

# ENHANCING EFFICIENCY AND SECURITY IN HEALTHCARE IOT: A NOVEL APPROACH FOR FOG COMPUTING RESOURCE OPTIMIZATION USING TGA-RNN

Submitted: 21<sup>st</sup> May 2024; accepted: 23<sup>rd</sup> July 2024

Rahul Jaywantrao Shimpi, Vibha Tiwari

DOI: 10.14313/jamris-2025-037

## Abstract:

*Fog computing, a new computing paradigm that has gained popularity, brings calculations closer to data sources from healthcare facilities. The healthcare industry is the driving force behind the growth of Internet of Things (IoT)- driven Fog computing, which improves network performance and efficiency, particularly in the safe and effective aggregation and transmission of healthcare data. This requires optimizing resource allocation and addressing overflow issues. This study introduces a novel approach that combines Task Group Aggregation (TGA) with a Recurrent Neural Network (RNN) to assess Quality of Service (QoS) characteristics and detect overloaded servers. The TGA method is utilized to effectively manage data movement to Virtual Machines (VMs), thereby alleviating congestion and improving system stability. Furthermore, it utilizes the Chaotic Fruit Fly Optimization Algorithm (CFOA), a neural computing system, to optimize service and user separation based on individual characteristics in the context of secure healthcare data aggregation and transmission within IoT networks. The integration of TGA with CFF enhances the detection of overflow problems within the RNN framework, enabling proactive management of resource allocation. The proposed work is evaluated by using the Java programming language, and the results demonstrate the effectiveness of the Fog computing overflow control model in mitigating congestion and optimizing resource scheduling, thereby facilitating the efficient and secure aggregation and transmission of healthcare data within IoT networks.*

**Keywords:** Fog Computing, Internet of Things, Task Group Aggregation, Recurrent Neural Network, Chaotic Fruit Fly Optimization Algorithm.

## 1. Introduction

The emergence of IoT technologies is attributed to the increasing number of user devices and the expansion of distributed processing capabilities [1]. The IoT is a contemporary technology that has broad applications across many industries, including healthcare [2]. The entire healthcare industry's adoption of this technology holds promise, as it enables hospitals to operate more efficiently and provides patients with access to better care [3, 4]. While other features are still crucial for the healthcare industry, remote patient monitoring is one of the main applications of Healthcare IoT (HIoT) that preserves money and millions

of lives. To protect patient privacy, provide effective treatment, and ensure patient safety and confidentiality, safeguards for HIoT systems are critical [5, 6]. Healthcare is undergoing a digital transformation because of Cloud IoT and the ability of Cloud Computing (CC) to access data from anywhere at any time, and data in IoT is also expanding at an exponential rate [7, 8]. The cloud will face tremendous computing demand if all this traffic is transferred there [9]. Fog computing has therefore emerged to do three things: (i) utilize the idle resources of IoT devices; (ii) decrease network flow in cloud centers, and (iii) shorten job delays. Fog networks have been used extensively thus far in areas such as industrial automation, smart cities, smart buildings, and the Internet of cars [10–12]. Thus, integrating deep neural networks with IoT devices results in a new generation of apps that are able to support a whole new level of interaction between people and their physical environment through sophisticated sensing and understanding activities [13,14]. One particular kind of neural network that is useful for processing sequential data is the RNN, which improves the understanding and representation of input data by capturing contextual information [15]. Using past data and learning patterns, the RNN method calculates the QoS for all services depending on several traffic loads, network topology, and connected devices [16]. In cloud-fog computing frameworks, an artificial ecosystem-based optimization is created as an alternative job scheduling model to enhance the QoS for IoT. It has demonstrated itself to be the most successful and effective technique for solving cloud-fog TS optimization problems, yielding the best results. It needs to be expanded to address other goals relevant to cloud-fog environments, such as energy usage and financial expenditures, to facilitate the scheduling of IoT tasks [17, 18]. A multi-objective Gray Wolf Optimization approach is presented in [19] for the better management of Fog computing service scheduling and activation. The scheduling problem is the primary application of this method. Unfortunately, to function as a balancing agent, it also needs to employ a different algorithm. Specifically, this approach is used to oversee and handle activations. The multi-objective optimization method MOCSA, related to the cuckoo search meta-heuristic algorithm, was expanded to address the combinatorial optimization problem. When compared to other state-of-the-art approaches in terms of objective functions, it exhibits a noticeable advantage

in terms of average total power usage and average overall latency. Nest is a possible solution and a crucial part of the CSA algorithm. The performance of the method is greatly affected by encoding on the nest [20]. An enhanced Harris Hawks Optimization (HHO) algorithm for scheduling scientific workflows has been presented. This approach is linked to a discrete opposition, known as DO-HHO. It outperforms state-of-the-art efforts in reducing offloaded jobs on CC, lost deadlines, and SLA violations. The DO-HHO algorithm has to be expanded upon and its efficacy assessed for more scheduling difficulties [21, 22].

A Whale Optimized Resource Allocation (WORA) method was developed in [23]; it utilizes less energy and achieves better results in terms of make span and the task's successful completion ratio. It is necessary to consider certain measures, such as throughput and delay rate, when assessing the algorithm's performance. Increasing a task's processing speed also means using more energy; therefore, look for ways to do jobs quickly while using the least amount of energy. Consequently, a Lyapunov framework-based heuristic Particle swarm optimization algorithm was presented (LPSO). A balance is struck among the energy used by IoT nodes for processing, transmission energy, and Fog node computing energy, ensuring that jobs are completed with the least amount of energy consumption possible, as task time and queue stability are ensured. Because it consumes less energy than the other methods, the number of tasks completed was somewhat fewer [24]. A method for Lyapunov optimization that functions with limited buffers without compromising asymptotic optimality was proposed in [25]. It achieves asymptotically decreased time-average cost and boosts throughput. However, because of the small buffers of real-world IoT devices, the problem of interest was not directly addressed by Lyapunov optimization. Multireplicas Pareto ant colony optimization (MRPACO), a solution related to ant colony optimization, was presented in [26]. MRPACO worked well and appears to be dependable in reaching convergence and widespread solutions. Notably, additional points obtained using the baseline approaches predominate over a small number of points obtained using MRPACO. To cluster over healthcare IoT datasets, a Chaotic Biogeography-Based Optimization technique employing Information Entropy (CBO-IE) is used. The BO technique yields effective results for numerical optimization and is ideally suited for exploring and exploiting large search spaces. However, the data clustering technique and numerical optimization differ somewhat [27]. The Cascaded long short-term memory (CSO-CLSTM) method, which is related to the Crow Search Optimization algorithm, is employed by [28] to diagnose diseases. Smart healthcare systems use the CSO-LSTM model as a suitable illness diagnosis tool. Some of the shortcomings of the CSO include its high likelihood of getting stuck in local optima and its slow search precision. Therefore, it was solved by optimizing service and user separation based on individual features within the context

