \mu_e - \mu_e(t-1) \in [-1\ 1]$, and battery SoC is described as $\mu_{SoC} = SoC(t)/100) \in [0\ 1]$ fuzzy membership output is the reference C-rate $\mu_{i_{ivr}} \in [0\ 1]$.

Table 1 illustrates the normal rate charging conditions in which the current and voltage rise exponentially toward the battery charging voltage and maximum charging capacity. Table 2 illustrates the accelerated charging condition securing the maximum charging current and voltage. Table 3 illustrates the tapered charging condition where the current drops slowly from the maximum charging current to zero when the battery SoC reaches 100% and maintains the set point charging voltage.

5. Results and Discussion

The proposed system was validated using MATLAB/Simulink and simulated based on the charger

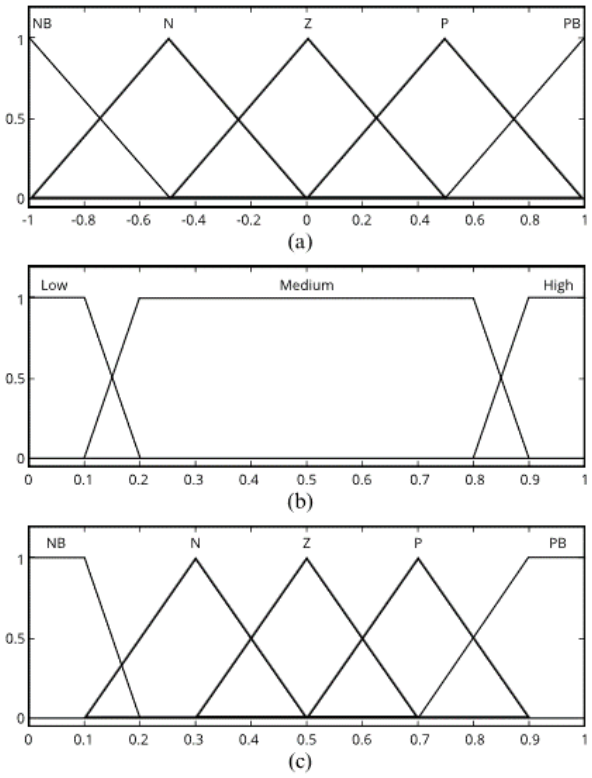


Figure 5. Fuzzy membership function (a) input error and Δerror, (b) input SoC, (c) Output

Table 1. The fuzzy rule for low SoC level

SoC: Low		error				
		NB	**N**	**Z**	**P**	**PB**
Δ error	**NB**	Z	Z	Z	Z	Z
	N	Z	N	N	Z	P
	Z	Z	Z	Z	P	PB
	P	Z	Z	P	PB	PB
	PB	Z	P	PB	PB	PB

Table 2. The fuzzy rule for medium SoC level

SoC: Medium		error				
		NB	**N**	**Z**	**P**	**PB**
Δ error	**NB**	PB	PB	PB	PB	PB
	N	PB	PB	PB	PB	PB
	Z	PB	PB	PB	PB	PB
	P	PB	PB	PB	PB	PB
	PB	PB	PB	PB	PB	PB

Table 3. The fuzzy rule for high SoC level

SoC: High		error				
		NB	**N**	**Z**	**P**	**PB**
Δ error	**NB**	NB	NB	NB	NB	NB
	N	NB	NB	NB	NB	N
	Z	NB	NB	NB	NB	N
	P	NB	NB	NB	N	N
	PB	NB	NB	N	N	N

parameters outlined in Table 4. Furthermore, the proposed charging topology has also been validated by hardware implementation.

