FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 5, May, 2025, pp 322 – 327
DOI: https://doi.org/10.33003/fjs-2025-0905-3685

AN OPTIMIZED LSTM MODEL FOR PHISHING WEBSITE DETECTION USING PARTICLE SWARM OPTIMIZATION AND HYPEROPT TECHNIQUES

***Ukashatu Adamu, Umar Iliyasu and Tasiu Suleiman**

Department of Computer Science, Faculty of Computing Federal University Dutsin-Ma.

*Corresponding authors' email: ukashaadamaliyu@gmail.com

ABSTRACT

Phishing remains a prevalent cybersecurity threat that exploits human trust to steal sensitive information. Traditional detection methods, such as blacklisting and rule-based approaches, often fail to adapt to the rapidly evolving nature of phishing websites. In contrast, machine learning and artificial intelligence offer powerful solutions by identifying phishing patterns based on URL structures and website behavior. While hyperparamter tuning is a crucial step in machine learning, its impact on phishing detection models remain under-examined, highlighting a need for more research in this area. This study addresses this gap by developing an LSTM-based phishing detection model and optimizing it using two hyperparameter tuning techniques: Particle Swarm Optimization (PSO) and HyperOpt. The results demonstrate that HyperOpt outperforms PSO, achieving an accuracy of 93.12% compared to 92.00% with PSO. This superiority is attributed to Bayesian optimization and the Tree-structured Parzen Estimator (TPE), which enable more efficient hyperparameter selection. The findings emphasize the importance of hyperparameter tuning in improving phishing detection accuracy and enhancing cybersecurity defenses against evolving threats.

Keywords: Phishing website, Hyperparameter, Hyperparameter tuning, Particle swarm, HyperOpt

INTRODUCTION

Phishing is a common cybersecurity threat that uses human trust to deceive individuals into sharing sensitive information (Ali, 2017). Traditional methods like blacklisting and rule-based techniques are often insufficient due to their inability to adapt to the rapidly evolving landscape of phishing websites. Machine learning and artificial intelligence have emerged as powerful tools for identifying phishing websites based on patterns and traits extracted from URLs and website behaviors (Kulkarni and Brown, 2019). The rapid growth of online platforms and e-commerce has intensified the impact of phishing attacks, with cybercriminals constantly refining their methods. URL-centric features, such as domain age, special characters, and HTTPS utilization, play a crucial role in detecting phishing websites (Mahajan and Siddavatam, 2018). However, static feature-based frameworks often struggle to adapt to zero-hour phishing incidents, necessitating the need for dynamic and scalable machine learning paradigms capable of real-time detection.

Various machine learning methodologies, including classical models like Random Forests, Support Vector Machines (SVM), Decision Trees, and Convolutional Neural Networks (CNN) have been explore for classification and predictive tasks (Ndabula, Olanrewaju, and Echobu, 2023). CNNs have been used to convert feature vectors into visual representations, presenting an innovative method for phishing detection through deep learning (Ajik, Obunadike, and Echobu, 2023). However, many of these studies tend to rely on default model parameters for phishing detection, which can restrict the models from reaching their optimal performance and achieving higher detection accuracy (Kulkarni, 2023). This study aims to address these challenges by utilizing deep learning algorithms combined with superior feature extraction and selection methodologies.

Literature Review

The research conducted by Sonowal and Kuppusamy (2020) developed PhiDMA, a robust phishing detection model for general users and visually impaired individuals. The study introduced a multi-layered phishing detection approach, incorporating filters like whitelists, URL features, lexical signatures, string matching, and accessibility score analysis to identify phishing sites. The methodology involved creating a browser plugin and testing its performance using real-world data from publicly available repositories. The model achieved a high accuracy of 92.72%, outperforming other techniques.

Safi and Singh (2023) analyzed 80 research papers on phishing website detection techniques, focusing on heuristic, machine learning, and deep learning approaches. The study found machine learning, particularly Random Forest, was the most used algorithm, with datasets like PhishTank and Alexa being prominent. Convolutional neural networks achieved the highest detection accuracy of 99.98%. However, the study faced limitations, such as insufficient coverage of newer datasets and challenges in evaluating generalization across different phishing attack types.

Also, Palaniappan et al. (2020) used machine learning to detect malicious domains using domain name features, host-based attributes, and web-based data. They proposed a logistic regression-based classifier trained on 20,000 domain names, extracting features like DNS-based attributes, lexical patterns, web rankings, and blacklists. The model achieved a modest accuracy of 60%, but the study highlighted the potential for improved models and multi-class classification.

