FUDMA Journal of Engineering and Technology
ISSN: 3092-9385
https://fjet.fudutsinma.edu.ng/index.php/fjet/
Volume 2, Issue 1, 2026, 21-30

Development of Ensemble SVM–LSTM Model for Phishing Website Detection

Abubakar L. IBRAHEEM[1*], John K. ALHASSAN[2], Noel D. MOSES[3], Suleiman AHMAD[4]

[1*,2,3,4]Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria

[1*]labbo.pg202318993@st.futminna.edu.ng, [2]jkalhassan@futminna.edu.ng, [3]noelmoses@futminna.edu.ng, [4]ahmads@futminna.edu.ng

Abstract

Phishing is a criminal mechanism employing social engineering techniques to exploit human vulnerabilities and technical loopholes. This is to deceive users into divulging sensitive information, which are in turn used for fraudulent activities. Meanwhile, many machine learning approaches have been proposed in the literature, they often struggle with scalability, adaptability to emerging threats and the trade-off between detection accuracy and computational efficiency. This study aims to enhance phishing website detection through the implementation of an ensemble model that integrate Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) networks. The methodology involves collecting a diverse dataset of phishing and legitimate URLs with a total of 343,525 records. This includes phishing URLs downloaded in CSV file format from PhishTank and Kaggle comprising a total of 167,582 records and legitimate URLs downloaded from the UC Irvine Machine Learning Repository and Kaggle consisting of 175,943 records. Static and Sequential features were extracted. 49 features were extracted and 15 were selected as the most relevant features using Recursive Feature Elimination (RFE) and univariate statistical tests. An ensemble architecture integrating SVM and LSTM networks was then trained using the selected features, employing stratified k-fold cross-validation. The results demonstrate that the proposed approach achieves high detection accuracy of 97.58%, precision 93.54%, recall 96.1% and F1- Score 95.78%, outperforming traditional models and various benchmark classifiers. The findings highlight the effectiveness of combining static and sequential features within an ensemble framework to improve the generalization and robustness of phishing detection systems. Exploring deep learning architectures like CNNs, LSTMs, and GANs in a hybrid framework will likely boost detection capabilities against evolving cyber threats.

Keywords: *Phishing, Cybercrime, Cyber Threats, Identity theft, Ensemble Learning.*

1.0 Introduction

The internet has emerged as a critical global infrastructure, underpinning modern societal functions and enabling unprecedented access to information, commerce, and entertainment [1]. However, the internet also serves as primary platform for malicious actors to perpetrate cybercrime. Among these threats, phishing websites represent a significant risk. The Anti-Phishing Working Group (APWG) defines phishing as "a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials"[2]. Such attacks can lead to severe consequences, including privacy breaches, identity theft, financial losses, and systemic erosion of trust in digital systems.

Cybercriminals employ several deceptive tactics by creating phishing websites and spreading malicious links through various online platforms such as Facebook, X, formally known as Twitter and email. These links are often accompanied by messages designed to instill fear, create a sense of urgency or present an enticing financial offer, compelling their victims to act as a matter of urgency [3]. When an unsuspecting user clicks on the link and enters sensitive credentials, attackers gain unauthorized access to valuable personal data, including financial information, usernames and passwords; this stolen information is subsequently exploited for illicit activities, such as blackmail and financial fraud. Users fall victim to phishing attacks due to a limited understanding of URLs, uncertainty about trustworthy websites, hidden or redirected URLs, time constraints and habitual browsing and an inability to distinguish between legitimate and phishing websites [4] .

Phishing is a multifaceted issue that is challenging to comprehend and analyze, as it encompasses both technical and social attack vectors, with no universal solution to completely prevent or eradicate it [4]. This complexity underscores the necessity of developing intelligent and adaptive solutions that can be sustained over time. While various methodology-based approaches have been proposed, they often encounter a significant challenge: a high rate of false positives, where legitimate web pages are mistakenly classified as phishing sites [5]. This issue primarily arises from the limitations of existing methods, such as reliance on static techniques like blacklisting and whitelisting, the absence of human intelligence and expert oversight, and constraints in timeliness and scalability [6].

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

1.1 Problem Statement

Phishing remains one of the most prevalent and evolving cybersecurity treats, exploiting human vulnerabilities and technical loopholes to deceive users into divulging sensitive information [7]. Traditional phishing detection methods, such as blacklisting and rule-based systems, often fall short due to their reliance on static databases, inability to detect novel threats and high false-positive rates [8]. Machine Learning (ML) techniques have shown promise in phishing detection by identifying patterns in malicious URLs, website features and user behaviours. However, existing ML-based approaches often struggle with scalability, adaptability to emerging threats and the trade-off between detection accuracy and computational efficiency [9].

