Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050815

ARTICLE

Optimized Phishing Detection with Recurrent Neural Network and Whale Optimizer Algorithm

Brij Bhooshan Gupta[1,2,3,*], Akshat Gaurav[4], Razaz Waheeb Attar[5], Varsha Arya[6,7], Ahmed Alhomoud[8] and Kwok Tai Chui[9]

[1]Department of Computer Science and Information Engineering, Asia University, Taichung, 413, Taiwan

[2]Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, Maharashtra, 411057, India

[3]Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, 248007, India

[4]Ronin Institute, Montclair, NJ 07043, USA

[5]Management Department, College of Business Administration, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[6]Department of Business Administration, Asia University, Taichung, 413, Taiwan

[7]Department of Electrical and Computer Engineering, Lebanese American University, Beirut, 1102, Lebanon

[8]Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha, 91911, Saudi Arabia

[9]Department of Electronic Engineering and Computer Science, Hong Kong Metropolitan University (HKMU), Homantin, Hong Kong, China

*Corresponding Author: Brij Bhooshan Gupta. Email: bbgupta@asia.edu.tw

Received: 19 February 2024 Accepted: 13 June 2024 Published: 12 September 2024

ABSTRACT

Phishing attacks present a persistent and evolving threat in the cybersecurity land-scape, necessitating the development of more sophisticated detection methods. Traditional machine learning approaches to phishing detection have relied heavily on feature engineering and have often fallen short in adapting to the dynamically changing patterns of phishing Uniform Resource Locator (URLs). Addressing these challenge, we introduce a framework that integrates the sequential data processing strengths of a Recurrent Neural Network (RNN) with the hyperparameter optimization prowess of the Whale Optimization Algorithm (WOA). Our model capitalizes on an extensive Kaggle dataset, featuring over 11,000 URLs, each delineated by 30 attributes. The WOA's hyperparameter optimization enhances the RNN's performance, evidenced by a meticulous validation process. The results, encapsulated in precision, recall, and F1-score metrics, surpass baseline models, achieving an overall accuracy of 92%. This study not only demonstrates the RNN's proficiency in learning complex patterns but also underscores the WOA's effectiveness in refining machine learning models for the critical task of phishing detection.

KEYWORDS

Phishing detection; Recurrent Neural Network (RNN); Whale Optimization Algorithm (WOA); cybersecurity; machine learning optimization

Copyright © 2024 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Phishing attacks have a significant historical context that dates back to the mid-1990s. The term "phishing" was first coined in 1997, signifying a new form of cyber fraud [1,2]. These attacks involve luring individuals to fake websites that mimic legitimate ones, aiming to deceive users into providing sensitive information. Over time, phishing attacks have evolved from rudimentary email scams to sophisticated social engineering tactics and technological integrations [3,4]. These attacks have shifted from generic fraud attempts to more targeted schemes based on trends, facts, and opportunities [5].

The impact of phishing attacks has been substantial, with many victims suffering harm due to these malicious activities [6,7]. Phishing is recognized as a prevalent cybersecurity threat, leading to identity theft and financial losses for individuals and organizations [8,9]. The attackers behind these schemes have continuously adapted their methods, targeting not just the systems but also exploiting human vulnerabilities [10].

Efforts to combat phishing attacks have led to the development of various detection and prevention techniques. Machine learning algorithms, such as ensemble learning and deep learning neural networks, have been employed to enhance phishing detection capabilities [11,12]. Additionally, security indicators and authentication measures have been explored to protect users from falling victim to phishing attempts [13,14].

As phishing attacks continue to pose a severe cybersecurity problem, organizations and researchers are actively working on improving defences against these threats. Intelligent anti phishing frameworks and hybrid classification approaches have been proposed to bolster detection mechanisms [15,16]. Furthermore, the application of lightweight phishing detection sensors driven by deep learning highlights the ongoing efforts to enhance security measures against evolving phishing tactics [17].

Phishing attacks significantly impact individuals and organizations, leading to financial losses, compromised data, and reduced trust in online communications. These attacks constantly increase, causing substantial economic losses to individuals and organizations [18,19]. During a phishing attack, cybercriminals present themselves as trusted entities, tricking individuals into divulging sensitive information such as personally identifiable information, banking details, and passwords [20,21]. Phishing attacks can compromise individuals and enterprises through social interaction alone, making them a serious threat to both parties [22,23].

The impact of phishing attacks on organizations is substantial, with approximately 65% of organizations in the United States falling victim to successful phishing attacks [24,25]. These attacks can lead to financial losses, reduced work efficiency, abuse of network resources, and the spread of malware, including viruses, worms, and trojans [26]. Moreover, phishing techniques have been adopted by advanced persistent threat (APT) groups to target high-profile organizations, leading to severe consequences such as data breaches and reputational damage [27]. Individuals are also significantly affected by phishing attacks, with the vulnerability to these attacks increasing, particularly among the younger generation [28]. Phishing attacks target individuals' websites, cloud storage sites, and government websites, leading to compromised personal and sensitive information [29]. Personality traits have been found to correlate with phishing susceptibility, which highlights the need for new methods to protect individuals from such attacks [30]. The impact of phishing attacks extends beyond financial losses and compromised data. These attacks also lead to a loss of trust in online communications, as phishing emails seriously threaten individuals and organizations [31]. The Anti-Phishing Work Group (APWG) reported a significant increase in phishing attacks, with an average of more than 92,500 phishing attacks per month in the fourth quarter of 2016 [18]. Furthermore, phishing

attacks capitalize on human errors and target vulnerabilities, leading to the exploitation of sensitive data and compromised systems [32].

Phishing attacks profoundly impact individuals and organizations, leading to financial losses, compromised data, and a loss of trust in online communications. These attacks target vulnerabilities and human errors, making them a significant threat in the digital age. Individuals and organizations need to be aware of the evolving nature of phishing attacks and implement effective countermeasures to mitigate their impact.

2 Related Work

The detection of phishing attacks has been a critical area of research, leading to the proposal of various frameworks and models aimed at effectively identifying and mitigating these cyber threats, as represented in Table 1. These frameworks leverage diverse techniques such as machine learning, deep learning, and ensemble methods to enhance the accuracy and efficiency of phishing attack detection.

Shin et al. [27] proposed a framework based on the ATT&CK Matrix, which aids organizations in countering malicious threats posed by advanced persistent threat (APT) groups. This framework contributes significantly to private and public sectors, providing a robust defence mechanism against sophisticated phishing campaigns.

Thakur et al. [33] introduced a comprehensive model for characterizing the behaviour of phishing attacks. They proposed a new framework for describing awareness, measurement, and defence strategies against phishing-based attacks. This framework aims to enhance the understanding of phishing attacks and improve the defence mechanisms against them. Smadi et al. [34] presented a novel framework that combines neural networks with reinforcement learning to detect phishing attacks in real-time, marking a significant advancement in online phishing email detection. This dynamic, evolving neural network-based framework represents a pioneering approach to combat phishing attacks. Mao et al. [35] proposed a framework for phishing page detection using machine learning classifiers, mainly focusing on protecting Android mobile devices from new phishing activities. This framework leverages a machine learning detection engine to identify and prevent phishing attempts on mobile platforms effectively.

Sumathi et al. [36] introduced a Hybrid Ensemble Feature Selection (HEFS) framework for machine learning-based phishing detection systems, aiming to enhance the accuracy and robustness of phishing attack detection. This framework represents a significant advancement in applying machine learning techniques to combat phishing threats. Aassal et al. [37] developed a benchmarking framework called 'PhishBench,' enabling the systematic evaluation and comparison of existing features for phishing detection under identical experimental conditions. This benchmarking framework provides a standardised approach to assess the effectiveness of various phishing detection techniques. Asiri et al. [38] surveyed intelligent detection designs of HTML URL phishing attacks, reviewing a set of frameworks such as web browser extensions and phone applications for detecting phishing attacks. This survey provides valuable insights into the diverse frameworks for detecting HTML URL phishing attacks.

Catal et al. [39] conducted a systematic literature review on deep learning applications for phishing detection, revealing that Keras and TensorFlow were the most preferred deep learning frameworks. However, 46% of the articles did not mention any specific framework, indicating the diversity of approaches in the field. Anjali et al. [40] proposed a deep learning model supported by 1D CNN for detecting phishing websites, showcasing the potential of deep learning techniques in effectively

identifying malicious websites. Almousa et al. [41]. Pursued the development of parsimonious deep learning models and hyperparameter optimization to achieve high accuracy and reproducible results for phishing website detection, highlighting the importance of optimizing deep learning models for effective detection. Al-Ahmadi et al. [42] introduced a deep learning approach called PUCNN, which solely depended on the website URL, demonstrating the potential of URL-based deep learning models for phishing detection.

