https://doi.org/10.31436/iiumej.v26i1.3337

DESIGN OF INTELLIGENT FEATURE SELECTION TECHNIQUE FOR PHISHING DETECTION

SHARVARI SAGAR PATIL*, NARENDRA M. SHEKOKAR, SRIDHAR CHANDRAMOHAN IYER

Department of Computer Engineering,
SVKM's Dwarkadas J. Sanghvi College of Engineering, Mumbai, India.

**Corresponding author: sharvarichorghe@gmail.com*

(Received: 23 June 2024; Accepted: 21 September 2024; Published online: 10 January 2025)

ABSTRACT: Phishing attacks lead to significant threats to individuals and organizations by gaining unauthorized access. The attackers redirect the users to fake websites and steal their credentials and other confidential data. Various techniques are employed to detect phishing using machine learning algorithms or static detection techniques that use blacklisting of web URLs. The attackers tend to change their approach to launch an attack, making it difficult for traditional phishing detection techniques to safeguard the user. The performance of conventional detection methods relies on exhaustive data and features selected for classification. Features selected for designing detection systems majorly contribute to the performance of the detection system. Phishing detection techniques rely mainly on static features that are selected based on traditional feature selection or ranking techniques. This paper proposes an innovative approach to phishing detection by designing a feature selection technique using reinforcement learning. A novel reinforcement learning agent is designed that uses a dynamic, adaptive, and data-driven approach to improve classifier performance in phishing detection. The technique is designed to select the features using the RL agent dynamically. We have evaluated our technique using the real-world phishing dataset and compared its performance with the existing techniques. Based on the evaluation, our proposed methodology of dynamic feature selection gives the best accuracy of 99.07 % with the random forest classifier model. Our work contributes to advancing phishing detection methodology by developing a dynamic feature selection technique.

ABSTRAK: Serangan pancing data membawa ancaman besar kepada individu dan organisasi dengan mendapatkan akses tanpa kebenaran. Penyerang akan mengalihkan pengguna ke laman web palsu dan mencuri maklumat log masuk serta data sulit yang lain. Pelbagai teknik digunakan bagi mengesan pancing data menggunakan algoritma pembelajaran mesin atau teknik pengesanan statik yang menggunakan URL laman web yang disenarai hitam. Penyerang cenderung mengubah pendekatan mereka untuk melancarkan serangan, menjadikan teknik pengesanan pancing data tradisional sukar bagi melindungi pengguna. Prestasi kaedah pengesanan konvensional bergantung kepada data menyeluruh dan ciri-ciri yang dipilih untuk pengelasan. Teknik pengesanan pancing data kebanyakannya bergantung pada ciri-ciri statik yang dipilih berdasarkan kaedah pemilihan atau penarafan ciri tradisional. Kajian ini mencadangkan pendekatan inovatif bagi pengesanan pancing data dengan mereka bentuk teknik pemilihan ciri menggunakan pembelajaran peneguhan. Ejen pembelajaran peneguhan baru, direka menggunakan pendekatan yang dinamik, adaptif, dan berasaskan data bagi memperbaiki prestasi pengelas dalam pengesanan pancing data. Teknik ini direka untuk memilih ciri-ciri secara dinamik menggunakan ejen RL. Teknik ini dinilai menggunakan dataset pancing data sebenar dan dibanding prestasinya dengan teknik sedia ada. Berdasarkan penilaian, metodologi pemilihan ciri dinamik ini memberikan ketepatan terbaik sebanyak

99.07% dengan model pengelasan rawak. Kerja ini merupakan sumbangan kepada kemajuan metodologi pengesanan pancing data dengan membangunkan teknik pemilihan ciri dinamik.

KEYWORDS: *Reinforcement Learning; Feature Selection; Phishing Detection; Machine Learning*

1. INTRODUCTION

Phishing is an attack where the attacker pretends to be a trusted entity to gain access to confidential information from internet users. The attackers generate genuine-looking URLs and send them to the victim through various channels like emails, messages, and other communication mediums. Based on The State of Phishing 2024 report, there is a 341% increase in malicious phishing links [1]. The launch of Chat GPT has also led to the rise of phishing attacks. The reports suggest that since the launch of ChatGPT in November 2022, there has been a surge of 4151% in phishing messages. In the Internet crime report published by the Federal Bureau of Investigation, phishing is among the top 5 cybercrimes reported in 2023 [2]. Multiple detection strategies have been designed that rely on static blacklisted datasets and other machine learning-based techniques. These techniques depend on the dataset and the features used while developing the detection model.

The technology for detecting phishing web pages has evolved continuously over the period. Various conventional methods for phishing detection include the blacklist method, heuristic feature method, visual similarity features based, and machine learning approaches [3]. Researchers have also employed advanced detection methods like deep learning. Traditional methods mainly depend on analyzing and modeling manually extracted features from various sources, including URL features, page content, and webpage structure. Recently, deep learning has been investigated and utilized for phishing webpage detection because of its strong ability to extract features automatically. Bio-inspired optimization algorithms like Particle Swarm Optimization have been used for feature selection [4]. Feature selection using ranking methods like omitting redundant features and filtering methods have been proposed by researchers on the phishing dataset [5]. However, as phishing webpages become more evasive and rapidly updated, these traditional methods require increasingly detailed analyses and the extraction of more features. This results in many feature dimensions and possible correlations between features.

To address the aforementioned issue, the proposed Intelligent Feature Selection Technique (IFST) utilizes artificial intelligence and machine learning to identify and prioritize features most indicative of phishing behavior dynamically. IFST aims to provide a robust defense against known and emerging phishing threats by intelligently adapting to evolving attack strategies and patterns.

Most relevant and informative features are selected using feature selection techniques. The features contribute to the accurate classification of the test cases. These techniques aim to improve model performance and decrease computational complexity. Various methods are used to select beneficial features, such as filtering, wrapping, and embedding techniques [4]. The significance of features is calculated using formulas like correlation between the variables, information gain calculated for each feature, and chi-square metrics. In developing embedded systems, feature selection is not done in a separate stage. It is an integral part of model building. Other advanced techniques the researchers use are genetic algorithms, particle swarm optimization, and recursive feature elimination. Despite the advantages of feature selection techniques, filter methods are formula-based, resulting in data loss. Wrapper and embedded methods might result in the computational complexity of the resulting model. Apart from this,

https://doi.org/10.31436/iiumej.v26i1.3337

these techniques generate dataset-dependent results that might not work optimally in real-world applications.

This paper introduces a novel idea of using reinforcement learning (RL) to select feature subsets. Reinforcement learning is an agent and environment-based approach. An agent learns to interact with the environment based on the action it takes and the rewards received based on the policy. One of the benefits of using RL is that balance can be achieved between exploration and exploitation strategy by adjusting the value of parameter epsilon. The idea of an agent taking action is considered a random selection of feature subsets. The agent gets a reward based on the accuracy of the classifier for a given feature subset. At the end of the episode, the reward is calculated for each feature subset selected randomly by the agent. In summary, we have developed a dynamic subset selection reinforcement learning agent that selects the subset of features and learns based on the rewards it gets after each episode. The objectives of this research are as follows:

- To evaluate the performance of the traditional feature selection techniques on the phishing dataset and compare their performance with the proposed methodology.
- To develop a novel reinforcement learning agent that dynamically selects the subset of features.
- To analyze the performance of the classification models with correlated features and the change in performance after eliminating the correlated features.

2. RELATED WORKS

Feature selection is a significant data preprocessing technique that aids in developing effective and efficient classification models. Various researchers have previously proposed different feature selection and ranking techniques. This section discusses the various methodologies proposed by the researchers in feature selection techniques for phishing detection.

In [7], the researchers introduce a sophisticated hybrid feature selection mechanism methodology using boosting algorithms to detect phishing websites. A multilayer learning algorithm based on Boosting was developed using hybrid feature selection technology to select relevant features. Extreme Gradient Boosting method (XGB), Classification Boosting (CatB), Light Gradient Boosting Machine method (LGBM), and average values are calculated to select the final feature subset. Datasets generated after excluding the irrelevant features are given as input to the classifiers at different levels. There are four models in the first layer: XGBoost, LGBM, CatBoost, and AdaBoost. The next layer includes three models: XGBoost, CatBoost, and AdaBoost. Finally, XGBoost implements the last meta-learner method. Phishing detection is done by taking input from the lower layer. The test results show that the model's accuracy increases from 96.16% to 98.95% without special options. Accuracy with custom selection is between 96.18% and 98.80%.

In [8], the authors highlight the challenges associated with phishing website classification. The major challenges the researchers have focused on are the evolving tendency of phishing attacks and the huge number of features used for classification. Filter and wrapper methods are evaluated based on the performance of a specific classifier. Two approaches are used in this research. The first approach is based on removing redundant features. In the second approach, the author selects features using the filter method. The dataset used in this research is from the University of California. The two approaches were evaluated by performing classification using algorithms like Random Forest, Multilayer Perceptron, and Naïve Bayes.

In [9], the author uses the Uni-variate feature selection (UFS) technique. Using this technique, features were selected and given as input to an ensemble learning classifier. This classifier was developed by combining Cat Boost, Gradient Boost, and Random Forest. It was observed that the accuracy was improved using this methodology. The accuracy without feature selection ranges from 96.16% to 98.95%, and with relevant features ranges from 96.18% to 98.80%. The study highlights the significance of ensemble learning and Uni-variate feature selection in optimizing the performance of phishing detection schemes. The authors suggest that the Uni-variate feature selection methodology is used to improve the response time. The research suggests that the features were selected based on their statistical significance level.