Table 4. The charger simulation parameters

Parameter	**Value**
RMS Input Voltage	220-230V 50Hz (AC)
AC Transformator	1:2
Rectifier	2 kW
Switching device	MOSFET N-type Rd = 0.01 Ω Rs = 1e5 Ω
Inductor L_s	1e-3H
Capacitor output C_O	200e-6F
Voltage output V_o	48-84V
Output current i_o	0-20A

5.1. Simulation Implementation Result

The proposed system was developed using MATLAB/Simulink, incorporating charger parameters delineated in Table 4. A nickel-manganese-cobalt (NMC) battery [25], configured in a 20s8p topology (nominal voltage of 72 V and a capacity of 20.4 Ah), was utilized for testing purposes. Comparative analyses were conducted against the PID CC-CV and PI CC algorithms. These assessments evaluated charging current and voltage dynamics, transient response, and total charging duration to ensure a rigorous and unbiased performance validation.

Figure 6 demonstrates the transient voltage response to an 84 V setpoint. The proposed method effectively achieves voltage stabilization within 1.25 ms, exhibiting a response time twice as rapid as the PID CC–CV algorithm, which stabilizes at 2.5 ms. In contrast, the PI-CV method fails to reach convergence, displaying a steady-state error of 2.3%. These results underscore the superior transient response of the proposed method.

Table 5 illustrates that the current regulation performance across the entire SoC range is quantitatively assessed. The proposed system maintains a current deviation of ± 4 A at 20% SoC while operating at a 1 C-rate (20.4 A) during the accelerated charging phase. Furthermore, the current throttling rate is 0-0.8 C-rate and 0-0.5 C-rate at 0% and 80% SoC, respectively. As depicted in Figure 7, the proposed method effectively mitigates risks associated with battery thermal run and cell damage. The system achieves current tapering to 0 A at 100% SoC.

Charging time comparisons presented in Figure 7 reveal that the PI-FLC algorithm accomplishes a 0 to 100% SoC charging duration of 57.75 minutes. This achievement represents a significant reduction of 67.9% and 75.9% compared to the PID CC-CV (180 min) and PI-CV (240 min) methods, respectively. The performance of the proposed system exceeds that of traditional PID CC-CV chargers currently implemented in commercial E2W systems [26], thus demonstrating its viability for industrial applications.

5.2. Hardware Implementation Result

The proposed hardware implementation of the 3Phase IBC-based charger was developed to validate theoretical performance metrics and demonstrate

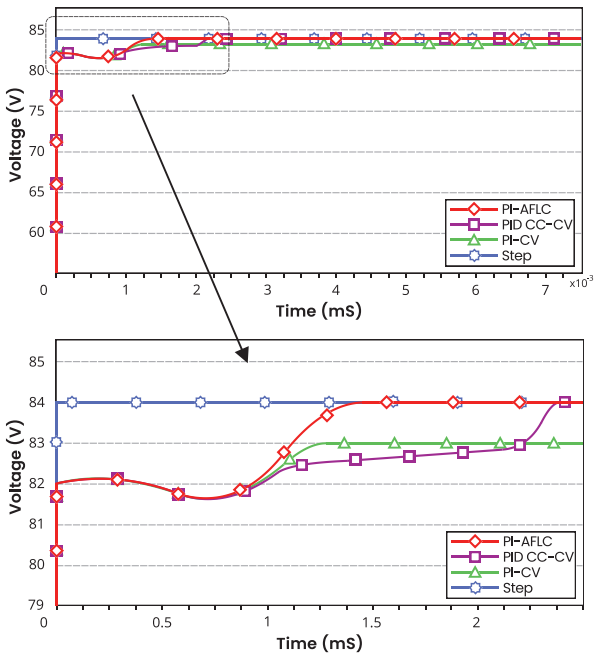


Figure 6. The comparison of charging voltage characteristics

Table 5. The charging current characteristics of the proposed algorithm

SoC (%)	Peak Current (A)	Average Charging Current (A)	C-Rate	Charging Voltage (V)
0	13	10	0.5	84
20	24	20	1	84
40	23	20	1	84
60	22	20	1	84
80	15	10	0.5	84
100	0	0	0	84

