# Interactive Learning Model For Feature Selection Using Classifiers

**Md Sirajul Huque[1*], V.Kiran Kumar[2]**

[1, 2] *Dept of Computer Science and Technology, Dravidian University, Andhra Pradesh, India, [1]sirajulhuque1985@gmail.com, [2]kirankumar.v@rediffmail.com*
***Corresponding Author:** Md Sirajul Huque*
**E-mail: [1]sirajulhuque1985@gmail.com,*

**KEYWORDS**
Classification, mutli-agent based learning, feature selection, autoencoder, Gaussian mixture model.

**ABSTRACT**
A lot of attention to feature selection has been focused in the multi-directional field of machine learning to enhance prediction accuracy through feature set analysis and lowering dimensionality. Finding the best features from many feature spaces is challenging, even if several attention techniques have been investigated for feature selection. Therefore, an agent-based interactive learning model is proposed for feature selection with achieving maximum feature subset. At the outset, state-level feature selection is taken in an interactive learning framework, where agents create the environment's based on the state with corresponding actions and build a stable representation of it to feed into interactive learning. The range of feature subset space is explored by agent activity using the interactive learning technique. An interactive learning (IL) model is considered using an exploration or current strategy. The experiment is demonstrated per the proposed model with specific data sets where the suggested strategy significantly improved over more conventional approaches. According to the comparison results, Ada boost performs better (Train score - 1.00). It outperformed competing classifiers on the Parkinson's dataset (testing score:0.93, accuracy:0.93, mean score: 0.67) regarding performance.

## 1. Introduction

In machine learning, feature selection is crucial for predicting distinct classes or subclasses. The primary objective of feature selection is to be most beneficial for a downstream prediction task [1, 2]. Reducing dimensionality, decreasing training time, increasing generalizability, preventing overfitting, boosting predictions, and making them easier to grasp and interpret are all possible with effective feature selection. Our research focuses on computerized feature selection related to extended effective future prediction challenges. In order to optimize prediction tasks, filter techniques [3] employ the feature ranking using a given score. Regarding embedded approaches, predictive models are quite strict with their structural assumptions. LASSO prioritizes features with non-zero weights, for instance. Feature selection is not a simple procedure; it requires (i) a technique for feature evaluation scores, (ii) the ability to rapidly create the optimal subset of features, and (iii) enable prediction models.

Very few researchers have created reinforcement learning algorithms that find the best long-term decisions while learning [4]. These characteristics substantially improve the capacity to automate feature sub-space finding. Most prior work on automated feature selection relied on a single decision-making agent in [5,6]. In these models, a single agent decides whether to pick or reject all N characteristics. Therefore, for that model, the size of this agent's action space is $2^N$. Using a similar method, evolutionary algorithms [7,8,9] frequently locate local optima. Therefore, this research aims to present an approach to automated feature selection based on interactive learning model. This model faces new problems, such as (a) How can the problem's perspective be changed to where IL can reduce the action space? (b) How to improve the accuracy as per the state's action in IL? (c) how can we optimise the discovery of the finest features?

We have considered the feature selection challenges inside an interactive learning-based approach framework to address these concerns. Specifically, we begin by assigning one agent to each feature; the agents' task is to choose which features belong to them and which don't. After that, we provide a strategy incorporating feature-label relevance and feature-feature redundancy to incentivize accurate prediction. This kind of collaboration and rivalry amongst agents is necessary for efficient feature discovery. We also offer improved methods using the dynamically selected subset of features to derive a constant-length representation vector. As per current models [10], during the experience replay, the agent draws on samples stored in its memory, including several training data types, to train the model. Because it's necessary to consider every possible state, all of the memory samples in the automated control area are typically considered in interactive learning.

As part of their exploration strategy, interactive learning agents explore their environment for rewards to improve their exploration trajectories. However, as the size of the state space increases, its exploration efficiency decreases dramatically. Reducing the exploration space is achieved using an interactive learning (IL) approach [11,12]. If we incorporate this, we can change the state's representation and steer Illinois into more productive research paths.

We have considered state representation strategies for each feature where each feature is tested for selection or not as per our proposed model i.e., multi-agent interactive learning model. In this model, we have considered specific parameters such as agent, actions, environmental setting, states, rewards etc to analyze the performance of selected features. Few mathematical models have taken to represent the above parameters to proceed the performance of the feature selection approach. We have considered two datasets for experiments, and different methods have been used to perform the feature selection approach. From various methods, Ada boost performs well than other methods.

This study primarily contributes to the following areas:

(1) An interactive learning approach framework is utilized to reframe the feature selection issue. A unique incentive structure is constructed to direct the agents' cooperation and competition.
(2) we have considered two investigate the integrity of the state representation using meta-descriptive statistics.

