A Reinforcement Learning Framework for Adaptive Detection of Phishing Attack

Sharvari Patil[1,*], Narendra M. Shekokar[1], Aditya Surve[1], Priyanka Ramchandran[1]

[1]Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

Emails: sharvarichorghe@gmail.com; narendra.shekokar@djsce.ac.in; surveaditya521@gmail.com; priyankaar25@gmail.com

Abstract

Phishing is one of the most dominant forms of cybercrime, with over half a billion incidents occurring annually. It remains one of the most insidious forms of fraud due to its effectiveness. Phishing attacks are on the rise with increasingly deceptive tactics, often leading unwitting victims to divulge personal information. Phishing frauds also involve website phishing, which mimics legitimate sites. Despite the best user training and practices, people still fall for these frauds. The methodology of detecting phishing attacks using the blacklisting approach was not very effective since these URLs are active for a limited period. Hence, Machine Learning methods were used for detecting the phishing attempt. Machine learning solutions are not adaptive to changes in the approach and are biased towards the developed solution. In addition, there is a need to develop a solution to this constantly evolving phishing attack. The proposed system is an attempt to use reinforcement-learning methodology as the solution to detect phishing. It has trained an adaptive intelligent learning system based on previous experiences using the Q-learning algorithm. The system focuses on dynamically selecting the relevant features and the classification model. The agent is trained to select optimal features and classification models dynamically based on Q-learning algorithm. In contrast to static methods, the proposed system continuously adapts its strategy of combinations feature subsets and classification models as defense against the rapidly evolving attacks. The system aims to supplement existing cybersecurity measures with an adaptable tool capable of countering sophisticated phishing schemes. The experimental analysis shows that the proposed methodology attained an accuracy of 99.25%, demonstrating its high performance in phishing detection.

Keywords: Reinforcement Learning; Phishing Detection; Internet Security; Feature Engineering; Website Classification; Q-Learning; Adaptive Learning

1. Introduction

The Internet has been of great convenience in the past few decades for various activities like banking, communication and entertainment. Despite the advantages of the internet, it leads users to serious security issues like phishing, viruses and disclosure of confidential data. Phishing is an attack where the attacker uses their technical skill to launch an attack. Phishing is a very popular technique used to get access to confidential data. According to the Zscaler ThreatLabz 2024 Phishing Report, India ranks third globally in the number of phishing attacks, with over 79 million incidents recorded in 2023 alone. This accounts for 33.12% of all phishing attempts in the Asia Pacific & Japan region, making it the most targeted country in this area [1]. In addition, the Anti-Phishing Working Groups Phishing Activity Trends Report showed that five million phishing attempts were observed in 2023. Despite a decline in the second quarter, phishing incidents surged towards the end of the year, with the APWG documenting 1,077,501 attacks in the fourth quarter alone [2]. Phishing has thus become a significant threat to Internet users.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

A phishing website is a platform used by cybercriminals with malicious intentions, such as stealing credentials or perpetrating financial fraud. Individuals who have clicked on a malicious link within a deceptive email often access these websites. Cybercriminals use diverse methods to deploy phishing websites. One typical scenario, as depicted in Figure 1 involves a user landing on a fake login page where the user inputs their credentials. These credentials are subsequently harvested by phishers, thereby leading to account takeovers. Sometimes they are prompted to pay for the products that remain undelivered. Phishing websites can download harmful files automatically. These occur in the background or the user is forced to download them. In multiple cases, the attackers also use HTTPS and SSL certificates to develop fake websites making it easy to trap the users.

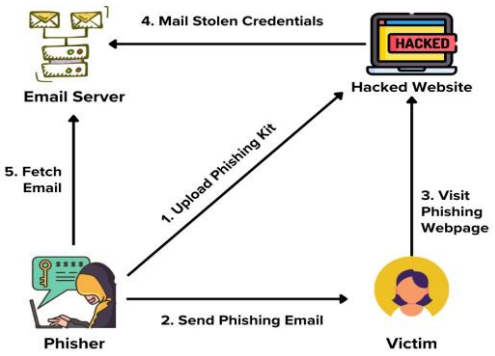

Figure 1. Phishing Cycle

Commonly used techniques to detect phishing are using Blacklists and Whitelists. Browsers commonly integrate these lists to safeguard users from phishing outbreaks. A Google service, Safe Browsing helps client applications verify URLs. This service verifies the URLs by comparing them with Google has blacklisted URLs. This list is regularly updated for potentially harmful websites [3]. Phishing detection using this static approach is easy to implement with less response time and a low false positive rate. These static methods are ineffective in identifying zero-day phishing attacks, where the fraudulent website remains active for only a short period. Researchers have proposed machine Learning, Deep learning and other heuristic-based approaches in the past [22]. Machine Learning and Deep learning model's performance is based on the attributes used to develop the model. The selected features are static in nature and can be easily exploited by attackers. Attackers possess knowledge of the classification models functionality, enabling them to evade the deployed security mechanisms. Traditional methods, which rely on static feature sets, often might fall short against sophisticated, rapidly evolving phishing tactics. [16]

We have proposed a Reinforcement Learning (RL) framework to overcome these challenges. This method is intended to dynamically determine the best grouping of attributes and machine learning models for detecting fake websites. The Reinforcement Learning framework consists of an agent that takes action in the specified environment. Each agent's action changes the environment state. The State, Action, Reward, and Transition dynamics are some components of the environment for agent interaction. State space refers to all possible states in which the agent can be in the environment; it includes particular arrangements of the environment that provide the information needed for a specific course of action selection by the agent. We have defined the action space as the possible actions that an agent can perform in any given state [14,21]. A reward function's purpose is to give feedback from the environment to teach agents which behaviors maximize cumulative rewards over time [14,21]. Transition dynamics explain how one state changes after certain actions by the actor have been taken into account. Actions are governed by policies, which are strategies or rules guiding an agent at any given moment [18].

To design a framework using Reinforcement Learning for phishing detection States, Actions and Rewards are defined. State Space is determined by the feature subset selected. Action Space includes the classification model selection by the agent. The environment consists of the features to be selected from the dataset and the classification models. The proposed framework uses the Q-learning algorithm to define the agent's policy. The algorithm maps states to actions, giving directions on which actions should be taken by the agent in various states to maximize cumulative rewards over time. The Q-learning algorithm develops a policy for an agent. It instructs on what decisions to make at its current state to maximize cumulative rewards over time. To do this, the Q-table is exploited to store values of q that represent expected future rewards when those specific actions are taken from particular states. Thus, applying Q-learning to phishing detection leverages the adaptability and dynamic learning capabilities of reinforcement learning, offering a robust defense against evolving phishing tactics. After every episode, the Q-values are updated based on the output. The model remains effective even against zero-day phishing attacks.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

To summarize the above, we develop an RL agent that dynamically selects the features and classification model as a defense mechanism against phishing attacks. In particular, a variety of widely used classifiers, such as Random Forest Classifier, Decision Tree Algorithm, Gradient Boosting Algorithm and Logistic Regression are compared and chosen. In contrast to conventional static selection techniques, the suggested method dynamically modifies model selection according to how well the models perform across arbitrary batches of features. An attempt is made to identify the key components in each chosen classifier that have the greatest influence on model accuracy. Throughout this iterative process, a Q table is updated according to how well the models and features perform in each episode. Eventually, after several iterations, the best performing model and features are identified.

The significant contributions of this research are as follows:

(1) We have designed a reinforcement learning agent that selects the features and the classification model dynamically using a Q-learning algorithm.

