VOLUME 18, N° 1 2024
Journal of Automation, Mobile Robotics and Intelligent Systems

# Using Reinforcement Learning to Select an Optimal Feature Set

Submitted: 23rd February 2022; accepted: 25th January 2023

*Yassine Akhiat, Ahmed Zinedine, Mohamed Chahhou*

DOI: 10.14313/JAMRIS/1-2024/6

**Abstract:**
*Feature Selection (FS) is an essential research topic in the area of machine learning. FS, which is the process of identifying the relevant features and removing the irrelevant and redundant ones, is meant to deal with high dimensionality problems to select the best performing feature subset. In the literature, many feature selection techniques approach the task as a research problem, where each state in the search space is a possible feature subset. In this paper, we introduce a new feature selection method based on reinforcement learning. First, decision tree branches are used to traverse the search space. Second, a transition similarity measure is proposed so as to ensure exploit-explore trade-off. Finally, the informative features are the most involved ones in constructing the best branches. The performance of the proposed approaches is evaluated on nine standard benchmark datasets. The results using the AUC score show the effectiveness of the proposed system.*

**Keywords:** *Feature selection, Data mining, Decision tree, Reinforcement learning, Dimensionality reduction*

## 1. Introduction

With the advent of high-dimensional data, typically many features are irrelevant, redundant and noisy for a given learning task as they have harmful consequences in terms of performance and/or computational cost. Moreover, a large number of features requires a large amount of memory or storage space. Applying data mining and machine learning algorithms in high-dimensional data usually leads to the downgrading of their performance due to overfitting problem [1, 2]. Given the existence of a large number of features, machine learning models become intricately complicated to interpret as their complexity increases leading to the restriction of the generalizability. Therefore, reducing the dimensionality of data has become indispensable in real world scenarios to successfully build understandable and accurate models that can improve data-mining performance and enhance models interpretability. Data mining can take advantage of dimensionality reduction tools which are integral parameters central to data pre-processing to reduce the highness of data dimensionality [3]. Dimensionality reduction can be categorized into feature extraction and feature selection (see figure 1) [4–6]. Feature extraction aims at transforming the original feature space to a new reduced one, where features lose their meaning due to the transformation [7–9, 9, 10]. In contrast to feature extraction, feature selection is the process of identifying the relevant features and removing the irrelevant and redundant ones with the objective of obtaining the best performing subset of original features without any transformation [11–13]. Thus, the constructed learning models using the selected subset of features are more interpretable and readable. This gives preference to the reliable applicability of feature selection as an effective alternative prioritized over feature extraction in many real-world datasets. The major reasons for applying the feature selection are the following:
- Making models easier to interpret.
- Reducing resources requirement (shorter training time, small storage capacity etc.).
- Avoiding the curse of dimensionality.
- Avoiding the over-fitting problem, thus, a better model.
- Improving accuracy: less noise in data means improved modeling accuracy.

In general, feature selection algorithms are categorized into: Supervised, Semi-supervised and Unsupervised feature selection [12, 14–18]. In this paper, we put more emphasis on supervised feature selection, which is a threefold approach, Filter, Wrapper [19–23], and Embedded [24–26] (see Fig. 1). Filter Methods rely on the relationship between features and class label (such as distance, dependency, correlation etc.) to assess the importance of features. This category is a pre-processing step, which is independent from the induction algorithm. Filters are known by their ease of use and low computational cost. On the contrary, the Wrapper approach generates models with subsets of features. Then, it uses prediction performance as a criterion function or a guiding-compass to orient the search for the best feature subset. This approach takes into account the interactions between features. Generally, Wrappers achieve better performance than some Filter methods. The Embedded approach performs feature selection by implication while simultaneously constructing models, which makes them less costly in terms of execution time than wrappers do.