The running summary of the remaining parts is as follows. Section 2 is explained for the background of related work in light of the rationale behind our study approach. Different components of the proposed model are considered in Section 3. The concept of multi-agent interactive learning with state representation is described in Section 4. Section 5 shows that the proposed model works by comparing its performance on two datasets. The papers are concluded in Section 6 with recommendations for further research.

## 2. Related Work

Feature selection models and the prediction of accurate features have been subject to many approaches. In filter methods, features are sorted according to their relevance scores, and features with the highest rankings are collected. Filter techniques' speed and cheap processing costs make them useful on large datasets. The following predictors as they pertain to feature selection. As opposed to filter strategies, wrapper methods aim to improve prediction performance [13]. Two examples of wrap-per methods are bound and branch algorithms [14,15]. Unfortunately, traversing the space of feature subsets is an NP-hard problem since its size rises exponentially with the number of features. Evolutionary algorithms can ensure locally optimal results while reducing the processing cost [7,8,9]. The two most common methods utilised in embedded systems are LASSO[16] and decision trees [17]. Embedded methods frequently clash with external predictors, even when they work well with integrated predictors. A novel approach that has recently demonstrated outstanding success in solving the feature selection problem is reinforcement feature selection, which employs reinforcement learning.

A reinforcement learning agent in the one-agent model adjusts its environment, gets feedback as a reward, and then utilizes this data to improve its future action decisions [18]. Interactions between agents and their environments are essential to the multi-agent formulation. Using an estimated path value, this technique considers all dimensions in a high-dimensional space and selects a path accordingly [19]. When it came to managing taxi fleets, this system worked well [4]. The issue is that these methods may miss important environmental data as they don't employ representation learning to establish their states. We also know that multi-agent reinforcement learning training speeds are inadequate due to the vast action space, even though these systems seldom consider strategies to improve training efficiency. Previous studies have utilized A single agent to create decisions [5,6]. This agent, on the other hand, must determine if all N features are selected. Thus, the number of actions were created on those features as $2^N$ size and compared with different algorithms as [7,8,9,24] for local optimum.

## 3. State level based Interactive Learning Model

Feature set analysis using a multi-agent interactive learning model is considered in this section. Figure 1 shows the proposed procedure or flow of work for interactive learning-based multi-agent feature selection with select and deselect approaches. We create a feature agent for each collection of features to start the selection procedure. Note that, here, each feature is considered as agent. After settling on a feature agent, the suggested model dictates whether features will be chosen individually or in groups. The interactive learning-based approach system includes agents, states, environments, rewards, techniques for awarding rewards, and agent actions; they were all built as part of the agent-based interactive learning model. Moreover, we considered incentive assignment methods that use feature correlations.

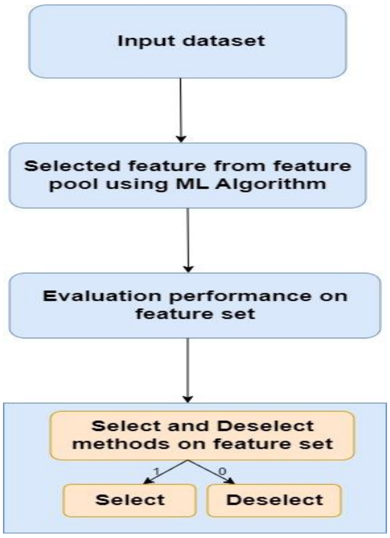


**Figure1: Existing procedure of feature selection.**

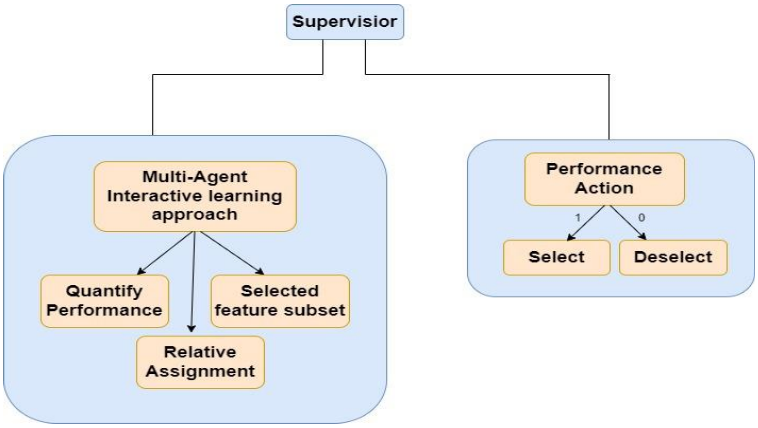


**Figure 2:Architecture of agent based interactive learning approach.**

We have developed different components to analyze select and non-select features from dataset. Descriptions of various components of figure 2 are as follows:

(i) Agent: We considered N features as N agents for our model. Specifically, each feature is intended to make the matching feature selection for a single agent.