(2) We have performed analysis on the size of the feature subset by testing the agent on different subset sizes to get the optimal subset size.

(3) We have tested the agent on the dataset and analysed their performances.

2. Related Work

The challenge of identification of phishing attack is a detection process that relies on the use of URL features, image-based analysis or static approach. As there is an increase in phishing attacks, the researchers to design a solution put in continuous efforts. Over the past years, various methodologies have been developed to detect phishing attacks. Static Phishing Detection, Visual Similarity-based detection, and Detection Models using Machine Learning algorithms are the various methodologies utilized for phishing detection. These section overviews the recent advances in phishing detection by discussing the literature studied.

During the literature study, it was observed that reinforcement learning was not widely used for phishing detection. Research [4,5,23,24] has used reinforcement learning for developing continuous learning solutions. Ariyadasa et al., in this work [4], presented an anti-phishing solution that combines innovative knowledge acquisition with integrated continuous learning support. Deep learning and reinforcement learning are combined by SmartiPhish to assess a webpage's popularity and make final decisions to identify phishing websites. As a result, a deep learning model is utilized to predict, using the URL and HTML features of a website, the likelihood that it is phishing, while the reinforcement-learning component makes the prediction based on the page's popularity and past information. The authors claim that SmartiPhish yields a 96.40% detection accuracy. They have also mentioned that the system shows an improvement of 5.65% over time.

Moitrayee Chatterjee et al. [5] presented a deep-learning-based Q network. This network adapts to the ever-changing phishing websites. As a result, the network can obtain the crucial features required to recognize these fraudulent websites with a 90.01% accuracy rate. The agents learn the value function from the input URLs to perform the classification. The phishing detection problem is converted to a sequential decision-making task using a deep neural network.

In [6] Mahdi Bahaghighat et al. have used multiple machine learning algorithms. The algorithms studied for URL detection in this approach were Support Vector Machine, K-Nearest Neighbours, Naive Bayes, Random Forest, and Extreme Gradient Boosting. They have pre-processed the dataset by applying SMOTE-ENN balancing algorithms and dropping the constant features. The comparative results of 6 classification models are stated in this paper. It demonstrates that the XGBoost classifier has the best overall performance, with a 99.2% total accuracy rate.

In [7], Qasem Abu Al-Haija et al. explore the range of machine learning algorithms to detect counterfeit websites by analyzing URL-based features. The study uses models such as Decision Trees, Random Forests, and Support Vector Machines. The findings indicate that ensemble methods like Random Forests are particularly effective, highlighting the use of machine learning methodologies to automate and improve the detection of phishing sites, thereby enhancing cybersecurity defences. The system was tested on a dataset of phishing websites, with the best model built using decision trees on a balanced dataset achieving a classification accuracy of 97.40%.

In [8] Asif Irshad Khan et al. described a novel method of preventing phishing attempts on social media users by utilizing artificial neural networks and multilayer Q-learning algorithms. The Logistic Bayesian Long Short-Term Memory model for the researchers to collect use malware analysis, pre-process, and analyse the URL data. The extracted observed malicious URL features ensure that the URL detection is implemented. The effectiveness of

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

the suggested anti-phishing module in shielding social media users from phishing attacks has been confirmed by its 94.33% detection accuracy.
Parvathapuram et al. proposed a method called SI-BBA [9] to use a cutting-edge deep learning technique based on swarm intelligence to identify and categorize phishing websites. A swarm intelligence approach was used to train the neural network. The authors to design the neural network, which categorizes the network URL websites, implemented the Swarm Intelligence Binary Bat Algorithm (SI-BBA). Their test results confirm the high degree of precision 94.8% with which the SI-BBA-based model detects phishing website attacks.

Lakshmanarao et al. [10] described a novel machine learning-based technique for identification of phishing websites. The authors have applied pre-processing techniques to remove null, missing values and outliers from the dataset. The Principal Component Analysis technique was used for Feature Extraction. They tested several classification techniques, including support vector machine, decision tree, random forest, AdaBoost, and gradient boosting. To improve the detection accuracy, they proposed two priority-based algorithms and used their results to determine the final classifier. The proposed model was evaluated based on the model's accuracy, which was 97%.

Saha et al. [11] proposed a system for detecting malicious URLs utilizing a deep learning-based approach to classify URLs. They have classified URLs into categories like phishing, legitimate, and suspicious websites. The model employs a multilayer perceptron technique for classification, which is trained on ten extracted features from given URLs of legitimate and phishing websites. These features include characteristics such as the keywords, URL length, and age of the domain. The system processes these features through multiple layers of the neural network to classify the websites. The system achieved an accuracy of 97.8% in detecting phishing attacks compared to other traditional machine-learning techniques. Table 1 summarizes the literature studied in this paper.

Table 1: Literature Survey Summary

Author	**Methodology Used**	**Future Scope**
Subhash Ariyadasa et al. [4]	Deep Learning and Reinforcement Learning	The RL agent gives the output based on the popularity statistics of the website and the deep learning architecture. The third party also decides reward function calculation.
Mahdi Bahaghighat et al. [6]	ML algorithm	Expanding the dataset and incorporating real-time detection capabilities.
Moitrayee Chatterjee et al. [5]	Multi-agent Reinforcement Learning	Optimizing the Markov Decision Process through performance tuning of the deep Q-learning algorithm. Feature Extraction, Evaluation and Ranking can be incorporated in the system.
Qasem Abu Al-Haija et al. [7]	Neural networks and decision trees.	Applying the model to a wider range of phishing scenarios and enhancing feature extraction techniques
Asif Irshad Khan et al. [8]	A multilayer Q-learning framework with CaspNet and swarm optimization.	Making the model adaptive to evolving threats, real-time across multiple platforms, and mobile-friendly.
Parvathapuram Pavan et al. [9]	Swarm Intelligence Binary Bat Algorithm (SI-BBA) and deep learning-based neural network for URL classification.	Refining hyperparameters, such as the number of epochs, learning rate, and batch size, to further improve the model’s accuracy.
Lakshmanarao et al. [10]	Machine learning models	Expanding the feature extraction process to improve model robustness against sophisticated phishing attacks.
Ishita Saha et al. [11]	Multilayer Perceptron (MLP) neural network	Adding more layers to the neural network and utilizing advanced architectures like backpropagation networks to improve phishing detection accuracy.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

Based on the literature studied it is observed that there is an inclination to develop phishing detection solutions using machine learning. Researchers have used ML algorithms to detect the attack by combining multiple algorithms and selecting the model with the highest results. Optimization algorithms were used like Swarm Intelligence to select features and then perform the classification task.

Few researchers have used deep reinforcement learning for phishing detection by using it along with neural networks. They have limited the use of the adaptability feature of reinforcement learning to the feature selection stage followed by a neural network for classification. The novel approach used in this research is the use of reinforcement learning agents in the selection of classification models and the adaptive selection of features. The approach of adaptive and random feature and classification algorithm selection ensures that the strategy used for detection is a black box for the attackers trying to bypass the defense system. Unlike the existing RL-based phishing detection methods, the RL agent based on the URL received for classification selects the features and classification algorithms in real time. The proposed methodology uses multiple performance metrics for evaluation and learning of the Reinforcement Learning Agent. The best feature subsets of each of the classification's algorithms are tracked during training of the RL Agent for improved decision-making for the agent.

3. Proposed System

A reinforcement-learning framework is proposed in this research for phishing detection. It is comprised of an agent, environment, policy function, Reward and Value function. The core elements of this framework are the agent and the environment. Policy, reward and the value function are the sub-elements [4]. The agent will learn its actions by interacting with the environment. In this interaction with the environment, the agent will take actions in the environment that will result in a change of state of the environment. The state change results from the action taken by the agent. Based on the outcome of this action, a reward is calculated. This reward helps the agent make better decisions in future interactions with the environment.