Anwekar and Agrawal (2022) focuses on detecting phishing websites using machine learning algorithms. They compare the performance of decision tree, random forest, and SVM classifiers based on accuracy, false positive, and false negative rates. The study uses a dataset of 36,711 URLs and features extracted from Alexa.com and Phishtank.com. The random forest algorithm outperforms others with 97.14% accuracy. However, the study highlights gaps like limited dataset diversity and over-reliance on specific features, suggesting future research could explore hybrid methods.

Machine learning methods were also explored by Renusree (2021) for detecting phishing websites using Feed-Forward Neural Networks. The study compares the performance of Decision Tree, Random Forest, and Support Vector Machine classifiers in detecting phishing URLs. Random Forest outperforms all three, highlighting the need for intelligent

anti-phishing solutions. The study suggests further refinement and integration with other approaches for comprehensive detection of zero-hour phishing attacks.

Garje et al. (2021) uses machine learning algorithms to detect phishing websites, classifying them as legitimate or phishing. They use K-Nearest Neighbors, Naive Bayes, Decision Trees, and Gradient Boosting to classify websites. The Decision Tree algorithm achieved the best performance, balancing recall and precision. However, the research highlights gaps in addressing zero-hour phishing attacks and calls for improved generalization in future models.

Priya (2023) explores machine learning algorithms for detecting phishing websites. The research evaluates various models, including Random Forest, Decision Tree, SVM, KNN, and CNN, to improve accuracy and efficiency. Using PhishTank datasets, the study found Random Forest as the most effective algorithm, achieving an accuracy of 97.14%. However, challenges like limited dataset size and the need for continuous model refinement are highlighted, emphasizing the need for continuous improvement.

Research on machine learning and deep learning models for phishing detection has shown promising results, but many studies overlook the importance of hyperparameter tuning. Hyperparameter tuning is effective in improving model accuracy, but its application in phishing detection remains limited. This study aims to bridge this gap by applying hyperparameter tuning techniques, specifically Particle Swarm Optimization (PSO) and HyperOpt, to improve phishing detection models, demonstrating its impact on enhancing accuracy and efficiency. By integrating dynamic attributes and hyperparameter optimization techniques, this research intends to overcome the shortcomings of existing methodologies and provide a scalable solution for phishing website identification. This initiative not only strengthens cybersecurity frameworks but also establishes a foundation for adaptive systems that can evolve with the threats they are designed to counter.

MATERIALS AND METHODS

The tools, procedures and methods deployed in the implementation of the model are highlighted and discussed as shown in figure 1.

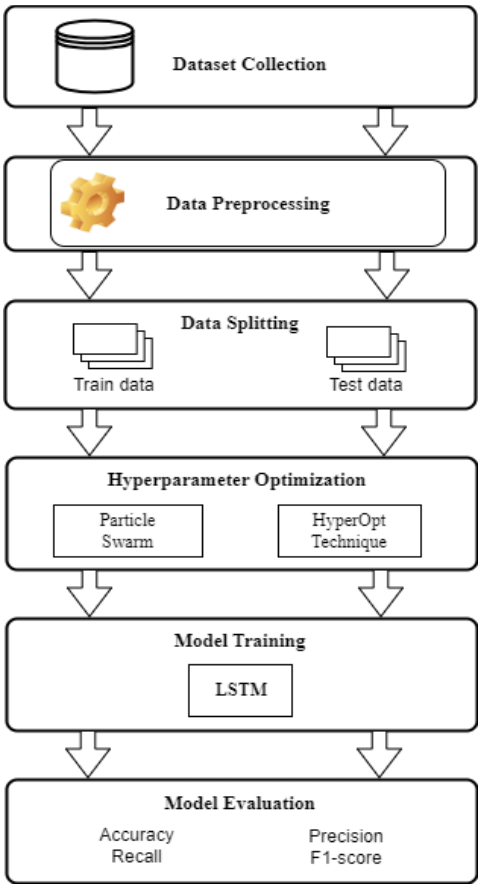

Figure 1: System Architecture

Data Collection and Annotation

This research uses an open source dataset called PhiUSIIL, published by (Prasad and Chandra, 2023) which contains 235,795 data on phishing websites. The dataset includes legitimate and phishing URLs in CSV format. The data was manually checked to ensure no missing labels and appropriate labels for the research. The data was assigned binary labels, with 134,850 mapped as legitimate and 100,945 as phishing URLs.