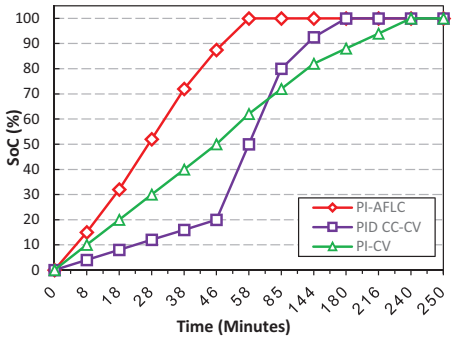


Figure 7. Charging time performance

practical applicability for E2W systems. The experimental setup encompasses modular architecture featuring an AC-DC rectification stage, a 3Phase IBC, precision sensing modules for real-time voltage and current monitoring, and a microcontroller unit (MCU) executing a hybrid PI-FLC algorithm. Operating from a standard 220–230 V AC (50 Hz) input, the system rectifies to a 30 V DC bus and delivers a regulated 21 V DC output to charge a 5-series, 1-parallel (5S1P) lithium-ion battery array.

Table 6. The charger hardware implementation parameters

Parameter	Value
RMS Input Voltage	220-230V 50Hz (AC)
Voltage Input	30 V
Current Input	0.4 A
Power Input	12 W
Voltage Output	21 V
Current Output	0.564 A
Output Capacitor	100 uF
Inductor	60 uH
Switching Device	MOSFET
Switching Frequency	30 kHz
Battery Array	5 series 1 parallel
Power Output	11.85 W
Charger Efficiency	98.8 %

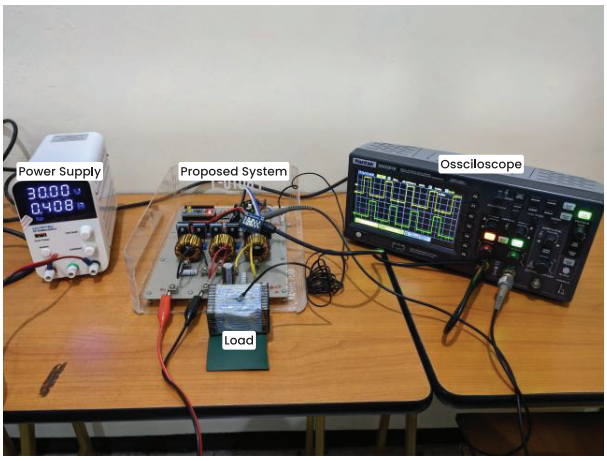


Figure 8. Hardware experimental setup

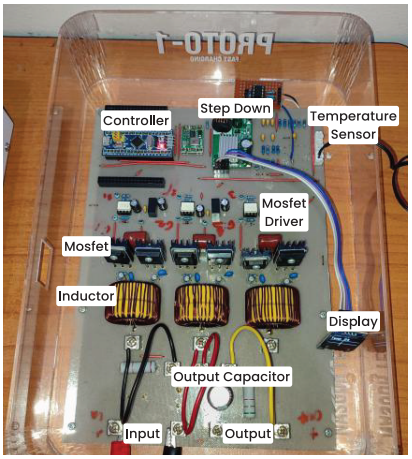


Figure 9. Hardware implementation of the proposed charging topology

The experimental setup is illustrated in Figure 8. The hardware parameters outlined in Table 6 demonstrate an input power of 12 W (30 V, 0.4 A), yielding an output of 11.85 W (21 V, 0.564 A) and achieving an exceptional efficiency of 98.8%. This performance is attributed to the low on-resistance of the MOSFETs, high-frequency operation at 30 kHz, minimizing core losses in the 60 µH inductor, and reduced ripple current enabled by the 100 µF output capacitor.