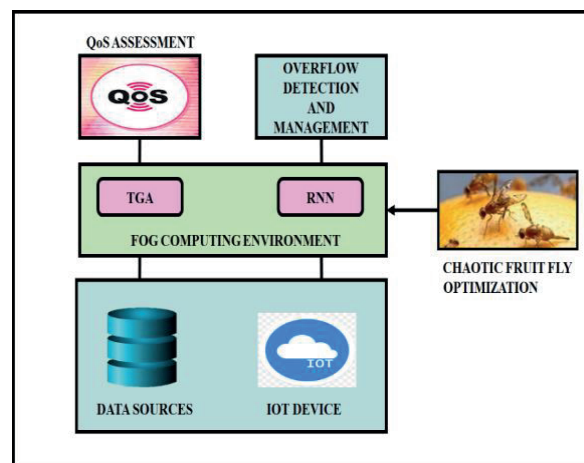


Figure 1. Block diagram of proposed method

of safe healthcare data aggregation and transmission within IoT networks using the CFFO, a neural computing system. This paper proposes a unique method to detect overloaded servers and evaluate QoS features by combining an RNN with TGA. By efficiently managing data migration to VMs, the TGA approach reduces congestion and enhances system stability. The main influences that shaped this investigation are as follows:

- By combining TGA and CFF, overflow issues within the RNN framework are detected more effectively, allowing for proactive resource allocation management.
- To evaluate the quality of service attributes and identify overloaded servers, it combines TGA with an RNN.
- The efficacy of the proposed overflow control approach for Fog computing in reducing congestion and streamlining resource scheduling, which makes it easier to aggregate and transmit medical data securely and effectively over IoT networks.

## 2. Proposed Methodology

The combination of IoT has produced significant insights, cost-effectiveness, and productivity, all of which have had an impact on quality of life. IoT is being used in healthcare to improve patient monitoring, reduce expenditure, and promote innovation in patient care. Due to the large volume of data gathered, it will be stored in the cloud for testing and decision-making purposes. The primary cause of Fog computing's rise is the CC scenario's high time-delay problem, which persists despite its numerous advantages. Fog computing is therefore ideal for applications that require rapid or real-time responses, particularly in the healthcare industry.

The input data is obtained from a data source and an IoT device used for monitoring purposes in healthcare. The fog computing environment consists of TGA and RNN methods used to assess QoS characteristics and detect overloaded servers. To efficiently manage data migration to VMs, reduce congestion, and enhance system stability, the TGA approach is

utilized. Furthermore, the CFOA is a neural computing system designed to maximize service and user separation based on unique attributes in the context of safe healthcare data transfer and aggregation over IoT networks. The proactive control of resource allocation made possible by the integration of TGA and CFF improves the detection of overflow issues within the RNN architecture. The efficiency of the Fog computing overflow management model reduces congestion. It enhances resource scheduling, making it easier for healthcare data to be aggregated and transmitted securely and effectively over IoT networks.

### 2.1. Health Care IoT

Health care is the act of receiving essential medical treatment to enhance one's well-being. Issues with traditional IT healthcare systems include expense, location, ongoing surveillance, apps, administrative work, processing lags, and accuracy. Various IT infrastructures have been used to implement numerous healthcare applications. For non-invasive healthcare, the current IT and healthcare solutions perform admirably. Medical devices that are installed on the body and continuously generate data relevant to organ health are used in non-invasive applications. The devices employ various processing and interpretation techniques to collect and analyze this data, and then take appropriate action to maintain the patient's health. However, several vital health applications, such as electrocardiography, electromyography, blood pressure monitoring, glucose level sensing, oxygen sensing, rehabilitation, blood temperature, and medication, require parametric information. IT support systems will assist in making health-related decisions based on the data that has been recorded. However, if decisions on such vital medical care are postponed in any way, it leads to disastrous outcomes. QoS is affected if healthcare systems, when implemented, are not responding promptly. The purpose is to raise the present IT-related health care application using a QoS method.

The combination of the words "Internet" and "things" has greatly advanced the field of ICT. This is a paradigm in which every device is internet-connected and has its own identity and address. It is a global network of interconnected devices used to refer to, process, store, and transmit environmental information. It appears that distinct viewpoints have converged as IoT is applied in various industries, with application scenarios including healthcare, personal and social, smart environments, logistics, and transportation. IoT in the healthcare industry can track items and people, recognize and authenticate individuals, and gather data spontaneously by detecting surroundings. Additionally, IoT devices can interact in diverse environments through various wired and wireless protocols. As CC processes enormous volumes of data, it results in significant end-to-end delays and issues with network utilization. IoT-based healthcare solutions continue to face challenges with security and efficiency. By offering processing and storage capacities at the

network's edge, fog computing is designed to handle the issues associated with CC.

### 2.2. Fog Computing in Healthcare

Fog is a distributed computing environment where applications and data are disseminate between multiple data sources and the cloud. A decentralized method called fog computing processes data nearer to the edge devices. Fog computing also helps with security because it divides the bandwidth flow and provides more firewalls to a network for added protection. All applications are still able to utilize CC, but fog computing will surely reduce cloud traffic to a greater extent. Furthermore, CC remains the only method available for saving and retrieving historical data. The amount and diversity of stream data have increased to previously unheard-of levels in recent years because of the rapid growth of IoT. IoT is a broad and adaptable network architecture that links objects with unique identities to provide a multitude of innovative application services. Even though IoT offers improved services, its limited processing and storage capacity prevent it from handling and saving the huge volume of data. The nuisance of IoT is eliminated by CC technology, which provides virtual resources on a pay-as-you-go basis. It provides countless processing and storage capacity options. Although numerous CC resources, services, and apps are available, latency issues hinder the complete utilization of some of these resources. Due to the rapid increase in internet-coupled smart devices and the volume of service demands, the network capacity is significantly stressed, and the QoS is decreased. Implementing low-latency apps is hindered by the large network latency between smart devices and the cloud. The most promising approach is fog computing, which offers CC benefits and greatly lowers latency by expanding cloud services to the network edge. In addition to facilitating the generation of new information and analyzing data from streams generated by clever IoT devices, it offers dispersed services. The purposes of fog computing are especially well-suited to applications for extensive health monitoring. The IoT is essential for consistently monitoring hospitalized patients' physiological states without requiring active caregiver participation. Wireless Body Area Networks (WBAN), the major IoT technology used in healthcare, primarily rely on applications for healthcare monitoring. Electromyography, Electrocardiography, glucose level sensing, blood pressure monitoring, oxygen sensing, rehabilitation, blood temperature measurement, and medication administration are among the physiological data that WBAN assists in the efficient and inconspicuous collection of. The prior research endeavour leverages CC technology for IoT devices to effectively facilitate the pervasive applications of healthcare.