To address these challenges, this study aims to implement an ensemble machine learning approach that integrates SVM with LSTM networks, which is a deep learning technique to improve phishing detection accuracy. SVMs are known for their effectiveness in handling high-dimensional data and distinguishing between phishing and legitimate websites [10] while deep learning models, particularly neural networks, excel at extracting complex patterns from large datasets [11]. By leveraging the strengths of both techniques, this approach seeks to enhance detection performance, reduce false positives and ensure adaptability to evolving phishing tactics. This study benchmarks its approach against existing machine learning-based phishing detection models, evaluating key performance metrics such as accuracy, precision, recall and F1-score.

2.0 Methodology

The study uses an experimental design to implement and evaluate the efficiency and effectiveness of an ensemble SVM and LSTM algorithm in accurately classifying URLs as either phishing or legitimate. The study will measure and analyze various URL features to understand their impact on the classification accuracy. The steps followed by the researcher in the study are thus:

2.1 Review of related literature

A thorough review of research carried out by other researchers in area of phishing detection was conducted, the followings are some related studies by different authors:

According to [12], the authors proposed a novel strategy for phishing prevention by leveraging deep learning techniques to enhance online security. The proposed system utilizes machine learning models to predict the likelihood of a website being malicious by analyzing its URLs and Universal Resource Identifiers (URIs). Various classification techniques, including logistic regression, principal component analysis and prioritization methods, were employed to optimize model performance. Among these, logistic regression was identified as the most effective approach. Experimental evaluation demonstrated that the system achieved an accuracy of approximately 98% in detecting fraudulent websites. As online threats continue to escalate, the study confirms the efficacy of the proposed techniques in safeguarding users against phishing attacks.

[13], proposed a model that classifies websites as either legitimate or phishing based on their visual content and URLs. The model employs a CNN to extract key features from webpage images and structures, thereby facilitating the detection of phishing threats. The model achieved an accuracy rate of 99.67%, demonstrating its effectiveness in identifying online phishing attempts. [14] introduced a method that leverages Generative Adversarial Networks (GANs) to generate phishing samples based on URLs. The adversarial examples produced by their approach were capable of deceiving black-box phishing detection systems, including those that utilize sophisticated techniques such as intra-URL similarity analysis. Their findings highlight the effectiveness of GANs in crafting adversarial phishing instances capable of bypassing advanced security mechanisms.

To improve the accuracy of fraudulent website detection, [15] incorporated a feature selection algorithm into a majority voting-based ensemble learning framework. Their method was evaluated against several classification models, including RF and LR. The experimental results demonstrated that their approach could achieve detection accuracies of up to 95%, surpassing the performance of existing technologies, which typically yield accuracy rates ranging between 70% and 95%.

In a study conducted by [16] seven machine learning models LR, KNN, SVM, NB, DT, RF, and GBM were developed for phishing detection. The experimental results indicated that the Gradient Boosting model, particularly when used in combination with Random Forest, outperformed the other approaches in terms of classification accuracy and overall effectiveness.

Recent advances in deep learning and NLP techniques have significantly improved the effectiveness of spam and phishing detection systems [17]. [18] proposed a Semantic Convolutional Neural Network (SCNN) model that combines CNN with a semantic layer utilizing Word2Vec to embed words into vector representations. Their model was evaluated on two datasets Twitter and SMS spam achieving accuracy rates of 94.40% and 86.5%, respectively. These results demonstrated the potential of integrating semantic understanding into deep learning models for improved classification performance.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

Similarly, [19] developed a deep learning approach for identifying spam tweets using a Multi-Layer Perceptron (MLP) and Word2Vec embeddings. Tested on a large dataset of over 376,000 spam and 73,000 non-spam tweets, their method achieved an accuracy of 99.35%. This high accuracy indicates the effectiveness of combining NLP techniques with deep learning classifiers in social media spam detection.

Focusing specifically on phishing emails, [18] evaluated various machine learning algorithms including NB, SVM, RF, DT and KNN on a phishing email dataset. Their results showed that RF achieved an impressive accuracy of 99.45%, though the study was limited to phishing emails and not extended to other spam types.

Deep learning models have been further applied to spam detection in SMS and social media contexts. For instance, [20] used RNN and LSTM networks for spam SMS filtering, attaining a peak accuracy of 98% with relatively low processing time. Enhancing LSTM's performance. [21] conducted a study on SMS spam detection across two languages, namely English and Arabic, employing a hybrid CNN-LSTM approach. Their findings demonstrated that the CNN-LSTM model outperformed alternative techniques, achieving an accuracy of 98.37%. [22] proposed a spam detection model that incorporated the LSTM algorithm, applied to real-time Twitter data for both spam detection and sentiment analysis. The study compared several approaches, including traditional machine learning methods such as Naïve Bayes and SVM, alongside deep learning techniques. Experimental results revealed that the LSTM-based model delivered superior performance, achieving an accuracy of 98.74% in spam detection and 73.81% in sentiment analysis.