The reinforcement-learning framework of phishing URL classification consists of a Dataset, Pre-processing of that dataset and the RL Framework. The pre-processed dataset is loaded in the reinforcement-learning framework. The random feature subset is selected which is considered as the state of the phishing environment in the RL framework. After the selection of the feature subset, the next step is to take action by the agent. The actions that an agent takes in this RL framework for phishing detection are the selection of the classification model. Calculation of accuracy is done considering the model selected and the feature subset, which acts as a reward function for the agent. The agent then updates its Q-table to take future actions based on the learning. An overview of the proposed system is shown in Figure 2.

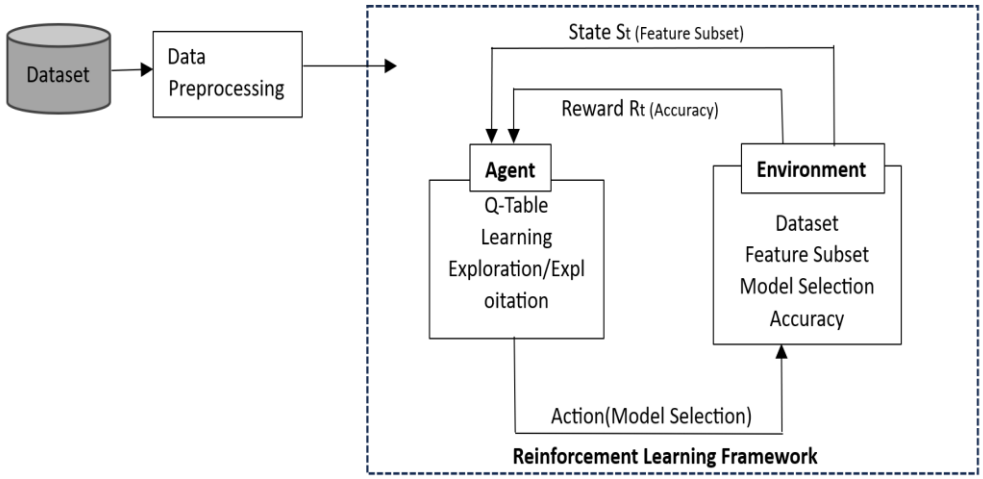

Figure 2. Overview of Reinforcement Learning Framework for Phishing Detection

Figure 3 elaborates on the flow of the stated architecture for phishing detection utilizing Q Learning for feature selection and model optimization.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

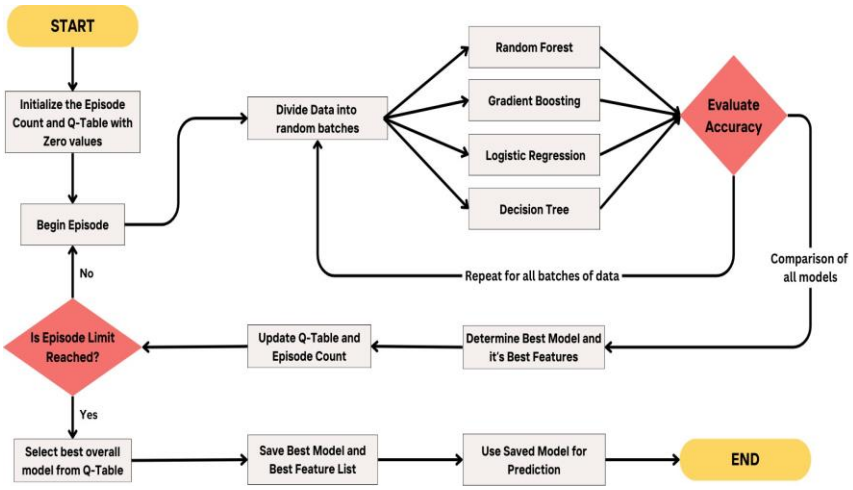

Figure 3. Flow of the Proposed System

3.1 Dataset

This study makes use of a dataset to perform testing from The Phishing Websites Dataset [11], which was compiled by Grega Vrbančič. It comprises two versions: a comprehensive set with 88,647 instances, including 58,000 genuine and 30,647 fake sites, and a smaller set with 58,645 instances, of which 27,998 are legitimate and 30,647 are phishing. 111 features, providing detailed attributes that are crucial for developing effective machine-learning models for phishing detection [12], describe each instance. The wide range 111 features make it suitable for optimal feature selection. The dataset has been used by multiple researches studied in the literature making it suitable for comparative study of the proposed approach. This dataset was chosen due to its size, diversity, and range of features. The dataset was generated using PhishTank for the phishing websites and for legitimated URLs are from the websites that are publicly available, community labeled and from the Alexa top ranking websites. The attributes in this dataset include feature categories like the URL Properties, Domain Properties, URL directory properties, file properties, parameter properties and resolving URL and external services.

3.2 Data Preprocessing

Initially, a comprehensive check was performed for missing values in the dataset. There were no missing data points identified. We employed an 80/20 train-test split on the original dataset to create separate training and testing sets.

3.3 Reinforcement Learning Framework

A framework developed using reinforcement learning gets trained using trial and error method. By receiving feedback from its actions in the form of positive or negative signals. These signals, either rewards or punishments, aim to maximize the overall reward function. By learning from its mistakes, RL provides artificial intelligence that closely resembles natural intelligence. This work proposes a reinforcement learning (RL) framework for optimizing feature selection and model choice in phishing detection. The problem is framed as a sequential decision-making process. The RL framework allows the agent to learn which features are most informative for distinguishing between legitimate and phishing websites and identifies which learning model performs best for the task of phishing detection [17]. Figure 4 shows a typical RL framework.

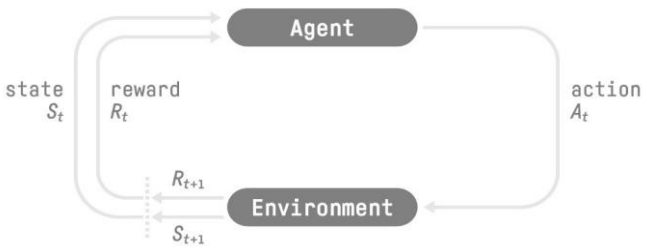

Figure 4. Reinforcement Learning Framework [19]

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

The RL framework for phishing detection has an agent that takes actions (ML model selection) depending on the present state (feature subset selected) of the RL environment. The components of the RL agent include the State Representation, Action Space, Policy Function, Q-Table, Reward Function and the Learning Process. Figure 5 shows the components of the agent designed in the proposed RL framework.

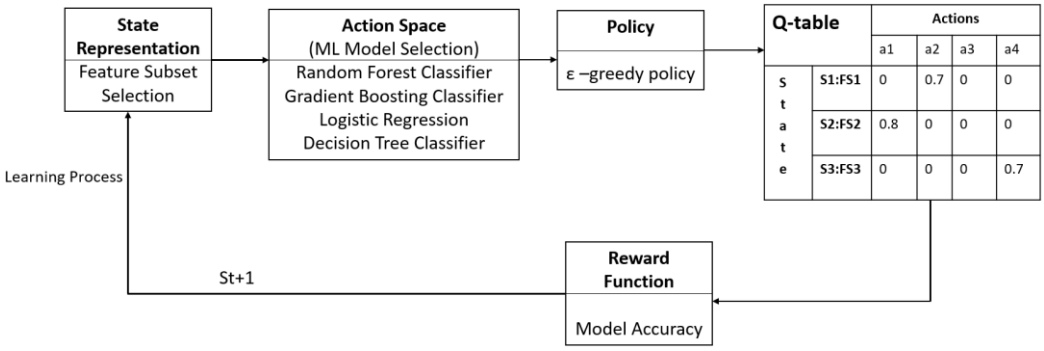

Figure 5. Components of Phishing Detection Agent

State Representation:

The state is represented by different subsets of features extracted from a dataset. These subsets are randomly selected from the dataset. Each subset of features serves as a state from which the agent selects actions to train and evaluate.