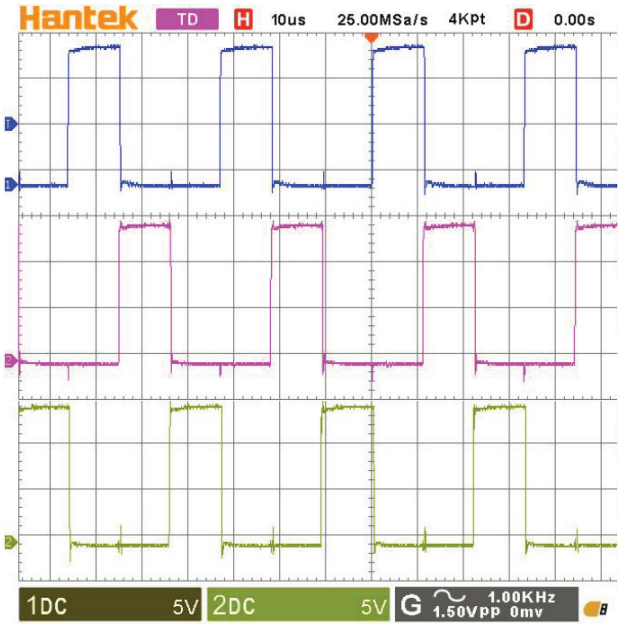


Figure 10. Three phases interleaved PWM signal result 30Khz

Figure 9 provides a granular view of hardware realization, highlighting the three-phase interleaved PWM signals generated by the MCU. The interleaved operation at 30 kHz ensures continuous input current, minimizes input/output voltage ripple, and distributes thermal stress evenly across the three phases. This configuration enhances power density and mitigates electromagnetic interference (EMI). The stable PWM waveform alignment with the control algorithm's dynamic adjustments confirms the robustness of the PI-FLC in adapting to real-time battery SoC and voltage fluctuations.

Figure 10 depicts the steady-state input voltage waveform, which remains consistently regulated at 30 V DC despite variations in load conditions. This stability is critical for maintaining the integrity of the AC-DC rectification stage and ensuring reliable power delivery to the IBC. Meanwhile, Figure 11 illustrates the output voltage waveform, showcasing the charger's ability to maintain a precise 21 V DC under dynamic loading. Notably, the absence of significant overshoot or oscillations during transient responses validates the hybrid PI-FLC algorithm's effectiveness in enforcing tight voltage and current thresholds, even during rapid SoC changes. The interleaved architecture and optimized LC filter achieve minimal voltage ripple at the output.

Charging efficiency (η) and charging time (t) are critical metrics in EV charging systems. Efficiency quantifies power conversion effectiveness while charging time reflects the system's ability to deliver energy rapidly. This study compares recent converter topologies and control algorithms, proposing a novel 3-phase IBC with PI-FLC for enhanced performance. The charging efficiency is described in (11)