(ii) Actions: Features are selected for the i[th] feature agent when the $Act_i = 1$, and deselected when the action $Act_i = 0$

(iii) Environment Setting: The feature subspace, which stands for a predetermined collection of features, serves as the setting as our proposed model. A feature agent's actions modify the state representation for feature subspace whenever they select or deselect a feature.

(iv) State: Description of the chosen subset of features is the purpose of state s. Using meta-descriptive statistics, we investigate three distinct approaches to extracting the representation of s. We will go deeper into these three methods of state representation.

(v) Reward: The reward is considered to select a feature set using various factors through specific parameters. This measurement is the weighted average of three factors: (i) the feature subset's predictive accuracy ($A_{cc}$), (ii) its selected feature redundancy ($R_{df}$), and (iii) its selected feature relevance ($R_{vf}$).

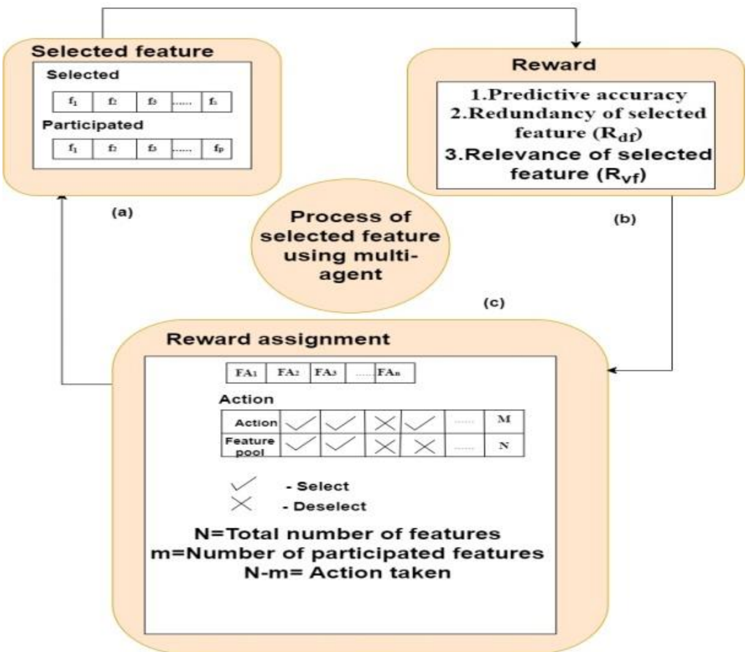


**Figure 3: Feature agent participate and it's corresponding issues.**

Method for assigning rewards: We devise a scheme to distribute the whole reward among all the feature agents. According to the plan, the agents' interactions are considered cooperative and competitive, and each one takes responsibility for a certain activity that reflects the entire compensation. All feature agents' participation is acknowledged, and they are all rewarded for their work. In Figure 3, we have considered the distribution of rewards. Here is the strategy considered: The action is defined as N-M, where N is the total number of features and M is the set of features involved. For example, the deselect characteristics are represented by N=4, M=3, and N-M=1. It is necessary to act based on the existing situation. Feature selection will be finalized based on the present circumstances. The maximum number of features will be chosen based on the M value. Participating in specific actions and using them for the present reward R are described by feature agents ($FA_i$), which include $FA_1$, $FA_2$, and $FA_m$.

## 4. Methodology for Multi-Agent Strategies

We have considered a feature subspace discovery system based on a multi-agent interactive learning model and use some methods to improve the state representation, speed up the suggested framework, and measure the reward.

### *A. Agent's State Representation Methods*

Our proposed system has a number of phases to explore feature subspaces, as shown in Figure 3. There are two parts to every exploration step: the control and training phases. From figure 3 (a & b), we considered the control phase for the selected feature subspace. During the control stage, the feature agents execute their activities according to their policy networks. Each feature agent updates their feature content using select/deselect approach. At the same time, feature agents are rewarded for whatever they do. Each agent is assigned reward from this strategy.

During training, agents work individually to teach their policy using experience replay. At time t, each agent's memory is updated with a freshly formed tuple $\{s_i^t,\ a_i^t\ r_i^t, s_i^{t+1}\}$, which includes the current state ($s_i^t$), the action($a_i^t$), the reward ($r_i^t$) and the subsequent state $s_i^{t+1}$. The $i^{th}$ agent trains its Deep Q-Network (DQN) using mini-batch data to maximize reward, according to [21].

$$Q(\theta_t) = \ r_i^t + \gamma\, max\ Q(s_i^{t+1}, a_i^{t+1}\ |\theta_i^{t+1}) \tag{1}$$

Here, θ denotes the parameters for the Q network and γ stands for the discount factor. When a number of predetermined conditions are satisfied or convergence occurs, the feature subspace is explored further.

### ***B. Measuring Reward***

We considered different action rewards R with the help of the following parameters: (i) $A_{cc}$, (ii) $R_{df}$ (iii) $R_{vf}$.