The conventional Fog Computing methods focus on various application scenarios to offer multiple options for reducing service latency. These approaches are still in their infancy, though, as they aim to offer ubiquitous healthcare computing services in the real world. To provide a smart fog gateway

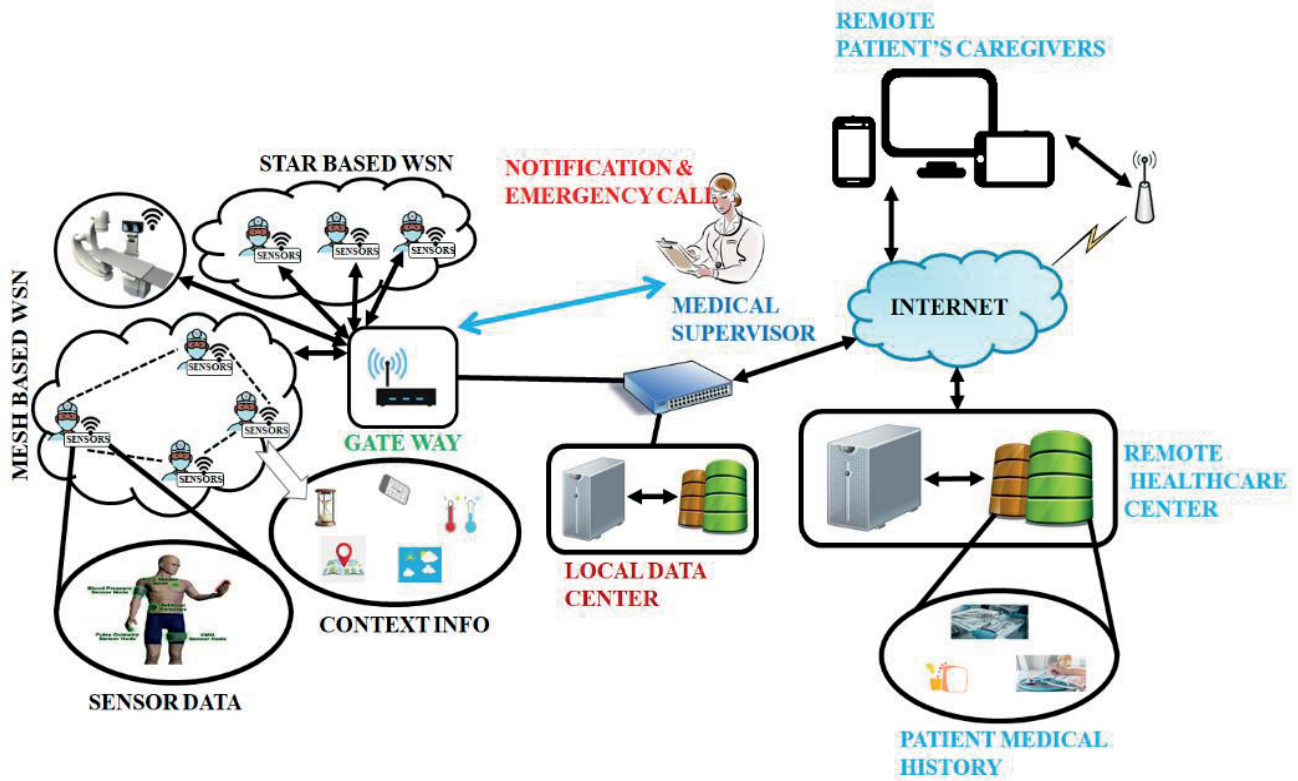


Figure 2. IoT-based Health Monitoring Architecture

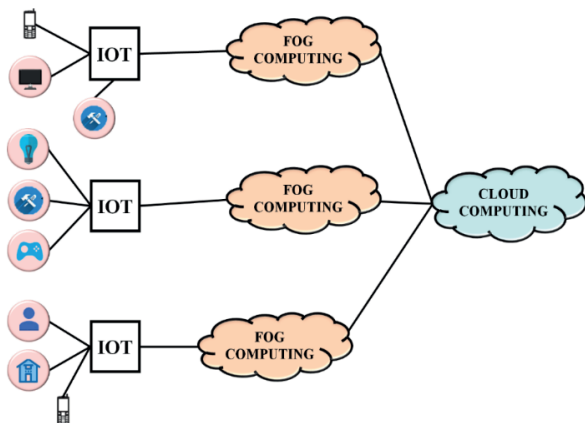


Figure 3. Fog computing architecture

and maximize cloud resources for HIoT inquiries, this strategy focuses on offering decision-making and smart partitioning using linear decision trees in fog scenarios. It strives to improve resource use and reduce response latency while delivering the service to healthcare applications, as specified by SLAs. The best layer of the IoT is CC, which is a paradigm for hosting services online. Fog computing resources are positioned at the top of the devices. The fundamental requirements of fog computing is effective resource management, which supports the provision of high-quality IoT services. In this case, security for IoT devices is instead provided by a Fog node, which handles these cryptographic operations. Fog processing and storing occur closer to the data source at the edge of the network, which

is important for real-time control. Fog acts as a bridge between a distant server and computer hardware. It determines what data was processed locally and what will be forwarded to the server. Fog functions as an intelligent gateway that eliminates clouds, facilitating more effective data processing, analysis, and storage. The fog computing architecture is depicted in Fig. 3, where IoT devices are situated in the bottom layer, and Fog computing expands the system's efficiency and is also used to confirm increased security. For resource optimization, Fog computing combines TGA with an RNN to assess QoS characteristics and detect overloaded servers.

### 2.3. TGA with RNN

The integration of TGA with RNN evaluates QoS characteristics and identifies overloaded servers. In order to efficiently manage data migration to VMs, reduce congestion, and enhance system stability, the TGA approach is employed. Algorithm 1 depicts the TGA algorithm.

A type of deep learning technique is the RNN, which is used to minimize the traffic flow in the network. LSTM networks are a type of RNN, and the three gates of an LSTM network are the input, forget, and output gates, which enable it to recall past data and make it an excellent choice for handling time-series data. LSTM networks are examples of recurrent neural networks. Fig. 4 depicts the baseline model architecture. The input layer, the hidden layer (LSTM & hidden layer (Dense) and the output layer make up the three layers of the LSTM model. The three neurons in this work are represented by the vectors  $x, y$  and

**Algorithm 1: TGA Algorithm**

**Input** :  $U \rightarrow$  Number users

$C \rightarrow$  Number of servers

**Output** :  $AS \rightarrow$  List of Allocated server

**Start**

Define  $RTF = [sCPUsMemsBW]$  //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