$$\eta = \frac{P_{chrg}}{P_{in}} \cdot 100\% \tag{11}$$

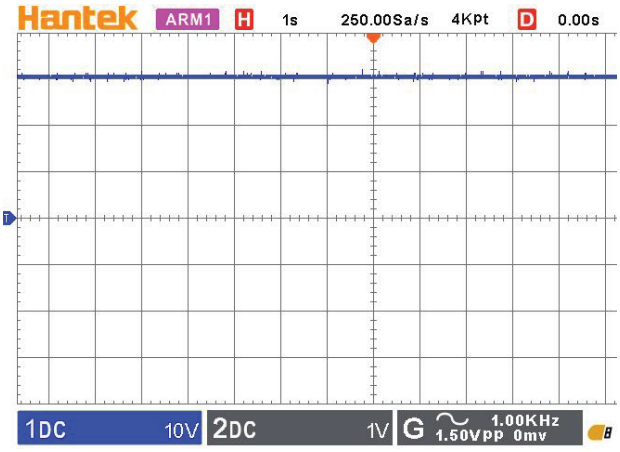


Figure 11. Charger input voltage

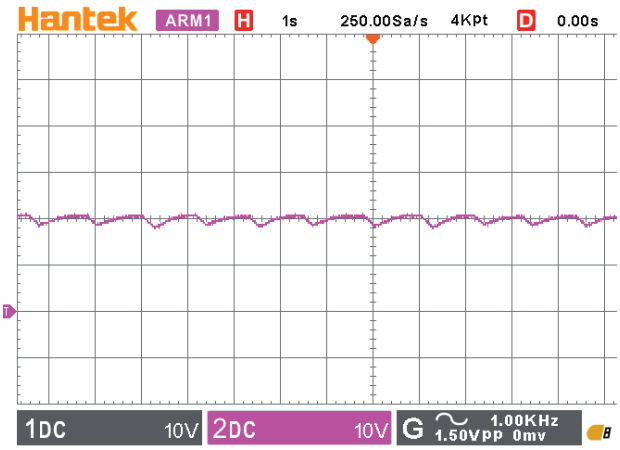


Figure 12. Charger output voltage

Where P_{chrg} is the charging power and P_{in} is the charger input power. Charging time from 0% to 100% SoC depends on battery energy capacity (E_{bat}) and effective charging power ($P_{chrg} \times \eta$) mathematically modelled as (12).

$$t = \frac{E_{bat}}{P_{chrg}\,\eta} \cdot 60(\text{min}) \tag{12}$$

Eq. (11) and (12) are then used to calculate the efficiency and charging speed of the charger in various previous methods which are then included in Table 7. This table presents a comparison between the previous charging methods and the proposed method, proving the efficacy of the proposed method.

Table 7 compares various power converters and control algorithms in EV charging systems from 2020 to 2025. Various converters from previous studies were compared to evaluate their performance in charging batteries. The use of diverse algorithms resulted in variations in charging efficiency. The battery specifications are sometimes different, by calculating the charging time using Eq. (12), although each algorithm is tested using different battery specifications, Eq. 12 will provide the same charging time value according to the performance and capacity of the charger. Efficiency ranges from 92.3% to 98.8%, with charging times between 65 and 175 minutes.

Table 7. Comparison between the proposed method and previous methods

Ref.	Year	Converter Type	Algorithm	Efficiency (%)	Charging Time 0-100% SoC (min)
[27]	2012	Boost	Neural Network	97.8	85
[28]	2018	Flyback	PSO	94.2	100
[29]	2019	Quasi-Resonant	Hysteresis Control	97.5	70
[15]	2020	Cuk	DCM	-	120
[30]	2020	Buck-Boost	FLC	95.5	90
[31]	2021	SEPIC	PID	92.3	110
[32]	2022	Interleaved Boost	FLC	98.5	65
[17]	2023	Buck	ANFIS	-	175
[33]	2023	Zeta	Sliding Mode Control	98.1	80
[34]	2024	LLC	GA	96.7	95
[35]	2024	Dual Active Bridge	MPC	97.9	75
Proposed	2025	3-Phase IBC	PI-FLC	**98.8**	**57.75**

The leading model for 2025, the 3-Phase Interleaved Boost Converter (IBC) with PI-FLC, achieves 98.8% efficiency and the fastest charging time of 57.75 minutes, demonstrating the advantages of the interleaved topology with PI-FLC.

6. Conclusion

The proposed high-performance fast charger for E2W, utilizing a 3 Phase IBC controlled with PI-FLC algorithm, significantly enhances charging efficiency and speed. The PI-FLC optimizes charging current and voltage dynamically by integrating real-time battery current, voltage, and SoC data. Achieving fast charging of 57.75 minutes for a 72 V 20 Ah NMC battery, this system demonstrates a 67.9% and 75.9% reduction in charging time compared to PID CC-CV (180 min) and PI-CV (240 min), respectively. Moreover, the proposed method outperforms several cutting-edge charging methods, such as a Zeta converter with SMC with 98.1% efficiency and 80 min charging time, Quasi-Resonant converters with hysteresis control with 97.5% efficiency and 70 min charging time, Interleaved Boost with FLC 98.5% efficiency and 65 min charging time, and Dual Active Bridge with MPC 97.9% efficiency and 75 min charging time. The hardware implementation demonstrated 98.8% efficiency during testing at a 0.22 C rate. The 3-phase IBC topology minimizes output current ripple and EMI, while the adaptable PI-FLC algorithm prevents overcharging by precisely regulating voltage and current. By addressing the balance between rapid charging requirements and battery longevity, this innovation presents a scalable, effective, and safe solution for the evolving demands of the E2W ecosystem, ultimately contributing to the sustainability of electric mobility.