Ozcan et al. [43] proposed a hybrid DNN-LSTM model for detecting phishing URLs, showcasing the effectiveness of deep learning models in identifying phishing attempts through a combination of deep neural networks and extended short-term memory techniques. Vaitkevicius et al. [44] highlighted the introduction of novel deep learning approaches for solving the problem of phishing website detection, emphasizing the continuous advancements in deep learning techniques for combating cyber threats. Al-Sarem et al. [45] emphasized the widespread investigation of deep learning methods for detecting phishing websites, indicating the growing interest in leveraging deep learning to enhance online platform security.

Altaher et al. [46] proposed an intelligent ensemble learning approach for phishing website detection based on weighted soft voting, showcasing the potential of ensemble deep learning models in improving the detection of phishing websites.

Table 1: Comparative analysis of phishing detection frameworks

Reference	Focus and methodology	Contributions
[27]	ATT&CK matrix-based framework for countering APTs	Defence against sophisticated phishing
[33]	Behavioral model for phishing attacks	Awareness, measurement, and defence strategies
[34]	Neural networks with reinforcement learning	Real-time phishing email detection
[35]	Machine learning classifiers for android devices	Protection against new phishing activities
[36]	Hybrid Ensemble Feature Selection (HEFS) framework	Enhanced accuracy in phishing attack detection
[37]	'PhishBench' benchmarking framework	Systematic evaluation of phishing detection features
[38]	Survey on intelligent detection designs	Insights into detection frameworks
[39]	Literature review on deep learning applications	Deep learning framework preferences
[40]	1D CNN deep learning model	Effective phishing website detection
[41]	Parsimonious deep learning models	Optimisation for effective phishing detection
[42]	PUCNN deep learning approach	URL-based phishing detection
[43]	Hybrid DNN-LSTM model	Detecting phishing URLs
[44]	Novel deep-learning approaches	Advances in phishing website detection
[45]	Investigation of deep learning methods	Security enhancement of online platforms
[46]	Intelligent ensemble learning approach	Improved phishing website detection

3 Proposed Approach

In our proposed approach, depicted in Fig. 1, the phishing detection process begins with collecting URLs from user traffic. A URL Extractor component facilitates this initial data acquisition, which systematically captures URLs users access over the Internet. These URLs serve as the input data for the feature selection mechanism, which is a critical phase where relevant characteristics of the URLs are identified for subsequent analysis.

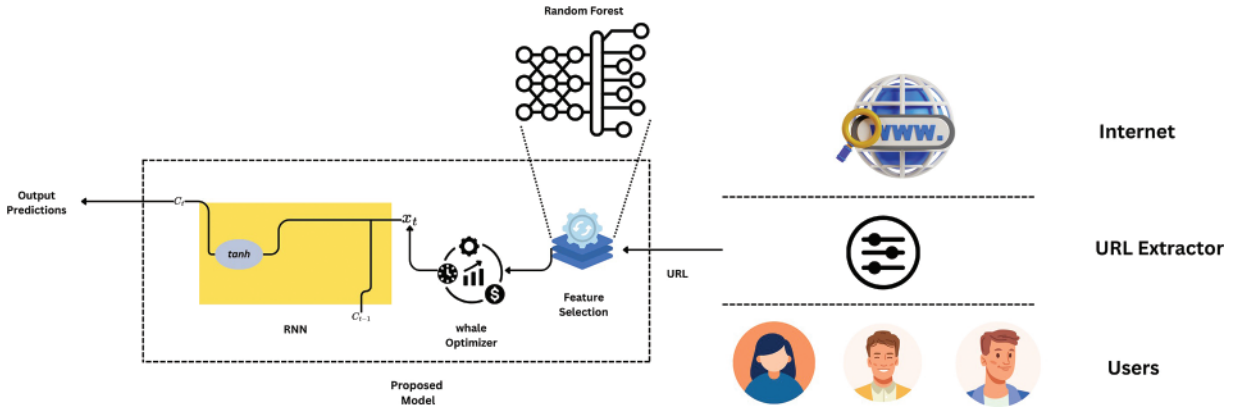

Figure 1: Proposed model

The feature selection process is powered by a Random Forest model, chosen for its efficacy in handling high-dimensional data and its robustness in selecting significant features. Once the relevant features are extracted, they are fed into our optimized Recurrent Neural Network (RNN) model. The RNN, known for its proficiency in processing sequential and time-series data, is specifically structured to identify complex patterns within the URL features indicative of phishing activity.

To refine the RNN's performance, we introduce the Whale Optimization Algorithm (WOA). The WOA iteratively adjusts the RNN's hyperparameters, such as the number of layers and the neurons within each layer, to optimise the detection accuracy. The intricate dance of the WOA's search agents, mirroring the whales' hunting strategy, seeks out the most effective configuration for the RNN by minimising a predefined loss function.

3.1 Feature Selection

In our methodology, we employed a Random Forest model to assess and rank the importance of various features in predicting phishing websites (Algorithm 1). Initially, the Random Forest model was trained on our dataset, comprising over 11,000 website URLs labeled as either phishing or legitimate based on 30 distinct website parameters. This training process involves constructing multiple decision trees and aggregating their predictions to improve the model's accuracy and robustness against overfitting, as represented in Fig. 2.

Upon training the model, we calculated each feature's importance. This calculation was based on the mean decrease in impurity criterion, a standard measure used in tree-based models to evaluate how each feature contributes to the homogeneity of the nodes and leaves in the decision trees. Essentially, features that lead to higher decreases in impurity when used for splitting in the trees are considered more important. After computing the importance scores for all features, we ranked them in descending order of importance. This ranking gave us a clear hierarchy of features based on their effectiveness in distinguishing between phishing and legitimate websites. The most important features contributed

most significantly to the accuracy of the Random Forest model, highlighting their relevance in the context of phishing detection.

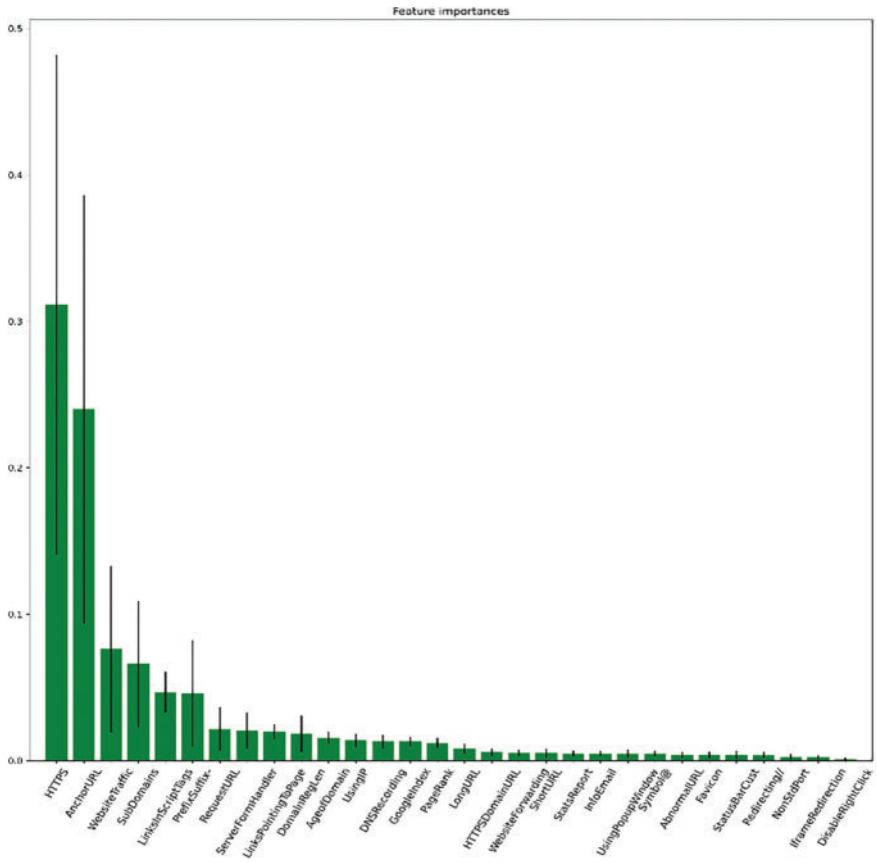

Figure 2: Important features

This feature ranking process, as outlined in our algorithm, plays a crucial role in our study. It not only identifies the key indicators of phishing activities but also aids in refining our model by focusing on the most informative attributes. Such an approach ensures that our phishing detection system is efficient and effective, leveraging the strengths of Random Forest models in handling complex classification tasks.