The literature reveals that many existing phishing detections approaches predominantly depend on URL signatures and static features, often neglecting the integration of multiple classifiers or diverse data types, which limits their detection accuracy and adaptability [23]. Furthermore, datasets used in prior studies tend to be small, static, or lack sufficient diversity, impairing the models' ability to generalize to emerging threats [24]. Additionally, there is limited exploration of hybrid deep learning architectures that combine models like CNNs and LSTMs to exploit their complementary strengths, which could enhance detection performance [25]. These gaps hinder the development of comprehensive, scalable, and resilient phishing detection systems.

This study proposes an ensemble approach combining SVM and LSTM networks to leverage both static and sequential features extracted from diverse datasets. The ensemble architecture aims to capitalize on the strengths of both models, addressing the limitations of previous approaches. Rigorous evaluation using multiple metrics will demonstrate significant improvements.

2.2 Data collection and Labeling

The PKU-PhishURL-2023 dataset was used for our study comprising of phishing and legitimate URLs. It has a total of 343,525 records. The datasets were downloaded in CSV file format. Phishing URLs consist of 167,582 records from PhishTank and Kaggle, while legitimate URLs consist of 175,943 records from UC Irvine Machine Learning Repository and Kaggle. The PhishTank and Kaggle are reputable platforms known for curating up-to-date phishing data. Each phishing instance was labeled with a class value of 1, denoting its malicious nature. This labeled data is crucial for supervised machine learning models to distinguish phishing patterns effectively. Similarly, the segment of the dataset comprising legitimate URLs, were labeled with a class value of 0, signifying its authenticity.

2.3 Data preprocessing

The datasets for the study were preprocessed using Sckit-learn and TensorFlow python libraries to ensure that they are clean, consistent and ready for training. This process removes noise, fills in missing values and corrects inconsistencies to improve learning accuracy [26]. The steps followed include:

1. Data Cleaning: The process of data cleaning was used to remove outliers, handle missing values, remove duplicate records and correct inconsistencies in our datasets.
2. Data Integration: The datasets were combined to increase data volume and conflicts between sources were resolved.
3. Data Reduction: our datasets were reduced through feature selection and dimensionality reduction while preserving key patterns.
4. Data Transformation: The datasets were converted into formats suitable for training. This process include normalization (example; scaling values between 0 and 1), discretization, and smoothing.

2.4 Data Preparation

For data preparation, two steps were employed as follows:

A. Feature extraction: feature extraction was used to transform raw data into numerical features that was employed by machine learning models. This was achieved using SelectKBest and Recursive Feature Elimination Python functions. For the purpose of this study, the following features were extracted:

i. Address Bar Features: such as url_length, num_dots, num_hyphens, num_at, num_question_marks and num_equals.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

ii. Domain Based Features: These features include domain age, domain expiration and WHOIS availability (has_WHOIS).

iii. HTML and JavaScript-based features: such as num_iframes, num_forms, num_input_fields, num_scripts.

B. Feature selection: Features represent the attributes of a dataset that models use for learning. Not all features contribute to accuracy; some may be redundant or irrelevant. Feature selection, therefore, improves efficiency and accuracy by retaining only meaningful variables [27]. The following feature selection methods were applied in the study in order to obtain features with greatest impact on prediction. The methods employed for feature selection are:

i. Support Vector Machine: Feature selection was done using the SVM, which inherently performed a form of feature selection, it focused on the most relevant features. This was achieved by calculating the support vectors, which are the data points closest to the decision boundary. The algorithm assigned higher importance to these support vectors and features related to them are prioritized. By selecting only the most critical features, SVM helps reduce computational complexity and overfitting, thus improving model performance.

ii. Recursive Feature Elimination: The Recursive Feature Elimination (RFE) method iterates to remove less significant features, focusing on those that enhance predictive accuracy. RFE ranked the feature importance, removed the least important features and rebuilt the model until a desired feature subset was obtained.

iii. Univariate Feature selection: This method selected the features that have the strongest relationship with the target variable based on statistical tests.

3.0 Proposed Approach

The proposed architecture of the study presents an ensemble model that integrates SVM and LSTM networks for the detection of phishing websites. This approach leverages the strengths of both machine learning and deep learning paradigms to enhance detection accuracy and generalization. The system begins with the collection of phishing and legitimate URLs from publicly available datasets repositories. These URLs undergo data preprocessing, data preparation which involves feature extraction and selection where both static and sequential features such as URL structure, domain properties and content patterns were derived. Figure 1, depicts the proposed architecture of the study.