Data Preprocessing

Data preprocessing was conducted to prepare the dataset for effective model training. Initially, string conversion was applied to ensure all URLs were properly formatted. Label encoding was then used to preserve the original labeling structure of the dataset. The data was subsequently vectorized, transforming raw inputs into numerical representations suitable for machine learning algorithms. Lastly, outlier detection was performed to identify and remove odd values that could negatively impact the model's performance.

Data Spliting
The preprocessed data was divided into a train and test set based on the parento rule, with 80% used for training models on phishing website features, and the remaining 20% for testing their performance in detecting such websites.

Hyperparameter Tuning
This research has adopted the Particle Swarm Optimization and HyperOpt techniques to be used in tuning the hyperparameters of the LSTM model. The LSTM hyperparamters tuned are discussed below;
LSTM Units: which determines the number of memory cells in each LSTM layer, thereby, directly affecting the model's ability to learn and retain sequential patterns (Hochreiter & Schmidhuber, 1997). A range of 32 to 128 was specified for the LSTM units.
Learning_rate: controls how much the model adjusts its weights during backpropagation (Goodfellow et al., 2016). A learning_rate ranging from -3 to 3 was specified for the LSTM algorithm.

Model Building
The model was trained using the Long-Short Term Memory (LSTM). The entire process was implemented on Google Colab, a free cloud-based Jupyter Notebook service that enables Python code writing and execution. Colab is highly functional due to its cloud hosting, allowing smooth code execution without personal computer resources. It is shareable, easy to access, and supports a large number of Python libraries. Its GPU and TPU memories are suitable for deep learning models.

Model Evaluation
The models were evaluated for their effectiveness in detecting phishing websites based on training data features. The evaluation assessed if the models made accurate predictions and if there was any overfitting or underfitting due to class imbalance. Key metrics used included:
Accuracy: evaluates how well the model's predictions match the actual labels.
Recall: measure how well the model identifies all actual positive instances, ensuring that it captures as many relevant cases as possible.
Precision: evaluates the accuracy of positive predictions made by the model, thereby, reduces false positives.
F1-score: finds a balance between precision and recall, especially when there is an imbalance between false positives and false negatives.
Confusion matrix: is used to gain a detailed breakdown of the models' prediction performance by showing how many instances were correctly and incorrectly classified into different categories. Table 1 shows a depiction of the confusion matrix.

Table 1: Confusion Matrix

		Predicted Class		
		Yes	**No**	**Total**
Actual Class	Yes	TP	FN	P
	No	FP	TN	N
	Total	P'	N'	P + N

RESULTS AND DISCUSSION
The PhiUSIIL dataset, published by Prasad and Chandra, was annotated into legitimate and phishing classes. Out of 235,795 URLs, 57.19% were legitimate, while 42.81% were phishing. The distribution of classes is shown in figure 2.

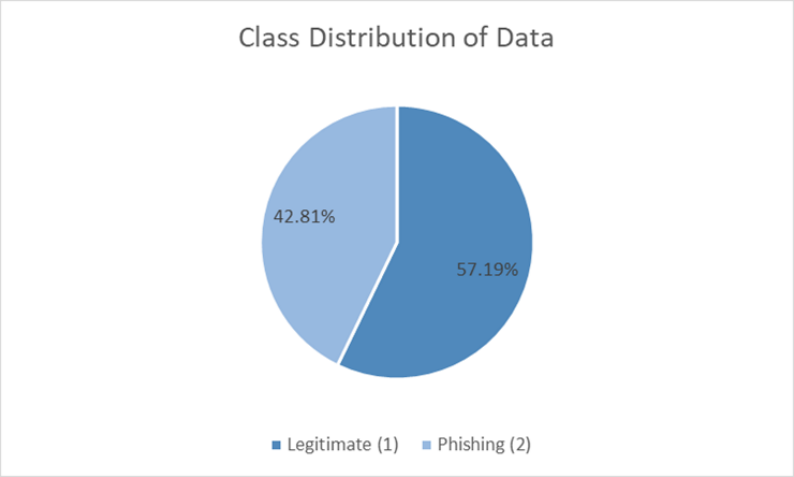

Figure 2: Class Distribution of Data

The LSTM model trained on 80% of the PhiUSIIL dataset achieved 90.00% accuracy, 89.00% precision, 88.00% recall, and 88.00% f1-score, with a comprehensive overview provided in figure 3.