$UtilRate = DemR \times TimInt$  // Resource Utilization Rate

If  $\max(UtilRate) \leq \max?(RTF)$

$AvgPTimeSer(i) = DemR(i)/(1-$

$UtilRate(i) AvgPTimeAllSeri(i) = (AvgPTimeSer(i) \times TimInt(i))/TimInt(i)$

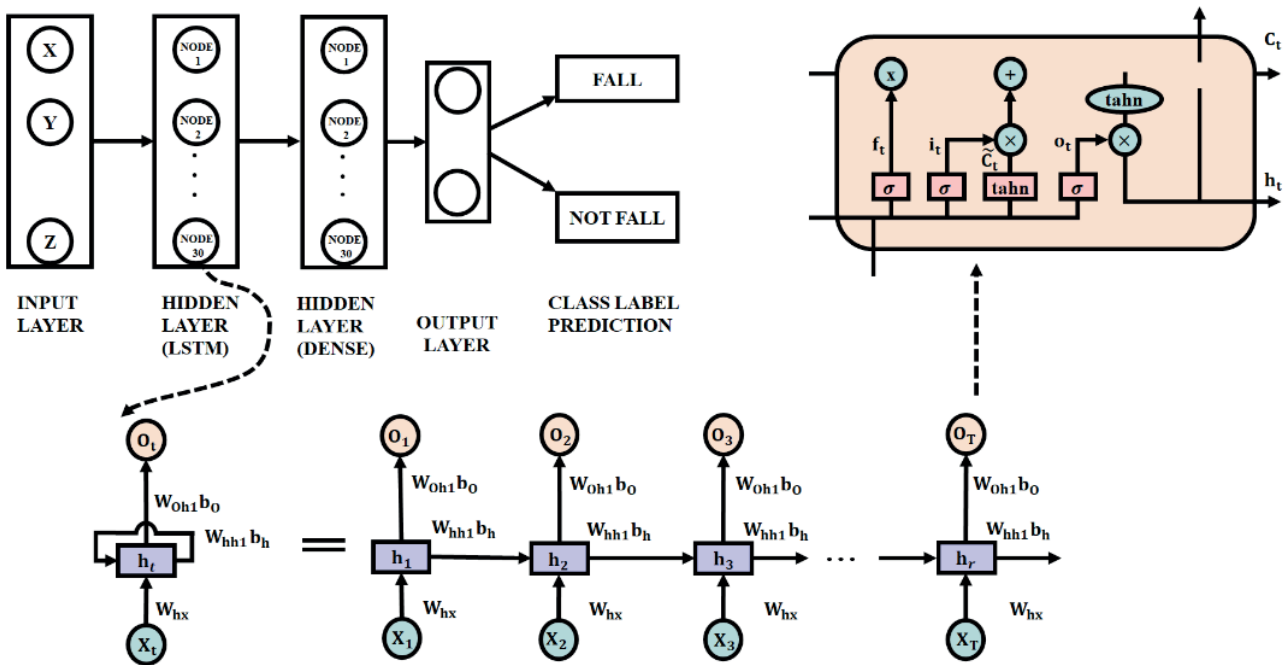
$AS - LIST = \text{ceil}(cserver \times rand)$

End - If

End - For

Return :  $AS - LIST$  as a list of allocated servers

End - Algorithm



**Figure 4.** LSTM model

z. The input layer in this work receives the raw data. This output layer is transferred onto the succeeding hidden layers, which consist of the dense layer with 30 neurons and the first LSTM layer with 30 neurons. Lastly, the 2-node output layer computes the class label in probabilities using the softmax function. Since the LSTM model is typically trained using the well-known backpropagation algorithm, its performance is greatly influenced by the number of time steps (n) utilized during model training. For example, the model won't be able to capture the entire range of the activity if it was trained using a tiny number of steps in the past. Conversely, if the method is trained with

numerous time steps, a large amount of irrelevant data is considered. The LSTM model's individual cell activities are explained by equations (1) through (5). More precisely, vectors and matrices are represented by the following input variables:  $x_t, h_{t-1}$  and  $c_{t-1}$ . The input and recurrent connections are denoted by  $W_q$  and  $U_q$  weights, respectively. Furthermore, these matrices  $q$  have an index that corresponds to the forget gate (f), input gate (i), output gate (o), or memory cell (c).

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

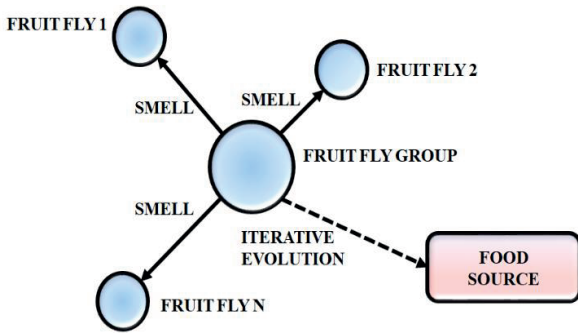


Figure 5. Iterative foraging process of fruit flies

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t o c_{t-1} + i_t o \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t o \sigma_h(c_t) \quad (5)$$

In this case,  $C_o = 0$  and  $h_o = 0$  are the beginning values. The operator refers to the Hadamard product, and the time step is represented by the index  $t$ . Moreover, the functions represented by the values of  $\sigma_g$ ,  $\sigma_c$  and  $\sigma_h$  are the sigmoid, hyperbolic, and hyperbolic tangent functions. Along with TGA and RNN, the CFOA is utilized to solve issues or enhance existing solutions.

#### 2.4. Chaotic Fruit Fly Optimization Algorithm for Optimization Purposes

The CFOA enhances the Fruit Fly Optimization Algorithm (FOA) by interpolating the new parameter joined with chaos, and it has an extremely quick convergence rate. FOA with a Chaotic map indicates ascendancy in terms of reliability of global optimality and algorithm success rate.

##### 1) Basic fruit fly optimization Algorithm

FOA is a new metaheuristic method inspired by the behavior of fruit flies, designed to solve optimization problems and find global optima. The character of fruit flies in the wild served as the model for the fruit fly optimization algorithm. When compared to other similar species, fruit flies are superior, particularly when it comes to employing their olfactory and eyesight traits to find food. An individual is looking for and discovering food sources near the fruit fly swarm during the scent foraging phase. The fragrance concentration that matches the fitness value for each food source is next assessed. The swarm is steered towards the greatest smell concentration value assigned during the vision foraging phase. Fig. 5 depicts the fruit fly's foraging behavior.

There are six distinct steps that make up FOA, and they are as follows: (i) establish the size of the population and the maximum number of iterations; (ii) use Eq. (6) to randomly establish the location of the fruit fly swarm;

$$\delta_j = \text{lower\_bound} + (\text{upper\_bound} - \text{lower\_bound}) \times \text{rand}() \quad (6)$$

(iii) Utilize Equation 7 to randomly produce several fruit flies near the fruit fly swarm in order to build a

population;

$$x_{i,j} = \delta_j \pm \text{rand}(), j = 1, \dots, n \quad (7)$$

(iv) assess the total population of fruit flies to determine the smell concentration value of each individual in the swarm; (v) identify the fruit fly with the highest smell concentration and then use the vision search process to guide the swarm to find this best individual; and lastly (vi) complete the algorithm if several generations is arrived, or proceed to step (iii) else.

The factors  $x_{i,j}$  and  $\delta_j$  affect the olfaction and vision search stages, and are responsible for producing food sources. The final algorithm answer is determined by the significant way in which they are calculated. Several studies have demonstrated that the use of non-standard distributions (such as the Gauss or uniform distributions) improves the performance of random-based optimization techniques. Furthermore, the periodicity and non-repetition of chaos can compel an algorithm to run its overall searches more quickly. These are the primary drivers behind the creation of enhanced FOA, which are discussed next. However, this approach has drawbacks such as a poor global optimization capability and produces population individuals in an unduly simplistic manner. An algorithm for CFOA is proposed to address this issue.

##### 2) Chaotic fruit fly optimization Algorithm

This part introduces a new parameter boosted by chaos to present a novel FOA. The CFOA is explained in the following manner.

###### 2.1) Algorithm Initialization

The initial location of the swarm significantly influenced rapid convergence and the ultimate result. CFOA selects the optimal randomly generated solution from PS to determine the initial position of the fruit fly swarm as the first improvement. When the experimental run is over, the computation of the initial swarm position produces a better algorithm solution and faster convergence.