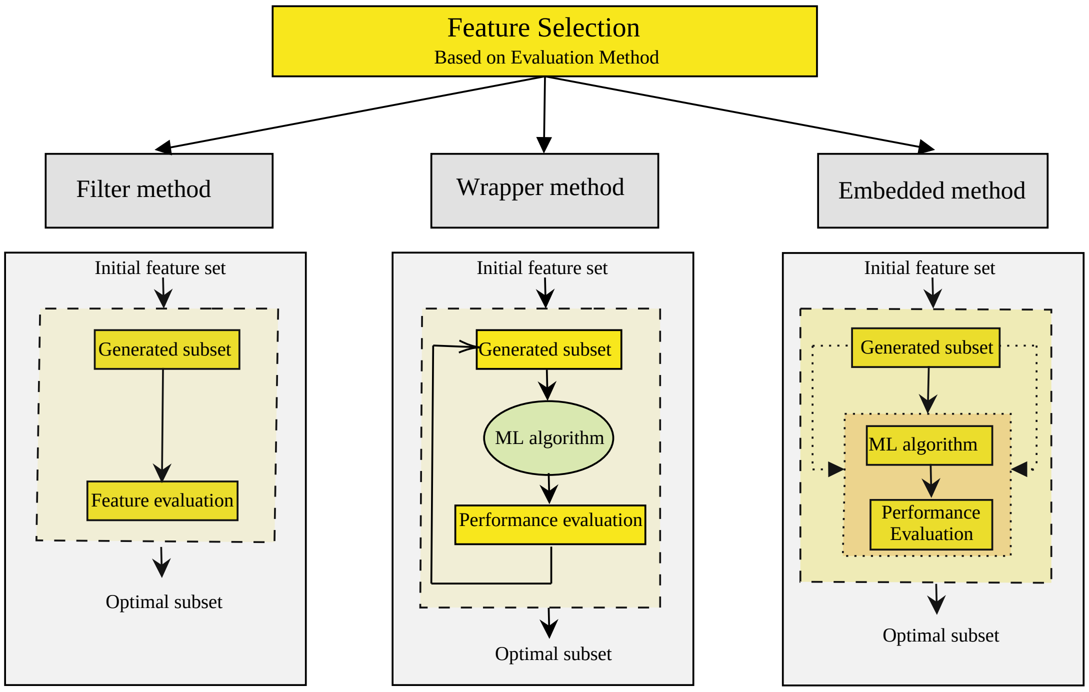


**Figure 1.** Feature selection categorization

### 1.1. Research Objectives

In this paper, we introduce a new feedback system based on reinforcement learning to solve the feature selection problem. The system keeps exploring the state space while it is moving through the available space of features to select the best subset. In this system, we have used the decision tree branches. Therefore, each subset is represented by a branch. The main idea of the proposed feature selection algorithm is to select the applicable subset of features, which are mostly involved in constructing efficient branches. In its preliminary outset, the system endeavors to build the first branch without any pre-installed knowledge (exploring the environment). As iterations transpire in linearly successive alternation, the system accumulates experiences that furnish the ground for constructing better branches (diverse, relevant, etc.) using the propounded Transition Similarity Measure (TSM). Out of the best branches, we select the most utilized features in creating them (See the Fig. 2). The contributory aspirations and the quintessential mainsprings of this study are fourfold.

1) A reinforcement learning-based method is developed to be used in selecting the best subset of features.
2) The proposed system traverses the state space to select the informative subset using a modified version of decision tree branches. Since the transition between states (feature subsets) is controlled using Decision tree branches, the proposed system is straightforwardly accessible. As a result, the spotlighted solution, through effective implementation of the suggested feature selection method, our proposed system is rendered comprehensively interpretable.
3) Transition similarity measure (TSM) is intended to maintain the progressive sustainability the system's environmental exploration by creating new branches and simultaneously exploiting what has learned to avoid redundancy and maximize diversity.
4) The proposed system can be adapted to any problem (it is not dependent on a specific dataset) because our feature selection problem is considered as reinforcement learning.

The remainder of this paper is organized as follows: section two presents the related works. Section three is devoted to the problem and our introduced contributions. In the fourth section, the results of the proposed system are introduced. As to the last section, it is put forward to conclude this work.

## 2. Related Works

Extensive research and deeply thorough breakthroughs have been disclosed in the domain of feature selection as an ever-evolving field of study [3, 13, 19, 27]. In this section, some wrapper algorithms similar to the fundamental one encompassed by this paper are briefly reviewed. The first algorithm is ubiquitous in FS state of the art, which is forward selection [28, 29]. (1) it starts with an empty subset; (2) adds to the subset the feature that increases its performance ; (3) repeats Step 2 until all features have been examined or until no better performance is possible ; (4) returns the subset of features that yields maximum performance on the validation set [30]. This method is fast and effective, but it tends to over-fit the validation set. In [20], the authors proposed a new algorithm entitled ensemble feature selection, which significantly reduces this problem.

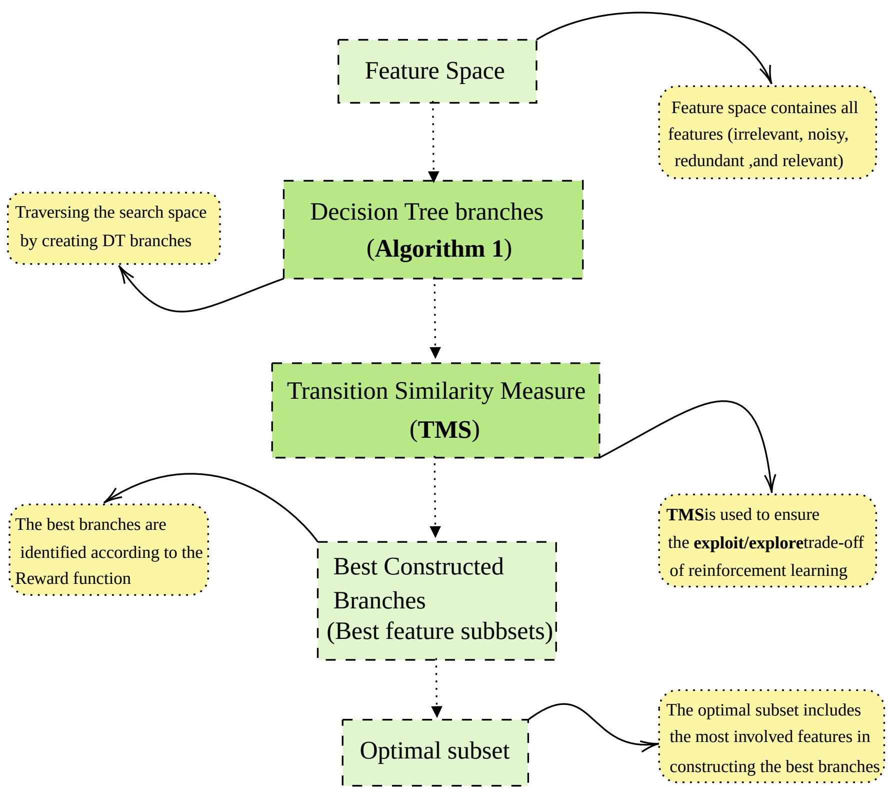

Feature Space
Feature space containes all features (irrelevant, noisy, redundant ,and relevant)
Decision Tree branches (Algorithm 1)
Traversing the search space by creating DT branches
Transition Similarity Measure (TMS)
TMSis used to ensure the exploit/exploretrade-off of reinforcement learning
The best branches are identified according to the Reward function
Best Constructed Branches (Best feature subbsets)
Optimal subset
The optimal subset includes the most involved features in constructing the best branches

**Figure 2.** Flow-chart of the proposed feedback feature selection system

First, for each feature, they train different models using different classification algorithms. Then, they store them in a library of models. Second, they use a selection with replacement technique [30] to find the optimal subset of models that, when averaged together, achieves excellent performance. Another wrapper method based on graph representation is proposed in [14], where the node degree is used as a criterion to select the best features subset among the whole features space. This algorithm consists of two phases: (1) Choosing features to be used in graph construction. (2) Constructing a graph in which each node corresponds to each feature, and each edge has a weight corresponding to the pairwise score among features connected by that edge. Finally, the best features are the nodes with the highest degree. In [31], a pairwise feature selection (FS-P) has been introduced, features are evaluated in pairs using decision tree classifier. First, it ranks features individually. Second, it involves the machine learning algorithm (Decision tree) to evaluate pairs of features. In [32, 33], a well-known wrapper approach is presented, Recursive Feature Elimination using Random Forest (RFE). RFE performs feature selection recursively. At the first iteration, the model (Random forest) is trained on whole set of attributes. After ranking features according to the model's importance, the least important features are eliminated. As iteration takes place, the considering set of features become smaller and smaller until the desired number of features is reached.