Actions: Actions correspond to the selection of machine learning models to be trained and evaluated on each subset of features.

Reward: The reward provides feedback to the agent about its actions in the environment. The reward is derived from the accuracy of machine learning models that are trained on different subsets of features. Higher accuracy corresponds to a higher reward, influencing the Q-value updates. This accuracy serves as feedback to the agent, guiding the update of Q-values in the Q-table through the Q-learning update rule. By maximizing rewards, the agent learns to select optimal models for detecting phishing websites effectively over multiple training episodes.

Policy: The policy dictates how the agent selection of the machine-learning model will be used based on the current feature subset state. The policy used is the epsilon-greedy. The value of Epsilon (ε) determines the probability of exploration vs. exploitation. During the exploration phase of the agent, it chooses a random action that is selection of ML model with a probability of ε, enabling it to discover potentially better options regardless of the current Q-values. While during Exploitation, it chooses the action with the highest Q-value for the current state, which is the feature subset that is chosen with a probability of (1 - ε), making use of its learned experience to maximize the reward.

To ensure exploration and exploitation epsilon is initialized to 1.0, meaning the agent will initially explore completely randomly. The Discount factor (γ) is set to 0.995, indicating that epsilon will decrease gradually over time, making the agent less exploratory and more exploitative as it learns. The learning rate (α) is set to 0.01, ensuring that even in the later stages of learning, the agent retains a small probability of exploring new actions. Between 0 and 1, a random number is generated. The agent explores by choosing a random action if this number is smaller than epsilon. In the absence of such a strategy, it selects the action that has the highest Q-value given the present condition. After each episode, the epsilon is decayed by multiplying it with the Discount factor. This approach gradually reduces the exploration rate as the agent gains more understanding of the environment. By beginning with a high exploration rate and then shifting towards exploitation, the agent effectively learns to choose the best models for detecting phishing websites based on feature subsets. At the same time, it still occasionally explores new possibilities to ensure thorough learning.

3.3.1 Q-Learning Algorithm for Phishing Detection

The Q-learning algorithm, a form of reinforcement learning, was utilized to optimize both feature selection and model performance for phishing detection. This algorithm operates by learning a policy that tells an agent which action to take under various circumstances. Specifically, it maintains a Q-table where each entry represents the

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

quality of a particular action in a given state, quantified by expected future rewards [17]. The proposed method initializes the Q-table with zeros and employs a decayed ε-greedy strategy to balance initial exploration and eventual exploitation of knowledge. The agent was trained in two phases. In phase one, the agent was trained to select features randomly. During each episode in this phase, the agent starts with a random set of features and iteratively performs actions to add or remove features. After each action, the corresponding phishing detection model was trained and evaluated, and the Q-value for the state-action pair was updated based on the observed reward. In phase 2 for model training, features were selected by the agent dynamically but using a sequential method. State Space and reward function used is the same as in the first phase.

The Q-table is updated using the following formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \quad (1)$$

This formula ensures that the Q-value is updated to reflect the expected future rewards from taking action a in state s. Figure 6 shows a general framework of the Q-Learning algorithm.

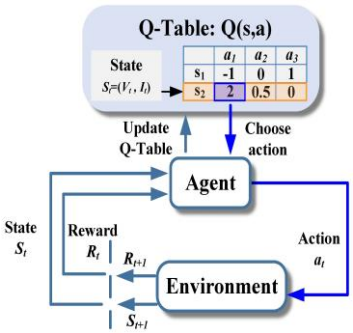

Figure 6. Working of Q-Learning Algorithm

The Q-table is a fundamental component used by the agent to make decisions based on experiences in various states. The agent learns from its experience of the reward that it has received based on the actions taken. The Q-table is a matrix where each row represents a state of the environment and each column represents an action taken by the agent. In the proposed RL framework, we have considered feature subset selection as a State of Environment. Actions taken by the RL agent are the selection of any one of the classification models. Q-table stores Q-values, which estimate the expected reward for each state-action pair. In this system reward function is equivalent to the accuracy achieved for the model selected by the agent. Q-values are updated using the Q-learning formula based on the accuracy of the model on the current subset of features. In the proposed system, the agent for a state where a random feature subset is selected can take four actions. The Q-values are updated during the training process based on the reward received for taking an action in a particular state. The process of updating the Q-value is explained in detail as follows:

1. Initially the Q-table is initialized to zeros. The learning rate (α) is set to 0.1, the discount factor (γ) is 0.95 and Epsilon (Exploration Rate) to 1.0.

Q-table	Action 0	Action 1	Action 2	Action 3
State 0	0	0	0	0
State 1	0	0	0	0

2. The agent starts in State 0(random feature subset) and chooses Action 0 (e.g., Random Forest Classifier).
3. Assuming that the environment in State 0 is a feature subset with columns 0 to 14 and the target column phishing. The agent trains the Random Forest Classifier on this subset and evaluates its performance. The accuracy obtained (reward) for Action 0 in State 0 is 0.75 (e.g., 75%).
4. The agent will move to the next state (State 1), which will correspond to the next feature subset (e.g., columns 15 to 29 and phishing).
5. Q-value will be calculated using the formula for State 0, Action 0.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

$$Q(s,a) = Q(s,a) + \propto [r + \gamma \max_{a'} Q(s',a,) - q(s,a)$$

Q (s, a) represents the new value.

Q (s, a) = 0 + 0.1*[(0.75 + 0.95 * 0)- 0]
Q (s, a) = 0 + 0.1*[0.75 – 0]
Q (s, a) = 0 + 0.075
Q (s, a) = 0.075

6. New Q table.

	Action 0	Action 1	Action 2	Action 3
State 0	0.075	0	0	0
State 1	0	0	0	0

7. The above steps are repeated for 100 episodes to update Q-values of different feature subsets and classification models based on the rewards calculated. The Q-table eventually guides the agent in selecting the best models for different feature subsets to maximize the detection accuracy of phishing websites.

The agent iterates through multiple episodes where it selects a subset of features, trains and evaluates each model on the selected subset, Updates Q-values based on the observed accuracy and identifies the best model and the feature subset. Episodes are terminated upon reaching the maximum number of steps. Following the completion of the training phase, the optimal state-action values were identified within the Q-table. This identification was achieved through the selection of state-action pairs exhibiting the highest Q-values. These state-action pairs, corresponding to the most effective feature combinations for phishing detection, were then utilized to train final models. The subsequent evaluation of these final models was conducted on a test subset, specifically designed to assess their generalization performance. Q-Learning algorithm dynamically steered the overall feature and model selection process by continuously checking the effective and ineffective feature combinations and model selections through performance evaluations. The combination of features and classifiers in the Q-learning by this agent was to bring up the best possible detection, such as the 15 top features that provided the highest accuracy with the chosen model. Through an adaptive approach, selected features and models were acquired, providing not only proficiency in detecting phishing websites but also adaptation to growing patterns in data; this will enhance the overall robustness and effectiveness of the phishing detection system [19].