###### 2.2) Chaos FOA

The chaos approach is referred as a practical way to balance exploration and exploitation, ensuring algorithmic performance stability. One of the most crucial factors affecting algorithm performance in FOA is the calculation of food sources. This metaheuristic technique's basic implementation assumes uniform distribution randomization of the variables  $x_{i,j}$ . This is often a poor decision, particularly when dealing with complex, multimodal, and nonlinear issues. An alpha is a new parameter introduced and utilized to generate food sources to increase the overall speed and convergence of FOA. Specifically, changed Eq. (7) to include a chaotic variable as below:

$$x_{i,j} = x_{i,j} + \text{alpha}(x_{i,j} - x_j^*), i = 1, \dots, PS, j = 1, \dots, n \quad (8)$$

Where the optimal solution as of present is  $X^*$ . Make people move in a disorderly manner in the direction of the best possible answer thus far. When compared to FOA with Levy distribution and basic FOA,

**Algorithm 2: Procedure of Chaotic FOA**


---

```

// Algorithm initialization
Set the population size  $PS$  and maximum number of iterations  $I_{max}$ 
// Initialize fruit fly swarm location in the search space  $n$ 
For  $i = 1, \dots, PS$ 
 $X_{i,j} = lower\_bound + (upper\_bound - lower\_bound) \times rand(), j = 1, \dots, n$ 
EndFor
 $\Delta \leftarrow arg(\min_{i=1,2,\dots,PS} f(X_i))$  //Set Swarm location
// Set optimal solution and iteration counter
 $X^* = \Delta$ 
 $Iter = 0$ 
Repeat
// Smell – based (osphresis) foraging phase
For  $j = 1, \dots, PS$ 
//Generate food source  $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ 
 $alpha = Chaos()$  //Determine chaotic parameter
 $x_{i,j} = x_{i,j} + alpha(x_{i,j} - x_j^*), i = 1, \dots, PS, j = 1, \dots, n$ 
//Limit the result
If  $x_{i,j} > upper\_bound$  then
 $x_{i,j} = Upper\_bound$ 
Endif
If  $x_{i,j} < lower\_bound$  then
 $x_{i,j} = Lower\_bound$ 
End if
EndFor
//Vision – based foraging phase
 $X_{best} = arg \min_{i=1,2,\dots,PS} (f(X_i))$ 
//Find global best solution
if  $f(X_{best}) < f(\Delta)$  then
 $\Delta = X_{best}$ 
End if
if  $f(\Delta) < f(X^*)$  then
 $X^* = \Delta$ 
End if
Until the maximum number of iteration is reached :  $Iter = Iter_{max}$ 

```

---

this turned out to be a significant advantage. Algorithm 2 presents the full process of a novel CFOA.

### 3. Results and Discussion

To reduce congestion and improve resource scheduling, this research implies the fog computing overflow control model. This will enable the safe and effective collection and transfer of medical data across IoT networks. By combining TGA and CFF, overflow issues in the RNN framework can be detected more effectively, enabling proactive resource allocation management. Additionally, the Java programming language is used to obtain the results of the associated techniques. Java client-server in fog computing is created at various layers of the fog architecture, including fog nodes, fog servers, gateway devices, cloud services, and management/orchestration systems, to enable efficient and scalable edge computing solutions.

Table 1 displays the simulation parameters of the method. FOA with Chaotic map shows superiority in terms of resource scheduling efficiency, response time, energy consumption, and algorithm success rate.

**Table 1.** Simulation parameters

Simulation parameters	Value
Radius of the FN	100 m
Data size for task	[0.1 – 1] Mbits
Computation capacity of the FN	[0.7 – 1] GHZ
Computation capacity of the IoT devices	20 GHZ
Local computing energy consumption	$[2 - 4] * 10^{-11} J/Cycle$
FN computing energy consumption	$1 * 10^{-11} J/Cycle$

Figure 6 illustrates the monitored healthcare data, including temperature and blood pressure values. Here, the IoT is used to monitor the parameters of the human body, and one of the sample outputs is provided above.

According to user requests varying from 5 to 100, as seen in Fig. 7, the proposed method has an average success rate. The average success rate drops as the number of requests from the user increases. However, the CFOA has a better success rate than other methods.

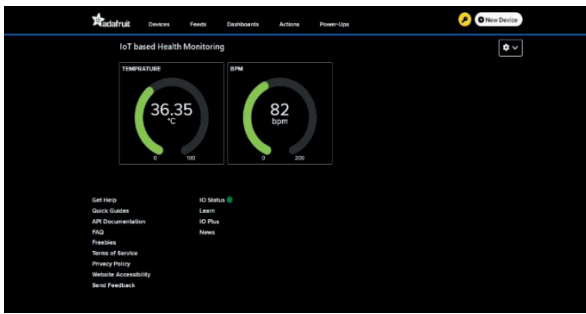


Figure 6. Monitored healthcare data

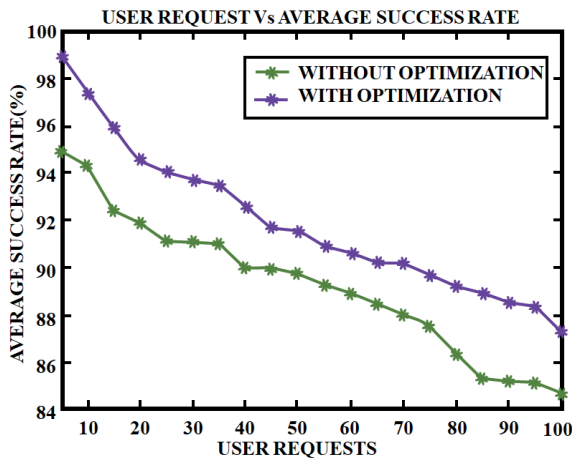


Figure 7. Average Success Rate

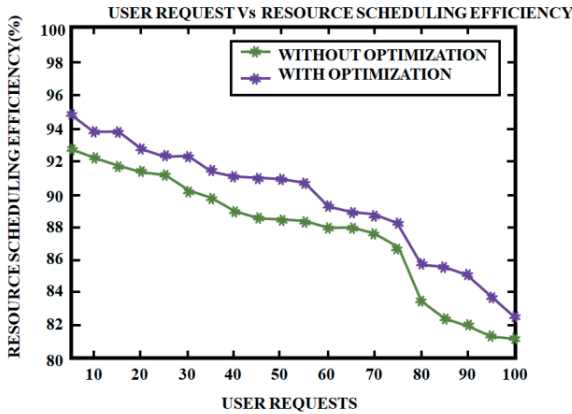


Figure 8. Resource Scheduling Efficiency

The efficiency of resource scheduling for user requests is shown in Fig. 8, which considers a range of customer demands from 5 to 100. The findings show that as the number of requests increases, the effectiveness of resource scheduling varies between 80% and 96%. Although the CFOA has better resource scheduling efficiency than other methods.

After accounting for user demands, Fig. 9 illustrates how energy consumption varies from 1 to 100 users. It was found that as customer demand increases, so does energy usage. However, the CFOA utilizes lower energy than other methods.