Random forests are among the most popular machine learning algorithms [34]. Thanks to its performance, robustness, and interpretability, RF has proved the frequency of its beneficial applicability. They can select informative variables [11]. RF performs feature selection using mean decrease impurity and means decrease accuracy criteria [35]. Mean decrease impurity is used to measure the decrease in the weighted impurity in trees by each feature. Therefore, the features are ranked according to this measure. Mean decrease accuracy is a measure of the feature impact on model accuracy. The values of each feature are permuted first. Then, we measure how this permutation decreases model accuracy. The informative features decrease the model accuracy significantly, while unimportant features do not.

As opposed to the traditional feature selection (FS) formalization and the inspiration generated from the reinforcement learning approach, the feature selection problem can be effortlessly handled with the profitable reliability of our proposed system. The feature space using our approach can be seen as a Markov decision process (MDP) [36, 37], where each subset of features is represented by a state (decision tree branch). Our system explores the state space while it exploits the gathered experiences so far using the proposed transition similarity measure (TSM). In [38], the authors proposed a method based on reinforcement learning (RL) for selecting the best subset. First, they use an AOR (average of rewards) criterion to identify the effectiveness of a given feature in different conditions. AOR is the average of the difference between two consecutive states in several iterations. Second, they introduce an optimal graph search to reduce the complexity of the problem.

The way our system traverses from one state to another is handled using decision tree branches to represent each state, as mentioned before. In its totality, this technique is similar to the way RF creates branch. The RF method creates multiple trees. For each tree, only a random subset of input variables is used at each splitting node. Therefore, the final trees of RF are independent of each other, and they do not learn from the previously created trees. On the other hand, our system can learn from prior attempts. At each iteration, it explores new branches and exploits the assimilated knowledge to create highly-performative and qualitative ones in the subsequent iteration.

## 3. Feedback Feature Selection System

This paper foregrounds to the brings a new feature selection system based on reinforcement learning; the proposed system principally comprises three parts. First, decision tree branches are used to traverse the search space (features space) to create new rules (branches or feature subsets) and select the best feature subset. Second, a transition similarity measure (TSM) is introduced to ensure that the system keeps exploring the state space by creating new branches and exploiting what it has learned so far to circumvent the problematic implications or the drawbacks of redundancy. Finally, the relevant features are the most involved ones in constructing the branches of high quality. For further illustrative explications, the subsequent section will accessibly resurface the general framework of reinforcement learning and delineate the know-how dimensions in which our system can synthesize the benefits of this powerful approach.

### 3.1. Reinforcement Learning Problem

RL is the most active and fast-developing area in machine learning and is one of three basic machine learning approaches, alongside supervised learning and unsupervised learning. RL consists of the following concepts: Agent, environment, actions, and reward. The agent takes action A and interacts with an environment to maximize the total reward received R. At iteration t, the agent observes state St from the environment. In return, the agent gets a reward Rt. The agent takes action $A_t$. In response, the environment provides the next state $S_{t+1}$ and reward; the process continues until the agent will be able to take the right actions that maximize the total reward. The agent must balance between exploiting what has been learned so far and continuously exploring the environment to gather more information that may help in maximizing the total reward.

- **Agent**: An agent takes actions. In our case, the agent is the proposed feature selection system.
- **Actions** is the ensemble of all possible moves the agent can make, for our system, the actions are the nodes that may be used to create a branch.
- **Environment** is the feature space through which the system moves. It receives the system's current state and action as input; then, it returns the reward and the next state of the system.
- **State** is the current situation where the agent finds itself. In our context, this is the current node of the branch.

As the reinforcement concepts are transparently tackled and highlighted, the following steps may unfold in depth with the constitutive mainstay or the technical infrastructure of our proposed algorithm.

The feature selection system (agent) scrutinizes the environment, and then starts with a single node arbitrarily without any pre-stockpiled knowledge (exploration phase), which branches into possible outcomes. Each of those outcomes leads to the next nodes (action). To indicate how effective the chosen action is, a difference between two consecutive states is produced. Since the depth is not yet reached, the system keeps adding one node at a time in order to create a branch. As iterations take place, the system assembles experiences and becomes able to take actions that maximize the overall reward R. As a yielded offspring, branches of high quality are created. A transition similarity measure is proposed to establish a balanced equipoise between exploiting what has been learned so far to choose the next action that maximizes rewards, and continuously exploring the feature space to achieve long-term benefits. The way we construct the branch is the same as the decision tree (c4.5), the difference is when we add a node to the branch, we retain only the best branch with the highest threshold. The following steps give more precise information about creating a branch.

### 3.2. Steps to Create a DT Branch

We start with a random feature as the root of the branch. As long as the branch did not reach the desired depth or min sample leaf yet, the system keeps adding to the branch one node at a time. The added node is the one we obtained using the feature and its threshold that produces the highest AUC score (Area Under the Curve ROC). The idea behind using depth and min simple leaf parameters as stopping criteria is to avoid as much as possible the over-fitting problem. The most common stopping method is min sample leaf, which is the minimum number of samples assigned to each leaf node. If the number is less than a given value, then no further split can be done, and the node is considered a final leaf node. Besides, the depth of the branch is very useful in controlling over-fitting because the deeper the branch is, the more information captured by the data and more the splits it has, which leads to predict well on the training data. However, it fails to generalize on the unseen data.