Predictive Accuracy: We must investigate and select a suitable feature subset per the predictive model. We suggest measuring the incentive using the prediction model's accuracy $A_{cc}$. In fact, a high reward for activities is considered for the specified feature subset when predictive accuracy is high and a low reward for actions that result in poor predictive accuracy.

Feature Subspace Properties: We suggest considering the features of the chosen feature subset in addition to using the prediction accuracy as a reward. Mutual information, represented by I, can quantify the information's importance and its redundancy. Specifically,

$$I(x, y) = \sum_{i,j} p(x_i, y_j)\ log\ log\ \left(\frac{p(x_i y_j)}{p(x_i)p(y_j)}\right), \tag{2}$$

The joint distribution of $x_i$ and $y_i$ is denoted by p(x, y) as $i^{th}$ and $j^{th}$ feature, and p(x) and p(y) are the marginal distributions of x and y, respectively. The sum of the pairwise mutual information among features measures the information redundancy, abbreviated as $R_{df}$. So, $R_{df}$ is determined by

$$R_{df} = \frac{1}{|S^2|}\sum_{x_i, x_j \in S} I(x_i; x_j) \tag{3}$$

Here, $i^{th}$ feature and feature subset (S) are used to determine $R_{df}$ and proceed for quantify information as $R_{vf}$. The formal expression for $R_{vf}$ is:

$$R_{vf} = \frac{1}{|S|}\sum_{x_i \in S} I(x_i; c) \tag{4}$$

where c is the label vector.

***C. Improving State Representation***

With M data samples and N characteristics, we get an M*N dataset D. For the $j^{th}$ exploration phase, let $n_j$ be the number of features that were picked. The dimension of the data matrix S that has been chosen is denoted as $M * n_j$, changes as the investigation progresses. The target networks always insist on a fixed-length vector for the state representation, s. We take the chosen data matrix S and get its descriptive statistics using quartiles, Q1, Q2, and Q3, respectively.

***D. Improving Sampling approach***

Replaying past experiences is a typical tool for improving the efficiency of training neural networks in reinforcement learning [10]. Following each action, the most up-to-date sample a tuple including the action (a), reward (r), current state (s), and next state (s') is stored in memory to replace the oldest sample. We concentrated on discovering methods to leverage high-quality data to expedite feature subspace research. Prior studies have tackled this problem by finding ways to increase the chances of getting good samples [20,21]. According to the Gaussian mixture model (GMM), our approach to this problem is as follows: [23]. In the first algorithm, we observe that a collection of memory samples {} is used as input for each agent. First, the memory samples are separated into two categories: $T_0$ and $T_1$. Next, we choose the best examples, representing the top p percent of each group, according to the selected action (Act=0 for some samples and Act=1 for others), and arrange them in T by reward (r). We use GMM algorithm [23] to proceed with our model using an Expectation Maximization (EM) approach [22]. If the agent wants to learn faster, it can use a tiny portion of the new high-quality data set. According to [23], the specifics of the GMM-based method are laid forth for the processing of batch samples.

## 5. Experimental Analysis

***A. (i) Datasets***

We have experimented with the proposed model using two real-world datasets for feature selection.
(a)Concrete Dataset [25]: we considered 1030 records and 9 features from the concrete dataset. This dataset is used to construct different buildings, roads, bridges, etc.
(b)Parkinson’s dataset [26]: we considered 193 records and 23 features from Parkinson’s dataset. This dataset is used to define whether the person is affected by Parkinson’s disease or not.

***(ii) Computational Environment***

We have used Python 3.2, Google Colab, on T4 GPUs (8 x 80 GB), 8 GB RAM for the experimental environment.

***B. Evaluation Metrics***

The following measures are considered to demonstrate the performance of the proposed model. We have considered various evaluation factors through specific parameters. Given that TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively, for all classes, the total accuracy is represented as $A_{cc} = \frac{TP+TN}{TP+TN+FP+FN}$. We employ different metrics when evaluating a classifier's performance on a test dataset. The last three metrics evaluate the label's categorization performance from various angles. We also considered other metrics such as Precision, Recall, F-1 Score which are explained as below.

$$Precision\ (P) = \frac{TP}{TP+FP} \tag{5}$$

$$Recall\ (R) = \frac{TP}{TP+FN} \tag{6}$$

$$F1 - Score = \frac{2*P*R}{P+R} \tag{7}$$

Where, P is the precision and R is the recall.

***C. Specific approach***

We evaluate the efficiency of our suggested interactive Learning Feature Selection (ILFS) approach compared to two existing ones: Choose K-Best or KNN and LASSO.

a) Choose K-Best: After sorting features according to their $x^2$ scores with the aim vector, the K-Best Selection [3] chooses the top K features. For the sake of the tests, we'll set K equal to the total number of MARLFS features chosen.
b) LASSO: To pick features and reduce the feature space, LASSO [16] uses the penalty, which eliminates feature variables with zero coefficients. A value of 1.0 is used in the experiments as the hyperparameter for LASSO's regularization weight λ.