3.2 Whale Optimization Algorithm

The Whale Optimization Algorithm (WOA) is a metaheuristic optimization technique inspired by the hunting behavior of humpback whales. Initially proposed in 2016, the algorithm mimics the hunting strategies of whales, such as encircling and predation, to achieve global optimization [47,48]. WOA has gained popularity due to its ability to address a wide range of optimization problems across various domains, including mathematics, computer science, and engineering. Researchers have applied WOA to diverse optimization tasks, showcasing its versatility and effectiveness. For instance, WOA has been utilized in economic patch problems in power systems, environmental-constrained economic dispatch, and parameter estimation of software reliability growth models [48,49]. Moreover, the algorithm has been employed in tasks like image denoising, face image forgery detection, and

blind source separation, demonstrating its applicability in image processing and signal processing domains [50].

The algorithm's success lies in its ability to balance exploration and exploitation, which is crucial for achieving optimal solutions in complex optimization landscapes [48]. Researchers have also proposed enhancements to the original WOA, such as incorporating simulated annealing strategies, inertia weight methods, and parallelisation techniques to improve its performance [51–53]. Due to these advantages of WOA, we used it in our proposed approach. It has been applied to our model to search for the optimal set of hyperparameters efficiently. The WOA begins by initialising a population of candidate solutions called search agents. These agents explore the search space through a simulated process of encircling prey and bubble.

Algorithm 1: Feature importance calculation using random forest

Input: Training dataset $D = (x_i, y_i)_{i=1}^{N}$ with features $X = x_1, x_2, \ldots, x_M$ and labels $Y = y_1, y_2, \ldots, y_N$
Output: Ranked list of features by importance
procedure TrainRandomForest(D)
    Initialize Random Forest model RF
    Train RF on dataset D
    return RF
end procedure
procedure CalculateFeatureImportance(RF)
    Initialize empty list *importance*
    for each feature f in X **do**
        Calculate importance of f using RF (e.g., mean decrease in impurity)
        Append importance to *importance* list
    end for
    return *importance*
end procedure
procedure RankFeatures(*importance*)
    Sort *importance* in descending order
    return Sorted list of features by importance
end procedure
$RF \leftarrow$ TrainRandomForest(D)
importance $\leftarrow$ CalculateFeatureImportance(RF)
rankedFeatures $\leftarrow$ RankFeatures (*importance*)
print *rankedFeatures*

Net feeding behavior, which mathematically guides the update of their positions in the search space. The best search agent, determined by the objective function's fitness evaluation, leads the algorithm's exploration and exploitation phases.

Over a pre-defined number of iterations, each agent updates its position relative to the position of the best search agent discovered thus far. If a search agent encounters a better solution, it becomes the new leader, guiding the swarm's subsequent movements. This iterative process continues until the algorithm converges on an optimal solution or reaches the maximum number of iterations, resulting in the best set of hyperparameters for the RNN model.

The WOA's efficacy in navigating complex, multidimensional search spaces and avoiding local optima is attributed to its balanced mechanism of exploration and exploitation, which is crucial for the adaptive tuning of hyperparameters in deep learning models. The implementation of WOA in our study, as outlined in the algorithmic structure, significantly contributed to the high classification performance of the RNN model by identifying an optimal hyperparameter configuration that accommodates the nuances of the phishing detection task (Algorithm 2).

3.3 Recurrent Neural Network Model

The Recurrent Neural Network (RNN) training for our phishing detection model follows a structured algorithmic approach (Algorithm 3). The process begins with initialising the RNN parameters, including the weights and biases of the network's layers. These parameters are set to random values to break symmetry and facilitate learning. Once initialised, the model enters the training phase, which processes the input sequences through a series of forward passes. In each forward pass, the RNN computes the hidden states sequentially for each time step, leveraging the information from the previous state and the current input. This temporal dependency allows the RNN to capture dynamic temporal behavior, which is characteristic of sequence data.

Algorithm 2: Whale Optimization Algorithm (WOA)

Input: Objective function $f(\vec{x})$, Number of search agents n, Maximum iterations T
Output: Best solution $f(\vec{x})^*$
procedure InitializeAgents(n)
  for $i \leftarrow 1$ to n **do**
    Initialize position $\vec{x}_i$ randomly
  end for
  return $\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n$
end procedure
procedure FindBestAgent($\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n$)
  $\vec{x_1}^* \leftarrow$ agent with the best fitness value according to $f(\vec{x})$
  return $\vec{x}^*$
end procedure
procedure MainLoop($\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n,\ T$)
  for $t \leftarrow 1$ to T **do**
    for each agent $\vec{x}_i$ **do**
      Update position $\vec{x}_i$ based on $\vec{x}_i^*$ and $f(\vec{x})$
      Calculate fitness of $\vec{x}_i$
      if fitness of $\vec{x}_i$ is better than $\vec{x}_i$ **then**
        $\vec{x}_i{}^* \leftarrow \vec{x}_i$
      end if
    end for
  end for
  return $\vec{x}_i^*$
end procedure
$\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n \leftarrow$ InitializeAgents(n)
$\vec{x}_i^* \leftarrow$ FindBestAgent($\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n$)
$\vec{x}_i^* \leftarrow$ MainLoop($\vec{x_1}, \vec{x}_2, \ldots \vec{x}_n,\ T$)

Algorithm 3: Recurrent neural network training algorithm

Input: Training dataset $D = :(X^i Y^i)_{i=1}^N$, Learning rate η, Number of epochs E
  Output: Trained RNN model parameters
    procedure InitializeParameters
    Randomly initialize the weights and biases in RNN layers
    return Initialized parameters Θ
end procedure
procedure ForwardPass(X, Θ)
      for each time step t in X **do**
          Compute hidden state h_t using previous hidden state h_{t-1} and current input x_t
          Compute output o_t from hidden state h_t
      end for
      return All outputs O and final hidden state h_T
end procedure
 procedure ComputeLoss(O, Y)
      Compute the loss L between the predicted outputs O and true values Y
      return Loss L
end procedure
procedure BackWardPass(L, Θ)
      Compute gradients of loss L with respect to parameters Θ
      Update parameters Θ using gradients and learning rate η
      return Updated parameters Θ
end procedure
procedure TrainRNN(D, η, E)
       $\Theta \leftarrow$ InitializeParameters
      for epoch $e \leftarrow 1$ to E **do**
          for each $(X^{(i)}, Y^{(i)})$ in D **do**
              O, $h_T \leftarrow$ ForwardPass($X^{(i)}$, Θ)
              $L \leftarrow$ ComputeLoss($O, Y(i)$)
              $\Theta \leftarrow$ BackwardPass(L, Θ)
          end for
      end for
      return Trained parameters Θ
end procedure
$\Theta \leftarrow$ TrainRNN(D, η, E)

After calculating the network's final output, a loss function is employed to measure the discrepancy between the predicted sequence and the true output labels. The loss function is crucial for evaluating the model's performance at each epoch. The computed loss is then used to perform a backward pass. During backpropagation, gradients of the loss concerning the model parameters are calculated, and the parameters are updated accordingly. This update is guided by the learning rate, which controls the step size in the gradient descent optimisation. The model undergoes multiple training epochs, iteratively improving its parameters through successive forward and backward passes. Each epoch refines the model's parameters (weights and biases), enhancing its ability to classify input sequences accurately. The training continues until the RNN converges to a set of parameters that minimises the loss function. This indicates that the model is adequately trained to detect phishing

websites with high precision and recall. The efficacy of this training algorithm is validated by the performance metrics obtained post-training, which demonstrate the model's ability to generalise and perform accurate predictions on unseen data.

4 Results and Discussion

4.1 Dataset Representation

The dataset utilized in our study was sourced from Kaggle [54]. This dataset comprises over 11,000 website URLs, each annotated with 30 distinct parameters describing various website aspects. These parameters were carefully selected to capture features relevant to identifying phishing activities. Additionally, each website in the dataset is labelled with a class indicator, where '1' denotes a phishing website and '0' indicates a non-phishing (legitimate) website, as represented in Fig. 3.