4. Results and Discussion

Features play an important role in phishing detection. Therefore, experiments were conducted with feature subset sizes varying from 10 to 15 features. The RL agent was designed to detect phishing on feature subsets with sizes between 10 to 15. We have identified the best feature subset size, feature subset and the classification model from these experiments. In this section, we will discuss our results in detail.

4.1. Evaluation Metrics

The employed evaluation metrics encompassed accuracy, precision, recall, and F1-score, providing a comprehensive assessment of model efficacy in real-world scenarios.

$$Precision = \frac{TP}{TP + FP} \text{ [15]} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \text{ [15]} \quad (3)$$

$$f1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \text{[15]} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \text{ [15]} \quad (5)$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives [15]. The experimental results of the reinforcement-learning agent using the dataset for a feature subset size equal to 10 are shown in Table 2. The highest accuracy obtained with a feature subset size of 10 is for Random Forest Classifier with 98.98%.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

Table 2: Performance of RL Agent for Feature Subset Size=10

Model	Feature Subset Selected(k=10)	Accuracy
Random Forest	"domain_length", "time_response", "qty_dot_params", "qty_vowels_domain", "qty_tld_url","ttl_hostname","time_domain_activation","qty_slash_url", "qty_space_url", "qty_asterisk_params"	98.98
Decision Tree	""qty_at_url", "qty_equal_params", "asn_ip", "time_domain_activation", "qty_exclamation_file", "qty_dot_url", "directory_length", "qty_asterisk_domain", "ttl_hostname", "url_shortened".	98.68
Gradient Boosting	"qty_exclamation_domain", "asn_ip", "time_response", "qty_hashtag_domain", "time_domain_activation", "qty_space_params", "qty_underline_url, "directory_length", "domain_google_index", "qty_comma_domain"	94.20
Logistic Regression	"qty_questionmark_params", "qty_dot_file", "qty_percent_url", "qty_redirects", "qty_slash_directory", "params_length", "qty_comma_file", "time_domain_activation", "qty_tilde_url", "length_url"	91.27

The RL Agent was trained and tested with feature subset size 11. In this test case also Random Forest Classifier gave an accuracy of 99.12% which was better compared to the other classification algorithms. The feature subset selected by this agent is similar to the previous agent with a feature subset size of 11. Table 3 gives the list of features selected by the RL agent when the subset size was set to 11.

Table 3: Performance of RL Agent for Feature Subset Size=11

Model	Feature Subset Selected(k=11)	Accuracy
Random Forest	"directory_length", "domain_length", "qty_tld_url", "qty_questionmark_params", "server_client_domain", "qty_percent_directory", "asn_ip", "qty_nameservers", "qty_comma_domain", "time_domain_activation", "qty_mx_servers"	99.12
Decision Tree	"email_in_url", "qty_and_params", "domain_length", "time_domain_activation", "time_response", "qty_slash_params", "qty_mx_servers", "directory_length", "qty_space_url", "qty_hyphen_file", "qty_at_url"	98.66
Gradient Boosting	"qty_dot_file", "qty_dollar_url", "qty_questionmark_params", "directory_length", "qty_slash_domain", "qty_hyphen_url", "qty_percent_url", "asn_ip", "time_domain_activation", "qty_dollar_file", "qty_space_directory"	94.21
Logistic Regression	"qty_dollar_params", "qty_exclamation_file", "qty_tld_url", "qty_slash_directory", "qty_exclamation_directory", "time_domain_activation", "tls_ssl_certificate", "domain_in_ip", "qty_mx_servers", "url_shortened", "qty_comma_directory"	91.36

In the next phase, the subset size was increased to 12 and the agent was trained to select the feature subset of size 12. This experimentation also resulted in the Random Forest Classifier as the top model with an accuracy of 99.23 %. The details of features selected in this experiment are listed in Table 4.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

Table 4: Performance of RL Agent for Feature Subset Size=12

Model	Feature Subset Selected(k=12)	Accuracy
Random Forest	"qty_hyphen_file", "domain_length", "qty_equal_params", "domain_google_index", "directory_length", "qty_slash_url", "time_domain_activation", "qty_percent_file", "qty_asterisk_directory", "qty_underline_params", "asn_ip", "ttl_hostname"	99.23
Decision Tree	"qty_dot_params", "asn_ip", "qty_asterisk_url", "qty_underline_url", "directory_length", "server_client_domain", "ttl_hostname", "qty_dollar_params", "time_domain_activation", "qty_slash_url", "domain_length", "qty_dollar_file"	98.87
Gradient Boosting	"tls_ssl_certificate", "url_google_index", "server_client_domain", "directory_length", "qty_dot_domain", "asn_ip", "domain_spf", "time_domain_activation", "qty_comma_url", "qty_slash_directory", "qty_percent_params", "qty_comma_domain"	94.59
Logistic Regression	"qty_exclamation_domain", "qty_plus_domain", "qty_tilde_url", "qty_plus_params", "qty_space_domain", "time_domain_activation", "length_url", "qty_comma_directory", "qty_hyphen_directory", "directory_length", "qty_underline_domain", "qty_hashtag_domain"	91.30

Random Forest Classifier outperformed the other classifier models with a feature subset of size 13. The total accuracy achieved with this subset size was 99.12%. The feature subsets selected for various models and their accuracy are given in Table 5. The Gradient Boosting Classifier algorithm attained an accuracy of 94.84 %, the Logistic Regression Model with an accuracy of 91.30 and the Decision Tree Model reached an accuracy of 98.72 %.

Table 5: Performance of RL Agent for Feature Subset Size=13

Model	Feature Subset Selected(k=13)	Accuracy
Random Forest	"qty_vowels_domain", "asn_ip", "qty_dollar_url", "qty_slash_params", "time_domain_activation", "qty_exclamation_file", "length_url", "qty_ip_resolved", "domain_spf", "qty_underline_directory", "qty_plus_params", "ttl_hostname", "qty_questionmark_file"	99.12
Decision Tree	"directory_length", "asn_ip", "qty_space_params", "time_domain_activation", "qty_and_directory", "qty_exclamation_domain", "qty_slash_url", "email_in_url", "domain_spf", "qty_slash_file", "server_client_domain", "qty_equal_params", "domain_length"	98.72
Gradient Boosting	"qty_plus_file", "qty_comma_params", "qty_comma_file", "qty_exclamation_file", "qty_underline_url", "time_domain_activation", "qty_tilde_file", "qty_hyphen_directory", "qty_ip_resolved", "asn_ip", "qty_tilde_directory", "qty_dot_domain", "directory_length"	94.84
Logistic Regression	"qty_hashtag_url", "qty_questionmark_url", "qty_redirects", "ttl_hostname", "domain_in_ip", "qty_dot_url", "length_url", "qty_slash_url", "qty_underline_params", "qty_hashtag_params", "qty_dot_file", "time_domain_activation", "tld_present_params"	91.20

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

The experimental results for the agent with a feature subset size equal to 14 show that the Random Forest Classifier achieved an accuracy of 99.25 %. The decision Tree classifier also achieved an accuracy of 98.74% similar to the Random Forest Classifier. Table 6 gives an overview of the outcome of the RL agent.