### 3.3. Reward Function

A reward function R [38] is used to calculate the score at each level of the branch by computing the difference between the score of the current branch and its score after a new node is added (DS). The DS indicates how useful the newly added feature is.

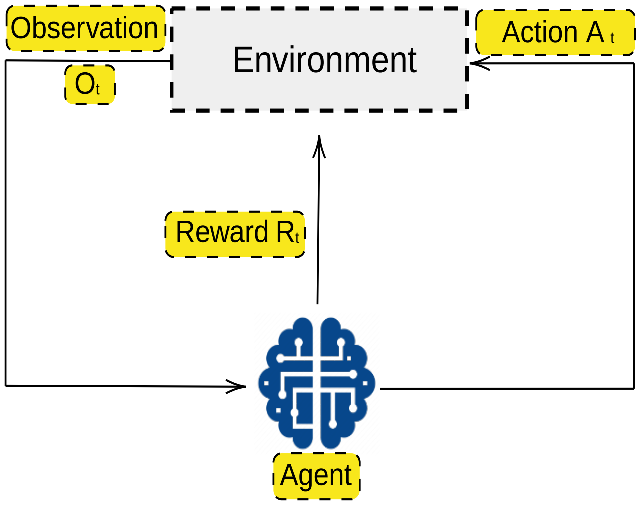


**Figure 3.** Reinforcement learning framework

**Algorithm 1**: Create a DT branch

```
1: Create the root node and choose the split feature. Choose the first feature randomly.
2: Compute the best threshold of the chosen feature.
3: Split the data on this feature into subsets in order to define the node.
4: Compute the AUC score on left and on right of the node, then, we keep the branch with the best AUC score.
5: Add the children node to root node.
6: Choose the next best feature.
7: Repeat from STEP 2 to STEP 5 until the desired depth or min sample leaf of the branch is reached.
```

This function is defined as follows:

$$(AUC_{next} - AUC_{current}) \times \log\ (\|Subset_{current}\|) \tag{1}$$

Where $AUC_{next}$ and $AUC_{current}$ is the score of the current branch and the score after adding a new node, $Subset_{current}$ is the length of samples used to split an internal node.

### 3.4. Transition Similarity Measure

#### Definition (Transition)

A transition is the process in which something changes from one state to another. In our system, the transition is the link between two successive nodes of the same branch.

#### Transition Similarity Measure

We proposed a transition similarity measure (TSM) to ensure that our system keeps exploring the state space, learning new rules, and preventing the redundant branches. For each branch, we stock all transitions with the corresponding samples used to split each internal node. Since the algorithm is iterative, different branches may share the same transitions, which is not a problem. In the case when the majority of the samples (higher than a given threshold) are equally used by those transitions of different branches, those two transitions are deemed similar, which is a huge problem. Allowing similar transitions to be in different branches can lead to constructing redundant and useless branches.

Therefore, the system keeps learning the same rules and branches. This means that the system will be expensive in terms of execution time, while the system should be less resources consuming (run time and storage requirement), and the branches should be strong and diverse.

The similarity between two transitions is computed by the following formula:

$$TSM = \frac{|S1 \cap S2|}{\|Subset_{current}\|} \tag{2}$$

Where $\|S1 \cap S2\|$ is the number of shared samples between two transitions.

### 3.5. The Proposed FS method

Since the proposed algorithm is iterative, the number of iteration N is given as the input. The reward function is set to zero at the beginning. Our system starts with an empty set F, and at each iteration, the system creates a new branch and adds it to F. If the next subset (branch) is already experienced by the system (seen by the system), the system uses this gathered experiences in the upcoming iterations. Otherwise, the system keeps exploring new rules, new patterns, and new branches.

### 3.6. Starting Example

To explain the proposed algorithm further, we suggest the following example. We suppose that we have a dataset of 10 features. The figure bellow (Fig. 4) contains the whole space of features. The purpose is

**Algorithm 2**: Feedback feature selection system pseudo-code

```
1:  Input:
2:  N: number of iteration
3:  S: Similarity
4:  Output:
5:  R: Reward
6:  for iteration=1 to N do:
7:    F={} to store subsets (branches)
8:    Step1: Create the root node (Algorithm 1)
9:    Step2: Find all possible transitions ($P_t$)
10:   Add the created node to F    for $T_i$ in $P_t$ do:
11:   for $T_i$ in $P_t$ do:
12:     if $T_i$ exist in F then:
13:       Compute the similarity between the two transitions using TSM
14:       if similarity higher than S then:
15:         f = New node (keep learning and exploring the environment )
16:         R{F}= $(AUC_{next} - AUC_{current}) \times log\ (|Subset_{current}|)$
17:       else:
18:         $F \cup f$: Add the chosen node to the branch
19:       end
20:     end
21:     Step3: Repeat until the desired depth and min sample leaf is reached
22:   end
23: end
24: Return Reward R
```

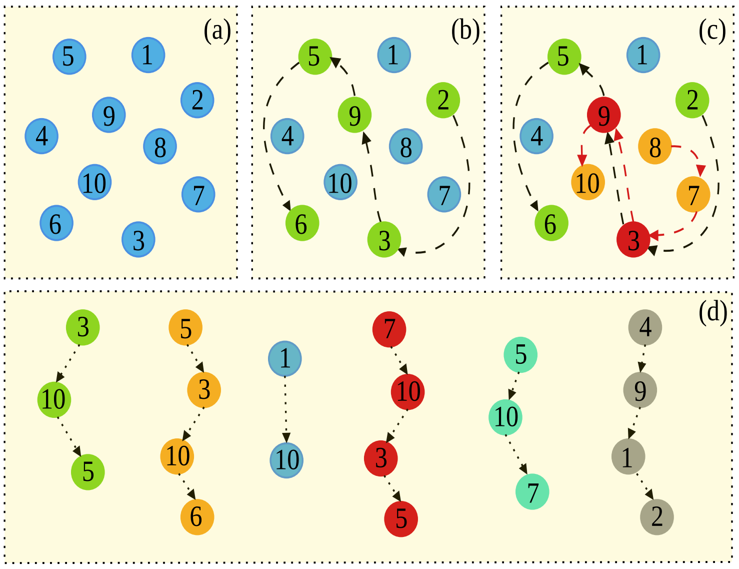


**Figure 4.** FBS proposed algorithm main steps

to select the best subset of features using the proposed system.