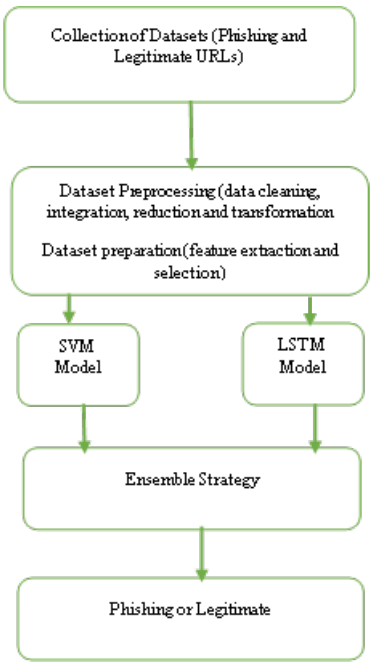

Figure 1: Proposed architecture

4.0 Model Equation

To formulate mathematical model equation for the SVM classifier decision function, some assumptions were made: Let X^*_s be the input static feature derived from our dataset, a_i is the support vector weights, y_i is the corresponding support vector labels, x_i is the support vector, $\acute{K}(x_i,x)$ is the Kernel function and b is the bias term, where each instance is represented as $x_i \in X^*_s$.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

4.1 The SVM classifier computes the decision function as

$$f_{svm}(X^*_s) = \sum_{i=1}^{N_{sv}} \alpha_i y_i K(x_i, X^*_s) + b, \quad (1)$$

where (X^*_s) is the static feature, a_i is the support vector weight, y_i is the corresponding support vector labels, x_i is the support vector, $\acute{K}(x_i,x)$ is the Kernel function and b is the bias term.

Predicted class label is given by:

$$p_{svm} = \sigma(f_{svm}(X^*_s)), \quad (2)$$

where p_{svm} is the predicted class label, σ is the sigmoid function and X_s^* is the static feature computed by SVM classifier**.**

The kernel function, $\acute{K}(x_i,x)$ is computed for each support vector x_i and static feature, X_s^* as input in (1). The sum of the support vector weights and the corresponding support vector labels are computed, then multiplied by the kernel function. The bias term is added to obtained result.

To obtain the SVM probability output, the results of equation (1), is multiplied by a sigmoid function in equation (2). The sigmoid function used binary classification which gives phishing probability with value between 0 and 1.

4.2 Sequential Feature Classification using LSTM

The LSTM networks classify sequential features by processing as inputs sequentially and outputting a class probability.

Given: Selected sequential X_t^* features (sequence data),

LSTM network was trained to output probability estimates in equation (3)

$$p_{lstm} = f_{lstm}(X^*_t) \in (0,1) \quad (3)$$

where p_{lstm} is the probability estimate and X^*_t is the sequential feature.

The LSTM predicted class label is given as:

$$\hat{y}_{lstm} = \begin{cases} 1, & if\ p_{lstm} \geq 0.5 \\ 0, & otherwise \end{cases} \quad (4)$$

4.3 Ensemble Decision Fusion

The predictions from both classifiers were combined by adopting a weighted ensemble combining approach in equation (5):

$$P_{ensemble} = W_s.P_{SVM} + W_t.P_{LSTM} \quad (5)$$

where *Pensemble* is the ensemble prediction, W_s is the weight of the static feature, P_{SVM} is the probability estimate of static feature, W_t is the weight of the sequential feature and P_{LSTM} is the probability estimate of sequential feature

The final classification is given by equation vi below:

$$\hat{Y} = \begin{cases} 1, & \text{if } p_{ensemple} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

In the final classification, if the value obtained is 1, that shows that the URL is phishing. Contrarily, if the value obtained is 0, that shows that the URL is legitimate.

4.4 Evaluation metrics

In order to assess the performance of the ensemble approach, the following evaluation metrics will be employed:

i) **Accuracy:** Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted instances to the total number of predictions:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

Where TP is the True Positives, TN is the True Negatives, FP is the False Positives and FN is the False Negatives

ii) **Precision:** Precision quantifies the model's ability to correctly identify phishing and legitimate URL. High precision means fewer false alarms. It given by equation vii.

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (8)$$

iii) **Recall** ***(Sensitivity or True Positive Rate):*** Recall reflects the model's ability to detect all actual Phishing URL.

$$\textbf{Recall} = \frac{TP}{TP+FN} \quad (9)$$

iv) **F1-Score:** The F1-Score provides a harmonic mean of precision and recall, offering a balance between them.

$$\textbf{F1-Score} = \frac{2 \times \text{Precsion} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

5.0 Results Analysis

5.1 Feature Selection Result Analysis

i) Support vector machine: Features selected are: url_length, num_hyphens, num_digits, num_letters, domain_length, domain_age, domain_experation, num_input_fields, num_scripts, num_external_scripts, num_internal scripts. Total of 11 features were selected. The feature importance is shown in figure 2