AUTHORS

Subiyanto – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: subiyanto@mail.unnes.ac.id.

Rizky Ajie Aprilianto* – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: rizkyajiea@mail.unnes.ac.id.

Mario Norman Syah – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: marionormansyah@mail.unnes.ac.id.

Bagaskoro Saputro – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: bagaskoro.s@mail.unnes.ac.id.

Abdurrakhman Hamid Al-Azhari – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: abdurrakhmanhamid@mail.unnes.ac.id.

Nektar Cahayasabda – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: nektarcahayasabda@gmail.com.

Bayu Adi Pambudi – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: bayuadipambudi@gmail.com.

Faiq Mananul Faqih – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: faiqmanal77@gmail.com.

Icha Arifah Annisa – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: ichaarifah03@gmail.com.

Dwi Bagas Nugroho – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: dwibagasn@gmail.com.

Siva Khaaifina Rachmat – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: sivaakhaaifina@gmail.com.

Dewi Anggriani – Department of Electrical Engineering, Universitas Negeri Semarang, Semarang, 50229, Indonesia, e-mail: dewianggriani480@gmail.com.

*Corresponding author

References

[1] H. Tu et al., "Extreme Fast Charging of Electric Vehicles: A Technology Overview," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, 2019, pp. 861–878; doi: 10.1109/TTE.2019.2958709.

[2] A. Ghosh, "Possibilities and Challenges for the inclusion of the Electric Vehicle (EV) to Reduce the Carbon Footprint in the Transport Sector: A Review," *Energies (Basel)*, vol. 13, no. 10, 2020, pp. 1–2; doi: 10.3390/en13102602.

[3] M. Bharathidasan, V. Indragandhi, and B. Aljafari, "Hybrid Controlled Multi-Input DC/DC Converter for Electric Vehicle Application," *International Transactions on Electrical Energy Systems*, vol. 2023, pp. 1–16; doi: 10.1155/2023/8308418.

[4] D.N. Huu and V.N. Ngoc, "A Three-Stage of Charging Power Allocation for Electric Two-Wheeler Charging Stations," *IEEE Access*, vol. 10, 2022, pp. 61080–61093; doi: 10.1109/ACCESS.2022.3181731.

[5] I. Veza et al., "Electric Vehicles in Malaysia and Indonesia: Opportunities and Challenges," *Energies (Basel)*, vol. 15, no. 7, 2022, pp. 1–4; doi: 10.3390/en15072564.

[6] R. Swapnil and V.N. Kalkhambkar, "Optimal Operation Model of Vehicle to Vehicle Charging System," *Journal of Electrical and Electronics Engineering*, vol. 15, no. 1, 2022, pp. 52–58; https://www.proquest.com/openview/6abce8708f539109954a48200386c0ae/1?pq-origsite=gscholar&cbl=54417

[7] P. Sharma et al., "A Comprehensive Study on Electrical Vehicle in Charging Infrastructure, Challenges and Future Scope," *Electric Vehicles, Green Energy and Technology*, N. Patel et al., eds., Springer, 2021, pp. 1–285; doi: 10.1007/978-981-15-9251-5_16.

[8] P. Sharma, A. Kumar Sharma, and A.K. Vyas, "Comparative Study of DC-DC Converter With Different Control Techniques," *Recent Advances in Renewable Energy Sources*, P. Sharma, ed., RARES2021, 2021, pp. 1–6; doi: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3808564#

[9] F. Mumtaz et al., "Review on Non-isolated DC-DC Converters and their Control Techniques for Renewable Energy Applications," *Ain Shams University*, 2021; doi: 10.1016/j.asej.2021.03.022.

[10] J. Anzola et al., "Review of Architectures Based on Partial Power Processing for DC-DC Applications," *Institute of Electrical and Electronics Engineers Inc*, 2020; doi: 10.1109/ACCESS.2020.2999062.