**Table 1:** Evaluation of Feature Selection Algorithms on concrete dataset

| **Algorithms** | **Training score** | **Testing score** | **Accuracy** | **Mean score** |
|---|---|---|---|---|
| **Random Forest** | 0.87 | 0.98 | 0.87 | 32.34 |
| **Gradient Boost** | 0.94 | 0.88 | 0.88 | 31.32 |
| **Ada boost** | 0.82 | 0.76 | 0.88 | 31.32 |
| **KNN** | 0.76 | 0.9 | 0.75 | 64.43 |
| **Bagging** | 0.97 | 0.866 | 0.866 | 34.89 |
| **SVM** | 0.72 | 0.65 | 0.65 | 90.42 |
| **XGBoost** | 0.992 | 0.88 | 0.88 | 30.19 |
| **Decision Tree** | 0.98 | 0.88 | 0.9 | 27.06 |

***D. Overall Performance***

We considered different algorithms for evaluating test and training data sets from two core data sets. So, we got the test, training, and mean scores, as shown in Table 1. Using two datasets, we also compared our model with different algorithms such as Random Forest, GradientBoost, Adaboost, KNN, Bagging, Support vector Machine, XGboost, Decision Tree.

***E. Robustness Check***

To get accurate predictions, you need both predictors and feature selection. We test it on many predictors to determine if the feature subset we examined is stable and if our strategy can consistently outperform other baseline approaches. This allows us to see how our strategies perform when faced with challenges. This experiment uses many predictors, such as XGBoost, decision trees, support vector machines, and the random forest (RF) predictor. Tables 1 and 2 indicate that our ILFS outperforms the suggested techniques for all predictors. However, our technique beats such baselines when feature selection is done using LASSO, and prediction is conducted using other classification models.

### *F. Use of Reward Function*

We considered the reward function's design following our concept. Specifically, we examine different cases: (i) The reward function may be expressed as Acc when measuring accuracy alone, Rv and Rd when measuring relevance and redundancy alone, and $A_{cc}+R_{vf}+R_{df}$ when measuring all three.
Since it directs the inquiry towards improving accuracy, Acc is the second-best reward function. Rvf and Rdf are both inadequate. This is because, as an unsupervised indicator of incentives, none directly correlate with the accuracy of predictions. Accumulated supervised and unsupervised indicators $(A_{cc}+R_v+R_d)$ provide the best outcomes.

### *G. Study of State Learning from Representations*

We have adopted a state-learning strategy to achieve state-level feature performance under our model. Two examples are considered: (i)The state-level feature analysis is approached using the meta-descriptive statistics (MDS) method, and (ii)the state-level encoding is accomplished using the auto-encoder (AE). The most effective result for feature selection was achieved by combining the two methods mentioned above.

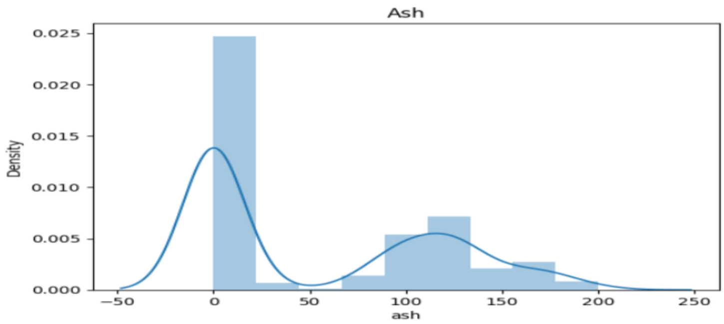


**Figure 4: Evaluation result as per quartile for ASH**

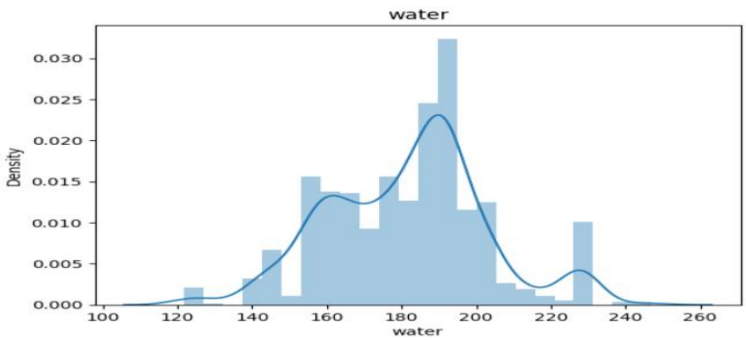


**Figure 5: Evaluation result as per quartile for water**

In this part, we considered Q1 and Q3 quantile for Cement dataset with Q1= = 192.375, Q3 = 350.00 and inter quantile (IQR) = 157.625. So, as per our experiments, we got lower outlier limit in cement: -44.0625 and an Upper outlier limit in cement: 586.4375. Similarly, we got quantile water results as Q1 = 164.9, Q3 = 192.0 and IQR = 27.0999. In this case, the Lower outlier limit in cement 124.2500, and Upper outlier limit in cement 232.6499. We considered the quartile values for all features for cement. We mentioned only two as figure 4 and 5.