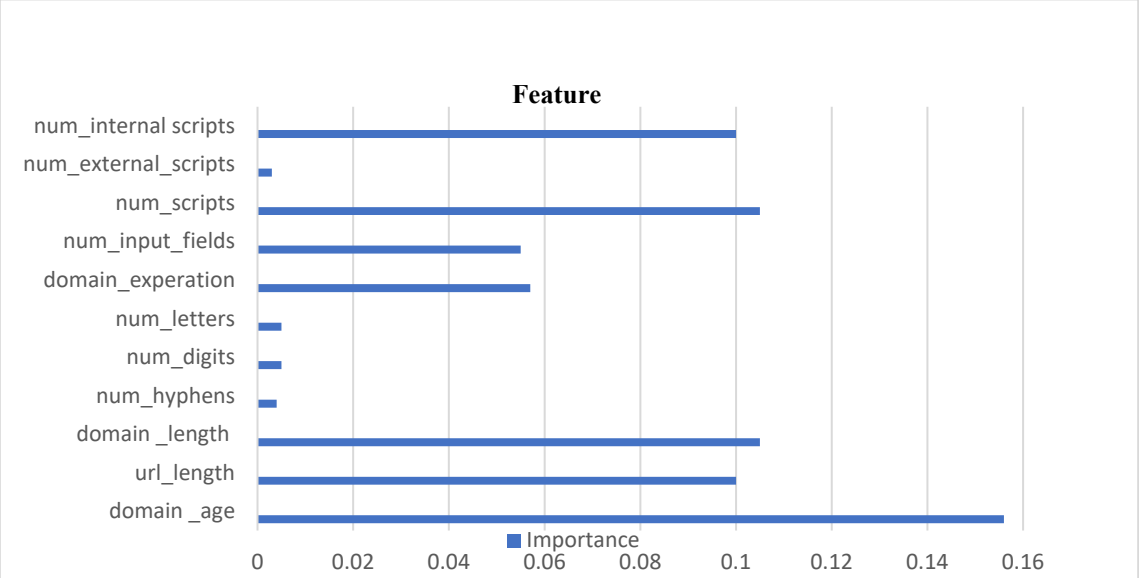

Figure 2: SVM Feature importance

ii) Recursive feature elimination: From the dataset, the feature that were selected using RFE method include num_hyphens, num_at, num_equals, num_percent, num_phishing_words, prefix_suffix, num_subdomain, has_whois, num_iframe, num_forms. Total of 10 features were selected. The features are ranked as shown in figure 3

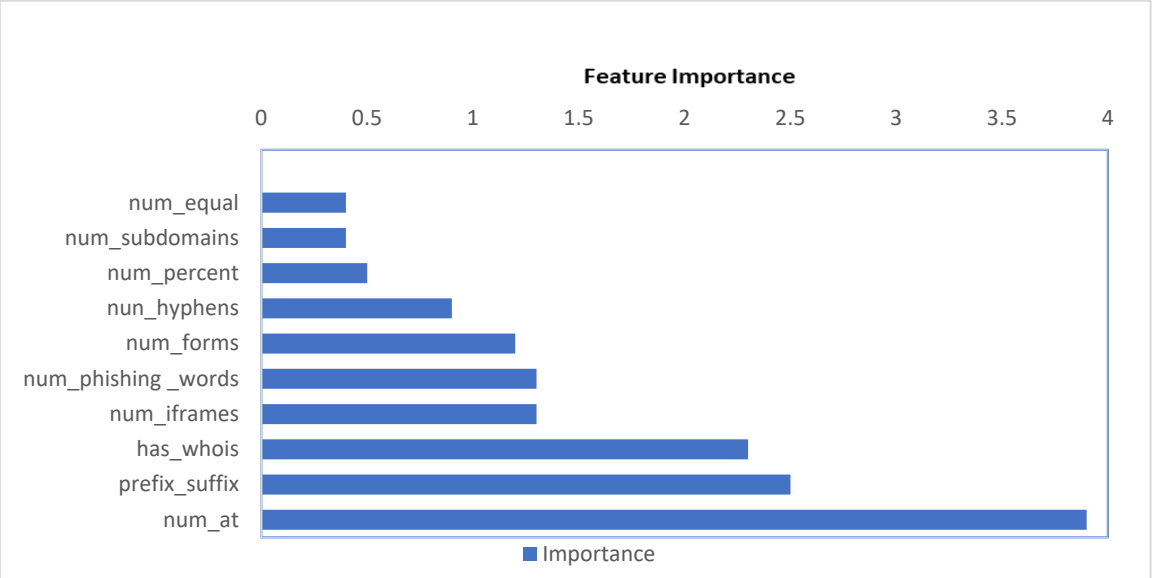

Figure 3: RFE feature importance

iii) Univariate Feature selection: The top 12 features selected using the univariate method include url_length, num_percent, num_digits, num_letters, domain_length, domain_age, domain_expiration, num_forms, num_input_fields, num_scripts, num_external_scripts, num_internal_scripts. The features' importance is as shown below in Figure 4.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