In [10], the researchers have proposed an explainable machine learning model. This model is responsible for the prediction of phishing websites and also explains. After the classification is done, this model explains the results. A novel multidimensional extension of the Gini Coefficient, Lorenz Zonoids, is proposed in this literature for feature selection. Initially, the features were selected based on a statistical formula-based method. The contribution of features in detection was also considered based on explainable model results. Validation of features was done in two stages: Exploratory analysis followed by Gini Index calculation.

The major motive of the research in [11] is to reduce the data dimensionality. The authors have proposed a technique that is a combination of filter methods. In the next stage, wrapper methods are used to build a model to detect websites. The results have shown improvement in the accuracy of the model. This methodology of selecting features from fewer features has improved computational time. The filter methods used in this research are Heatmap Correlation, ANOVA test, and Chi-square test. The top 12 features using all three formulas were considered. Researchers have used this approach in the view that the values obtained are independent of ML classifiers. In the last step, the researchers combined the three subsets obtained and applied them to the heuristic-based wrapper method using Logistic Regression, Random Forest, and Naïve Bayes models.

In [12], the authors have proposed a technique that selects features based on majority voting and consensus. The author proposes using random forest, gradient boosting, and LGBM for feature selection. These three techniques vote for each feature in the dataset, and based on the majority voting, the feature is either selected or rejected. These selected features are given as input to the detection model. The detection model is developed by combining Adaboost and LightGBM. It was observed that there was a drop in the detection time of the URLs.

In [13], the author analyzes the traditional feature selection algorithms and performance evaluation using ML algorithms. The performance of filter, wrapper, and embedded methods is compared. This survey paper has done a detailed analysis of how feature selection and dimensionality reduction algorithms affect the performance of the detection mechanisms. The metrics used for calculating the performance were Precision, Recall, K-fold Cross-validation, AUC-ROC, and execution time.

The research in [14] has focused on improvement of the recall values and reducing the number of false negatives. The authors have proposed a technique that classifies the URLs by combining deep learning and a genetic algorithm for searching the best feature subset. The deep learning algorithm used in this approach is a convolutional recurrent network. It was observed that combining a convolutional recurrent network with a genetic algorithm for feature selection improved the accuracy and recall. The results using a convolutional recurrent network without feature selection were used for performance comparison.

In [15], the authors proposed a technique for selecting relevant features that combined the correlation and recursive feature elimination techniques to identify features based on URL

characteristics. The authors combine recursive feature elimination with correlation, information coefficient correlation, and Spearman correlation coefficient. In this study, two datasets with 48 and 87 features were used. It was observed that it works effectively with small feature subsets. These techniques were evaluated based on the performance of Decision Tree, Random Forest, Support Vector Machine, and AdaBoost algorithms.

In [16], researchers proposed a model that recognizes relevant features using recursive feature elimination. The author has proposed a system consisting of five modules. In the first module, the author focuses on pre-processing the dataset, wherein missing and inconsistent data points are removed. The second module is related to finding the relation between the features using correlation and principal component analysis. The next stage involves automatically selecting several features using the Extra Tree Classifier. Different ensemble algorithms are compared to generate the best set of features.

In [17], authors have considered the use of psychological manipulation to trap users into phishing to design a multi-stage detection model that effectively detects fake websites in the real world. This research work covers the features of "Counterfeiting," "Affiliation," "Stealing," and "Evaluation." The technique focuses more on fast filtering and accurate detection. The legitimate websites are removed during the filtering stage, followed by a supervised detection model. The multistage model consists of several stages, each focusing on different aspects of phishing website detection. Initially, the model extracts features from web pages using the CASE framework, capturing diverse characteristics indicative of phishing behavior. Subsequently, these features are processed through multiple classifiers, each specializing in different aspects of phishing detection. Initially, whitelist-based filtering is done to remove the top-ranked websites. This filtering is based on the traffic of the DNS recursive server. In the second stage, the researchers used only a section of falsifying information on the web page, known as fast counterfeit filtering. In the final stage, classification is done based on the multiscale CASE features. Various algorithms are used in the experiments, including AdaBoost, sequential minimal optimization (SMO), and random forest.

In [18], the researchers used the optimization algorithm to extract important features. The authors claim that using Particle Swarm Optimization (PSO) for relevant feature selection improves the performance of the phishing recognition model. The feature weights are adjusted iteratively in PSO based on their contribution to identifying the phishing webpage as genuine. The research asserts that machine learning models are improved by employing a feature weighting technique based on Particle Swarm Optimization (PSO). The irrelevant features were removed based on their weights, ranging from 7 to 57%. The remaining features were used to perform classification operations. The study evaluated the proposed technique by comparing it with the performances of the Backpropagation neural network, SVM, k-nearest neighbor, random forest, naïve Bayes classifier, and decision tree. In Table 1, we summarize the methodology used, the features selected, and the performance of the methodologies discussed in the above literature.

The survey suggests that different approaches have been utilized to optimize the performance of the phishing detection methods. The approaches focus on combining traditional feature selection techniques or using metaheuristic optimization algorithms. The literature suggests that the feature selection improves the model accuracy. The research also highlights the importance of combining machine learning models with effective feature selection and optimization strategies to develop highly accurate, reliable, and interpretable phishing detection systems. The methods used by the researchers, like boosting-based hybrid selection mechanisms, filter and wrapper methods, and univariate feature selection, identify the important features from the dataset for correct classification. Dynamically evolving attack

https://doi.org/10.31436/iiumej.v26i1.3337

strategies may bypass the detection model trained using features selected based on a fixed strategy and classification algorithm.

Table 1. A Comparative Study of Research Papers: Methodology, Features, and Performance

Paper Reference	Methodology	Total Features Selected	Performance (Accuracy)	Research Gaps
L.R. Kalabarige [7]	Feature importance is calculated using XGB, CatB, and LGBM, and an average of feature importance is computed to select the final feature subset.	33	98.80	Further investigation into hybrid and evolutionary-based feature selection methods could lead to more optimized feature subsets.
Shabudin et al. [8]	Feature Selection by Omitting Redundant Features(FSOR)and Feature Selection by Filtering Method (FSFM)	FSOR: 22 FSFM: 9	FSOR: RF 97.1, MLP 96.5 FSFM: RF 95.3, MLP 95.0	Explore optimizing these methods or developing new techniques to enhance performance further.
K Adane et al. [9]	Uni-variate feature selection (UFS) technique on each ensemble learning classifier	DS-1: 69 of 87 DS-2: 27 of 31 DS-3: All 48	DS-1: 97.24 DS-2: 96.83 DS-3: 98.51	Hybrid approaches could further improve model accuracy and reduce computational time.
Calzarossa et al. [10]	Lorenz Zonoids, the multi-dimensional extension of Gini coefficient	6	AUC: 0.96	Testing is based on multiple classification models other than Random Forest.
Abulfaz Hajizada et al. [11]	Combination of multiple filter and wrapper methods for feature selections	12	Logistic Regression Random Forest Naïve Bayes	Investigation of the reasoning behind the choice of feature selection methods.
B Alotaibi et al. [12]	Voting-based Feature Selection.	23	98.63%	Investigation of other feature selection techniques or combinations to reduce detection time and improve accuracy.
Amit Singh et al. [13]	Filter, wrapper, and embedded feature selection method	23	Random forest gives the best accuracy of 98.067% and a precision of 0.982.	Explore additional feature selection and dimensionality reduction methods to enhance phishing detection systems.
Moedjahedy, J et al. [15]	Combination correlation and recursive feature elimination	10	Dataset 1: 97.06 Dataset 2: 95.88	Exercising the new evidence by providing the latest dataset.
Goud et al. [16]	Recursive feature elimination	29	93%	Other Feature selection techniques can be implemented.
Liu, D. J. et al [17]	CASE feature framework	"Counterfeiting," "Affiliation," "Stealing," and "Evaluation" features.	Recall /TPR: 0.8923 FPR: 0.0005 Precision: 0.9886 F1-Measure: 0.9380	Comprehensive utilization of multi-scale features.

WALEED ALI et. al[18]	Particle swarm optimization-based feature weighting	PSO-based feature weighting omitted between 7% and 57% of irrelevant features.	BPNN (95.88%), kNN (94.79%), RF (94.3%), and C4.5(94.033%).	An improved version of PSO can enhance performance and reduce the time required for feature evaluation and weighting.

Based on the studies, it was observed that feature selection impacts model accuracy. The researchers have used diverse methods like boosting-based hybrid selection mechanisms, filter and wrapper methods, and univariate feature selection to identify relevant features for accurate classification. The literature also showcases the use of ensemble learning approaches for phishing detection. Multi-layer stacked ensemble models, combining classifiers like XG-Boost, Cat-Boost, and Ada-Boost, are utilized to leverage the strengths of classifiers and improve overall performance. These ensemble models demonstrate high accuracy rates, ranging from 96.16% to 98.95% across different datasets.

It was observed that the models are trained on historical data that may not work with evolving phishing strategies. Studies have focused on specific feature selection techniques and machine learning models that might not adapt to constantly changing attack strategies. Since the technique is specific to a particular feature selection strategy and an ML model, it is easy for the attacker to bypass the security mechanism. To address these challenges, it is crucial to develop an effective feature selection method to dynamically select the features from the dataset and perform the classification.

3. PROPOSED METHODOLOGY

Previous research has applied different feature selection methods to improve the classification algorithm's effectiveness. These methods range from boosting-based hybrid mechanisms to filter and wrapper methods, aiming to select relevant features efficiently. Studies also explore the use of explainable machine learning models, where features are selected based on their contribution to the classification of fake web pages. Novel feature selection models have been proposed, highlighting the significant improvement in the performance of phishing detection systems.