1) First iteration 4(b): The system traverses the features space and creates the first branch without any prior knowledge. At each level of the branch, the system stores the AUC score using the reward function R. Moreover, it stores each transition (2 ← 3, 3 ← 9, 9 ← 5, 5 ← 6) and its corresponding subset of samples.

2) The second iteration 4(c): As we can see in the second iteration, the transition (3 ← 9) appeared for the second time. Here the TSM (transition similarity measure) should be involved. If two transitions of different branches are similar (nodes with green color), the system should not allow them to be in the next branches (the current branch included). The system has to explore the state's environment to find new rules to prevent the redundancy in creating branches.

3) The N iteration 4(d): After N iterations, the system is capable of identifying the best branches using the gathered experiences during each iteration. The top ranked branches constructed using the system are the illustrated in the subfigure 4(d).

**Table 1.** Characteristics of the benchmarking datasets.

| #No | Datasets | #Features | #Examples | Distribution | Class |
|---|---|---|---|---|---|
| 1: | Spambase | 57 | 4601 | 39% + / 61% - | 2 |
| 2: | Numerai | 22 | 96320 | 50% + 50% - | 2 |
| 3: | Clean | 167 | 6598 | 15% + / 85% - | 2 |
| 4: | SPECT | 32 | 80 | 33% + / 67% - | 2 |
| 5: | Caravan | 86 | 5823 | 6% + / 94% - | 2 |
| 6: | Ionosphere | 34 | 351 | 64% + / 36% - | 2 |
| 7: | Credit card | 24 | 30000 | 22% + / 78% - | 2 |
| 8: | Eye | 15 | 14980 | 45% + / 55% - | 2 |
| 9: | Sonar | 61 | 208 | 47% + / 53% - | 2 |

**Table 2.** Characteristics of the benchmarking datasets.

| #No | Datasets | Interation N | Depth | Similarity |
|---|---|---|---|---|
| 1: | Spambase | 360 | 4 | 0.95 |
| 2: | Numerai | 100 | 5 | 0.9 |
| 3: | Clean | 1000 | 10 | 0.65 |
| 4: | SPECT | 100 | 4 | 0.7 |
| 5: | Caravan | 650 | 8 | 0.6 |
| 6: | Ionosphere | 220 | 4 | 0.62 |
| 7: | Credit card | 200 | 7 | 0.8 |
| 8: | Eye | 110 | 6 | 0.9 |
| 9: | Sonar | 600 | 3 | 0.65 |

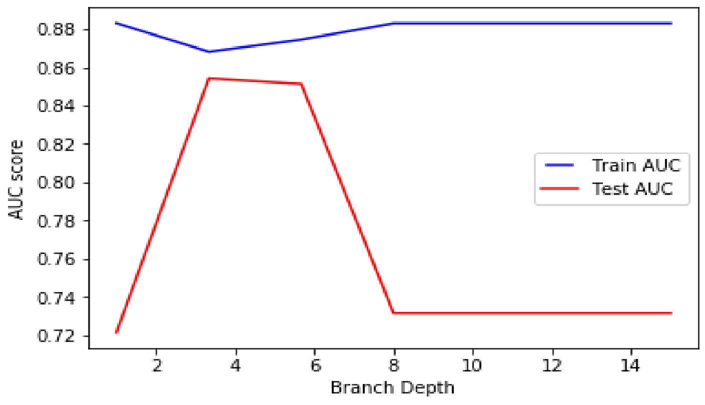


**Figure 5.** Over-fitting problem

From the above Figure 4, it is clear that the top subset of features is [3, 5, 10], because those features are involved the most in creating the best branches.

## 4. Experimental Results and Discussion

This experimental section attests to the efficiency of the proposed feedback feature selection system (FBS) in selecting the best features. Two benchmarks have been conducted, and then the profitable serviceability of our system is appraised by comparing it with two feature selection algorithms. The first one is the popular wrapper algorithm named Recursive Feature Elimination RFE (RFE-RF). The second one is the pairwise feature selection algorithm (FS-P), which is recently proposed and proved its effectiveness in identifying the best features [31].

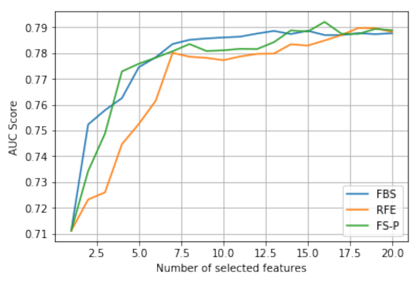
### 4.1. Benchmarking Datasets

In this paper, nine binary classification datasets have been employed in different experimental design aiming to evaluate the performance of the proposed feature selection method.

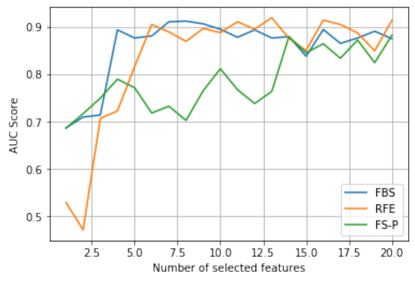
The datasets are chosen to be different in terms of class distribution (balanced or imbalanced), linearity, dataset shift, number of instances and variables. The datasets, which are publicly available, are collected and downloaded from UCI repository and kaggle platform [39]. An overview of the main characteristics of each dataset is illustratively tabulated in Table 1.