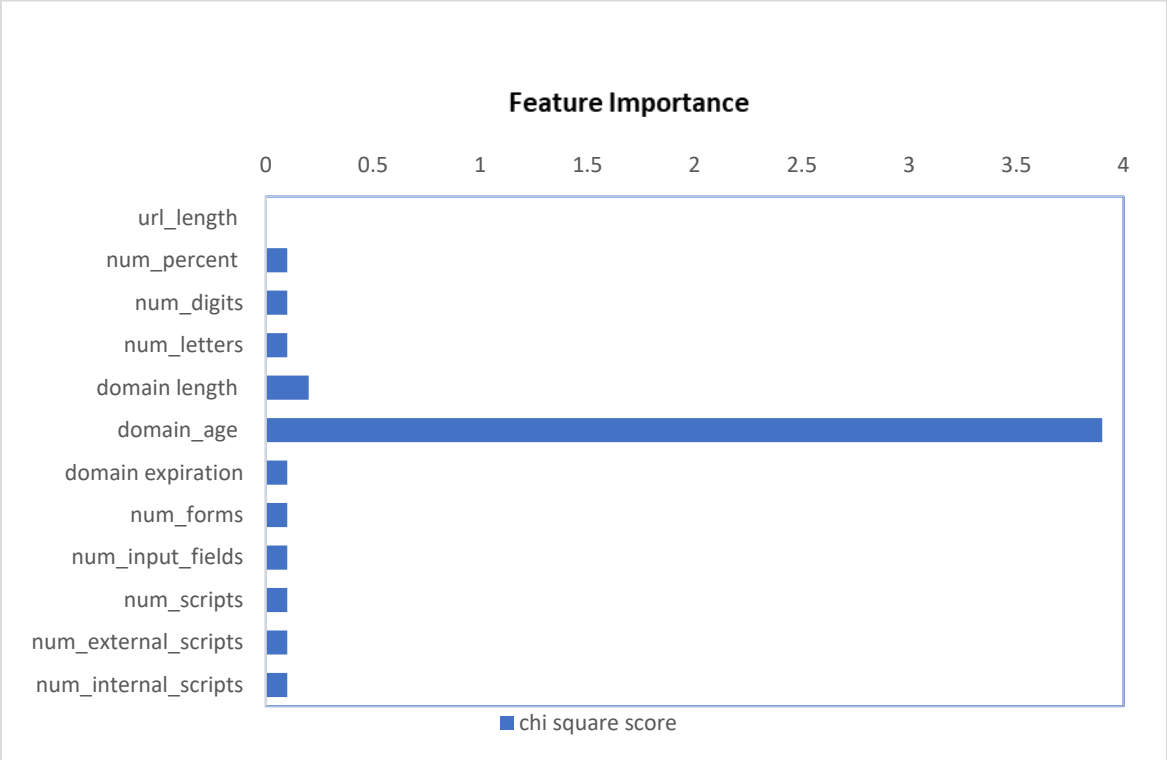

Figure: 4 Univariate feature importance

5.2 Feature ranking result analysis

The researcher selected only the features with a mean importance score of 0.15 or higher in order to train the ensemble approach. By setting this cut-off, the researcher was able to filter out less relevant features and focus on the ones that have the greatest impact on prediction. This not only sharpens the model's accuracy but also reduces the complexity of the dataset, making the model both more efficient and easier to interpret. The features that surpassed the threshold include: domain_age (0.7500), num_external_scripts (0.273706), prefix_suffix (0.273266), num_scripts (0.265107), domain_length (0.260522), num_at (0.250947), url_length (0.215948), has_whois (0.203078), domain_expiration (0.177514), num_digits (0.170888), num_internal_scripts (0.157757) and num_forms (0.157071). These are the most influential features based on their average importance scores across different selection methods.

To put these results into perspective, existing feature selection techniques were also reviewed to compare how many features have been applied in phishing website detection. The findings are summarized in Table 1.

Table 1: Comparison of feature selection with existing feature selected methods

Author	Feature Selection Method	Total No. of features	No. of Selected Features
[9]	Fisher's Score	49	30
	Information Gain	49	10
	Mean Absolute Difference	49	31
	Correlation Coefficient	49	23
	Variance Threshold	49	23
[28]	Feature selection by Omitting Redundant Features	48	22
	Feature selection by Filter Method	48	9
Researcher	Wrapper method (SelectKbest)	49	15

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

Author	Feature Selection Method	Total No. of features	No. of Selected Features
	Filter based method (RFE)	49	10
	Embedded method (SVM)	49	11

6.0 Result and Discussion

The selected features were divided into training and testing datasets using a 70:30 split ratio. The training set was used to build the LSTM model, experimenting with different unit sizes of 60, 80, and 100. After training, the test set was applied to evaluate the model's approach performance. To ensure a comprehensive assessment, we employed multiple evaluation metrics, including accuracy, precision, recall, F1-score and false positive rate. The results of these performance metrics are presented in Table 2.