The proposed system is based on reinforcement learning to select the feature subset. The idea behind reinforcement learning is the dynamic selection of feature subsets and learning based on the outcome and rewards. Reinforcement Learning is used for feature selection with the motive of automating the selection of features and developing a model that can decide the best feature subset dynamically. In addition, reinforcement learning is adaptive, and it continuously learns from the rewards and penalties it receives from the environment. Thus, reinforcement learning can systematically and effectively identify the most relevant features, improving model performance and efficiency.

To achieve this objective, a custom reinforcement learning environment was created. Fig. 1 represents the proposed system for feature selection using reinforcement learning. In the first stage of the proposed system, the data is balanced using hybrid data balancing algorithms, such as the Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) and one-sided selection (OSS). Data balancing is an essential data preprocessing stage to get an unbiased model. ADASYN generates synthetic data points of the minority class based on the density distribution of the data points. OSS focuses on removing noisy and borderline data

https://doi.org/10.31436/iiumej.v26i1.3337

samples. The balanced dataset is pre-processed to remove the missing values and duplicated data.

In the next phase of the proposed system, the dataset is given as input to the RL environment for training the agent. RL environment consists of the agent, action space, observation space, and classifier training component. Initializing the environment involves loading the dataset. In the next stage, action space is defined using Gym's discrete space where each action corresponds to selecting a feature from the dataset. The number of features available in the dataset determines the size of the action space. Gym's Discrete space represents a discrete set of integers. It is used to define action spaces in reinforcement learning environments. In the context of a phishing detection environment, the discrete space represents the set of possible actions an agent can take, where each action corresponds to selecting a feature from the dataset. For example, if our dataset contains 111 features, the Discrete action space has 111 possible actions, each representing the selection of a different feature. The details of the RL Environment are as follows:

- Agent: The "agent" is a decision-making entity that interacts with the environment to learn an optimal strategy for selecting features that maximize the performance of a phishing detection model. The agent's role is crucial in reinforcement learning, where it learns through interaction with the environment to maximize some notion of cumulative reward. The agent interacts with the environment based on its current policy, which could be exploratory or exploitative. In each interaction, the agent observes the current state of the environment. After observation, it selects an action by choosing a feature. It receives feedback from the environment as a reward based on the model's accuracy when using the selected feature. Later, it transitions to a new state, the next feature subset. The agent is a function program that generates subsets with a selected feature. The agent aims to figure out a policy that maximizes total rewards. ε-greedy policy is used for this purpose. The agent uses Q-learning to develop a policy that chooses the best characteristics to optimize accuracy.
- Q-Learning Algorithm- The agent first determines learning parameters, including the exploration rate (ε), learning rate (α), discount factor (γ), and epsilon decay parameters, then sets up a Q-table to hold probable rewards for each state-action combination. The agent employs an ε-greedy policy to balance random action selection for exploration and best-known action selection for exploitation. This tactic uses the agent's current knowledge to help it discover new, possibly lucrative activities. Every episode starts with the environment being reset to its initial state. From then on, the agent keeps choosing actions based on the current state, gets rewarded, and watches the next state until the episode ends. Following the Q-learning update rule, the agent adjusts its Q-values after every action based on the reward received and the anticipated future rewards.

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma\, maxQ(s',a') - Q(s,a)] \quad (1)$$

where s is the current state, a is the action taken, r is the reward received, s' is the next state, a' is the action that would be taken in the next state, α is the learning rate, controlling how much the new information overrides the old, γ is the discount factor, which balances immediate and future rewards. This update helps the agent to gradually improve its estimates of the value of actions in different states, leading to a better policy over time. Over multiple episodes, the agent's policy improves as it accumulates more information about which actions lead to higher rewards. The gradual reduction of epsilon over time ensures that the agent transitions from exploration to exploitation, utilizing its learned policy to maximize rewards. The agent undergoes training for 100 episodes.

The dataset has 111 features; at each step, the agent selects a feature from it and then combines it with 9 random features. The agent performs this action. This subset of size 10 is then used to train the classifier. The Q-learning algorithm is used to help an agent select the most important features for classification by learning from the outcomes of its previous actions. The agent aims to find feature subsets that maximize the classification accuracy. The Q-table of 111x111 is generated where rows represent different states. The agent's current state is the feature that has been selected so far, and columns represent the possible actions. Actions included which feature to select next. The agent can take 111 actions to choose any of the 111 features. The agent starts with a Q-table initialized to zeros, meaning it does not know which useful features. The working of the Q-learning algorithm for an episode is as follows:

Episode 1: The agent is in the initial state. It picks an action randomly due to high epsilon. Assume it selects feature f3. Now, the agent trains a Classifier using f3 and 9 other random features (for a subset of 10 features). Assume the randomly selected subset is [f3, f10, f22, f45, f67, f81, f89, f97, f102, f110]. The classifier is trained on this subset and tested. Let's say the accuracy is 75%. The accuracy (0.75) is the reward for selecting f_3. The agent updates the Q-value for selecting f_3 based on this reward. The Q-table is updated as follows:

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma\, maxQ(s',a') - Q(s,a)] \quad (2)$$

$$Q(s,f_3) = 0 + 0.1 \times (0.75 + 0.99 \times 0 - 0) = 0.075 \quad (3)$$

Episode 2: In this, another feature is selected randomly. The current state is after selecting f3. Let us assume feature 45 is selected at random. The selected subset is [f45, f3, f22, f67, f81, f89, f97, f102, f110, f10]. Assume the accuracy is 80%, so the reward is 0.80. The agent moves to the next feature selection step s'.The q-table value is updated for feature 45.

$$Q(s,f_{45}) = 0 + 0.1 \times (0.80 + 0.99 \times 0 - 0) = 0.08 \quad (4)$$

As episodes continue, more Q-values are updated. The agent selects features based on experience. The Q-values will be further refined based on both the immediate and future rewards that are discounted by gamma.

- Action Space: Agents interact with the environment by selecting actions from this discrete set, and the environment responds accordingly based on the chosen action. This discrete action space structure allows reinforcement learning algorithms to operate efficiently, especially when actions can be represented as simple integers. There are as many actions in the action space as features in the dataset.
- Observation Space: Another crucial component of the environment is the observation space, defined using Gym's Box space. This space represents a vector of values ranging from 0 to 1, with each value corresponding to a feature. This vector provides the agent with information about the current state of the environment. The shape of the Box space matches the number of features.
- Reward Function: The Reinforcement Learning environment relies significantly on this function, which gives the agent feedback depending on its actions. It guides the agent's training process. This reward function is embedded within the step method. The accuracy of the environment's classification serves as the reward for the agent's action. Higher accuracy indicates better performance, resulting in a higher reward for the agent. The reward function evaluates the agent's selected actions (feature selections) and provides a numerical reward signal indicating the quality of those actions. The purpose of the reward function is to quantify how well the agent's selected features contribute to accurately classifying phishing instances in the dataset. In the proposed system environment, the

https://doi.org/10.31436/iiumej.v26i1.3337

classification accuracy performed using the selected features is the basis for computing the reward. Higher accuracy implies better performance and, therefore, yields a higher reward, while lower accuracy results in a lower reward.

- Classifier Training: This step enables the environment to evaluate the actions taken by the agent and provide feedback on the same. The classification models are trained using a subset of the features from the dataset. The environment uses This classifier internally to classify the dataset based on the agent's selected features. The classifier's accuracy is the reward signal for the agent's actions, guiding its learning process. The trained classifier is a proxy for evaluating the quality of the agent's actions (feature selections) by assessing their impact on classification accuracy.

At the start of each episode, the reset function is initiated so that the environment is reset to its initial state, a random binary vector of feature values. These elements are initialized so that the environment may give the agent relevant feedback and help determine the best feature selection policy for phishing detection. The epsilon-greedy policy determines the action the agent chooses. Selecting a feature index to ascertain the features that have been chosen is an action. The feature subset is then created by the environment as a result of the activity. This feature subset is used to train the classifier, and the accuracy is determined. Accuracy is rewarded, and the chosen features advance to the next stage. Using the reward and the projected future rewards, the Q-value for the state-action pairs is updated in the next stage of the process. The agent transitions to the next state and accumulates the total reward. After the episode ends, all the features are selected at least once.