Fig. 10 illustrates how to respond to user requests within a suitable timeframe, accommodating queries

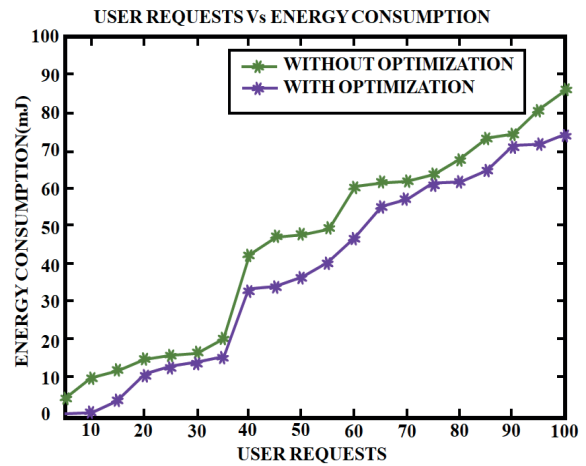


Figure 9. Energy Consumption

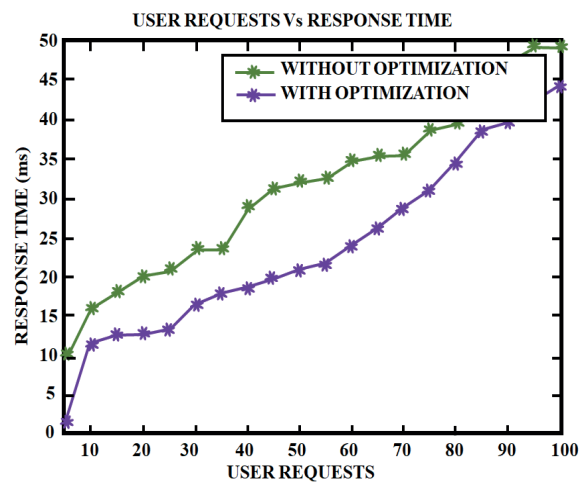


Figure 10. Response Time

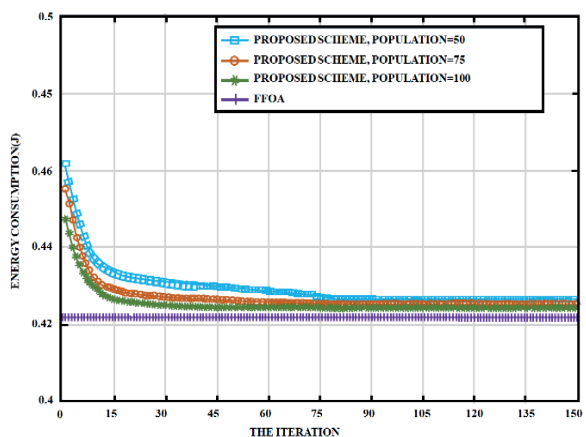


Figure 11. Energy consumption Vs the iterations

from 5 to 100 users. The observations have led to improvements in response times, as well as an increase in user requests. However, the CFOA has a lower response time than other methods.

The energy usage of the proposed CFOA in comparison to the iterations for various populations is shown in Fig. 11. The energy consumption rate for populations of fifty, seventy-five, and one hundred is



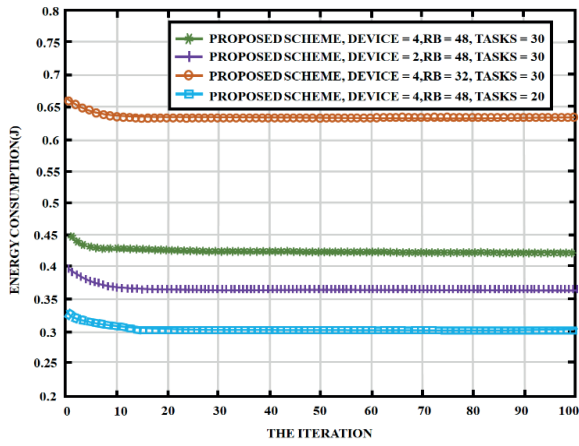


Figure 12. Convergence performance with devices/tasks/RBs

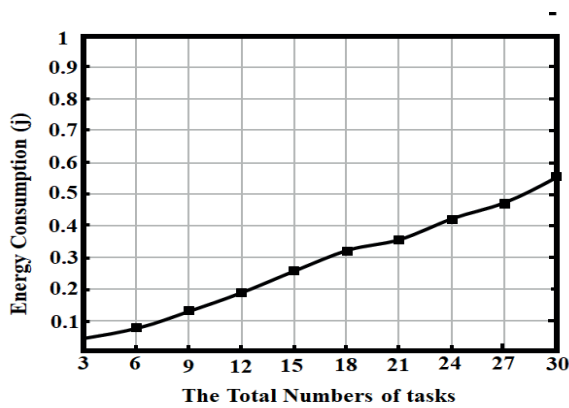


Figure 13. Energy consumption Vs the number of tasks

provided. Furthermore, the FFOA obtains the highly complex optimal solution. However, the CFOA has a low-complexity, suboptimal solution.

The effect of tasks, RBs, and device count on convergence performance is illustrated in Fig. 12. An increase in device count signifies higher energy consumption for convergence, while the impact of RBs and tasks remains similar. In the same vein, as tasks increase and devices and RBs become identical, so does the energy required for convergence. More RBs translate into more transmission resources for offloading, which improves convergence performance even when the devices and tasks are the same.

The energy consumption compared to the total task number is highlighted in Fig. 13. As the task number rises, so does the energy consumption. It leverages the FN’s powerful processing and storage capabilities to reduce energy consumption.

The energy consumption of the FN in relation to its computing capacity is shown in Fig. 14. Because the duties are dynamically assigned to the FN and IoT devices based on available resources, the proposed approach uses the least amount of energy.

The energy consumption versus the total number of RBs is displayed in Fig. 15. As the number of RBs increases, the energy consumption decreases more gradually. Additionally, the proposed strategy