[11] H. Ye et al., "High Step-Up Interleaved dc/dc Converter with High Efficiency," *Energy Sources, Part A: Recovery, Utilization and Environmental Effects*, vol. 46, 2020, pp. 4886–4905; doi: 10.1080/15567036.2020.1716111.

[12] P. Sharma, D.K. Dhaked, and A.K. Sharma, "Mathematical Modeling and Simulation of DC-DC Converters Using State-Space Approach," *Proceedings of Second International Conference*, J.C. Bansal, K. Deep, and A.K. Nagar, eds., Springer, 2020, pp. 1–737; doi: 10.1007/978-981-15-6707-0.

[13] P. Sharma et al., "Coati Optimized FOPID Controller for Non-Isolated DC–DC Converters in EV Charging Application," *IET Power Electronics*, 2024, pp. 2771–2784; doi: 10.1049/pel2.12798.

[14] P. Sharma et al., "Novel Current-Fed Bidirectional DC-DC Converter for Battery Charging in Electric Vehicle Applications with Reduced Spikes," *Electricity*, vol. 5, no. 4, 2024, pp. 1022–1048; doi: 10.3390/electricity5040052.

[15] K. Shreya et al., "CUK Converter Fed Resonant LLC Converter Based Electric Bike Fast Charger for Efficient cc/cv Charging Colution," *Journal of Applied Science and Engineering*, vol. 24, no. 3, 2020, pp. 331–338; doi: 10.6180/jase.202106_24(3).0008.

[16] Y.-S. Cheng et al., "Design and Implementation of Li-ion Battery Charger Using State-of-Charge Estimation with Fuzzy Temperature Control," in *2015 IEEE ICIT*, 2015, p. 799; doi: 10.1109/ICIT.2015.7125402.

[17] S.S. Hussein, A.J. Abid, and A.A. Obed, "ANFIS-Based New Approach for an Optimal Lithium-Ion Battery Charging Control," in *2023 IEEE 3rd International Conference in Power Engineering Applications: Shaping Sustainability Through Power Engineering Innovation, ICPEA 2023*, 2023, pp. 248–251; doi: 10.1109/ICPEA56918.2023.10093226.

[18] G. Balen et al., "Modeling and Control of Interleaved Buck Converter for Electric Vehicle Fast Chargers," in *2017 COBEP*, 2018, pp. 1–6; doi: 10.1109/COBEP.2017.8257412.

[19] L. Ntogramatzidis et al., "A Novel MIMO Control for Interleaved Buck Converters in EV DC Fast Charging Applications," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 4, 2023, pp. 1892–1900; doi: 10.1109/TCST.2023.3237497.

[20] M. Carbajal-Retana et al., "Interleaved Buck Converter for Inductive Wireless Power Transfer in dc–dc Converters," *Electronics (Switzerland)*, vol. 9, no. 6, pp. 1–15; doi: 10.3390/electronics9060949.

[21] S. Cuoghi et al., "Multileg Interleaved Buck Converter for EV Charging: Discrete-Time Model and Direct Control Design," *Energies (Basel)*, vol. 13, no. 2, pp. 1–8; doi: 10.3390/en13020466.

[22] A. Garg and M. Das, "High Efficiency Three Phase Interleaved Buck Converter for Fast Charging of EV," in *ICPEE 2021 - 2021 1st International Conference on Power Electronics and Energy*, 2021, pp. 1–5; doi: 10.1109/ICPEE50452.2021.9358486.

[23] K. Drobnic et al., "An Output Ripple-Free Fast Charger for Electric Vehicles Based on Grid-Tied

Modular Three-Phase Interleaved Converters," *IEEE Trans Ind Appl*, vol. 55, no. 6, 2019, pp. 6102–6114; doi: 10.1109/TIA.2019.2934082.

[24] Y.T. Yau, K.I. Hwu, and J.J. Shieh, "Minimization of Output Voltage Ripple of Two-Phase Interleaved Buck Converter with Active Clamp," *Energies (Basel)*, vol. 14, no. 16, 2021, pp. 1–30; doi: 10.3390/en14165215.