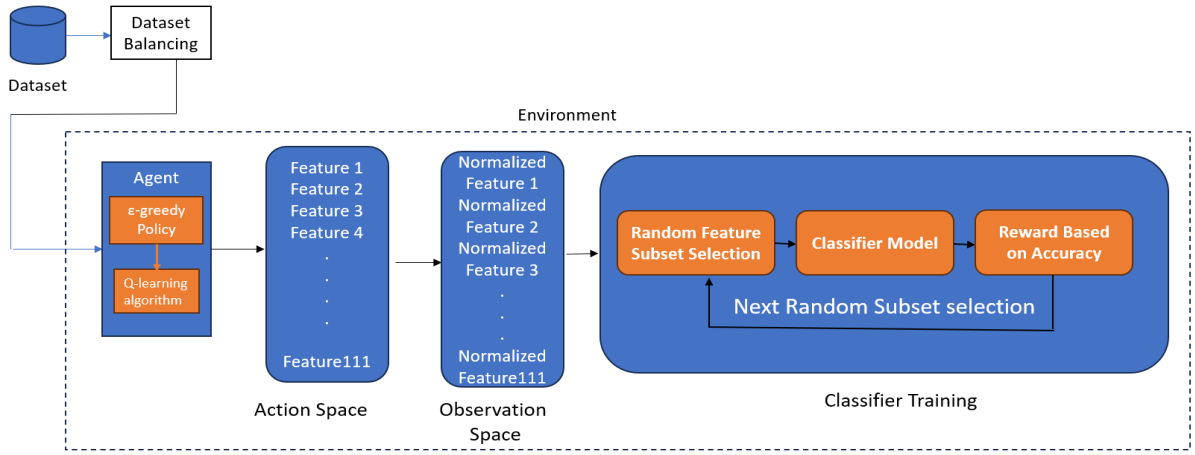

Figure 1. Feature Selection Using Reinforcement Learning

Traditional feature selection methods like the filter method and wrapper approach are also examined in the study. Fig. 2 depicts the methodology applied for analysis of feature selection techniques wherein the dataset is balanced using hybrid dataset balancing techniques followed by preprocessing of the dataset. Feature selection is done using ANOVA, Mutual Information, and Chi-square. Feature importances are also calculated using the SHAP explainable AI algorithm. Fig. 2 depicts the diagrammatic representation of the proposed system for feature selection using traditional feature selection techniques. Before applying the feature importance calculation techniques, redundant and quasi-constant features are removed from the dataset.

https://doi.org/10.31436/iiumej.v26i1.3337

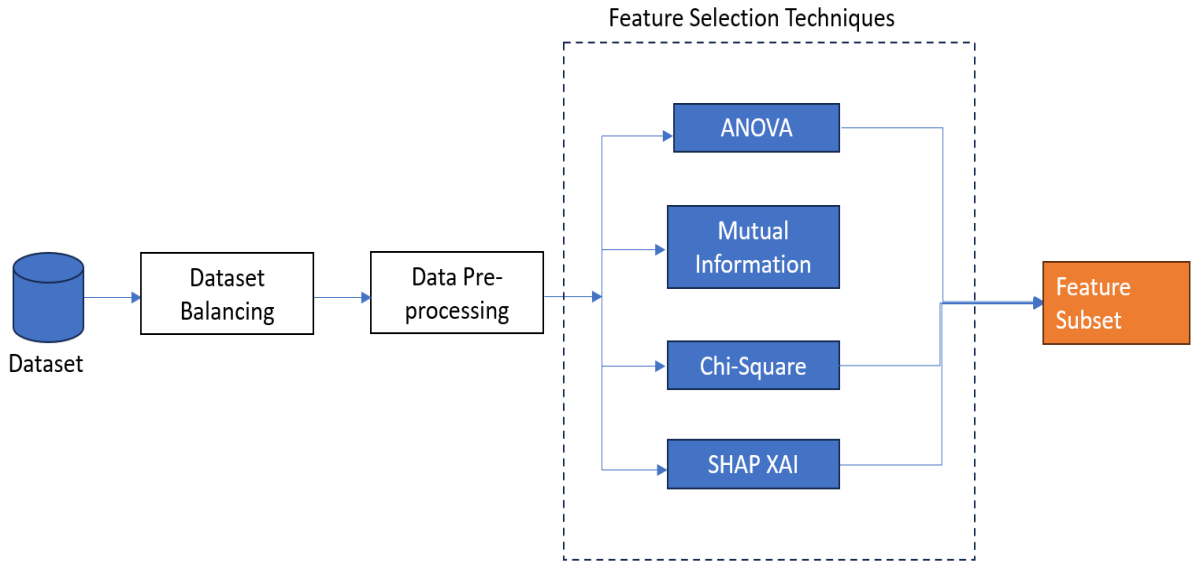

Figure 2. Feature selection using traditional methods

4. RESULTS AND DISCUSSION

This section will cover the evaluation of feature selection methods on phishing datasets. We will also discuss the study of comparative results of the proposed system and the traditional methodology using filter and wrapper methods.

4.1. Dataset Description

The dataset used for the experiments is accessible to the general public on Mendeley [16]. The features reported in the datasets were taken from publicly available lists of legitimate and phishing websites from which the data was obtained. 88,647 instances in total, of which 58,000 are genuine website instances (labeled as 0) and 30,647 are phishing website instances (labeled as 1). A total of 111 features are present. Fig. 3 gives an overview of the distribution of samples in the dataset.

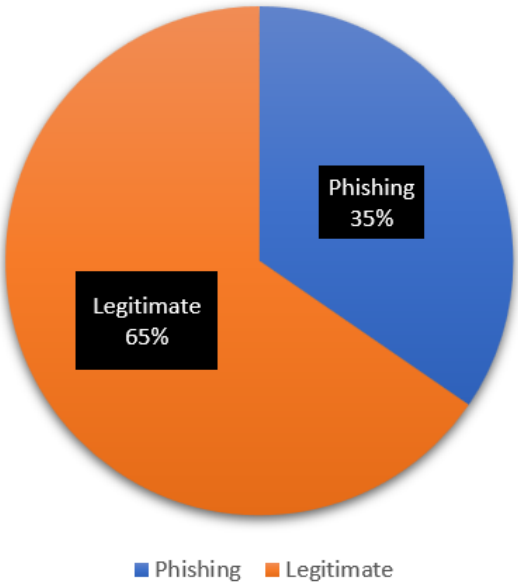

Figure 3. Dataset Distribution

4.2. Evaluation Metrics

The evaluation metrics Accuracy, Precision, Recall, and F-measure demonstrate the efficacy of the proposed approach. The metric used to assess the system's overall performance is accuracy. It is a ratio of the number of accurate predictions to the total number of predictions. [20].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

True positives are denoted as TP. True Positives are those predictions that specify phishing URLs correctly identified as phishing. False positives, or FPs, are the number of actual URLs the model identified as fake, while false negatives, or FNs, are the number of fake URLs identified as genuine. TN stands for true negatives, which are equivalent to URLs that are legitimate and predicted as phishing. The measure of precision evaluates the proportion of accurate identifications [18]. The ratio of True Positive samples to all predicted positive samples can be used to describe it. When the costs of False Positives are large, precision is a useful metric to assess [20]. Precision can be calculated using the given formula,

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Recall is the measure of our system correctly identifying True Positives [20]. Whenever the cost of the fake samples getting unidentified is high, recall is used to select the best model. Thus, recall tells us URLs that are actually phishing and correctly identified as phishing. The formula gives it,

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

It is important to select the performance metric for the classification model. Precision can be used when the cost of False Positives is High, that is, when the true URL is classified as fake. Recall is equally important for predicting phishing URLs since Recall calculates how many of the Actual Positives (Fake URLs) our model captured by labeling them as Fake. In the model, precision and recall are equally important. F1-score is the Harmonic mean of Precision and Recall [18].

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (8)$$

4.3. Baseline Algorithms

We compare the performance of the proposed feature selection using baseline algorithms like K-best selection, Recursive Feature Elimination, Feature selection using the SHAP algorithm, Mutual Information, and Chi-square.

4.3.1. K-best Selection

K-Best ranks the top 'k' attributes according to a grading system. This scoring method was developed based on statistical tests that measure the degree of correlation between each feature and the target variable. Every aspect is assessed separately using a predefined scoring system. Common scoring functions include ANOVA F-value, mutual information, and chi-squared. These scoring functions evaluate each feature's significance to the target variable—the proposed methodology used K-best Selection with ANOVA as a scoring function.

4.3.2. Recursive Feature Elimination (RFE).

This algorithm works in a recursive, wherein the least significant features are removed from the data until the target feature count is attained. RFE works in the following steps: Model Training, Feature Importance Ranking, Feature Elimination, and Model Retraining. First, the ML model is trained using all the features. The significance of every feature is calculated once the model has been trained. Depending on the approach utilized, this can be accomplished using alternative metrics or coefficients in linear models or feature importances in tree-based models. Next, the features that aren't as crucial are eliminated from the feature set. Using the smaller feature set, the model is retrained. Recursively, the stages of feature elimination, model retraining, and feature importance ranking are carried out until the required number of features is attained.

4.3.3. Feature selection using the SHAP algorithm

The SHAP (SHapley Additive exPlanations) algorithm is a feature selection technique that uses game theory to give each feature a score determined by how much it contributes to the prediction of a machine learning model. A unifying indicator of feature relevance, SHAP values consider the strength and direction of each feature's influence on predictions. SHAP values are first calculated for every feature in the dataset. The influence of each feature on the model's predictions for specific data points is represented by SHAP values. By assigning each feature the difference between the actual and expected predictions, they offer a local explanation for the predictions made by the model. The total relevance of every feature can be determined by averaging the computed SHAP values. Several summary statistics can be used for this aggregation, including mean absolute SHAP values, mean SHAP values, and variation of SHAP values throughout the dataset. Next, features are arranged according to their importance scores obtained from SHAP values. Retaining the top-ranked features for additional analysis or model training depends on the number of features to choose from. The most significant features are chosen, and only those are used to train a machine learning model.

4.3.4. Mutual Information-based Feature Selection

A technique for choosing features is called mutual information-based feature selection, and it involves calculating the mutual information or dependency between each feature and the target variable. Mutual information is a measure of the amount of information obtained about one variable through the other variable.

4.3.5. Chi-square-based Feature selection

Using a measure of each feature's dependence on a categorical target variable, this feature selection method helps to find the most relevant features for classification problems. The chi-square statistic is computed for every feature, which shows how strongly the feature is associated with the target. This information is used to rank the features, with the highest-ranked features being chosen for model training.

4.4. Performance

We compare our feature selection method using reinforcement learning regarding accuracy, precision, recall, and F1-measure on the phishing dataset. In an initial stage where feature selection was made using baseline algorithms, the constant features were removed from the dataset. It was observed that among 111 features, 14 features were constant. Table No. 2 gives the list of constant features identified. Constant features have the same value for all observations, whereas duplicate features contain redundant information with other features.

https://doi.org/10.31436/iiumej.v26i1.3337

Removing these redundant features is crucial for improving the efficiency and effectiveness of the machine learning model. In the next step, duplicate features were removed from the dataset.