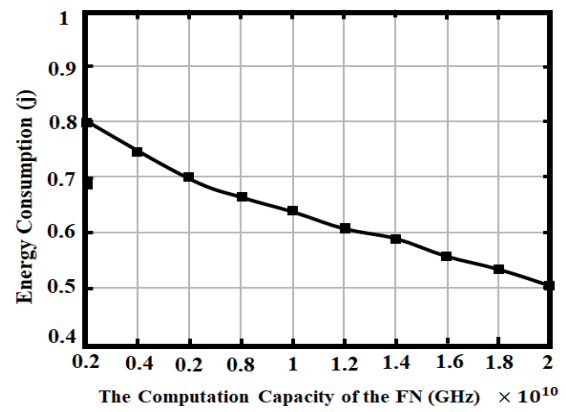


Figure 14. Energy consumption versus the FN’s computation capacity

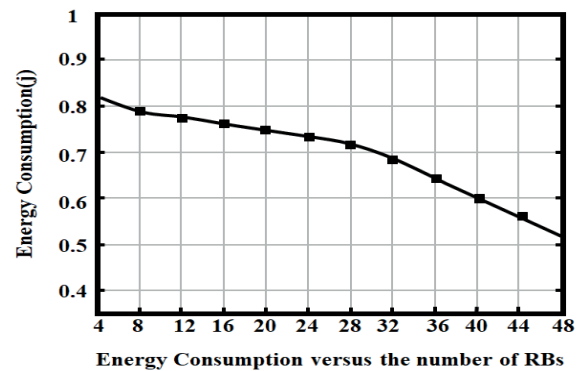


Figure 15. Energy consumption versus the number of RBs

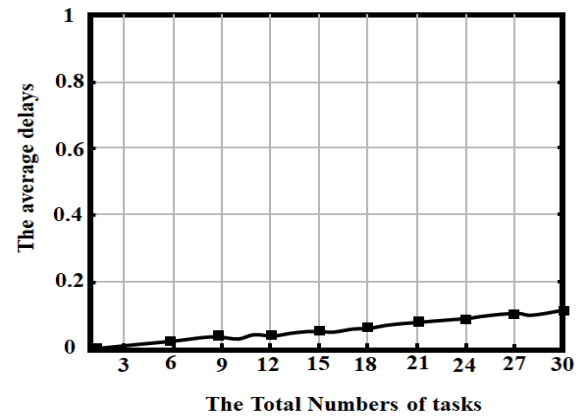
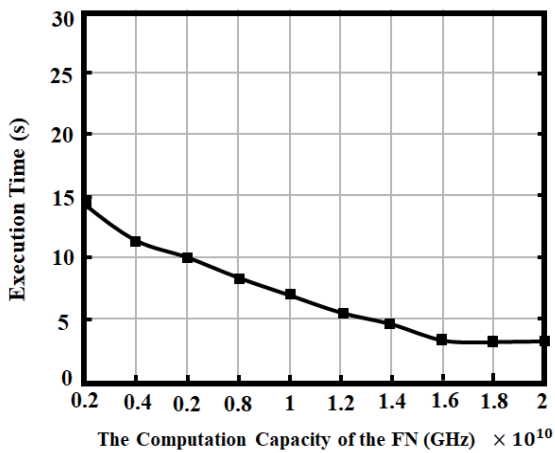


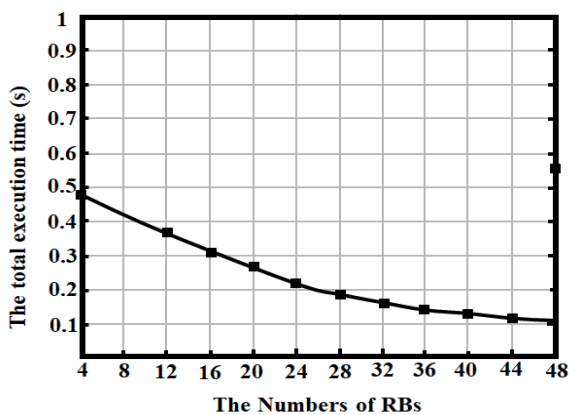
Figure 16. Average delay Vs the number of tasks

allocates jobs to the FN and achieves the lowest energy consumption as the number of RBs is steadily increased.

The average latency vs the total no. of tasks is displayed in Fig. 16. It is evident that as the number of tasks grows, so does the average delay. IoT devices did not process jobs with a greater number of variables in a short time. It causes a substantial delay and subpar performance. Moreover, the proposed model attains a reduced latency.



**Figure 17.** Average delay Vs the FN's computation capacity



**Figure 18.** Average delay Vs the number of RBs

**Table 2.** Comparison of Execution Time

Methods	Execution Time (s)
FOA	55 s
SSA	43 s
Proposed	7.1 s

The average delay is mapped against the FN's compute capability in Fig. 17. The average time of the local computing model is independent of the FN's computation capacity because all jobs are calculated locally at the IoT devices.

The total average delay versus the total number of RBs is highlighted in Fig. 18. As the number of RBs increases, all the proposed method's curves drastically decrease. This is because an increase in the available spectrum resources results in a significant reduction in the transmission time.

The comparison of execution time for FOA [29], SSA [30], and the proposed method is represented in Table 2. From that, the proposed method has the lowest execution time among other methods.

## 4. Conclusion

The proposed approach evaluates QoS attributes and identifies overloaded servers by fusing TGA and an RNN. By efficiently managing data migration to VMs, the TGA approach reduces congestion and enhances system stability. Furthermore, in the context of safe healthcare data aggregation and transmission inside IoT networks, the CFOA is a neural computing system to maximize service and user separation based on individual features. By combining TGA and CFOA, overflow issues in the RNN framework can be detected more effectively, allowing for proactive resource allocation management. The results are obtained by utilizing the Java programming language, and this method achieves an optimal solution with low complexity, improved convergence performance, enhanced resource scheduling efficiency, reduced energy consumption, and lower delay. It demonstrates how the fog computing overflow control model effectively reduces congestion and optimizes resource scheduling, making it easier for healthcare data to be aggregated and transmitted securely and efficiently over IoT networks.

## AUTHORS

**Rahul Jaywantrao Shimpi\*** – Research Scholar, Department of Electronics Engineering, Medicaps University, Indore, India, e-mail: rahul.shimpi@rediffmail.com.

**Vibha Tiwari** – Professor, Department of Electronics Engineering, Medicaps University, Indore, India, e-mail: vibha.tiwari@medicaps.ac.in.

\*Corresponding author

## References

- [1] C. Guerrero, I. Lera, and C. Juiz, "Genetic-based optimization in fog computing: Current trends and research opportunities", *Swarm and Evolutionary Computation*, vol. 72, 2022, 101094. <https://doi.org/10.1016/j.swevo.2022.101094>.
- [2] V. K. Quy, N. V. Hau, D. V. Anh, and L. A. Ngoc, "Smart healthcare IoT applications based on fog computing: architecture, applications and challenges", *Complex & Intelligent Systems*, vol. 8, no. 5, 2022, 3805–3815. <https://doi.org/10.1007/s40747-021-00582-9>.
- [3] S. Goyal, N. Sharma, B. Bhushan, A. Shankar, and M. Sagayam, "IoT enabled technology in secured healthcare: applications, challenges and future directions", *Cognitive Internet of Medical Things for Smart Healthcare: Services and Applications*, 2021, 25–48. [https://doi.org/10.1007/978-3-030-55833-8\\_2](https://doi.org/10.1007/978-3-030-55833-8_2).