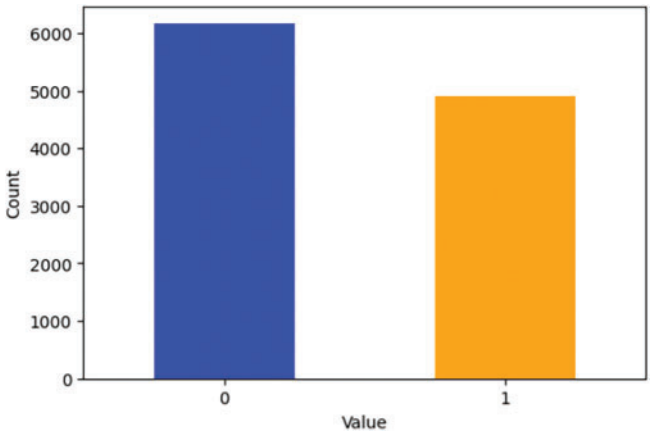

Figure 3: Distribution of class in the dataset

The correlation analysis of the selected features provides essential insights into the predictive attributes of phishing websites. As depicted in Fig. 4, the strongest positive correlation with the target variable was observed for the feature 'HTTPS', which yielded a correlation coefficient of approximately 0.715. This indicates that secure HTTP usage is a significant indicator of non-phishing websites. Following closely, 'AnchorURL' also showed a strong positive correlation, with a coefficient of about 0.693, suggesting that the use of anchor URLs is a prevalent characteristic of phishing sites. Other features displayed moderate.

Positive correlations, such as 'WebsiteTraffic' (0.346), 'SubDomains' (0.298), and 'PrefixSuffix-' (0.349), which underscore their relevance in the detection model. 'LinksInScriptTags' and 'RequestURL' exhibited similar correlation values around the 0.25 mark, reinforcing their utility as features for identifying phishing activities.

Conversely, the 'DomainRegLen' feature was negatively correlated with the target, showing a coefficient of −0.226, implying that longer domain registration lengths may be associated with legitimate websites.

Lesser yet noteworthy positive correlations were found with 'AgeofDomain' (0.121), 'GoogleIndex' (0.129), 'UsingIP' (0.094), 'DNSRecording' (0.076), and 'PageRank' (0.105). Although lower, these correlations indicate certain behaviors and characteristics of phishing websites.

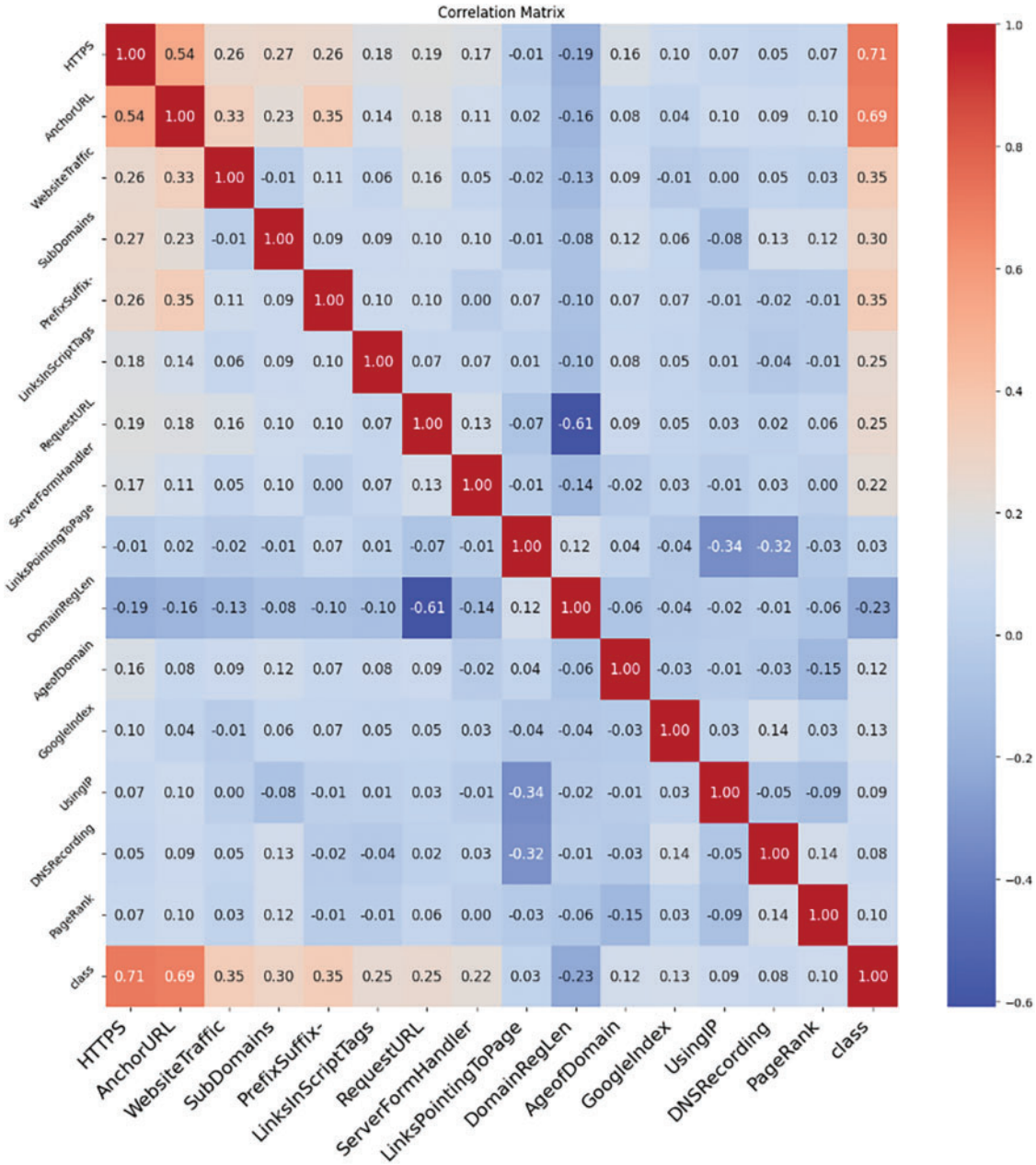

Figure 4: Correlation matrix

The 'LinksPointingToPage' feature showed an almost negligible correlation of 0.033, suggesting that it may not be a strong standalone predictor of phishing activity but could still contribute to the overall predictive power of the model when combined with other features.

The distribution and density of the values taken by each feature, categorized by class, are visually summarized in violin plots, as illustrated in Figs. 5 and 6. These plots elucidate the comparative distribution of features between phishing (Class 1) and legitimate (Class 0) websites.

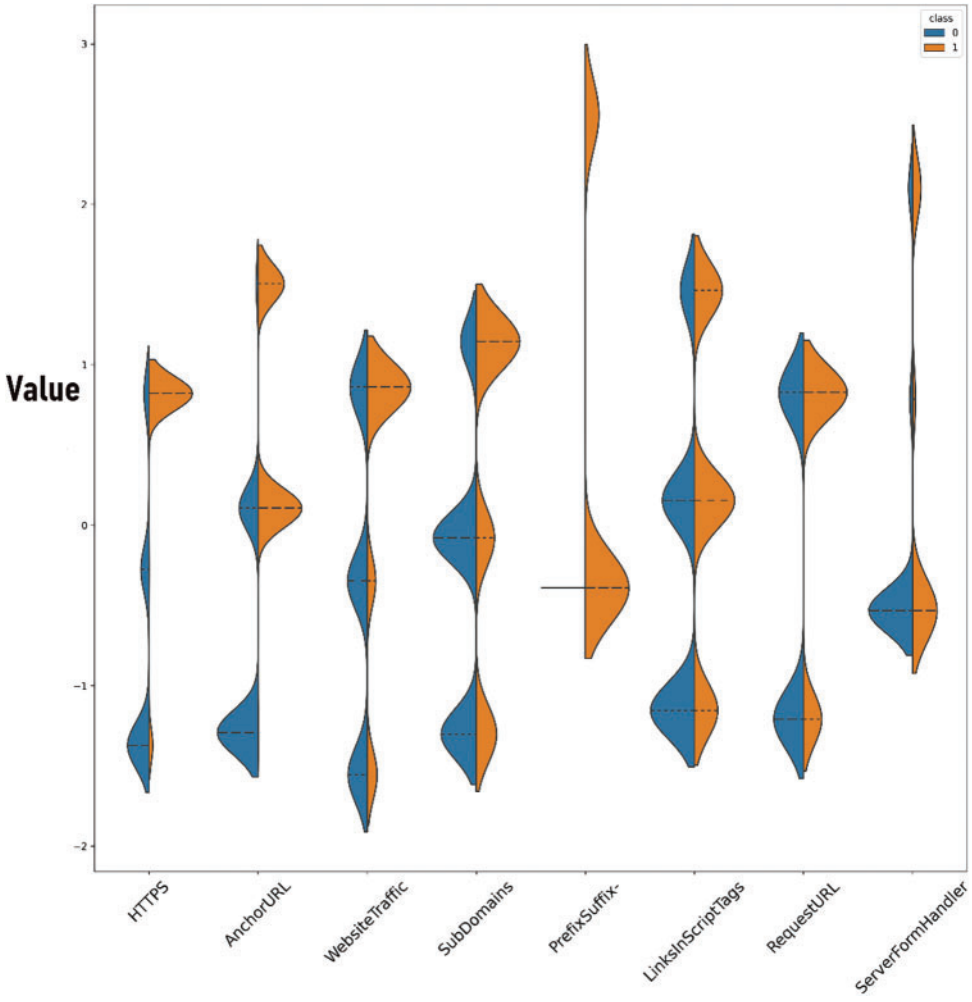

Figure 5: Violin plot of first 8 features

In Fig. 5, the 'HTTPS' feature demonstrates a distinct separation between the classes, with phishing websites showing a lower density of secure protocol usage. Similarly, the 'AnchorURL' feature shows a divergent distribution, underlining its discriminatory power. 'WebsiteTraffic' and 'SubDomains' also exhibit varying densities across classes, indicating their potential as discriminative features in phishing detection. The 'PrefixSuffix-' feature also denotes a clear distinction in its distribution between the two classes. This pattern is mirrored across 'LinksInScriptTags' and 'RequestURL', which display unique distributions for phishing websites. The 'ServerFormHandler' feature, while less distinct, still shows variability between the classes.