Table 6: Performance of RL Agent for Feature Subset Size=14

Model	Feature Subset Selected(k=14)	Accuracy
Random Forest	"qty_redirects", "qty_percent_params", "time_domain_activation", "qty_mx_servers", "qty_hyphen_directory", "domain_in_ip", "qty_dot_params", "qty_at_url", "domain_google_index", "qty_space_file", "asn_ip", "length_url", "directory_length", "time_response"	99.25
Decision Tree	"qty_dot_url", "qty_underline_url", "qty_slash_directory", "qty_plus_domain", "qty_at_domain", "time_domain_activation", "domain_length", "qty_and_url", "qty_slash_params", "file_length", "qty_hashtag_directory", "asn_ip", "time_domain_expiration", "qty_asterisk_domain"	98.74
Gradient Boosting	"qty_dot_file", "time_domain_activation", "tls_ssl_certificate", "qty_ip_resolved", "qty_comma_directory", "qty_at_file", "asn_ip", "qty_asterisk_params", "qty_tld_url", "directory_length", "qty_at_url", "qty_dot_domain", "qty_dollar_params", "qty_equal_domain"	94.76
Logistic Regression	"qty_asterisk_params", "length_url", "qty_slash_params", "file_length", "tls_ssl_certificate", "qty_slash_url", "qty_percent_domain", "qty_tilde_url", "qty_exclamation_domain", "qty_dot_domain", "time_domain_activation", "qty_ip_resolved", "qty_dot_file", "qty_equal_url"	91.80

In the last stage, the RL agent was trained to select features with a subset size of 15. In this stage, also Random Forest has outclassed all the other classification algorithms with an accuracy of 99.15 %. It was observed that the majority of the features selected by the agent were similar to the features selected by the agent with a feature subset size of less than 15. Table 7 gives the detailed results of the RL agent with a subset size equal to 15.

Table 7: Performance of RL Agent for Feature Subset Size=15

Model	Feature Subset Selected(k=15)	Accuracy
Random Forest	"qty_underline_directory", "qty_percent_file", "time_domain_activation", "qty_plus_file", "qty_redirects", "qty_hashtag_params", "time_response", "qty_percent_url", "qty_at_params", "directory_length", "qty_tilde_file", "asn_ip", "qty_dot_file", "qty_hyphen_url", "qty_plus_params"	99.15
Decision Tree	"qty_dollar_domain", "qty_slash_file", "time_domain_activation", "qty_and_params", "asn_ip", "qty_exclamation_directory", "directory_length", "qty_at_file", "qty_percent_file", "qty_slash_directory", "qty_plus_file", "qty_underline_url", "ttl_hostname", "domain_length", "qty_dollar_url"	98.89
Gradient Boosting	"qty_equal_domain", "time_response", "tls_ssl_certificate", "directory_length", "qty_tilde_file", "file_length", "asn_ip", "domain_length", "qty_underline_file", "email_in_url", "qty_dot_domain", "time_domain_activation", "qty_asterisk_url", "qty_dollar_url", "params_length"	95.10
Logistic Regression	"qty_equal_file", "domain_length", "time_domain_activation", "qty_redirects", "tls_ssl_certificate", "qty_percent_url", "params_length", "qty_slash_directory", "qty_dot_domain", "qty_vowels_domain", "qty_questionmark_domain", "qty_questionmark_url", "qty_and_file", "qty_nameservers", "time_response"	91.68

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

Comparative Performance Analysis

Experiments were performed using different approaches to feature selection. The proposed system was tested with sequential subset creation of feature subset. During this experimentation, it was observed that accuracy for subset sizes 10 to 14 gave results in the range of 89 to 92%. But with the feature subset of size 15, the accuracy increased drastically to 98.63% for the Random Forest Classifier. Figure 7 gives an overview of the accuracy of sequential feature selection.

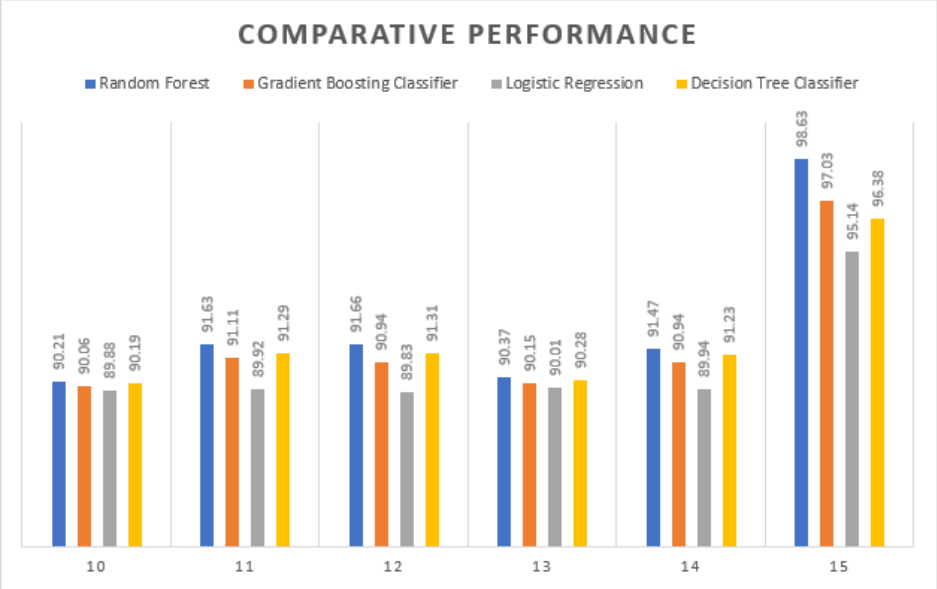

Figure 7. Comparative Performance of RL Agents for Different Feature Subset Size using Sequential Feature Selection.

While experimenting with the system with random feature subsets, it was observed that for different subset sizes, the Random Forest Classifier showed the best performance. The best accuracy achieved was 99.25 % for a feature subset of size 14. Thus, it can be concluded that an increase in the size of the feature subset does not have a significant improvement in the accuracy of the classifier. Figure 8 show the graphical summary of the performance of the RL agent for different subset size with random feature selection.

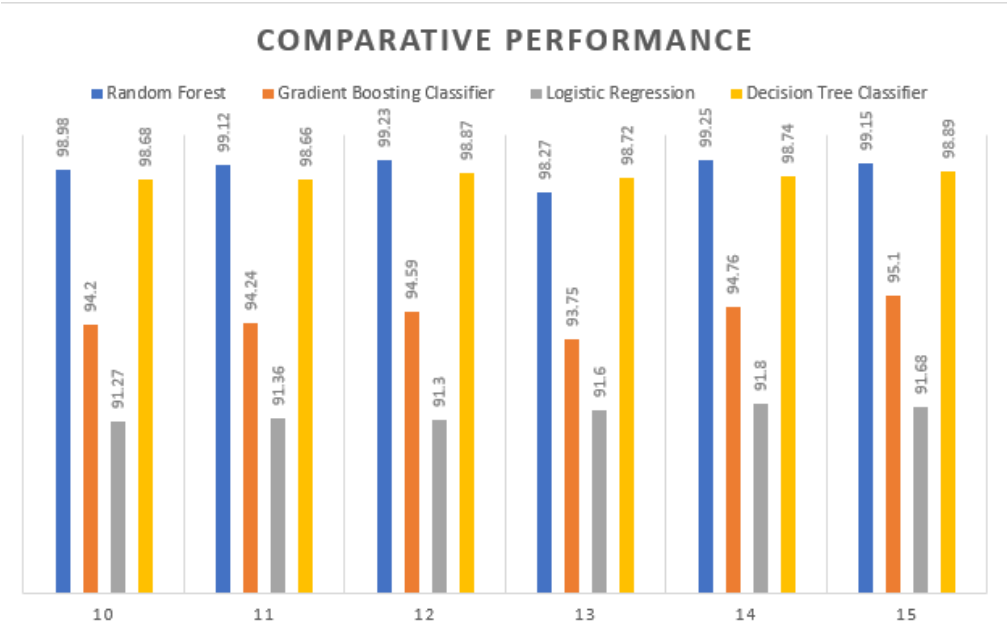

Figure 8. Comparative Performance of RL Agents for Different Feature Subset Size. (Random Selection)

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

The proposed system with random feature subset generation and pre-processed dataset was tested for different sizes of feature subset. During this experiment, it was observed that the random forest classifier outperformed the other models with an accuracy of 96.9%. The comparative analysis of this experiment is summarized in Figure 9.