- [4] M. Ijaz, G. Li, L. Lin, O. Cheikhrouhou, H. Hamam, and A. Noor, "Integration and applications of fog computing and cloud computing based on the internet of things for provision of healthcare services at home", *Electronics*, vol. 10, no. 9, 2021, 1077. <https://doi.org/10.3390/electronics10091077>.
- [5] S. M. Karunaratne, N. Saxena, and M. K. Khan, "Security and privacy in IoT smart healthcare", *IEEE Internet Computing*, vol. 25, no. 4, 2021, 37–48. <https://doi.org/10.1109/MIC.2021.3051675>.
- [6] S. Tanwar, and James, "Fog computing for health-care 4.0 environments", Switzerland: Springer International Publishing, 2021. <https://doi.org/10.1007/978-3-030-46197-3>.
- [7] F. Farid, M. Elkhodr, F. Sabrina, F. Ahamed, and E. Gide, "A smart biometric identity management framework for personalised IoT and cloud computing-based healthcare services", *Sensors*, vol. 21, no. 2, 2021, 552. <https://doi.org/10.3390/s21020552>.
- [8] N. Singh, M. Raza, V. V. Paranthaman, M. Awais, M. Khalid, and E. Javed, "Internet of Things and cloud computing", In *Digital Health*, 151–162. Academic Press, 2021.
- [9] A. R. Nasser, A. M. Hasan, A. J. Humaidi, A. Alkhayat, L. Alzubaidi, M. A. Fadhel, J. Santamaría, and Y. Duan, "Iot and cloud computing in health-care: A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes", *Electronics*, vol. 10, no. 21, 2021, 2719. <https://doi.org/10.3390/electronics10212719>.
- [10] C. Huang, H. Wang, L. Zeng, and T. Li, "Resource scheduling and energy consumption optimization based on Lyapunov optimization in fog computing", *Sensors*, vol. 22, no. 9, 2022, 3527. <https://doi.org/10.3390/s22093527>.
- [11] K. S. Awaisi, S. Hussain, M. Ahmed, A. A. Khan, and G. Ahmed, "Leveraging IoT and fog computing in healthcare systems", *IEEE Internet of Things Magazine*, vol. 3, no. 2, 2020, 52–56. <https://doi.org/10.1109/IOTM.0001.1900096>.
- [12] D. Kumar, A. K. Maurya, and G. Baranwal, "IoT services in healthcare industry with fog/edge and cloud computing", In *IoT-based data analytics for the healthcare industry*, 81–103, Academic Press, 2021. <https://doi.org/10.1016/B978-0-12-821472-5.00017-X>.
- [13] N. A. S. Al-Jamali, and H. S. Al-Raweshidy, "Smart IoT Network Based Convolutional Recurrent Neural Network With Element-Wise Prediction System", in *IEEE Access*, vol. 9, pp. 47864–47874, 2021. <https://doi.org/10.1109/ACCESS.2021.3068610>.
- [14] E. S. L. Ho, "Data security challenges in deep neural network for healthcare IoT systems", *Security and Privacy Preserving for IoT and 5G Networks: Techniques, Challenges, and New Directions*, 2022, 19–37. [https://doi.org/10.1007/978-3-030-85428-7\\_2](https://doi.org/10.1007/978-3-030-85428-7_2).
- [15] D. Li, W. Li, Y. Zhao, and X. Liu, "The Analysis of Deep Learning Recurrent Neural Network in English Grading Under the Internet of Things", in *IEEE Access*, vol. 12, 2024, 44640–44647. <https://doi.org/10.1109/ACCESS.2024>.
- [16] S. S. Sefati, B. Arasteh, S. Halunga, O. Fratu, and A. Bouyer, "Meet User's Service Requirements in Smart Cities Using Recurrent Neural Networks and Optimization Algorithm", in *IEEE Internet of Things Journal*, vol. 10, no. 24, 2023, 22256–22269. <https://doi.org/10.1109/JIOT.2023.3303188>.
- [17] M. Abd Elaziz, L. Abualigah, and I. Attiya, "Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments", *Future Generation Computer Systems*, vol. 124, 2021, 142–154. <https://doi.org/10.1016/j.future.2021.05.026>.
- [18] A. K. Sarangi, A. G. Mohapatra, T. C. Mishra, and B. Keswani, "Healthcare 4.0: A voyage of fog computing with iot, cloud computing, big data, and machine learning", *Fog Computing for Healthcare 4.0 Environments: Technical, Societal, and Future Implications*, 2021, 177–210. [https://doi.org/10.1007/978-3-030-46197-3\\_8](https://doi.org/10.1007/978-3-030-46197-3_8).
- [19] S. M. Hashemi, A. Sahafi, A. M. Rahmani, and M. Bohlouli, "GWO-SA: Gray Wolf Optimization Algorithm for Service Activation Management in Fog Computing", in *IEEE Access*, vol. 10, 2022, 107846–107863. <https://doi.org/10.1109/ACCESS.2022.3212439>.
- [20] Y. Ramzanpoor, M. H. Shirvani, and M. Golsorkhtabaramiri, "Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure", *Complex & Intelligent Systems*, vol. 8, no. 1, 2022, 361–392. <https://doi.org/10.1007/s40747-021-00368-z>.
- [21] D. Javaheri, S. Gorgin, J. A. Lee, and M. Masdari, "An improved discrete harris hawk optimization algorithm for efficient workflow scheduling in multi-fog computing", *Sustainable Computing: Informatics and Systems*, vol. 36, 2022, 100787. <https://doi.org/10.1016/j.suscom.2022.100787>.
- [22] H. Gezici, and H. Livatyali, "An improved Harris Hawks Optimization algorithm for continuous and discrete optimization problems", *Engineering Applications of Artificial Intelligence*, vol. 113, 2022, 104952. <https://doi.org/10.1016/j.engappai.2022.104952>.

- [23] R. Sing, S. K. Bhoi, N. Panigrahi, K. S. Sahoo, N. Jhanjhi, and M. A. AlZain, "A whale optimization algorithm based resource allocation scheme for cloud-fog based IoT applications", *Electronics*, vol. 11, no. 19, 2022, 3207. <https://doi.org/10.3390/electronics11193207>.
- [24] C. Huang, H. Wang, L. Zeng, and T. Li, "Resource scheduling and energy consumption optimization based on Lyapunov optimization in fog computing", *Sensors*, vol. 22, no. 9, 2022, 3527. <https://doi.org/10.3390/s22093527>.
- [25] C. Ren, X. Lyu, W. Ni, H. Tian, W. Song, and R. P. Liu, "Distributed online optimization of fog computing for internet of things under finite device buffers", *IEEE Internet of Things Journal*, vol. 7, no. 6, 2020, 5434–5448. <https://doi.org/10.1109/JIOT.2020.2979353>.
- [26] T. Huang, W. Lin, C. Xiong, R. Pan, and J. Huang, "An ant colony optimization-based multiobjective service replicas placement strategy for fog computing", *IEEE Transactions on Cybernetics*, vol. 51, no. 11, 2020, 5595–5608. <https://doi.org/10.1109/TCYB.2020.2989309>.
- [27] M. K. Ahirwar, P. K. Shukla, and R. Singhai, "CBO-IE: a data mining approach for healthcare IoT dataset using chaotic biogeography-based optimization and information entropy", *Scientific Programming*, vol. 2021, 2021, 1–14. <https://doi.org/10.1155/2021/8715668>.
- [28] R. F. Mansour, A. El Amraoui, I. Nouaouri, V. G. Díaz, D. Gupta, and S. Kumar, "Artificial intelligence and internet of things enabled disease diagnosis model for smart healthcare systems", *IEEE Access*, vol. 9, 2021, 45137–45146. <https://doi.org/10.1109/ACCESS.2021.3066365>.
- [29] S. Sutradhar, S. Karforma, R. Bose, and S. Roy, "A dynamic step-wise tiny encryption algorithm with fruit fly optimization for quality of service improvement in healthcare," *Healthcare Analytics*, vol. 3, 2023, 100177.
- [30] R. R. Irshad, A. Abdu Alattab, I. M. Alwayle, K. M. Alalayah, K. MG Noaman, M. A. Mahdi, and A. M. Aqlan, "A Novel Structure Optoelectronic Biosensor for Detection of Infectious Diseases Using SALP Swarm Optimized Artificial Neural Network Technique," *Journal of Nanoelectronics and Optoelectronics*, vol. 17, no. 8, 2022, 1154–1162.