Table 2. Constant Features

Sr. No	Feature Name
1	qty_slash_domain
2	qty_questionmark_domain
3	qty_equal_domain
4	qty_at_domain
5	qty_and_domain
6	qty_exclamation_domain
7	qty_space_domain
8	qty_tilde_domain
9	qty_comma_domain
10	qty_plus_domain
11	qty_asterisk_domain
12	qty_hashtag_domain
13	qty_dollar_domain
14	qty_percent_domain

Table 3. Quasi-constant Features

Sr.No	Feature Name	URL Variance
1	qty_exclamation_	0.00762844964915398
2	qty_space_	0.005278395261483356
3	qty_tilde_	0.00610379807390346
4	qty_comma_	0.005771091175729697
5	qty_hashtag_	0.003801380283231768
6	qty_dollar_	0.009946095859835575
7	qty_underline_domain	0.010148545461142985
8	qty_slash_domain	0.0
9	qty_questionmark_domain	0.0
10	qty_equal_domain	0.0
11	qty_at_domain	1.1280697598339478e-05
12	qty_and_domain	0.0
13	qty_exclamation_domain	0.0
14	qty_space_domain	0.0
15	qty_tilde_domain	0.0
16	qty_comma_domain	0.0
17	qty_plus_domain	0.0
18	qty_asterisk_domain	0.0
19	qty_hashtag_domain	0.0
20	qty_dollar_domain	0.0
21	qty_percent_domain	0.0
22	domain_in_ip	0.0022623045431979945
23	server_client_domain	0.004480789902101015
24	url_google_index	0.003438663212102363
25	domain_google_index	0.004000615427297923
26	url_shortened	0.00545242362148391

A total of 26 features were identified as Quasi-constant features. The features with very low variance are known as Quasi-constant features. These features may provide little to no discriminatory information for predictive modeling and can potentially lead to overfitting. Identifying and handling quasi-constant features is an important step in data preprocessing to

https://doi.org/10.31436/iiumej.v26i1.3337

improve the performance and interpretability of the machine learning model. Table 3 shows the list of quasi-constant features from the phishing dataset. Table No. 4 gives a list of duplicate features in the dataset.

Table 4. List of duplicate features

Sr. No	Feature Name
1	qty_hashtag_directory
2	qty_slash_file
3	qty_questionmark_file
4	qty_hashtag_file
5	qty_dollar_file
6	qty_slash_file
7	qty_questionmark_file
8	qty_hashtag_file
9	qty_dollar_file
10	qty_questionmark_file
11	qty_hashtag_file
12	qty_dollar_file
13	qty_hashtag_file
14	qty_dollar_file
15	qty_dollar_file

In the next stage, we have focused on correlated features. Correlated features are pairs of features in a dataset that exhibit some degree of linear relationship with each other. This relationship can be positive (both features increase or decrease together) or negative (one feature increases while the other decreases). Identifying correlated features is important in data preprocessing as highly correlated features can introduce redundancy and potentially lead to overfitting in predictive modeling. Removing correlated features in the context of phishing detection can positively and negatively affect performance, depending on factors such as the dataset, the algorithm used, and the specific features being removed. Eliminating correlated features may lead to loss of information, and removing them may overly simplify the model, leading to underfitting. The total correlated features identified were 42. The system performance was tested with correlated features and by removing correlated features.

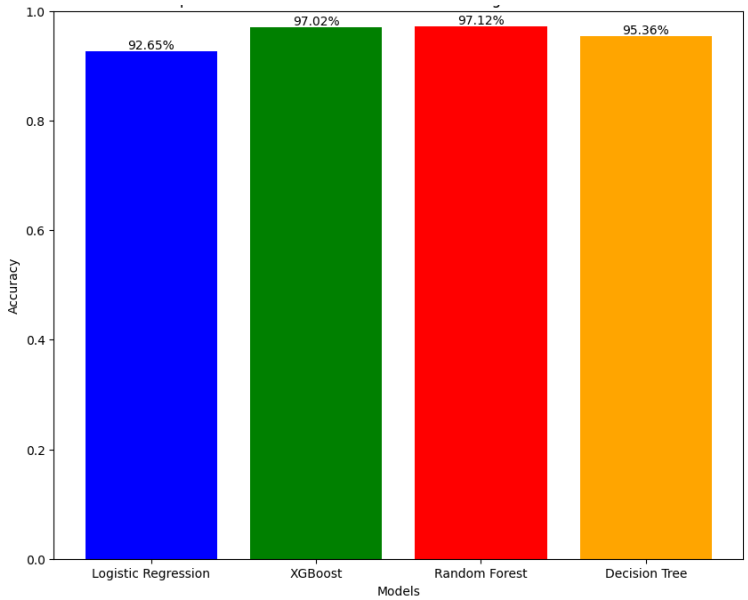

Figure 4. Comparison of Accuracy after removing Correlated Features.

https://doi.org/10.31436/iiumej.v26i1.3337

Fig. 5 shows the accuracy of the models with the correlated features, and Fig. 4 shows the accuracy of the models after removing the correlated features. For XG-Boost, Random Forest, and Decision Tree, accuracy was lower than that of correlated features. Accuracy was better in Logistic Regression for the dataset with correlated features.

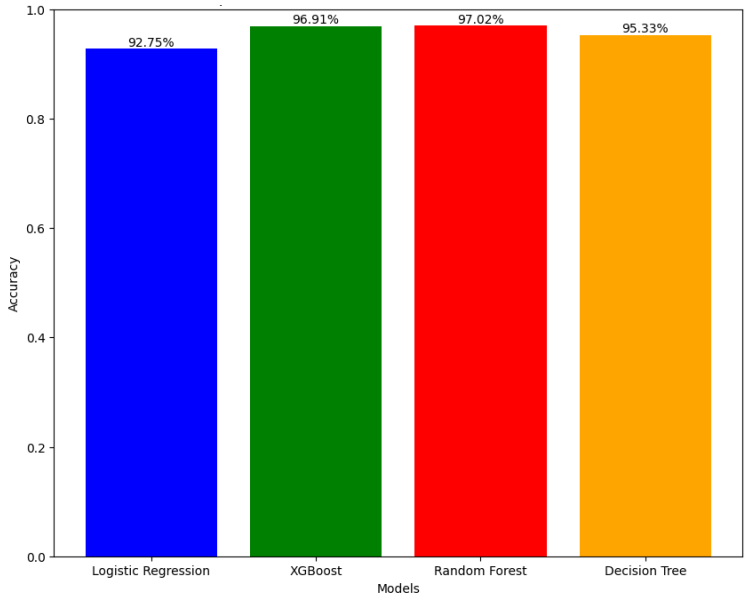

Figure 5. Comparison of Accuracy with Correlated Features.

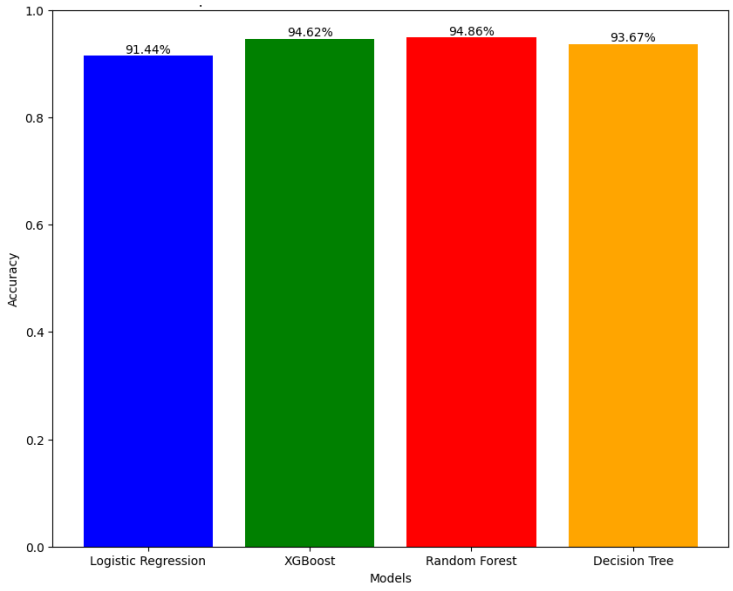

Figure 6. Comparison of Accuracy with ANOVA Feature Selection

While testing the K-best algorithm for feature selection, a total of 10 features were selected. The system utilizes the ANOVA F-value method via SelectKBest to select the top 10 features from the training dataset (X_train) based on their importance for classification. It then transforms the training and test datasets to contain only the selected features, maintaining consistency across the datasets. The selected feature names are captured and used to label the columns of the transformed datasets. ANOVA (Analysis of Variance) is a statistical technique that determines whether there are statistically significant differences between the means of three or more groups. In the context of feature selection, f_classif calculates the F-value for

https://doi.org/10.31436/iiumej.v26i1.3337

each feature by considering the variance between the classes (or groups) and the variance within the classes. 'qty_slash_url', 'qty_equal_url', 'length_url', 'qty_dot_domain', 'qty_dot_directory', 'qty_hyphen_directory', 'qty_underline_directory', 'directory_length', 'qty_questionmark_params', 'time_domain_activation' these features were selected using K-best Algorithm.