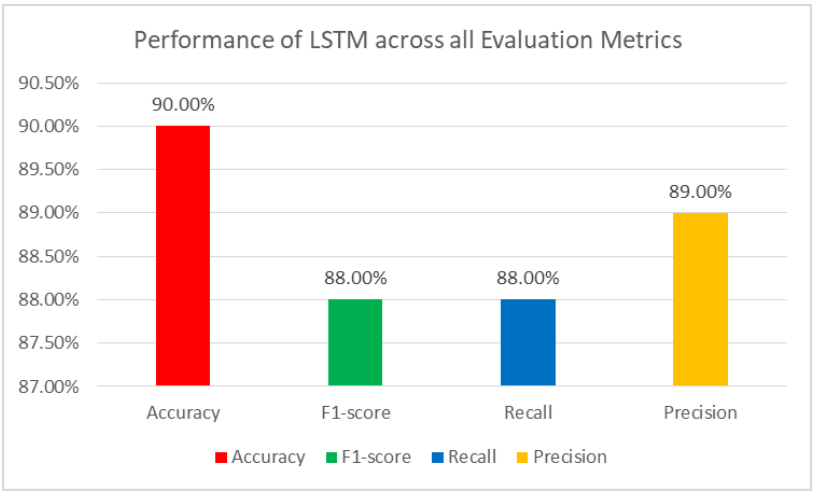

Figure 3: Result of the Performance of LSTM without Optimization

The LSTM model, optimized using Particle swarm optimization, achieved an accuracy of 92.00%, while precision, recall, and f1-score were all 91.00%. The performance is given in figure …

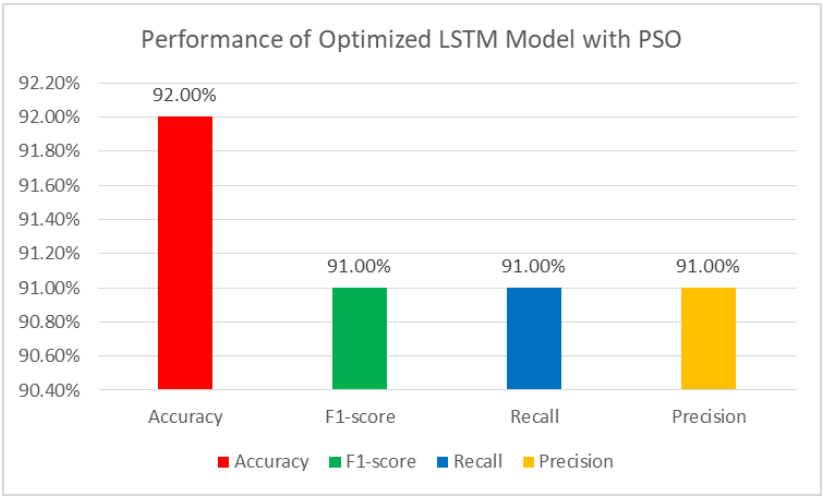

Figure 4: Result of the Performance of Optimized LSTM with PSO

The LSTM algorithm's hyperparameters were optimized using HyperOpt technique, resulting in an accuracy of 93.12%, precision of 93.02%, recall of 92.03%, and f1-score of 92.09%, as evaluated using the PhiSUIIL dataset.