### *H. Interactive learning model*

Tables 1 and 2 detail the two datasets considered for the proposed model, each using a different algorithm. Comparing the algorithms' performance on the two datasets reveals that the Parkinson's dataset outperformed the concrete dataset in terms of training score, testing score, and accuracy (figures 6-8). However, the results are inverted in figure 9, showing that the concrete dataset outperforms the Parkinson's dataset when using the identical techniques. As seen in figure (6-12), we examined the different comparative performances.

**Table 2: Evaluation of Feature Selection Algorithms on Parkinson's dataset.**

| Algorithms | Training score | Testing score | Accuracy | Mean score |
|---|---|---|---|---|
| **Random Forest** | 1 | 0.93 | 0.91 | 0.6 |
| **Gradient Boost** | 0.99 | 0.58 | 0.91 | 0.67 |
| **Ada boost** | 1 | 0.93 | 0.93 | 0.67 |
| **KNN** | 0.90 | 0.86 | 0.84 | 0.16 |
| **Bagging** | 0.99 | 0.88 | 0.88 | 0.15 |
| **SVM** | 0.80 | 0.83 | 0.83 | 0.16 |
| **XGBoost** | 0.99 | 0.32 | 0.86 | 0.13 |
| **Decision Tree** | 1 | 0.86 | 0.86 | 0.13 |

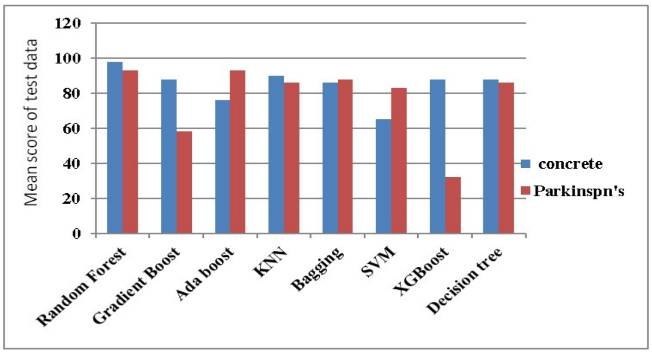


**Figure 6: Comparative results of ML methods for Testing data from dataset (Concrete and Parkinson's Disease dataset)**

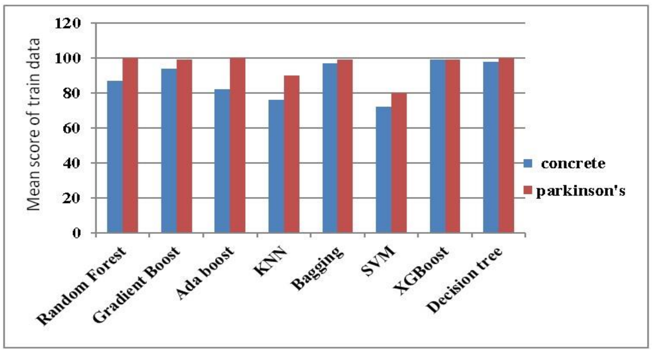


**Figure 7: Comparative results of ML methods for Training data on dataset (Concrete and Parkinson's Disease dataset)**

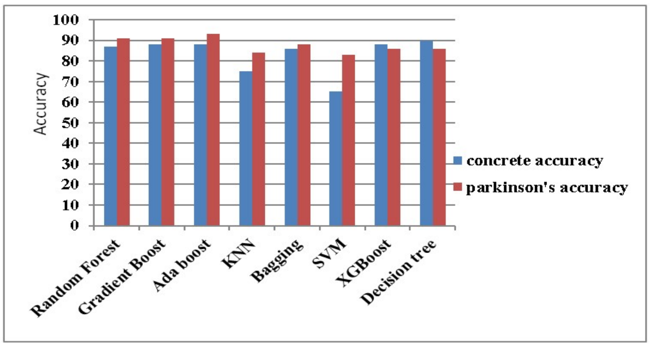


**Figure 8 Comparative results of ML methods for accuracy data on dataset (Concrete and Parkinson's Disease dataset)**

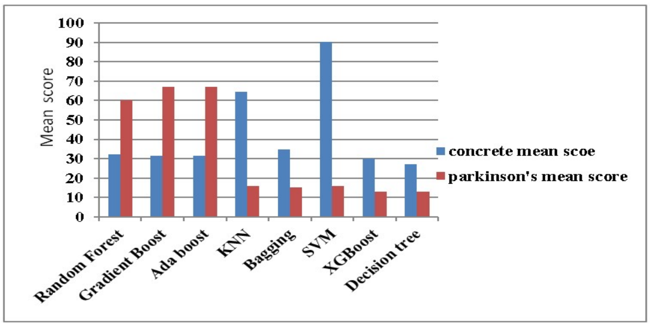


**Figure 9: Comparative results of ML methods for mean score data on dataset (Concrete and Parkinson's Disease dataset)**