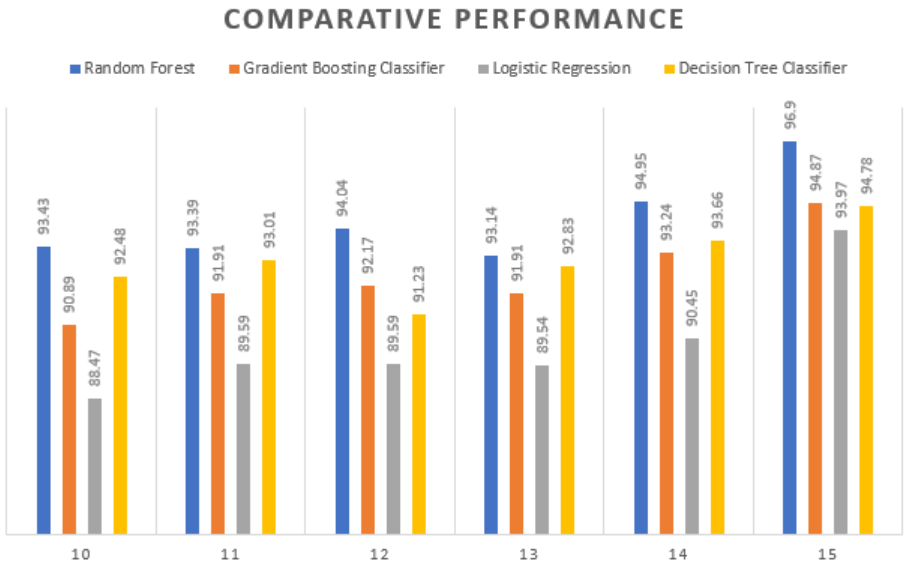

Figure 9. Comparative Performance of RL Agents for Different Feature Subset Size using Pre-processed Dataset.

Further experiments were performed by tuning the RL agent with Random Feature Subsets. The parameter tuned during this experiment was epsilon was set to 1.5 for more exploration-learning rate (α) is set to 0.05 to ensure exploration continues discount factor (γ) is set to 0.999 with the motive that slower the decay rate for a more gradual decrease in exploration. Figure 10 gives an overview of the results of this change in settings for the RL Agent with random feature subset selection for a subset size ranging from 10 to 15. The highest accuracy achieved in this setup was 93.24 by the Decision tree classifier for the subset size of 15.
Considering all the results it was observed that sequential feature subset selection using RL Agent gave the best accuracy of 98.64% for Random Forest Classifier with feature subset size of 15.

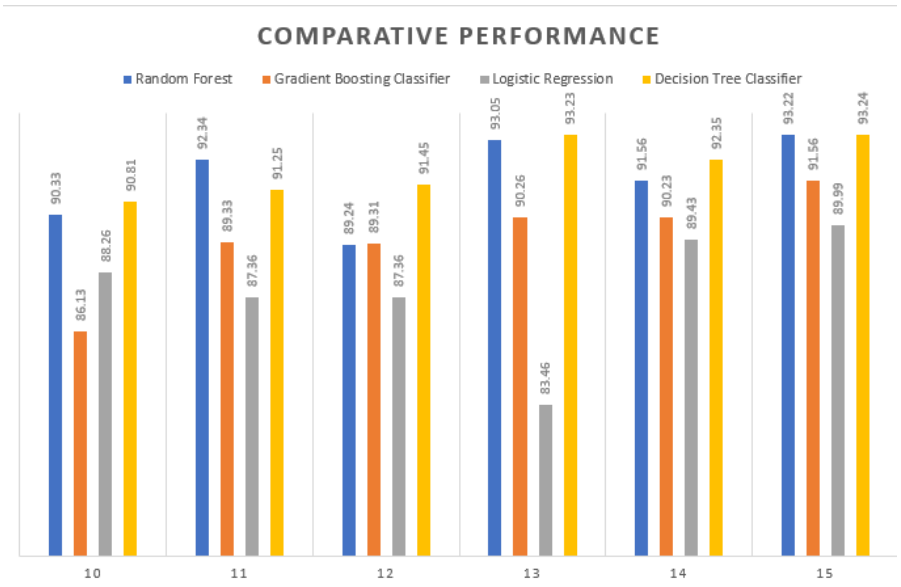

Figure 10. Comparative Performance of Tuned RL Agents for Different Feature Subset Size.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

Figure 11 gives the overview of the training time of the RL agents for features subsets ranging from size 10 to 15. Fluctuations are observed in training times with significant rise from 10 to 15. Maximum training time was observed at subset size11 where the time spikes to 27465 seconds. There is no linear relationship between the feature subset and the training time.

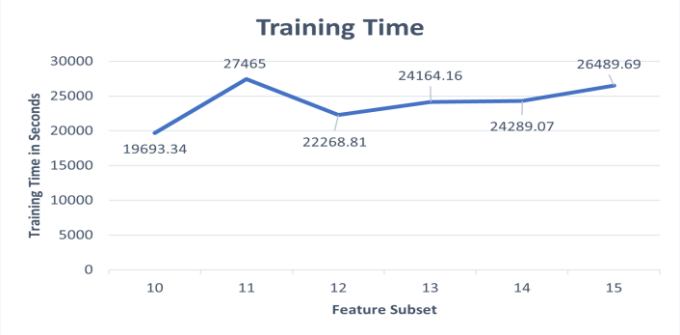

Figure 11. Training Time of the RL Agent ranging from Feature Subset Size 10 to 15

Table 8 comprehensively compares various phishing detection models, highlighting their employed algorithms, dataset descriptions, key features, and accuracy metrics. Notably, our study demonstrates the effectiveness of employing Q-Learning alongside feature and model selection techniques, achieving a remarkable accuracy of 99.25% using the Random Forest Classifier on the Phishing Websites Dataset. This performance surpasses that of other state-of-the-art models referenced in the literature, including those utilising deep learning architectures, swarm intelligence algorithms, and ensemble methods.