Table 2: Performance metrics for the ensemble SVM-LSTM approach

Metrics	Ensemble Model		
	60 unit	80 unit	100 unit
Accuracy	93.48%	93.59%	97.58%
Precision	93.46%	93.52%	93.54%
Recall	94.47%	93.81%	96.1%
F1-Score	93.2%	93.08%	95.78%
False Positive Rate	5.04%	3.98%	2.55%

Among the tested configurations, the Ensemble model with 100 units demonstrated the best performance across all evaluation metrics achieving highest accuracy of 97.58%. This indicates strong capability of the approach in correctly distinguishing between legitimate and phishing websites. Precision remained fairly stable across configurations, with the 100-unit model slightly ahead at 93.54%. Recall, however improved as the units increase, reaching 96.10% at 100 units, indicating the model's ability to correctly identify a higher proportion of phishing websites. F1-score, which balances precision and recall, also peaked with the 100-unit configuration at 95.78%, confirming strong overall classification performance. Similarly, the FPR decreased as the number of units increased, with the lowest value of 2.55% recorded at 100 units. This means the model generated fewer false alarms compared to the other configurations.

The results show that the proposed SVM-LSTM approach outperforms the baseline. It achieved an accuracy of 97.58%, with strong precision (93.54%), recall (96.1%), and an overall F1-score of 95.78% indicating a balanced and reliable performance in phishing detection. In comparison with the baseline paper by [9] in Table 3, using fisher score method (FS) reached an accuracy of 93.79%, but other performance metrics were not reported. Additionally, model presented by [28] also in Table 3, using filter method (FM) attained an accuracy of 97.30%. This highlights the effectiveness of the proposed ensemble approach in delivering more comprehensive and accurate results compared to other approaches. Only accuracy was used for the comparison because it was the only evaluation used by the baseline paper.

Table 3: Comparison of result with the baseline paper

Reference	Model	Accuracy	Precision	Recall	F1-Score
Proposed Model	SVM-LSTM	97.58%	93.54%	96.1%	95.78%
[9]	FS-LSTM	93.79%	-	-	-
[28]	FM-RF	97.30%	-	-	-

7.0 Conclusion

An ensemble phishing URL detection system was proposed as a key step in strengthening online security and protecting users from internet fraud. Machine learning algorithms are powerful techniques in developing models for detecting phishing attempts, but the performance of such models depends heavily on the dataset, selected features and chosen machine learning algorithms for training and testing. In this study, both legitimate and phishing URLs were acquired from available dataset repositories and various features were extracted and evaluated using feature selection methods to identify the most relevant ones. Using these optimized features, an SVM-LSTM model was trained and tested. The results showed that the model achieved strong accuracy of 97.58% with 100 units while keeping false positive rates low. This suggests that integrating multiple classifiers can enhance detection performance. The low false positive rate of 2.55% demonstrated the effectiveness of the approach in minimizing

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

incorrect phishing alerts, thereby reducing potential user inconvenience and enhancing trust in the system, The consistent high precision and recall values across different configurations highlight the balanced ability of the ensemble approach to accurately identify both legitimate and phishing URLs, contributing to more reliable online security measures. The results obtained implied that the optimized ensemble approach not only improves detection accuracy but also ensures practical usability with minimal false alarms. Exploration of other machine learning algorithms for developing intelligent phishing detection systems is recommended to improve model performance.

References

[1] X. Yu, "Phishing Websites Detection Based on Hybrid Model of Deep Belief Network and Support Vector Machine," *IOP Conf Ser Earth Environ Sci*, vol. 602, no. 1, 2020, doi: 10.1088/1755-1315/602/1/012001.

[2] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," *IEEE Access*, vol. 10, pp. 65703–65727, 2022, doi: 10.1109/ACCESS.2022.3183083.

[3] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J Ambient IntellHumanizComput*, vol. 10, no. 5, pp. 2015–2028, 2019, doi: 10.1007/s12652-018-0798-z.

[4] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network," *WirelCommun Mob Comput*, vol. 2021, 2021, doi: 10.1155/2021/8241104.

[5] H. Taherdoost, "Insights into Cybercrime Detection and Response: A Review of Time Factor," *Information (Switzerland)*, vol. 15, no. 5, 2024, doi: 10.3390/info15050273.

[6] M. Chitgopekar, A. R. Naik, and M. Vyas, "Artificial Intelligence and Large Language Models in Mental Healthcare: A Systematic Review," *Artificial and Cognitive Computing for Sustainable Healthcare Systems in Smart Cities: Volume 3*, pp. 195–215, 2024, doi: 10.1002/9781394297443.ch10.

[7] M. S. Kheruddin*et al.*, "Phishing Attacks: Unraveling Tactics, Threats, and Defenses in the Cybersecurity Landscape," *Authorea Preprints*, 2024.

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

[9] S. Ali, "Comparison of Feature Selection Methods in the Aspect of Phishing Attacks," no. November 2022, 2023.