Feature selection using mutual information was done on the dataset by calculating the mutual information score for each feature. Fig. 7 depicts the score for mutual information of the features in the dataset. Based on this score top 10 features were selected using mutual information. These features were 'qty_slash_url', 'length_url', 'qty_dot_directory', 'qty_hyphen_directory', 'qty_underline_directory', 'directory_length', 'qty_hyphen_file', 'file_length', 'asn_ip', 'time_domain_activation'.

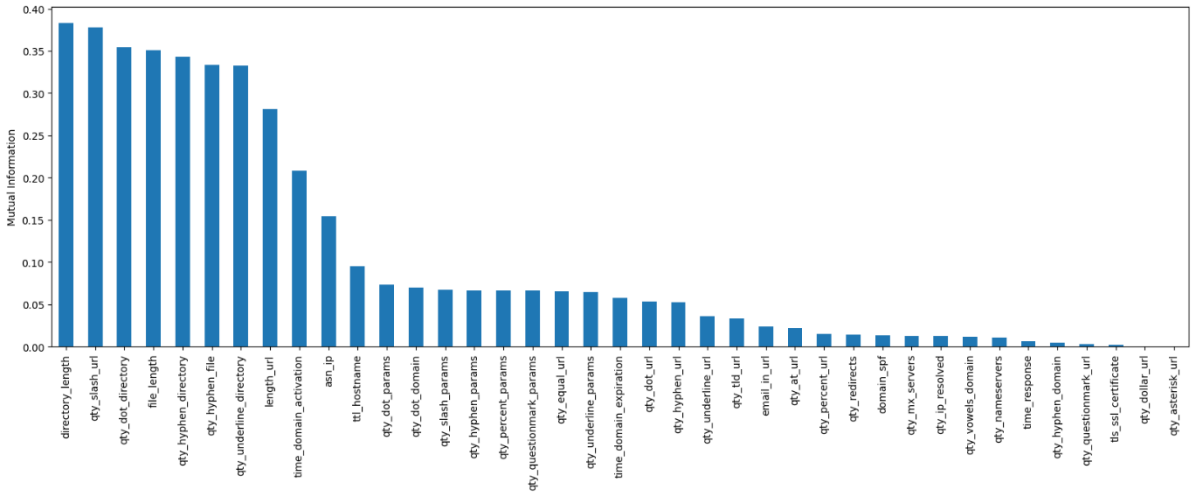

Figure 7. Mutual Information Scores

The dataset with features selected using mutual information was tested on different classification models. Fig. 8 gives an overview of the accuracy achieved after performing feature selection.

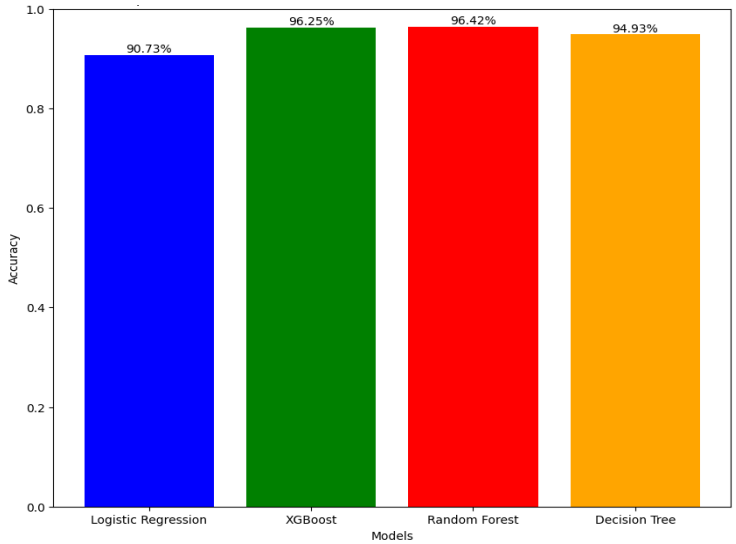

Figure 8. Comparison of Model Accuracies with Mutual Information-based select Features.

https://doi.org/10.31436/iiumej.v26i1.3337

The next baseline algorithm that was applied is Recursive Feature Elimination. In this approach, Recursive Feature Elimination (RFE) with a Random Forest classifier and Logistic Regression is done to select the top 10 features from the dataset. This approach offers a systematic way to identify the most relevant features for modelling tasks, aiding in dimensionality reduction and improving model interpretability. Table 5 shows the list of features selected by the RFE algorithm using Logistic Regression and Random Forest. Fig. 9 shows the comparison of accuracies obtained for RFE with logistic regression and RFE with Random Forest.

Table 5. Features Selected using RFE

Sr.No	Logistic Regression	Random Forest
1	'qty_hyphen_url'	'qty_slash_url'
2	'qty_slash_url'	'length_url'
3	'qty_at_url'	'qty_hyphen_directory'
4	'qty_asterisk_url'	'qty_underline_directory'
5	'qty_hyphen_domain'	'directory_length'
6	'qty_dot_directory'	'qty_hyphen_file'
7	'qty_hyphen_directory'	'file_length'
8	'qty_hyphen_params'	'asn_ip'
9	'qty_questionmark_params'	'time_domain_activation'
10	'email_in_url'	'ttl_hostname'

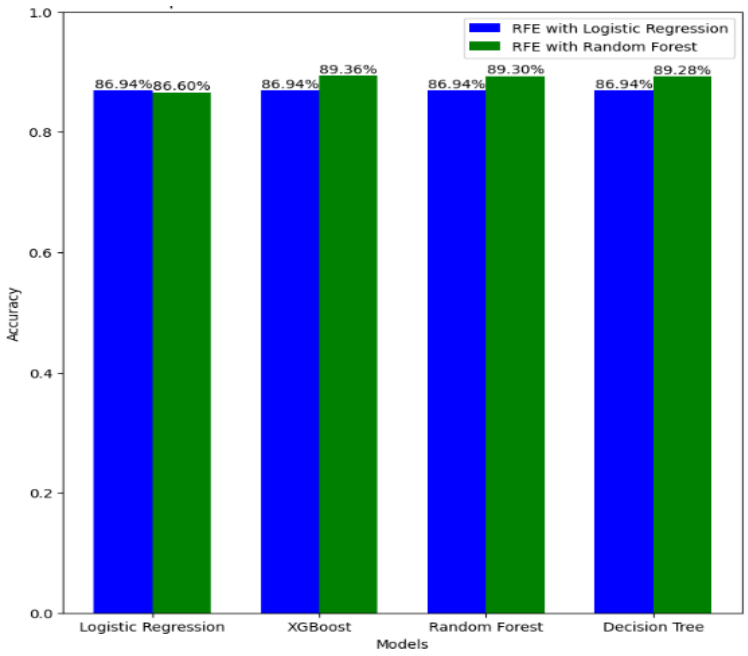

Figure 9. Comparison of Model Accuracies with RFE-Selected Features

The study uses the SHAP (SHapley Additive exPlanations) algorithm to explain the XG-Boost Classifier's output. It computes SHAP values for every feature after training the XG-Boost Classifier model on the training set. Machine learning models can be interpreted using SHAP (SHapley Additive exPlanations) values, which relate the model's output to its input features. They offer a single, Shapley-value-based way to quantify the significance of features. The SHAP values indicate the impact of each feature on the models' predictions. Understanding the importance of each feature to a particular prediction is aided by the explanation provided by SHAP values for individual predictions. The visualization for the XG-Boost Classifier is

https://doi.org/10.31436/iiumej.v26i1.3337

shown in Fig.10. The summarization of accuracies using the SHAP algorithm is depicted in Fig. 11.

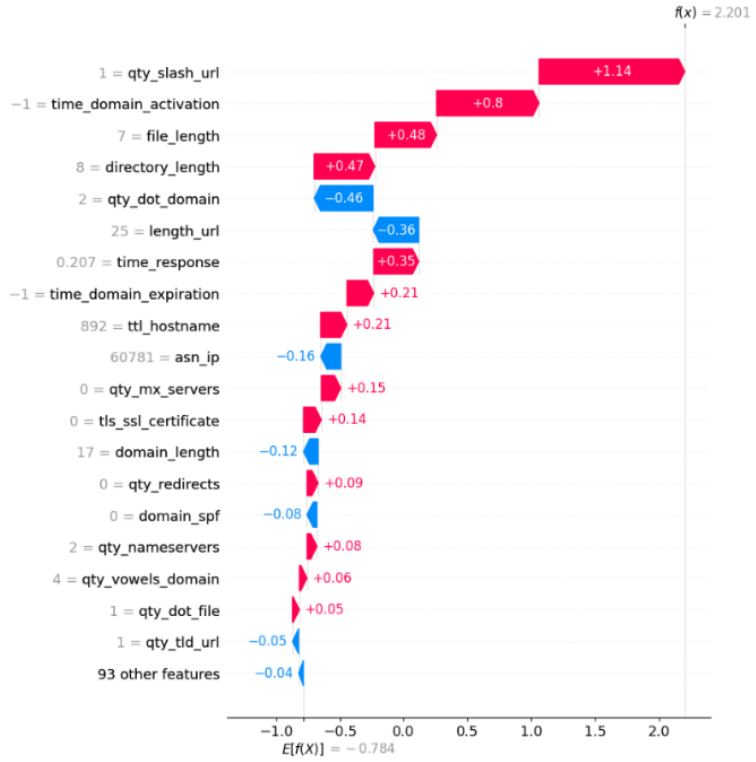

Figure 10. SHAP Values.