Figure 10 shows the results for a particular dataset, showing that XGboost outperformed the other algorithms in accuracy. Figure 11 also shows that the Parkinson's dataset yields the best results when using Adaboost. Figure 12 also shows that many algorithms run the assessment metrics on the Parkinson's dataset, with the best performance being the F1-score.

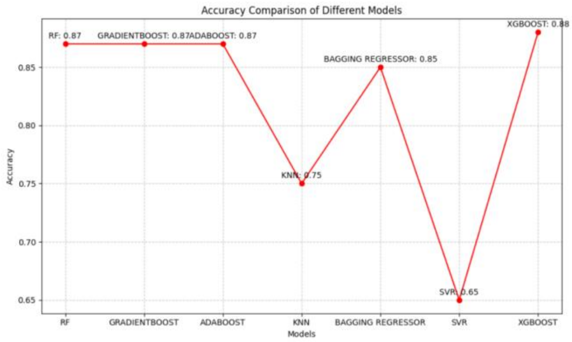


**Figure 10: Comparative accuracy among ML algorithms on Concrete dataset**

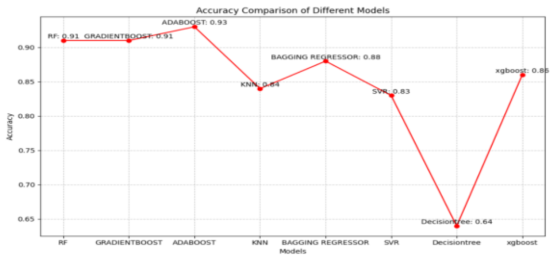


**Figure 11: Comparative accuracy among ML algorithms on Parkinson's dataset**

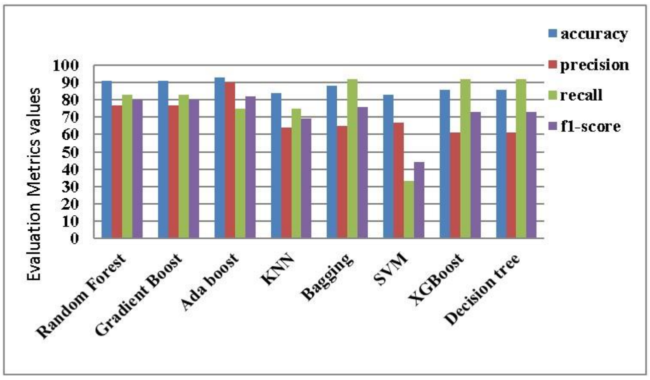


**Figure 12: Comparative performance on evaluation metrics using different algorithms on Parkinson's dataset**

We ran several tests and extracted features from various datasets based on the results shown above. Based on our different evaluation performances, we have selected 8 features from the concrete dataset and 10 features from the Parkinson's dataset.

## 6. Conclusions

Feature selection strategies for reducing dimensionality and improving accuracy according to our suggested model are examined in this work. Therefore, We have contemplated an interactive learning model that uses reward and reward assignments to address feature redundancy and relevance. Each feature acts as an agent for the model in our suggested model, coordinating their actions in response to the model's commands. We created an interactive learning model involving many agents to facilitate feature selection. To avoid overfitting, we exclude features from our model that do not get any data. We compared the performance of numerous strategies that assisted in feature selection from the dataset and attempted to use them to select features. Finally, we demonstrated that the proposed approach is practical by extensively testing two real-world datasets. Many areas of data mining and Machine learning models can benefit from our model. One of our long-term goals is to build an AI model that uses multi-action-based methods for feature selection for future work.

## Declaration Statements

a) Competing Interests: We have no conflict of interests.
b) Funding Information: We have no research funding.
c) Author contribution: Mr. Md Sirajul Huque has developed a architecture and implementation and Dr. V. Kiran Kumar create documentation for paper.
d) Data Availability Statement: As per requirement, we will provide.
e) Research Involving Human and /or Animals: We are human only for research.
f) Informed Consent: Since, we have developed our model, no need to take consent from others.