Fig. 6 continues this analysis with additional features. 'LinksPointingToPage' displays a marginal difference in distribution, suggesting a lesser, but still relevant, discriminatory capability. The negative values associated with 'DomainRegLen' in Class 1 websites reinforce this feature's inverse relationship with phishing likelihood, as shorter domain registration lengths are more common among phishing sites. 'AgeofDomain', 'GoogleIndex', 'UsingIP', and 'DNSRecording' present subtler differences in

their distributions but are nonetheless integral to the comprehensive model, contributing to the overall classification efficacy.

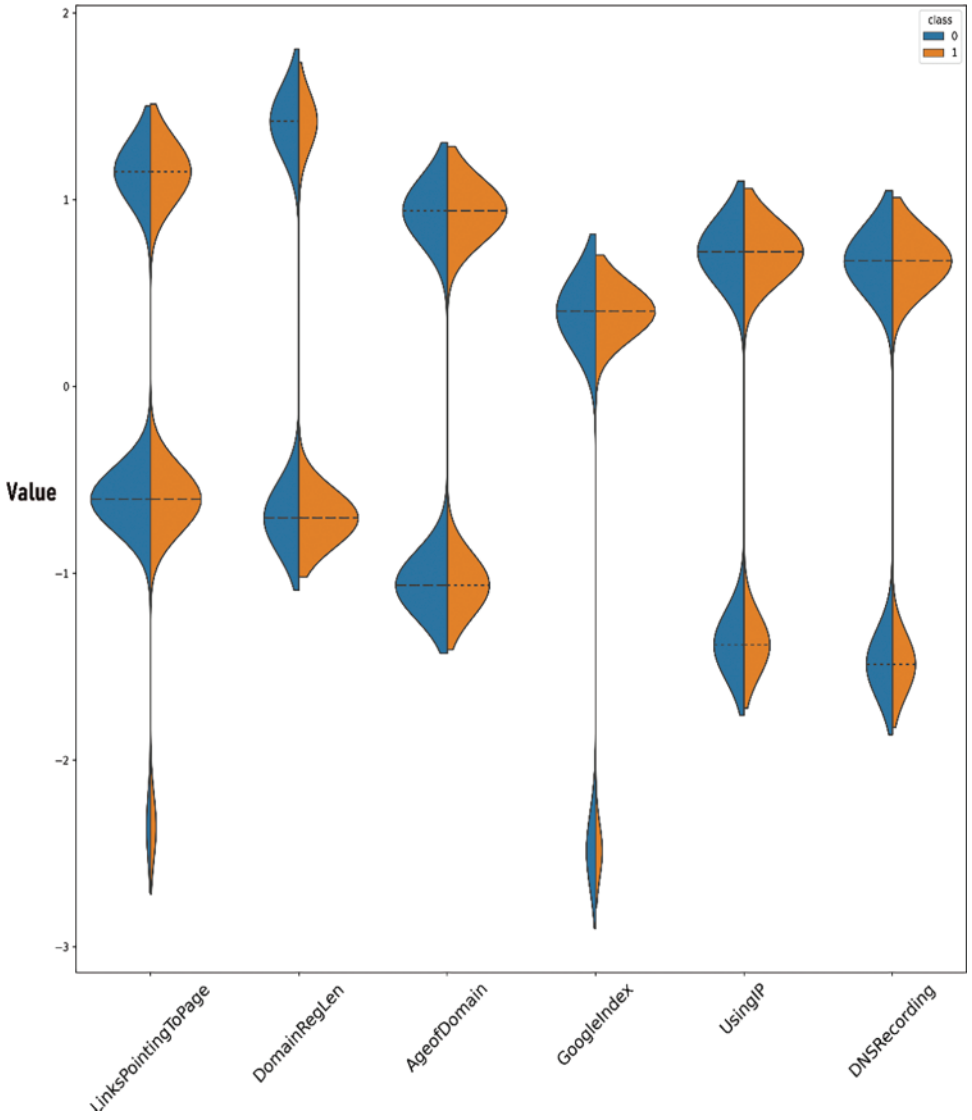

Figure 6: Violin plot of last 8 features

The distribution of each selected feature within the dataset is illustrated through a series of histograms presented in Fig. 7. These histograms visually represent the frequency of feature values across the entire collection of website URLs. The 'HTTPS' feature histogram shows a bimodal distribution, reflecting the dichotomy between secure and non-secure websites. Similarly, the 'AnchorURL' histogram indicates a strong bimodal presence, which aligns with its significant role in the classification of websites as indicated by the correlation analysis.

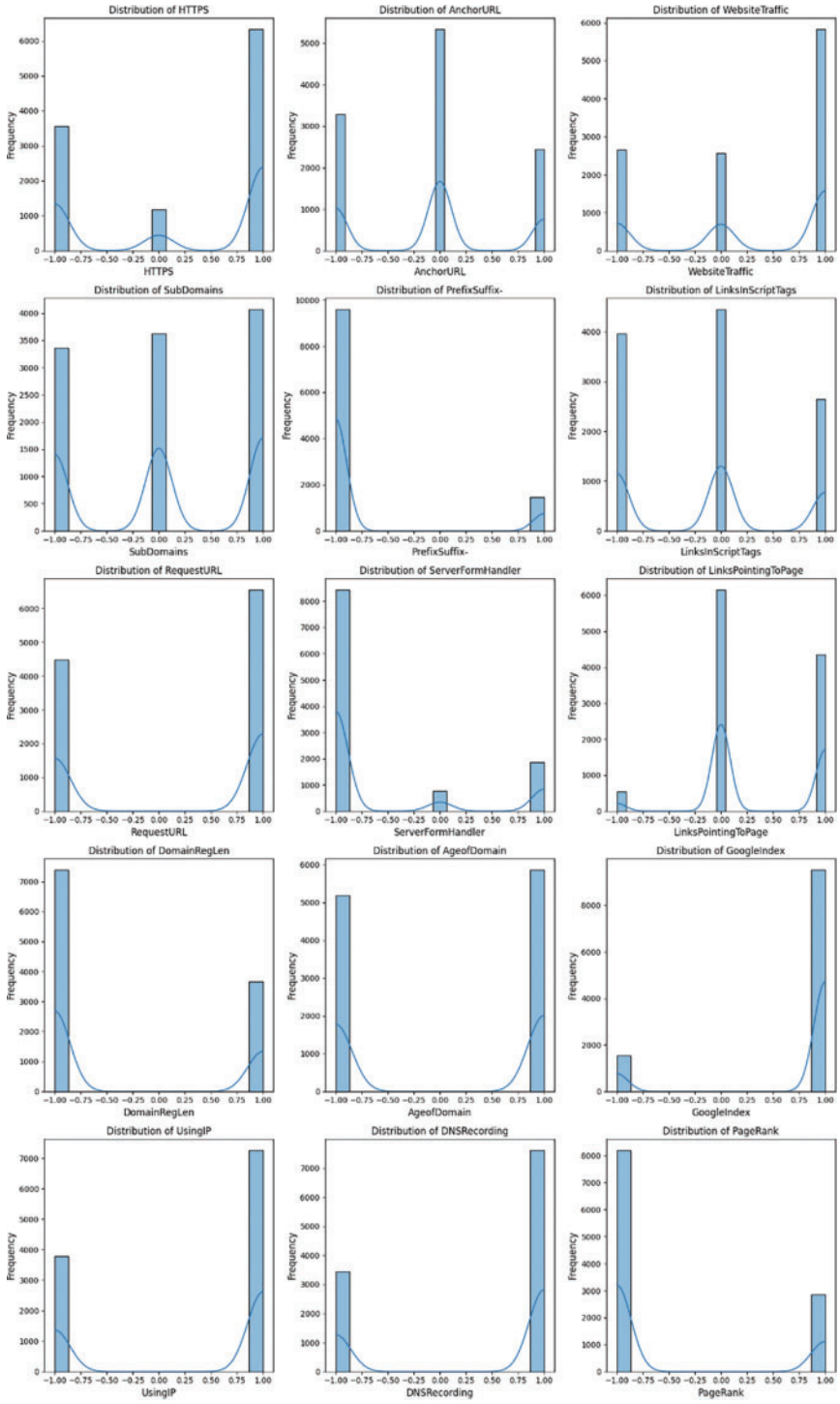

Figure 7: Histogram representation

The 'WebsiteTraffic' histogram exhibits a skewed distribution, suggesting that most websites in the dataset have a similar level of traffic, with a few outliers. This skewness is also evident in the

'SubDomains' feature, where most URLs have a lower count of subdomains. The 'PrefixSuffix-' feature shows a particularly striking bimodal distribution, which underscores its potential as a strong indicator of phishing activity when present.