### 4.2. Experiments Settings

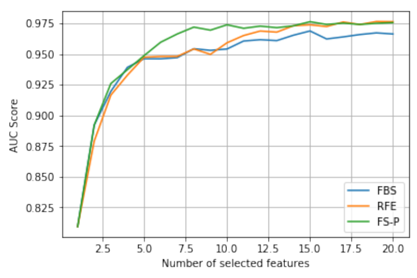
Two experimental endeavors are undertaken to estimate the workable prospects and the consequential ramifications of our proposed system. Initially, we



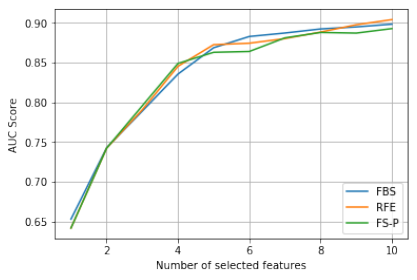
**(a)** Dataset: credit card



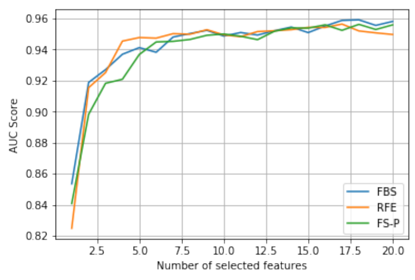
**(b)** Dataset: sonar



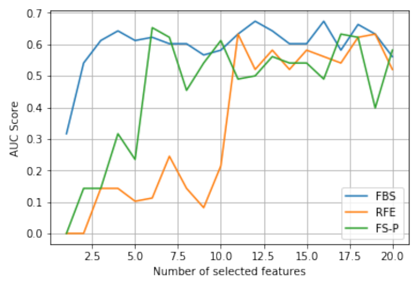
**(c)** Dataset: spambase



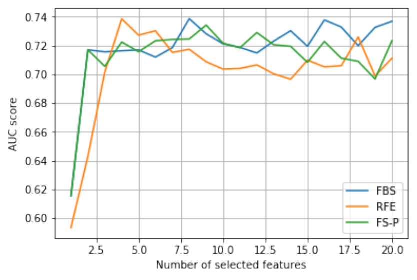
**(d)** Dataset: Eye



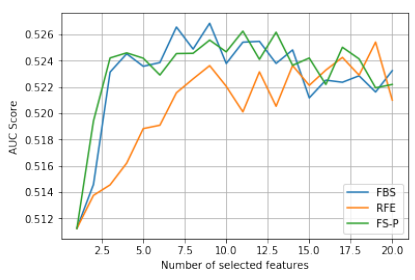
**(e)** Dataset: musk



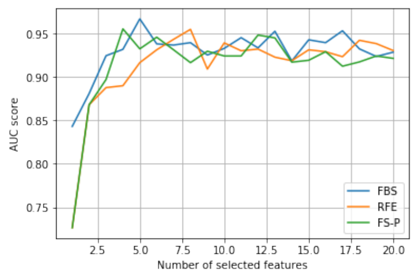
**(f)** Dataset: SPECT



**(g)** Dataset: Caravan



**(h)** Dataset: Numerai



**(i)** Dataset: Ionosphere

**Figure 6.** Performance of our system compared with the selected features selected by pairwise method on nine benchmark datasets

will empirically embody the applications of the proposed algorithms based on the datasets displayed in Table 1 in terms of Area Under the Roc Curve (AUC) where FBS is compared with the pairwise method, namely FS-P and with RFE.

In correlative parallelism with the previous step, the subsequent stage will demonstrate the eligible capability of the FBS system in encircling the practical subset as swiftly as possible through the exclusive employment of the few features supplemented by second benchmarking.

All datasets are segmented into two subsets; one subset is employed for training and testing the branches using cross-validation with 3-folds while the other subset is quarantined and cast aside (holdout set) and the performance of the final selected feature subset is evaluated on it. For the sake of a fair comparison, the final selected subset using FBS, FS-P, and RFE is evaluated using a Random Forest with a grid search strategy for the hyper-parameters. The AUC score is calculated using the out of bag (OOB) score of the random forest classifier. Since the benchmarking datasets used in this paper to evaluate the proposed system are unbalanced, the AUC metric is considered the best choice. Moreover, the AUC metric generally can be viewed as a better measure than accuracy [40].

## 4.3. Feedback System Parameters

The Feedback system parameters incorporate a systematic trilogy of changeable parameters which are in a dynamic alteration in accordance with each dataset.

- S is the similarity value.
- D is concerned with the indication of the branches' depth.
- N reflects the number of iterations.

To exemplify the probable changeability of these parameters. Datasets with large size, the N and D values should be higher since the best branches, in this case, should be deeper. The following table supplements a panoramic overview underlying the best parameters used for each dataset.

As clearly articulated in the aforementioned section, the choice of parameters is indispensable. The following graph delineates the influence of the depth parameter (D) on the quality of the constructed branches using the sonar dataset. This graphic plot displays a summative snapshot of the train and the test AUC scores after the gradually exponential variation of D parameter from 1 to 15 is fulfilled.

The recorded outcomes on the sonar dataset show clearly that the branches prone to over-fit for large depth values because the branches perfectly predict all of the train data (the blue line). However, they fail to generalize on unseen data (the red line). As can be visually observed, the best depth for the sonar dataset itself equals three (D=3).

### 4.4. Conducted Experiments

The proposed method is compared to the RFE and FS-P approachs in terms of prediction AUC score. In this manuscript, two empirically conclusive and thoroughgoing experiments are conducted.

1) First Experiments: To evaluate our proposed approach FBS, we compare the obtained performance (in terms of AUC score) by FBS with the wrapper method (Recursive feature elimination with random forest RFE) and with the pairwise algorithm FS-P.