## References

[1] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” Comput. Elect. Eng., vol. 40, no. 1, pp. 16–28, 2014.
[2] Bhuyan H. K., Vinay Kumar Ravi, Analysis of Sub-feature for Classification in Data Mining, IEEE Transaction on Engineering Management, Volume: 70, Issue: 8, Page(s): 2732-2746, 2023.
[3] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” J. Mach. Learn. Res., vol. 3, no. Mar.,pp. 1289–1305, 2003.
[4] K. Lin, R. Zhao, Z. Xu, and J. Zhou, “Efficient large-scale fleet management via multi-agent deep reinforcement learning,” 2018,arXiv:1802.06444.
[5] S. M. H. Fard, A. Hamzeh, and S. Hashemi, “Using reinforcement learning to find an optimal set of features,” Comput. Math. Appl.,vol. 66, no. 10, pp. 1892–1904, 2013.
[6] Bhuyan H. K., Narendra Kumar Kamila, Privacy preserving sub-feature selection in distributed data mining, Applied soft computing, Vol- 36, PP. 552-569, 2015.
[7] J. Yang and V. Honavar, “Feature subset selection using a genetic algorithm,” in Feature Extraction, Construction and Selection. Berlin, Germany: Springer, 1998, pp. 117–136.
[8] Bhuyan H. K., Narendra Kumar Kamila, Privacy preserving sub-feature selection based on fuzzy probabilities, Cluster computing, Vol-17, Issue-4, PP. 1383-1399, 2014.
[9] HK Bhuyan, V Ravi, MS Yadav, Multi-objective optimization-based privacy in data mining, Cluster computing 25 (6), 4275-4287, 2022.
[10] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, 2015, Art. no. 529
[11] F. Cruz, S. Magg, Y. Nagai, and S. Wermter, “Improving interactive reinforcement learning: What makes a good teacher?” Connection Sci., vol. 30, no. 3, pp. 306–325, 2018.

[12] Hemanta Kumar Bhuyan, CV Madhusudan Reddy, Sub-feature selection for novel classification, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE, Pages477-482, 2018.
[13] D. Guo, H. Xiong, V. Atluri, and N. Adam, “Semantic feature selection for object discovery in high-resolution remote sensing imagery,” in Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining, 2007, pp. 71–83.
[14] Bhuyan H. K., M Saikiran, Murchhana Tripathy, Vinayakumar Ravi, Wide-ranging approach-based feature selection for classification, Multimedia Tools and Applications, pages: Volume 82, Issue 15, pp 23277–23304, Jun 2023.
[15] R. Kohavi and G. H. John, “Wrappers for feature subset selection,”Artif. Intell., vol. 97, no. 1/2, pp. 273–324, 1997.
[16] R. Tibshirani, “Regression shrinkage and selection via the lasso,”J. Roy. Stat. Soc. Ser. Methodol., vol. 58, pp. 267–288, 1996.
[17] V. Sugumaran, V. Muralidharan, and K. Ramachandran, “Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing,”Mech. Syst. Signal Process., vol. 21, no. 2, pp. 930–942, 2007.
[18] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.
[19] Bhuyan H. K., Vinayakumar Ravi, M. Srikanth Yadav, Multi-objective optimization-based privacy in data mining, Cluster computing (Springer), Vol- 25, issue-6, pages 4275–4287 2022.
[20] Bhuyan H. K., Vinay Kumar Ravi, An Integrated Framework with Deep learning for Segmentation and Classification of Cancer Disease, Int J. on Artificial Intelligence Tools (IJAIT), Vol. 32, No. 02, 2340002, 2023.
[21] H. Wei, G. Zheng, H. Yao, and Z. Li, “IntelliLight: A reinforcement learning approach for intelligent traffic light control,” inProc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining,2018, pp. 2496–2505.
[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Roy. Stat.Soc. Ser. Methodol., vol. 39, pp. 1–22, 1977.
[23] Kunpeng Liu, Yanjie Fu, Le Wu, Xiaolin Li, Charu Aggarwal, and Hui Xiong, Automated Feature Selection: A Reinforcement Learning Perspective, IEEE Transactions on Knowledge and Data Engineering, VOL. 35, NO. 3, March 2023.
[24] S Dhawan, HK Bhuyan, SK Pani, V Ravi, R Gupta, A Rana, A Al Mazroa, Secure and resilient improved image steganography using hybrid fuzzy neural network with fuzzy logic, Journal of Safety Science and Resilience 5, 91-101, 2024.
[25] Kumar Singh, A.; Nigam, M.; Srivastava, R.K. Study of stress profile in cement concrete road of expansive soil due to swell pressure. Mater. Today Proc. Vol. 56, Page: 347–355, 2022.
[26] Jinil Persis Devarajan , V. Raja Sreedharan , and Gopalakrishnan Narayana Murthy, Decision Making in Health Care Diagnosis: Evidence From Parkinson’s Disease Via Hybrid Machine Learning, IEEE Transactions On Engineering Management, Vol. 70, No. 8, August 2023.