In the case of 'LinksInScriptTags', the distribution is more uniform yet still shows peaks that may correspond to standard scripting practices in phishing sites. The 'RequestURL' histogram leans towards lower values, implying that phishing URLs often contain fewer external links. 'ServerFormHan-deer', while less distinctly bimodal, exhibits a distribution that suggests it is a less common but still relevant feature in phishing URLs.

4.2 Model Performance Analysis

The optimization of hyperparameters for the Recurrent Neural Network (RNN) model was guided by the Whale Optimization Algorithm (WOA), leveraging the selected features to refine the model's predictive capabilities. As illustrated in Fig. 8, the loss and accuracy plots over 20 epochs demonstrate the model's performance during training. The training loss started at 0.440 and steadily decreased to 0.409, while the test loss closely followed, commencing at 0.435 and reducing to approximately 0.404. The convergence of these values indicates a robust fit to the data without significant overfitting. Simultaneously, the accuracy of the model showed consistent improvement. The training accuracy increased from approximately 90.58% at the initial epoch to 91.52% by the end of the 20th epoch. The test accuracy exhibited a similar upward trend, starting at 91.29% and reaching 92.41%. These metrics underscore the RNN model's capability to generalise well from the training data to unseen test data.

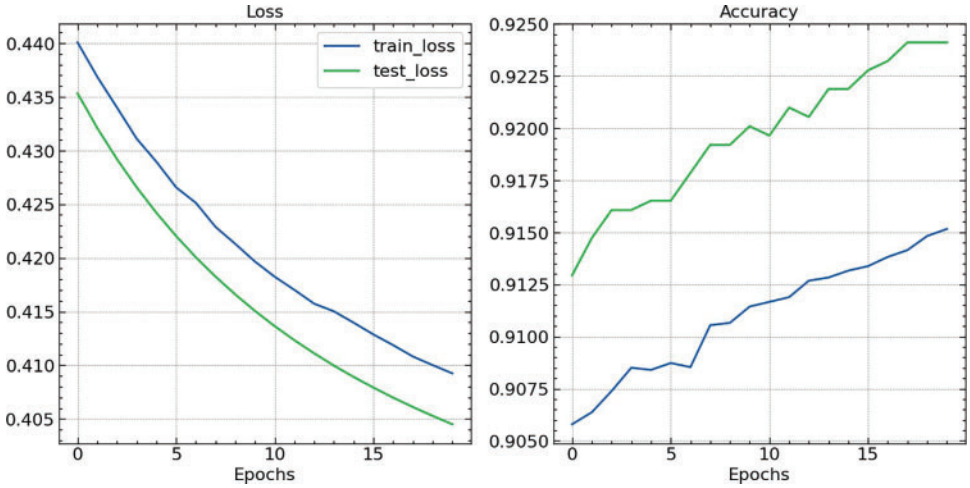

Figure 8: Accuracy and loss curves

The progressive improvement in both loss and accuracy, as depicted in the loss and accuracy graphs (Fig. 8), reflects the effectiveness of the selected features and the optimised hyperparameters in enhancing the model's learning process. The close alignment between training and test metrics throughout the training phase emphasises the model's stability and reliability, crucial qualities in a phishing detection system where precision and recall are paramount. The optimisation process via WOA thus not only facilitated a fine-tuned model and ensured its practical applicability in the real-world scenario of phishing detection.

The classification performance of the RNN model, post-optimization with the Whale Optimization Algorithm, is quantitatively presented in the classification report depicted in Fig. 9. The model achieved a precision of 0.92 for both classes, indicating a high level of accuracy in predicting both normal and phishing websites. The recall for regular websites was slightly lower at 0.90 compared to 0.94 for phishing websites, suggesting that the model was somewhat more effective at identifying phishing sites than normal ones. The F1-score, which is a harmonic mean of precision and recall, stood at 0.91 for standard websites and 0.93 for phishing websites, underscoring the balanced performance of the model across both classes. Overall Accuracy of the model was 0.92, which was consistent across the macro average and weighted average scores, indicating uniformity in the model's predictive capability across the dataset. The support, which is the number of true instances for each label, was 976 for standard websites and 1235 for phishing websites. These figures confirm that the dataset was somewhat imbalanced, favouring phishing website instances, yet the model maintained high precision and recall across the board.

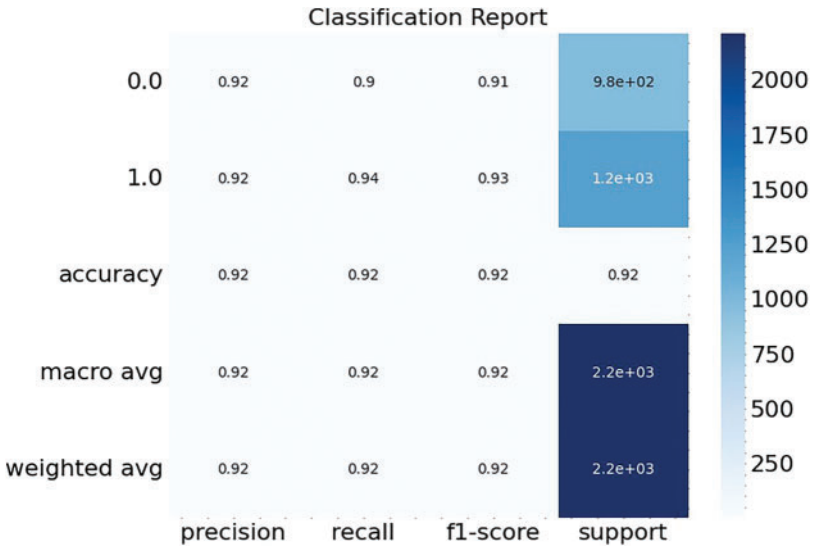

Figure 9: Classification report

The confusion matrix, as illustrated in Fig. 10, encapsulates the performance of the RNN model on the test set. The model correctly identified 880 out of 976 normal websites, resulting in a true positive rate of approximately 90.2% for normal websites. It also accurately classified 1161 out of 1235 phishing websites, equating to a true positive rate for phishing websites of about 94.0%. These results demonstrate the model's robustness in accurately identifying both classes. However, the model was not without its misclassifications. It incorrectly predicted 96 normal websites as phishing and 74 phishing websites as normal, accounting for false positive rates of 9.8% and 6.0%, respectively. Despite these misclassification, the model's overall accuracy, as depicted by the diagonal predominance in the confusion matrix, underscores its effectiveness.

4.3 Computation Complexity

The computational complexity of our proposed method, which integrates a Recurrent Neural Network (RNN) optimized by the Whale Optimization Algorithm (WOA), hinges on several key aspects. Firstly, the RNN's complexity is primarily determined by the number of parameters, which

includes the weights and biases across all layers and is influenced by the number of hidden units and layers in the network. For an RNN, the computational complexity per training epoch can be approximated as $O(T.N.H^2)$, where T is the sequence length, N is the number of samples, and H is the number of hidden units. This complexity stems from the RNN's need to process sequences of data, where each sequence element is passed through the network's layers sequentially.