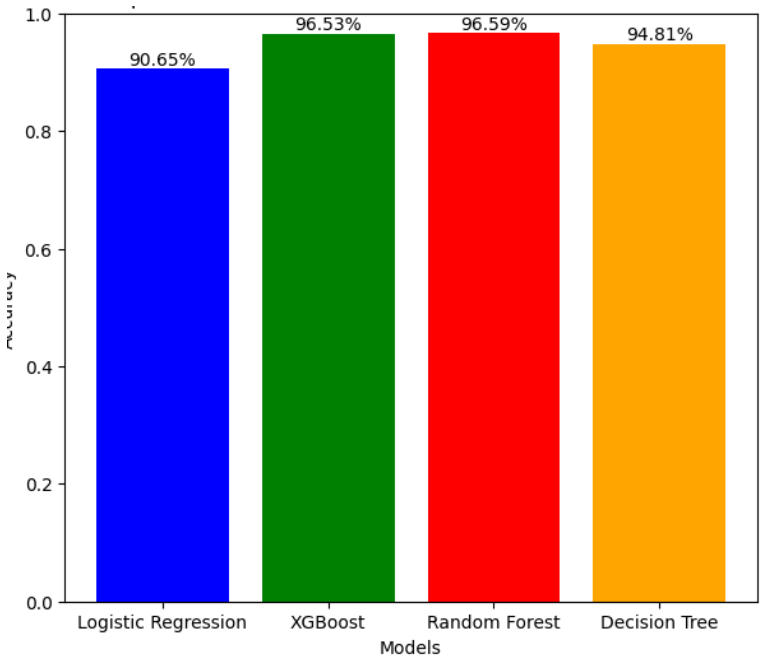

Figure 11. Comparison of Model Accuracies with SHAP-Selected Features

Fig. 13 shows the features selected based on the chi2 scoring function. It also shows the accuracy of the models using the selected features based on chi-square. Chi-square selects the features that are most relevant to the target phishing variable. The features selected using chi-square are shown in Fig. 12.

https://doi.org/10.31436/iiumej.v26i1.3337

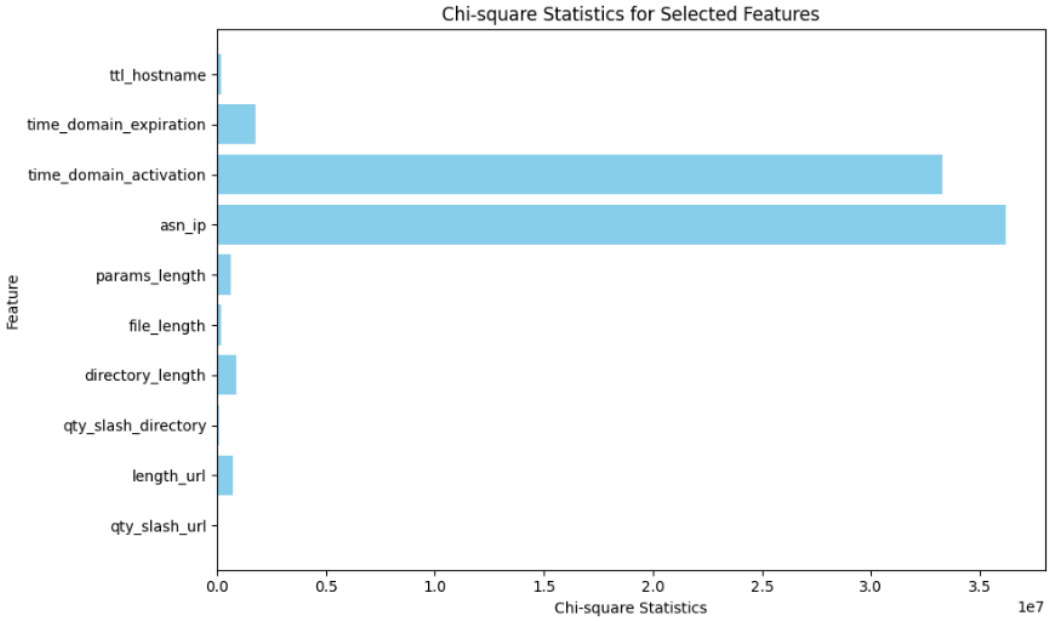

Figure 12. Chi-square values for top 10 features.

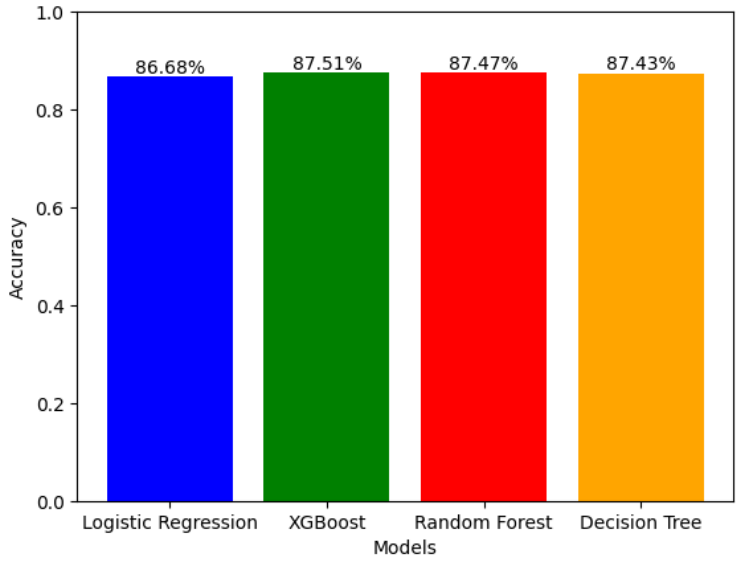

Figure 13. Accuracy of Models using feature selection with chi-square.

XG Boost algorithm gave an accuracy of 96.42 after 100 episodes of training. The selected features were 'time_domain_activation', 'qty_dot_domain', 'directory_length', 'params_length', 'qty_space_directory', 'tls_ssl_certificate', 'asn_ip', 'qty_asterisk_params', 'qty_dot_file', 'qty_tilde_directory'. The selected feature subset using Random Forest Algorithm was 'qty_redirects', 'asn_ip', 'qty_asterisk_params', 'qty_slash_url', 'file_length', 'qty_and_url', 'qty_equal_directory', 'qty_slash_directory', 'time_domain_activation', 'time_response' with best accuracy of 99.07 %. The selected feature subset using decision tree is 'qty_dot_params', 'time_response', 'qty_comma_file', 'qty_ip_resolved', 'domain_length', 'qty_comma_url', 'directory_length', 'qty_percent_params', 'time_domain_activation', 'qty_equal_params'. The accuracy achieved is 98.69%. Fig. 14 shows the comparison of accuracy for the algorithms.

https://doi.org/10.31436/iiumej.v26i1.3337

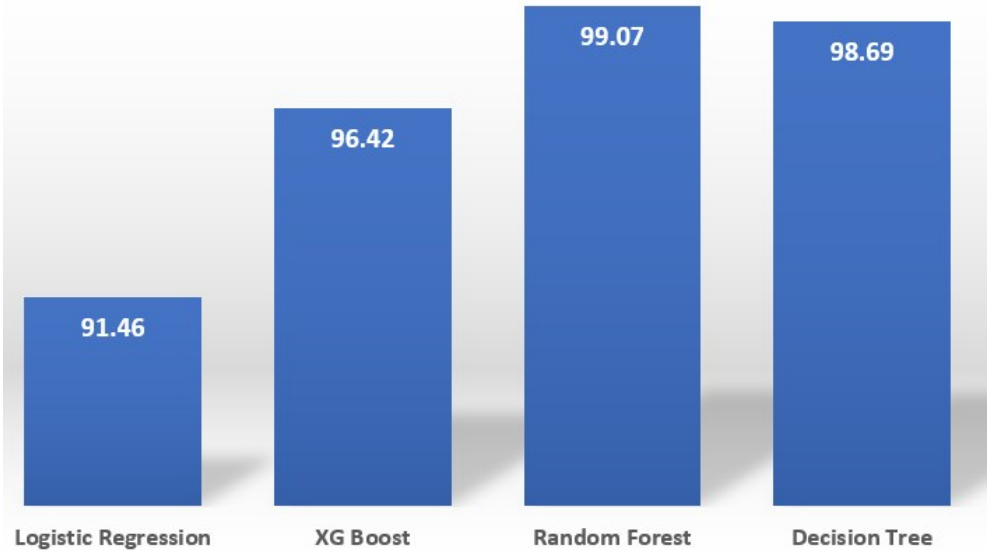

Figure 14. Cumulative Results

The proposed methodology was also evaluated based on other performance metrics like precision, recall, and f1-score. Table 6 shows the cumulative results obtained while testing the system. It compares the accuracy obtained for different feature selection techniques and the classification algorithms considered in this study. The proposed methodology significantly improves the accuracy of the Random Forest and Decision Tree, reaching near-perfect levels. This indicates that the methodology effectively enhances these models' ability to classify phishing websites, making them exceptionally reliable correctly. The proposed methodology consistently outperforms other feature selection techniques in terms of accuracy, particularly for ensemble methods like Random Forest and Decision Tree, which achieve near-perfect accuracy.