[25] A.I. Pózna, K.M. Hangos, and A. Magyar, "Design of Experiments for Battery Aging Estimation," *IFAC-PapersOnLine*, 2018, pp. 386–391; doi: 10.1016/j.ifacol.2018.11.733.

[26] H. Li et al., "Cooperative CC-CV Charging of Supercapacitors Using Multicharger Systems," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, 2020, pp. 10497–10508; doi: 10.1109/TIE.2019.2962485.

[27] L. Gan, U. Topcu, and S.H. Low, "Optimal Decentralized Protocol for Electric Vehicle Charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, 2012, pp. 940–951; doi: 10.1109/TPWRS.2012.2210288.

[28] M.C. Kisacikoglu, F. Erden, and N. Erdogan, "Distributed Control of PEV Charging Based on Energy Demand Forecast," *IEEE Trans Industr Inform*, vol. 14, no. 1, 2018, pp. 332–341; doi: 10.1109/TII.2017.2705075.

[29] S. Bhatnagar et al., "Charging System Analysis in Static and Dynamic Wireless Electric Vehicle," *International Journal of Advanced Research in Engineering and Technology (IJARET)*, vol. 10, no. 1, 2019, pp. 104–120; doi: https://www.doi.org/10.34218/IJARET.10.1.2019.011

[30] G. Madhur et al., "Fast Charging Electric Vehicle using Fuzzy Logic Controller," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, pp. 499–502; doi: 10.17577/IJERTV9IS050426.

[31] D.N. Huu and V.N. Ngoc, "Analysis Study of Current Transportation Status in Vietnam's Urban Traffic and the Transition to Electric Two-Wheelers Mobility," *Sustainability (Switzerland)*, vol. 13, no. 10, 2021, pp. 1–27; doi: 10.3390/su13105577.

[32] T. Saravanakumar and R.S. Kumar, "Fuzzy Based Interleaved Step-up Converter for Electric Vehicle," *Intelligent Automation and Soft Computing*, vol. 35, no. 1, 2022, pp. 1103–1118; doi: 10.32604/iasc.2023.025511.

[33] D.V. Pendam and T.M. Rofin, "Electric Vehicle and Charging Infrastructure Development: A Comprehensives Review Using Science Mapping and Thematic Analysis," 2023; doi: 10.1007/978-981-99-1019-9_26.

[34] V. N. Ngoc and D. N. Huu, "Optimal Valley-Filling Algorithm for Electric Two-wheeler Charging Stations," *Engineering, Technology and Applied Science Research*, vol. 14, no. 1, pp. 13072–13077, 2024; doi: 10.48084/etasr.6569.

[35] Z. Chen, Z. Zheng, X. Tang, and Y. Wu, "Model Predictive Controlled Dual Active Bridge Converter with Efficiency Optimization and Fast Dynamic Response," *Journal of Electrical Engineering and Technology*, vol. 19, no. 1, 2024; doi: 10.1007/s42835-023-01569-x.

Design, Implementation, and Performance Optimization of a ROS Based Autonomous Mobile Robot for Intralogistics in Manufacturing Facilities

Submitted: 6th November 2024; accepted: 24th February 2025

Neslihan Demir, Pinar Demircioglu, Ismail Bogrekci

DOI: 10.14313/jamris-2026-031

Correction note

The article entitled "Design, Implementation, and Performance Optimization of a ROS Based Autonomous Mobile Robot for Intralogistics in Manufacturing Facilities, Volume 20, No1 2021, page 93-102, DOI: https://doi.org/10.14313/jamris-2026-010 was corrected.

The correction concerns the author's email address. The correct author details are:

Neslihan Demir
Department of Industrial Engineering, Istanbul Aydin University
Email: neslihandemir2@aydin.edu.tr
ORCID: 0000-0001-8641-7787

This correction updates only the author's contact information and does not affect the content, results, or conclusions of the article.