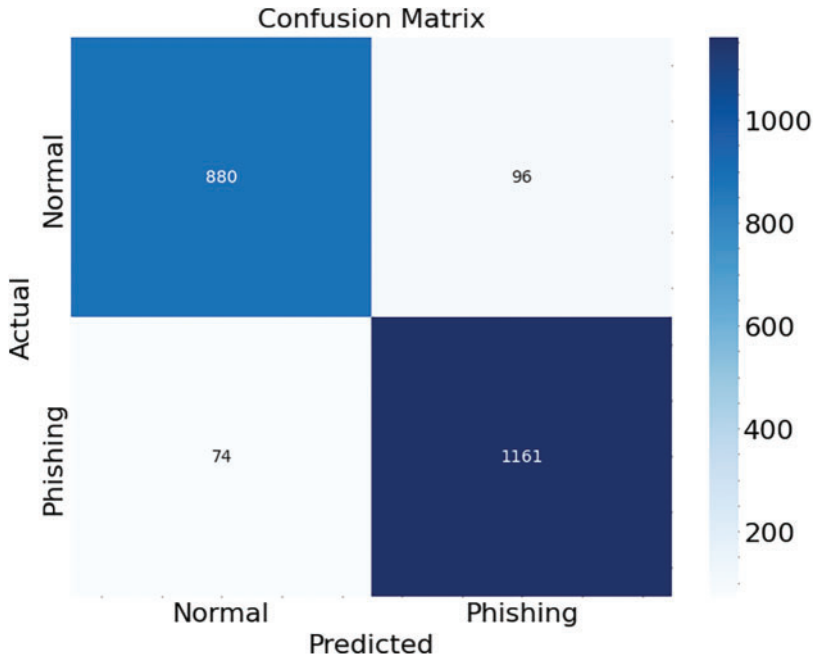

Figure 10: Confusion matrix

Secondly, the Whale Optimization Algorithm contributes to the overall computational load through its iterative search process for optimal hyperparameters. The complexity of WOA can be characterized by O(I.A), where I is the number of iterations and A is the number of agents (candidate solutions). Each iteration involves evaluating the performance of each agent, which, in the context of our method, means training and validating the RNN with a set of hyperparameters and calculating a performance metric.

The combination of these two methods means that the total computational complexity is influenced by the cost of training the RNN multiple times across the iterations of the WOA. Despite this, the optimization process is a one-time cost, with the outcome being an RNN model that is finely tuned for the phishing detection task. Moreover, our approach includes strategies to mitigate computational demands, such as parallel processing of WOA agents when feasible and early stopping criteria in RNN training and the WOA process to curtail unnecessary computations.

4.4 Comparative Analysis

In the context of phishing detection, various machine learning approaches have been explored, each with distinct attributes, as summarized in Table 2. Logistic Regression, known for its simplicity and speed, offers an interpretable model but is often outperformed by more complex algorithms when handling non-linear patterns. Support Vector Machines (SVM) are adept at discovering non-linear relationships but can be computationally intensive and less interpretable. Decision Trees stand out

for their interpretability and fast training times, yet their performance can vary significantly with large datasets. Naive Bayes is praised for its simplicity and speed but needs to improve in robustness, particularly with noisy data.

Table 2: Comparative analysis

Attribute	Logistic regression	SVM	Decision trees	Naive Bayes	Proposed model
Model complexity	Low	High	Medium	Low	High
Interpretability	High	Low	High	High	Medium
Feature scalability	Moderate	Low	High	Moderate	High
Handling of non-linear patterns	Low	High	Medium	Low	Very high
Training time	Fast	Slow	Fast	Fast	Slow
Sensitivity to imbalanced data	High	Medium	High	High	Low
Performance on large datasets	Good	Good	Variable	Good	Excellent
Requirement for feature engineering	High	High	Medium	High	Low
Robustness to noisy data	Low	Medium	Low	Medium	High
Online learning capability	Yes	No	Yes	Yes	Yes

Our proposed model, which employs a Recurrent Neural Network (RNN) optimized by the Whale Optimization Algorithm (WOA), demonstrates a significant improvement over these traditional approaches. It effectively automates feature representation learning, reducing the need for extensive feature engineering that traditional methods typically require. Moreover, the RNN-WOA model exhibits exceptional performance on large datasets, a key consideration given the voluminous nature of web traffic data. It is specifically designed to handle the intricate, non-linear patterns characteristic of phishing URLs, distinguishing itself from the simpler models that may falter in this aspect.

5 Conclusion

Our proposed framework successfully demonstrates the efficacy of using an RNN model with WOA for phishing website detection. The integration of these methodologies has led to significant advancements in predictive accuracy, achieving an overall model accuracy of 92%. The WOA proved instrumental in optimizing the RNN hyperparameters, enhancing model performance as evidenced by the precision, recall, and F1-score metrics. The study's results indicate that the combined approach is a robust solution for the increasingly complex task of identifying phishing threats, offering substantial improvements over traditional methods and setting a new benchmark for future research in cybersecurity defense mechanisms. In the future, we will test our proposed model on the different datasets. We will also focus on the scalability of our proposed work.

Acknowledgement: The research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R 343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Kingdom of Saudi Arabia, for funding this research work through the project number "NBU-FFR-2024-1092-02".

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R 343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Kingdom of Saudi Arabia, for funding this research work through the project number "NBU-FFR-2024-1092-02".

Author Contributions: Final manuscript revision, funding, supervision: Brij Bhooshan Gupta, Kwok Tai Chui; study conception and design, analysis and interpretation of results, methodology development: Akshat Gaurav, Varsha Arya, data collection, draft manuscript preparation, figure and tables: Ahmed Alhomoud, Razaz Waheeb Attar. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analysed during this study are included in this published article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. M. Ali and N. F. Mohd Zaharon, "Phishing—A cyber fraud: The types, implications and governance," *Int. J. Educ. Reform*, vol. 33, no. 1, pp. 101–121, 2024. doi: 10.1177/10567879221082966.

[2] A. K. Jain, B. B. Gupta, K. Kaur, P. Bhutani, W. Alhalabi and A. Almomani, "A content and URL analysis-based efficient approach to detect smishing SMS in intelligent systems," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 11117–11141, 2022. doi: 10.1002/int.23035.

[3] M. Kheruddin, "Phishing attacks: Unraveling tactics, threats, and defenses in the cybersecurity landscape," *TechRxiv*, Jan. 15, 2024. doi: 10.22541/au.170534654.48067877/v1.

[4] A. K. Jain and B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterp. Inform. Syst.*, vol. 16, no. 4, pp. 527–565, 2022. doi: 10.1080/17517575.2021.1896786.

[5] J. C. Gomes de Barros *et al.*, "Piracema: A Phishing snapshot database for building dataset features," *Sci. Rep.*, vol. 12, 2022, Art. no. 15149. doi: 10.1038/s41598-022-19442-8.

[6] P. Kumaraguru, S. Sheng, A. Acquisti, L. Cranor, and J. Hong, "Teaching johnny not to fall for phish," *ACM Trans. Internet Technol.*, vol. 10, pp. 1–31, 2010. doi: 10.1145/1754393.1754396.

[7] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, 2021. doi: 10.1016/j.comcom.2021.04.023.

[8] F. Tchakounté, L. Kanmogne Wabo, and M. Atemkeng, "A review of gamification applied to phishing," *Preprints*, 2020, Art. no. 2020030139. doi: 10.20944/preprints202003.0139.v1.

[9] V. Vajrobol, B. B. Gupta, and A. Gaurav, "Mutual information based logistic regression for phishing URL detection," *Cyber Secur. Appl.*, vol. 2, 2024, Art. no. 100044. doi: 10.1016/j.csa.2024.100044.