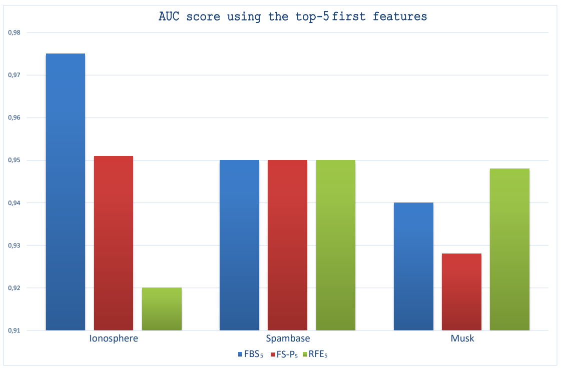
2) Second Experiments: This experiment is conducted to show the ability of the proposed system FBS in achieving the maximum performance using just a few features. For a fair comparison between FBS, FS-P, and RFE, we fix the generated subset size for all algorithms compared as follows: subset of size 5 ($FBS_5$, $FS-P_5$, $RFE_5$), a subset of size 10 ($FBS_{10}$, $FS-P_{10}$, $RFE_{10}$) and subset of 15 ($FBS_{15}$, $FS-P_{15}$, $RFE_{15}$).

### 4.5. Results and Discussion

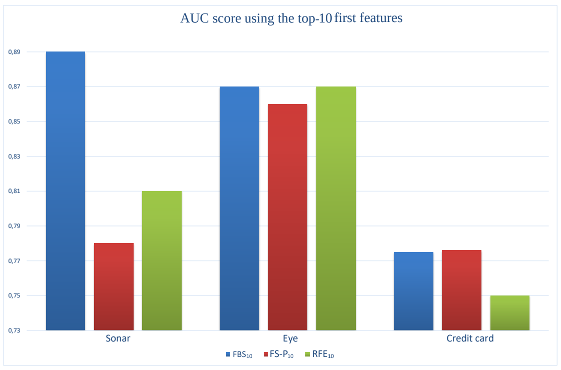
After selecting the feature subset, the same classifier (RF) is essentially mandatory to calculate the AUC score. The Random forest is utilized to determine the test performance for the top-ranked features of each employed dataset. The comparative juxtaposition between FBS, FS-P and RFE is accessibly represented in Figure 6 (First experiment).

As stated, our feature selection algorithm FBS exceeds and outstrips FSP and RFE considerably in almost all datasets, such as SPECT (Figure 6(f)), credit card (Figure 6(a)), ionosphere (Figure 6(i)), musk (Figure 6(e)), caravan (Figure 6(g)), and sonar (Figure 6(b)), except for spambase dataset ( 6(c)).

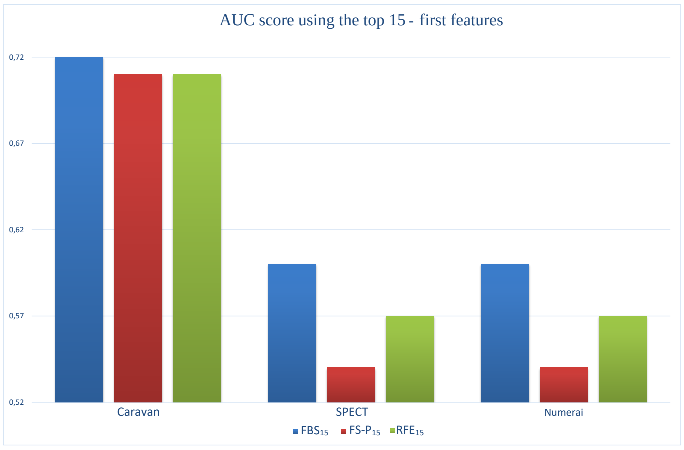
For the numerai dataset (Figure 6(h)), our method has a restrictively limited, if not downgraded performance at the beginning compared to RFE and FS-P. As our method does not select just the best-ranked feature as a starting point to prevent selecting a suboptimal subset but also attempt to maximize the overall performance of the selected subset taking into account



**(a)** Dataset: ionosphere, Spambase and Musk datasets



**(b)** Dataset: sonar, Eye and Credit card datasets



**(c)** Dataset: Caravan, SPECT and Numerai datasets

**Figure 7.** The performance of FBS, RFE and FS-P using feature subsets of 5, 10, and 15 features

the interactions between features. After the selection of the numerai dataset's fifth feature (Figure 6(h)), the aforementioned behavioral veracity is rendered observable, and FBS shows its drastically improved performance over FS-P and RFE.

Table 2 shows the best parameters used in our feedback system. The insightful bottom-line conclusion we can excerpt from the table is that the choice of the best parameters to use in each dataset is crucial, which means that the parameters should be carefully chosen to construct branches with high quality.

The purpose of the proposed feature selection method is not only to improve the classification performance but also to yield excellent performance using a minimum number of features (select the fewest possible number of features).

Figure 7 shows the number of selected features with the highest AUC score on nine benchmarks data sets. As it is illustrated through this benchmarking, FBS selects the proper features compared with FS-P and RFE almost in all datasets. One point to mention here is that the proposed feedback system can find the best subset using a minimum amount of features, as shown in Figure 7. Thus, the minimum resources requirement, fast execution, and better generalization.

## 5. Conclusion

In this paper, we have proposed a new feature selection method based on the decision tree branches concept to represent feature subsets. The proposed system deals with the FS problem as a reinforcement learning problem; the system tries to find a compromise between exploring the search space by experiencing new rules (creating new branches) and exploiting the gathered experiences so as to choose the right actions (relevant feature). The exploit/explore tradeoff is controlled by the proposed TSM. The proposed system can construct the best branches, hence, selecting the best subset of features.

To assess the effectiveness of the selected features using our proposed method, we have conducted an extensive set of experiments using nine benchmarking datasets. The results confirm that the proposed feedback feature selection system is not only effective at selecting the best performing subsets of features that produce the best performance but also choose the fewest number of features.