[10] S. Alnemari and M. Alshammari, "applied sciences," *Detecting Phishing Domains Using Machine Learning*, pp. 13, 4649, 2023, doi: https://doi.org/ 10.3390/app13084649.

[11] C. Xiao, E. Choi, and J. Sun, "Review Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review," vol. 25, no. June, pp. 1419–1428, 2018, doi: 10.1093/jamia/ocy068.

[12] Pavansai and G. G. sai Ziaul Haque Choudhury, "Classification of Phishing Website Using Hybrid Machine Learning Techniques," *Int J Innov Sci Res Technol*, vol. 8, no. 7, pp. 1385–1390, 2023.

[13] M. Elsadig*et al.*, "Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction," *Electronics (Switzerland)*, vol. 11, no. 22, 2022, doi: 10.3390/electronics11223647.

[14] A. Andreadis, "Missing Values Imputation on Multivariate Time Series in the field of Agriculture by Declaration of Authorship," no. July, 2022.

[15] G. Obaido*et al.*, "An Improved Framework for Detecting Thyroid Disease Using Filter-Based Feature Selection and Stacking Ensemble," *IEEE Access*, vol. 12, no. May, pp. 89098–89112, 2024, doi: 10.1109/ACCESS.2024.3418974.

[16] S. Pathan, O. Maddala, K. Naga Durga Saile, and P. Singh, "Phishing Websites Detection using Machine Learning," *Proceedings - 2nd International Conference on Advancement in Computation and Computer Technologies, InCACCT 2024*, vol. 7, no. 2, pp. 29–33, 2024, doi: 10.1109/InCACCT61598.2024.10551073.

[17] M. R. Al Saidat, S. Y. Yerima, and K. Shaalan, "Advancements of SMS Spam Detection: A Comprehensive Survey of NLP and ML Techniques," *Procedia Comput Sci*, vol. 244, pp. 248–259, 2024, doi: 10.1016/j.procs.2024.10.198.

[18] A. Sawsan, S. Amani, A. M. R. AlSobeh, and A. A. Magableh, "Beyond Word-Based Model Embeddings: Contextualized Representations for Enhanced Social Media Spam Detection," 2024.

[19] H. Loucif, "A Hybrid Deep Learning Approach for Spam Detection in Twitter," vol. 29, no. 1, pp. 117–123, 2024.

[20] L. Larasati, S. Saadah, and P. E. Yunanto, "Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Methods to Forecast Daily Turnover at BM Motor Ngawi," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 7, no. 1, p. 141, 2024, doi: 10.24014/ijaidm. v7i1.27643.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026

[21] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid CNN-LSTM model for SMS spam detection in arabic and english messages," *Future Internet*, vol. 12, no. 9, pp. 1–16, 2020, doi: 10.3390/FI12090156.

[22] A. P. Rodrigues *et al.*, "Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques," *ComputIntellNeurosci*, vol. 2022, 2022, doi: 10.1155/2022/5211949.

[23] H. Sebestyen, D. E. Popescu, and R. D. Zmaranda, "A Literature Review on Security in the Internet of Things: Identifying and Analysing Critical Categories," *Computers*, vol. 14, no. 2, 2025, doi: 10.3390/computers14020061.

[24] M. Almania, A. Zainal, F. A. Ghaleb, A. Alnawasrah, and M. Al Qerom, "Adaptive Intrusion Detection System with Ensemble Classifiers for Handling Imbalanced Datasets and Dynamic Network Traffic," *Journal of Robotics and Control (JRC)*, vol. 6, no. 1, pp. 114–123, 2025, doi: 10.18196/jrc.v6i1.23648.

[25] C. L. Cikambasi, L. M. Muriira, and R. M. Murungi, "Deep Learning Network Intrusion Detection with the Conv1d-Lstm Model: Integrating CNN and LSTM For Superior Performance," *International Journal of Professional Practice*, vol. 12, no. 4, pp. 41–49, 2024, doi: 10.71274/ijpp.v12i4.475.

[26] I. AbdulNabi and Q. Yaseen, "Spam email detection using deep learning techniques," *Procedia Comput Sci*, vol. 184, no. 2019, pp. 853–858, 2021, doi: 10.1016/j.procs.2021.03.107.

[27] B. Sun *et al.*, "Leveraging Machine Learning Techniques to Identify Deceptive Decoy Documents Associated with Targeted Email Attacks," *IEEE Access*, vol. 9, pp. 87962–87971, 2021, doi: 10.1109/ACCESS.2021.3082000.

[28] S. Shabudin, N. S. Sani, K. A. Z. Ariffin, and M. Aliff, "Feature selection for phishing website classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, pp. 587–595, 2020, doi: 10.14569/IJACSA.2020.0110477.

Received: 29/10/2025; Revised: 26/12/2025; Accepted: 03/01/2026; Published: 02/03/2026