Table 6. Cumulative Results for Accuracy

Accuracy	Logistic Regression	XG-Boost	Random Forest	Decision Tree
With Correlated Features	92.65%	97.02%	97.12%	95.36%
After the removal of Correlated Features	92.75%	96.91%	97.02%	95.33%
K-best (ANOVA)	91.44%	94.62%	94.86%	93.67%
Mutual Information	90.73%	96.25%	96.42%	94.93%
Chi-square	86.68%	87.51%	87.47%	87.43%
RFE	90.96%	91.94%	91.87%	91.86%
SHAP	90.62%	96.53%	96.59%	94.81%
Proposed Methodology	91.39%	92.45%	99.07%	95.49%

Table 7 shows the proposed system's comparative results based on the precision performance metrics. The proposed methodology has outperformed all the other approaches to feature selection for Random Forest and Decision Tree Classifiers. The high Precision suggests that the feature selection process in the proposed methodology is very effective at isolating the most relevant features for these ensemble methods, allowing them to make highly accurate positive predictions. For Logistic Regression, the precision value decreased. This reduction in Precision suggests that the proposed feature selection may include some features that introduce

https://doi.org/10.31436/iiumej.v26i1.3337

noise or are less compatible with the linear nature of Logistic Regression, leading to more false positives. With 94% accuracy in positive predictions, XG-Boost performs reliably, though slightly less so than Random Forest and Decision Tree. The proposed methodology excels in Precision for ensemble models but may introduce challenges for linear models like Logistic Regression, which could lead to a higher rate of false positives.

Table 7. Cumulative Results for Precision

Precision	Logistic Regression	XG-Boost	Random Forest	Decision Tree
With correlated features.	0.93	0.97	0.97	0.95
After the removal of Correlated Features	0.93	0.97	0.97	0.95
K-best (ANOVA)	0.91	0.95	0.95	0.94
Mutual Information	0.91	0.96	0.96	0.95
Chi-square	0.90	0.89	0.89	0.89
RFE	0.87	0.87	0.87	0.87
SHAP	0.91	0.97	0.97	0.95
Proposed Methodology	0.85	0.94	0.98	0.98

The Recall performance metric is critical in evaluating how well your models identify all phishing websites. Random Forest and Decision Tree both achieve the highest Recall with the proposed methodology. A Recall of 0.98 means these models successfully identify 98% of all phishing websites. This is particularly valuable in security applications, where failing to detect a phishing threat (false negatives) can have serious consequences. The high Recall indicates that the proposed methodology is highly effective at selecting features that help these ensemble models capture nearly all relevant phishing cases, making them robust and reliable. With a Recall of 0.95, XG-Boost detects 95% of all phishing sites. While slightly lower than Random Forest and Decision Tree, it still performs strongly. While still performing reasonably well, Logistic Regression might miss more phishing sites compared to the ensemble methods, which could be a concern in highly sensitive applications.

Table 8. Cumulative Results for Recall

Recall	Logistic Regression	XG-Boost	Random Forest	Decision Tree
With correlated features.	0.93	0.97	0.97	0.95
After the removal of Correlated Features	0.93	0.97	0.97	0.95
K-best (ANOVA)	0.91	0.95	0.95	0.94
Mutual Information	0.91	0.96	0.96	0.95
Chi-square	0.87	0.88	0.87	0.87
RFE	0.87	0.89	0.89	0.89
SHAP	0.91	0.97	0.97	0.95
Proposed Methodology	0.91	0.95	0.98	0.97

Random Forest and Decision Tree both achieve the highest F1-Score (0.98) with the proposed methodology. The F1-Score of 0.98 indicates that these models are accurate in predicting phishing websites (high Precision) and very effective at identifying nearly all actual phishing sites (high Recall). An F1-Score of 0.95 demonstrates that XG-Boost maintains a strong balance between Precision and Recall, though slightly less than the ensemble models. The F1-Score of 0.88 indicates a more noticeable trade-off between Precision and Recall for

Logistic Regression under the proposed methodology. While it balances these metrics reasonably well, it does not perform as strongly as the ensemble methods.

Table 9. Cumulative Results for F1-Score

F1-Score	Logistic Regression	XG-Boost	Random Forest	Decision Tree
With correlated features.	0.93	0.97	0.97	0.95
After the removal of Correlated Features	0.93	0.97	0.97	0.95
K-best (ANOVA)	0.91	0.95	0.95	0.94
Mutual Information	0.91	0.96	0.96	0.95
Chi-square	0.87	0.88	0.88	0.88
RFE	0.86	0.89	0.89	0.89
SHAP	0.91	0.97	0.97	0.95
Proposed Methodology	0.88	0.95	0.98	0.98

Based on the results obtained in the study, the proposed methodology works better than most traditional techniques. The results show that the proposed feature selection methodology using reinforcement learning outperforms all the traditional feature selection algorithms for decision tree and Random Forest classifier with an accuracy of 98.69% and 99.007%. For logistic regression, it was observed that it worked better than mutual information, K-best, Chi-square, RFE, and SHAP-based feature selection. The proposed feature selection algorithm showed better results than ANOVA, Chi-square, and RFE for the Random Forest Algorithm. The results show that the proposed methodology gave the best accuracy compared to all the other techniques with the Random Forest Classifier. It was also observed that removing correlated features might not always improve the system's performance.

5. CONCLUSION

This paper proposes a novel system that selects the features using a reinforcement learning algorithm. The methodology focuses primarily on the phishing dataset. Using the proposed method, we can achieve dynamic feature selection at each episode and improve accuracy based on the rewards the agent receives. We have studied the traditional feature selection algorithms and then analyzed their performance. The research analyzed the effect of correlated features in the dataset. It was observed that there was an improvement in the accuracies of XG-Boost, Random Forest, and Decision Tree models after the removal of the correlated features. The proposed methodology is tested using logistic regression, XG-Boost, random forest, and decision tree classification algorithms. We have tested the results using the accuracy gained for each classification algorithm. We have considered the feature subset size to be 10 and the number of episodes to be 100. Based on the above parameters, we have 91.46,96.42,99.07, and 98.69 accuracies for Logistic Regression, XGBoost, Random Forest, and Decision Tree, respectively. Considering the evaluation parameters' accuracy, precision, recall, and F1 score, the results show that the random forest classifier algorithm gives the best results. In the future, the system can be extended to different feature subsets by increasing the number of episodes. Also, multiple phishing datasets can be studied to get an appropriate feature subset and an optimal feature subset size.

REFERENCES

[1] Infosecurity Magazine, "341% Rise in Advanced Phishing Attacks," [Online]. Available: https://www.infosecurity-magazine.com/news/341-rise-advanced-phishing-attacks/.
[2] [Internet Crime Complaint Center (IC3), "2023 Internet Crime Report," [Online]. Available: https://www.ic3.gov/media/pdf/annualreport/2023_ic3report.pdf.
[3] Fredj, Ouissem Ben, et al. "An OWASP top ten driven survey on web application protection

methods." Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, November 4–6, 2020, Revised Selected Papers 15. Springer International Publishing, 2021.

[4] Alsenani, Theyab R., et al. "Intelligent feature selection model based on particle swarm optimization to detect phishing websites." *Multimedia Tools and Applications* 82.29 (2023): 44943-44975.

[5] Shabudin, Shafaizal, et al. "Feature selection for phishing website classification." International Journal of Advanced Computer Science and Applications 11.4 (2020).

[6] Hamim, Mohammed, et al. "A novel dimensionality reduction approach to improve microarray data classification." *IIUM Engineering Journal* 22.1 (2021): 1-22.

[7] Kalabarige, L. R., Rao, R. S., Pais, A. R., & Gabralla, L. A. (2023). A Boosting-based Hybrid Feature Selection and Multi-layer Stacked Ensemble Learning Model to detect phishing websites. IEEE Access.

[8] Shabudin, S., Sani, N. S., Ariffin, K. A. Z., & Aliff, M. (2020). Feature selection for phishing website classification. International Journal of Advanced Computer Science and Applications, 11(4).

[9] Adane, K., Beyene, B., & Abebe, M. (2023). Single and hybrid-ensemble learning-based phishing website detection: examining impacts of varied nature datasets and informative feature selection technique. Digital Threats: Research and Practice, 4(3), 1-27.

[10] Calzarossa, M. C., Giudici, P., & Zieni, R. (2024). Explainable machine learning for phishing feature detection. Quality and Reliability Engineering International, 40(1), 362-373.

[11] Abulfaz Hajizada and Sharmin Jahan. 2023. Feature Selections for Phishing URL Detection Using Combination of Multiple Feature Selection Methods. In 2023 15th International Conference on Machine Learning and Computing (ICMLC 2023), February 17–20, 2023, Zhuhai, China. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3587716.3587790

[12] Alotaibi, Bandar, and Munif Alotaibi. "Consensus and majority vote feature selection methods and a detection technique for web phishing." Journal of Ambient Intelligence and Humanized Computing 12 (2021): 717-727.

[13] Singh, Amit, and Abhishek Tiwari. "A study of feature selection and dimensionality reduction methods for classification-based phishing detection system." International Journal of Information Retrieval Research (IJIRR) 11.1 (2021): 1-35.

[14] Bu, S. J., & Kim, H. J. (2022). Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection. Electronics, 11(7), 1090.

[15] Moedjahedy, J., Setyanto, A., Alarfaj, F. K., & Alreshoodi, M. (2022). CCrFS: combine correlation features selection for detecting phishing websites using machine learning. Future Internet, 14(8), 229.

[16] Goud, N. Swapna, and Anjali Mathur. "Feature Engineering Framework to detect Phishing Websites using URL Analysis." International Journal of Advanced Computer Science and Applications 12.7 (2021).

[17] Liu, D. J., Geng, G. G., Jin, X. B., & Wang, W. (2021). An efficient multistage phishing website detection model based on the CASE feature framework: Aiming at the real web environment. Computers & Security, 110, 102421.

[18] Ali, Waleed, and Sharaf Malebary. "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection." IEEE Access 8 (2020): 116766-116780.

[19] Vrbančič, Grega, Iztok Fister Jr, and Vili Podgorelec. "Datasets for phishing websites detection." Data in Brief 33 (2020): 106438.

[20] Foundational Courses. Google for Developers Online. Available: https://developers.google.com/machine-learning/crash-course/classification