**AUTHORS**

**Yassine Akhiat*** – Department of Informatics, faculty of sciences dhar el mahraz, USMBA, Fez Morocco, e-mail: yassine.akhiat@usmba.ac.ma.

**Ahmed Zinedine** – Department of Informatics, faculty of sciences dhar el mahraz, USMBA, Fez Morocco, e-mail: ahmed.zinedine@usmba.ac.ma.

**Mohamed Chahhou** – Department of Informatics, faculty of sciences, UAE, Tetouan Morocco, e-mail: mchahhou@hotmail.com.

*Corresponding author

## References

[1] R. Roelofs, S. Fridovich-Keil, J. Miller, V. Shankar, M. Hardt, B. Recht, and L. Schmidt, "A meta-analysis of overfitting in machine learning," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 9179–9189.

[2] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, vol. 1168, no. 2. IOP Publishing, 2019, p. 022022.

[3] M. Li, H. Wang, L. Yang, Y. Liang, Z. Shang, and H. Wan, "Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction," *Expert Systems with Applications*, vol. 150, p. 113277, 2020.

[4] H. Liu, H. Motoda, and L. Yu, "A selective sampling approach to active feature selection," *Artificial Intelligence*, vol. 159, no. 1-2, pp. 49–74, 2004.

[5] Y. Akhiat, Y. Asnaoui, M. Chahhou, and A. Zinedine, "A new graph feature selection approach," in *2020 6th IEEE Congress on Information Science and Technology (CiSt)*. IEEE, 2021, pp. 156–161.

[6] D. M. Atallah, M. Badawy, and A. El-Sayed, "Intelligent feature selection with modified k-nearest neighbor for kidney transplantation prediction," *SN Applied Sciences*, vol. 1, no. 10, pp. 1–17, 2019.

[7] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*. Springer, 2008, vol. 207.

[8] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature extraction*. Springer, 2006, pp. 1–25.

[9] A. Yassine, "Feature selection methods for high dimensional data," 2021.

[10] Y. Manzali, Y. Akhiat, M. Chahhou, M. Elmohajir, and A. Zinedine, "Reducing the number of trees in a forest using noisy features," *Evolving Systems*, pp. 1–18, 2022.

[11] Y. Akhiat, Y. Manzali, M. Chahhou, and A. Zinedine, "A new noisy random forest based method for feature selection," *CYBERNETICS AND INFORMATION TECHNOLOGIES*, vol. 21, no. 2, 2021.

[12] S. Abe, "Feature selection and extraction," in *Support vector machines for pattern classification*. Springer, 2010, pp. 331–341.

[13] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[14] Y. Akhiat, M. Chahhou, and A. Zinedine, "Feature selection based on graph representation," in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*. IEEE, 2018, pp. 232–237.

[15] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection," *IEEE/ACM transactions on computational biology*

*and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2015.

[16] L. A. Belanche and F. F. González, "Review and evaluation of feature selection algorithms in synthetic problems," *arXiv preprint arXiv:1101.2320*, 2011.

[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[18] B. Nithya and V. Ilango, "Evaluation of machine learning based optimized feature selection approaches and classification methods for cervical cancer prediction," *SN Applied Sciences*, vol. 1, no. 6, pp. 1–16, 2019.

[19] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Computational Statistics & Data Analysis*, vol. 143, p. 106839, 2020.

[20] Y. Akhiat, M. Chahhou, and A. Zinedine, "Ensemble feature selection algorithm," *International Journal of Intelligent Systems and Applications*, vol. 11, no. 1, p. 24, 2019.

[21] L. Čehovin and Z. Bosnić, "Empirical evaluation of feature selection methods in classification," *Intelligent data analysis*, vol. 14, no. 3, pp. 265–281, 2010.

[22] Y. Asnaoui, Y. Akhiat, and A. Zinedine, "Feature selection based on attributes clustering," in *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*. IEEE, 2021, pp. 1–5.

[23] Y. Bouchlaghem, Y. Akhiat, and S. Amjad, "Feature selection: A review and comparative study," in *E3S Web of Conferences*, vol. 351. EDP Sciences, 2022, p. 01046.

[24] A. Destrero, S. Mosci, C. D. Mol, A. Verri, and F. Odone, "Feature selection for high-dimensional data," *Computational Management Science*, vol. 6, pp. 25–40, 2009.

[25] V. Fonti and E. Belitser, "Feature selection using lasso," *VU Amsterdam Research Paper in Business Analytics*, vol. 30, pp. 1–25, 2017.

[26] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[27] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[28] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[29] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE, 2002, pp. 306–313.

[30] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 18.

[31] A. Yassine, C. Mohamed, and A. Zinedine, "Feature selection based on pairwise evalution," in *2017 Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2017, pp. 1–6.

[32] B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," *Statistics and Computing*, vol. 27, no. 3, pp. 659–678, 2017.

[33] J. Kacprzyk, J. W. Owsinski, and D. A. Viattchenin, "A new heuristic possibilistic clustering algorithm for feature selection," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 8, 2014.

[34] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] H. Han, X. Guo, and H. Yu, "Variable selection using mean decrease accuracy and mean decrease gini based on random forest," in *2016 7th ieee international conference on software engineering and service science (icsess)*. IEEE, 2016, pp. 219–224.

[36] R. Sutton and A. Barto, "Reinforcement learning: An introduction. 2017. ucl," *Computer Science Department, Reinforcement Learning Lectures*, 2018.

[37] Y. Fenjiro and H. Benbrahim, "Deep reinforcement learning overview of the state of the art." *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 20–39, 2018.

[38] S. M. H. Fard, A. Hamzeh, and S. Hashemi, "Using reinforcement learning to find an optimal set of features," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1892–1904, 2013.

[39] M. Lichman, "Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california, school of information and computer science," *URL: http://archive. ics. uci. edu/ml*, 2013.

[40] F. F. Provost, "T., and kohavi, r. the case against accuracy estimation for comparing classifiers," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.