Table 8: Comparison of Phishing Detection Models

Paper Reference	Key Features Selected	Accuracy
Proposed Approach	**domain_length, qty_vowels_domain, directory-related attributes**	**99.25%**
[4]	URL and HTML features, webpage popularity	96.40%
[5]	Dynamically Evolving phishing features	90.01%
[6]	Various URLs and content attributes	>93%, XGBoost: 99.2%
[7]	URL patterns	97.40%
[8]	URL features,LB-LSTM analysis	94.33%
[9]	Swarm intelligence-based deep learning	94.8%
[10]	Priority-based feature selection	97%
[11]	Deep learning architecture	97.8%
[13]	URL and webpage content features	97.36%

Specifically, our model outperforms competing approaches such as SI-BBA, deep learning Q networks, and neural network-based systems, displaying its robustness in identifying phishing attacks. Furthermore, our model's superiority is evident in comparison to traditional machine learning techniques such as logistic regression, k-nearest neighbours, and naive Bayes. These results underscore the efficacy of our proposed methodology in enhancing phishing detection accuracy, thus offering significant advancements in cybersecurity research and practical applications. Figure 12 shows that the proposed adaptive model when applied to varying feature subset sizes demonstrates high predictive accuracy.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

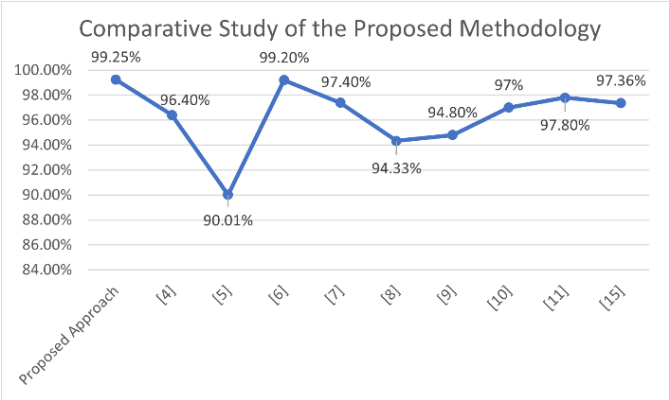

Figure 12. Comparative Performance of the Proposed Methodology

5. Conclusion

In this study, we developed a comprehensive phishing detection system leveraging a combination of machine learning algorithms and reinforcement learning techniques to enhance feature selection and model performance. The system demonstrated high efficacy in identifying phishing websites, achieving notable performance metrics across multiple classifiers. The Random Forest Classifier emerged as the most effective model, attaining the highest accuracy (99.25%) for the dataset used. A significant advantage of the proposed system is its capability to handle dynamic datasets. In real-world situations where phishing methods are always changing, the Q-Learning approach makes the model highly effective by enabling it to adapt to new and varied data patterns. The system's dynamic adaptation guarantees its continued relevance and efficacy, offering a strong defense against new phishing attacks. This study highlights the power of integrating machine learning with reinforcement learning to create effective and adaptive phishing detection solutions.

To increase detection accuracy and adjust to novel phishing techniques, future research could investigate the integration of extra features and sophisticated learning algorithms. The system can be improvised by implementing the methodology using two agents where one agent is responsible for feature selection and the other can used for classification algorithm selection. In the proposed methodology the though the agent is trained on subset sizes ranging from 10 to 15 but in future the subset size can be selected dynamically in real time by the agent and the agent can be trained to learn the optimal feature subset size.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

Data and Code availability: "Data and code will be made available on request."

References

[1] D. Desai and R. Hegde, "Phishing Attacks Rise 58% in the Year of AI: ThreatLabz 2024 Phishing Report," 2024. [Online]. Available: https://www.zscaler.com/blogs/security-research/phishing-attacks-rise-58-year-aithreatlabz-2024-phishing-report

[2] APWG, "APWG Q4 Report Finds 2023 Was Record Year for Phishing," 2023. [Online]. Available: https://apwg.org/apwg-q4-report-finds-2023-was-record-year-for-phishing/. [Accessed: Jul. 08, 2024].

[3] Google, "Safe Browsing," Google Developers, 2024. [Online]. Available: https://developers.google.com/safe-browsing. [Accessed: Jul. 08, 2024].

[4] S. Ariyadasa, S. Fernando, and S. Fernando, "SmartiPhish: a reinforcement learning-based intelligent anti-phishing solution to detect spoofed website attacks," *Int. J. Inf. Security*, vol. 23, pp. 1055–1076, 2024, doi: 10.1007/s10207-023-00778-9.

[5] M. Chatterjee and A. S. Namin, "Deep Reinforcement Learning for Detecting Malicious Websites," *arXiv preprint*, arXiv: 1905.09207, 2019.

[6] M. Bahaghighat, M. Ghasemi, and F. Ozen, "A high-accuracy phishing website detection method based on machine learning," *J. Inf. Security Appl.*, vol. 77, p. 103553, 2023, doi: 10.1016/j.jisa.2023.103553.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025

[7] Q. Abu Al-Haija and A. Al Badawi, "URL-based Phishing Websites Detection via Machine Learning," in *Proc. Int. Conf. Data Analytics Bus. Ind. (ICDABI)*, 2021, pp. 644–649, doi: 10.1109/ICDABI53623.2021.9655851.

[8] I. Khan and B. Unhelkar, "An Enhanced Anti-Phishing Technique for Social Media Users: A Multilayer Q-Learning Approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 1, 2024, doi: 10.14569/IJACSA.2024.0150103.

[9] P. P. Kumar, T. Jaya, and V. Rajendran, "SI-BBA – A novel phishing website detection based on Swarm intelligence with deep learning," *Mater. Today, Proc.*, vol. 80, Part 3, pp. 3129–3139, 2023, doi: 10.1016/j.matpr.2021.07.178.

[10] Lakshmanarao, P. S. P. Rao, and M. M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, 2021, pp. 1164–1169, doi: 10.1109/ICAIS50930.2021.9395810.

[11] Saha *et al.*, "Phishing Attacks Detection using Deep Learning Approach," in *Proc. Third Int. Conf. Smart Syst. Invent. Technol. (ICSSIT)*, 2020, pp. 1180–1185, doi: 10.1109/ICSSIT48917.2020.9214132.

[12] G. Vrbančič, "Phishing Websites Dataset," 2020. [Online]. Available: https://doi.org/10.17632/72ptz43s9v.1

[13] Subasi *et al.*, "Intelligent phishing website detection using random forest classifier," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, 2017, pp. 1–5, doi: 10.1109/ICECTA.2017.8252051.

[14] G. Barto, "Reinforcement learning," in *Neural Systems for Control*. Academic Press, 1997, pp. 7-30.

[15] Google for Developers, "Foundational Courses," Machine Learning Crash Course. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification

[16] J. Doshi *et al.*, "A comprehensive dual-layer architecture for phishing and spam email detection," *Comput. Secur.*, vol. 133, p. 103378, 2023.

[17] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88-102, 2018.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237-285, 1996.

[19] E. Cengiz and M. Gök, "Reinforcement learning applications in cyber security: A review," *Sakarya Univ. J. Sci.*, vol. 27, no. 2, pp. 481-503, 2023.

[20] A. Hammad *et al.*, "Deep Reinforcement Learning for Adaptive Cyber Defense in Network Security," in *Proc. Cognit. Models Artif. Intell. Conf.*, 2024.

[21] H. Kamal *et al.*, "Reinforcement learning model for detecting phishing websites," in *Cybersecurity and Artificial Intelligence: Transformational Strategies and Disruptive Innovation*. Cham: Springer Nature, 2024, pp. 309-326.

[22] Alzahrani, M. S. Alhassan, and M. A. Alzahrani, "A survey on phishing detection techniques: Challenges and future directions," *J. Inf. Security Appl.*, vol. 66, no. 1, p. 102756, 2023, doi: 10.1016/j.jisa.2023.102756.

[23] K. Liu, Y. Fu, L. Wu, X. Li, C. Aggarwal, and H. Xiong, "Automated feature selection: A reinforcement learning perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2272-2284, 2021.

[24] Maci, A. Santorsola, A. Coscia, and A. Iannacone, "Unbalanced web phishing classification through deep reinforcement learning," *Computers*, vol. 12, no. 6, p. 118, 2023.

DOI: https://doi.org/10.54216/JCIM.170215
Received: April 16, 2025 Revised: June 27, 2025 Accepted: August 26, 2025