[10] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, pp. 74–81, 2012. doi: 10.1145/2063176.2063197.
[11] C. M. Igwilo and V. T. Odumuyiwa, "Comparative analysis of ensemble learning and non-ensemble machine learning algorithms for phishing URL detection," *Fuoye J. Eng. Technol.*, vol. 7, no. 3, pp. 305–312, 2022. doi: 10.46792/fuoyejet.v7i3.807.
[12] G. Soon, L. C. Chiang, C. K. On, N. M. Rusli, and T. S. Fun, "Comparison of ensemble simple feedforward neural network and deep learning neural network on phishing detection," *Comput. Sci. Tech.: 6th ICCST 2019*, Kota Kinabalu, Malaysia, Springer Singapore, pp. 595–604, 2020.
[13] C. Marforio, R. J. Masti, C. Soriente, K. Kostiainen, and S. Capkun, "Hardened setup of personalized security indicators to counter phishing attacks in mobile banking," in *Proc. 6th Workshop Secur. Priv. Smartphones Mob. Devices (SPSM'16)*, New York, NY, USA, Association for Computing Machinery, 2016. doi: 10.1145/2994459.2994462.
[14] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, *2007 IEEE Symp. Secur. Priv. (SP'07)*, Berkeley, CA, USA, 2007, pp. 51–65. doi: 10.1109/SP.2007.35.
[15] S. Marimuthu, S. Gopalasamy, and J. BenâOthman, "Intelligent antiphishing framework to detect phishing scam: A hybrid classification approach," *Softw. Pract. Exp.*, vol. 52, pp. 459–481, 2021. doi: 10.1002/spe.3031.
[16] B. Hu, A. Gaurav, C. Choi, and A. Almomani, "Evaluation and comparative analysis of semantic web-based strategies for enhancing educational system development," *Int. J. Semant. Web Inform. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–14, 2022. doi: 10.4018/IJSWIS.
[17] Wei B *et al.*, "A deep-learning-driven light-weight phishing detection sensor," *Sensors*, vol. 19, 2019, Art. no. 4258. doi: 10.3390/s19194258.
[18] S. Baadel and J. Lu, "Data analytics: Intelligent anti-phishing techniques based on machine learn-ing," *J. Inform. Knowl. Manage.*, vol. 18, 2019, Art. no. 1950005. doi: 10.1142/S0219649219500059.
[19] A. Almomani *et al.*, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *Int. J. Semant. Web Inform. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–24, 2022. doi: 10.4018/IJSWIS.
[20] N. Stevanović, "Character and word embeddings for phishing email detection," *Comput. Inform.*, vol. 41, pp. 1337–1357, 2022. doi: 10.31577/cai_2022_5_1337.
[21] J. V. Tembhurne, M. M. Almin, and T. Diwan, "Mc-DNN: Fake news detection using multi-channel deep neural networks," *Int. J. Semant. Web Inform. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–20, 2022. doi: 10.4018/IJSWIS.
[22] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, 2021, Art. no. 154. doi: 10.3390/fi13060154.
[23] V. K. Chawra and G. P. Gupta, "Optimization of the wake-up scheduling using a hybrid of memetic and tabu search algorithms for 3D-wireless sensor networks," *Int. J. Softw. Sci. Comput. Intell. (IJSSCI)*, vol. 14, no. 1, pp. 1–18, 2022. doi: 10.4018/IJSSCI.
[24] A. Sumner, X. Yuan, M. Anwar, and M. McBride, "Examining factors impacting the effectiveness of anti-phishing trainings," *J. Comput. Inf. Syst.*, vol. 62, pp. 975–997, 2021.
[25] R. Alakbarov, "An optimization model for task scheduling in mobile cloud computing," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 12, no. 1, pp. 1–17, 2022. doi: 10.4018/IJCAC.
[26] U. Butt, R. Amin, H. Aldabbas, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex Intell. Syst.*, vol. 9, pp. 3043–3070, 2022. doi: 10.1007/s40747-022-00760-3.
[27] Y. Shin, K. Kim, J. Lee, and K. Lee, "Focusing on the weakest link: A similarity analysis on phishing campaigns based on the ATT&CK matrix," *Secur. Commun. Netw.*, vol. 2022, pp. 1–12, 2022. doi: 10.1155/2022/1699657.
[28] M. Aliyu, M. Bagarawa, A. Muâazu, and M. Umar, "Understanding phishing awareness among students in tertiary institutions and setting-up defensive mechanisms against the attackers," *Caliphate J. Sci. Technol.*, vol. 5, pp. 22–31, 2023. doi: 10.4314/cajost.v5i1.4.

[29] G. Mohamed, J. Visumathi, M. Mahdal, J. Anand, and M. Elangovan, "An effective and secure mechanism for phishing attacks using a machine learning approach," *Processes*, vol. 10, 2022, Art. no. 1356. doi: 10.3390/pr10071356.
[30] S. Eftimie, R. Moinescu, and C. Racuciu, "Spear-phishing susceptibility stemming from personality traits," *IEEE Access*, vol. 10, pp. 73548–73561, 2022. doi: 10.1109/ACCESS.2022.3190009.
[31] B. Weaver, A. Braly, and D. Lane, "Training users to identify phishing emails," *J. Educ. Comput. Res.*, vol. 59, pp. 1169–1183, 2021. doi: 10.1177/0735633121992516.
[32] N. Dholakia and P. Agrawal, "Review on phishing attack detection techniques," *Asian J. Converg. Technol.*, vol. 6, pp. 41–47, 2020. doi: 10.33130/AJCT.2020v06i02.008.
[33] J. Thakur and A. Pathan, "Innovations of phishing defense: The mechanism, measurement and defense strategies," *Int. J. Commun. Netw. Inform. Secur. (IJCNIS)*, vol. 10, pp. 19–27, 2018.
[34] 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. doi: 10.1016/j.dss.2018.01.001.
[35] J. Mao *et al.*, "Phishing page detection via learning classifiers from page layout feature," *EURASIP J. Wirel. Commun. Netw.*, vol. 2019, 2019, Art. no. 1745. doi: 10.1186/s13638-019-1361-0.
[36] K. Sumathi and V. Sujatha, "Deep learning based-phishing attack detection," *Int. J. Recent Technol. Eng.*, vol. 8, pp. 8428–8432, 2019. doi: 10.35940/ijrte.C6527.098319.
[37] A. Aassal, S. Baki, A. Das, and R. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *IEEE Access*, vol. 8, pp. 22170–22192, 2020. doi: 10.1109/ACCESS.2020.2969780.
[38] S. Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, "A survey of intelligent detection designs of html URL phishing attacks," *IEEE Access*, vol. 11, pp. 6421–6443, 2023. doi: 10.1109/ACCESS.2023.3237798.
[39] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Knowl. Inf. Syst.*, vol. 64, pp. 1457–1500, 2022. doi: 10.1007/s10115-022-01672-x.
[40] P. Anjali, P. Revati, J. Manorama, S. Suryawanshi, and U. Deepali, "A survey on detection of phishing websites using machine learning," *Int. J. Eng. Comput. Sci.*, vol. 3, pp. 48–51, 2021.
[41] M. Almousa, T. Zhang, A. Sarrafzadeh, and M. Anwar, "Phishing website detection: How effective are deep learningâbased models and hyperparameter optimization?," *Secur. Priv.*, vol. 5, 2022, Art. no. 23. doi: 10.1002/spy2.256.
[42] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, "PDGAN: Phishing detection with generative adversarial networks," *IEEE Access*, vol. 10, pp. 42459–42468, 2022. doi: 10.1109/ACCESS.2022.3168235.
[43] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Comput. Appl.*, vol. 35, pp. 4957–4973, 2021. doi: 10.1007/s00521-021-06401-z.
[44] P. Vaitkevicius and V. Marcinkevicius, "Comparison of classification algorithms for detection of phishing websites," *Informatica*, vol. 31, no. 1, pp. 143–160, 2020.
[45] M. Al-Sarem *et al.*, "An optimized stacking ensemble model for phishing websites detection," *Electronics*, vol. 10, 2021, Art. no. 1285. doi: 10.3390/electronics10111285.
[46] A. Altaher, "Intelligent ensemble learning approach for phishing website detection based on weighted soft voting," *Mathematics*, vol. 9, 2021, Art. no. 2799. doi: 10.3390/math9212799.
[47] A. Hegazy, M. Makhlouf, and G. El-Tawel, "Dimensionality reduction using an improved whale optimization algorithm for data classification," *Int. J. Mod. Educ. Comput. Sci.*, vol. 10, pp. 37–49, 2018. doi: 10.5815/ijmecs.2018.07.04.
[48] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125919–125928, 2019. doi: 10.1109/ACCESS.2019.2938857.
[49] K. Lu and Z. Ma, "A modified whale optimization algorithm for parameter estimation of software reliability growth models," *J. Algorithms Comput. Technol.*, vol. 15, 2021. doi: 10.1177/17483026211034442.
[50] D. Chu, H. Chen, and H. Chen, "Blind source separation based on whale optimization algorithm," in *Matec Web Conf.*, Nanjing, China, May 24–26, 2018, vol. 173, p. 3052.

[51] Y. Gao, H. You, and J. Xu, "Adaptive whale optimization algorithm with simulated annealing strategy and its application in magnetic target location," *Research Square*, 2022. doi: 10.21203/rs.3.rs-607714/v1.

[52] X. Shi, K. Li, and L. Jia, "Improved whale optimization algorithm via the inertia weight method based on the cosine function," *J. Internet Technol.*, vol. 23, no. 7, pp. 1623–1632, 2022. doi: 10.53106/160792642022122307016.

[53] Q. Chai, S. Chu, J. Pan, P. Hu, and W. Zheng, "A parallel WOA with two communication strategies applied in DV-Hop localization method," *EURASIP J. Wirel. Commun. Netw.*, vol. 2020, 2020, Art. no. 51. doi: 10.1186/s13638-020-01663-y.

[54] E. Chand, "Phishing website detector," Accessed: Mar. 19, 2024. [Online]. Available: https://www.kaggle.com/datasets/eswarchandt/ phishing-website-detector