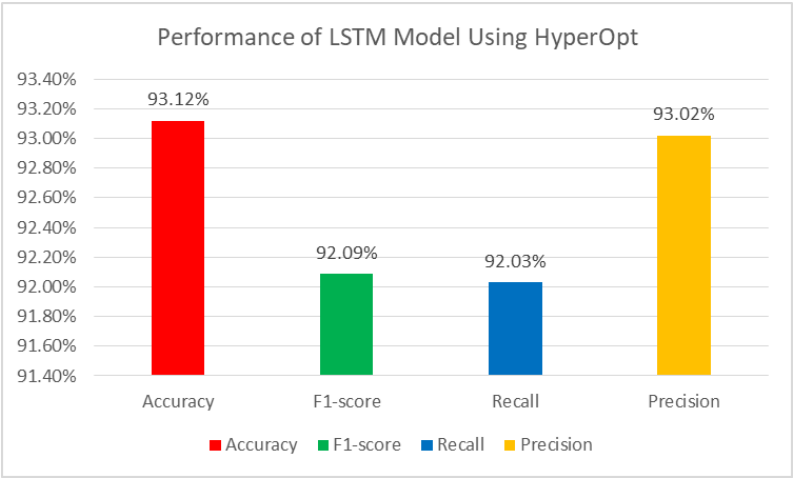

Figure 5: Result of the Performance of Optimized LSTM with HyperOpt

The comparison of the results for the LSTM model before and after hyperparameter tuning with both Particle swarm optimization and HyperOpt techniques shows that, optimizing LSTM with Particle Swarm Optimization (PSO) and HyperOpt significantly improves performance. The base model achieves 90.00% accuracy, but after PSO, it increases to 92.00%, improving F1-score, recall, and precision. HyperOpt further refines the model, achieving 93.12% accuracy and 93.02% precision, indicating better classification with fewer false positives. This is summarized in table 2 and visualized in figure 6.

Table 2: Comparison of LSTM Performance before and after Optimization Using PSO and HyperOpt

Metrics	LSTM	LSTM with PSO	LSTM with HyperOpt
Accuracy	90.00%	92.00%	93.12%
F1-score	88.00%	91.00%	92.09%
Recall	88.00%	91.00%	92.03%
Precision	89.00%	91.00%	93.02%

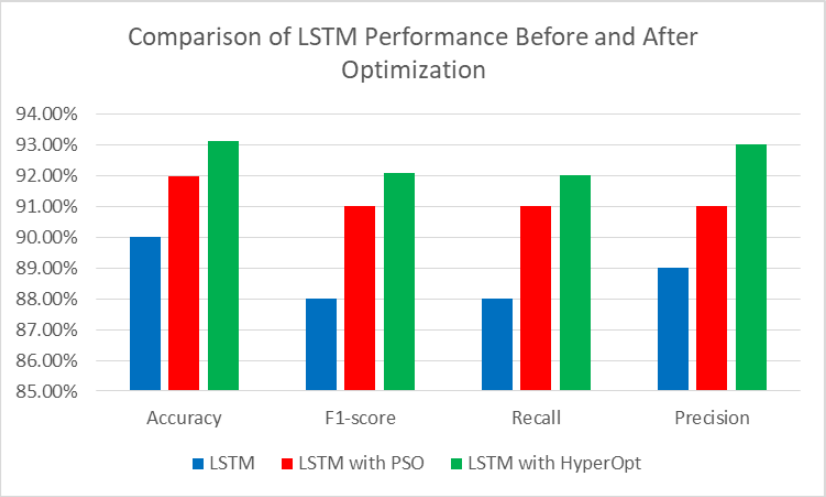

Figure 6: Comparison of LSTM Performance before and after Optimization Using PSO and HyperOpt

Discussion of Results

The LSTM model as shown in figure 3 was initially trained with default parameters, achieving an accuracy of 90.00%, an F1-score of 88.00%, a recall of 88.00%, and a precision of 89.00%. To improve its performance, Particle Swarm Optimization (PSO) was applied to fine-tune key hyperparameters, including the number of LSTM units, dropout rate, and learning rate. This optimization led to moderate improvements depicted in figure 4, increasing accuracy to 92.00%, while the F1-score, recall, and precision all improved to 91.00%. The confusion matrix revealed a reduction in false positives and false negatives, indicating that the model became more balanced and effective in classification after PSO-based tuning.

Further optimization using HyperOpt as seen in figure 5 yielded even better performance, with accuracy improving from 92.00% to 93.12%. The F1-score increased to 92.09%, recall to 92.03%, and precision to 93.02%, demonstrating a further reduction in misclassification errors. The confusion matrix showed a slight but notable decrease in false positives and false negatives, confirming that HyperOpt fine-tuned the LSTM model more effectively than PSO. These results clearly indicate that HyperOpt is superior to PSO for optimizing LSTM models, as it applies Bayesian optimization and the Tree-structured Parzen Estimator (TPE) to find more optimal hyperparameters. This refinement led to a highly optimized LSTM model capable of near-perfect phishing detection, proving that HyperOpt is a more efficient technique for enhancing deep learning models compared to PSO.

CONCLUSION

In order to tackle the ever increasing number of phishing websites and high level of cybersecurity threats over the internet, this research successfully developed an optimized model for phishing website detection by evaluating and improving LSTM model through hyperparameter tuning techniques. This study highlights the importance of hyperparameter tuning in enhancing deep learning models for phishing website detection. While the initial LSTM model achieved 90.00% accuracy, optimization using particle swarm optimization improved performance to 92.00%, demonstrating that tuning hyperparameters like LSTM units, dropout rate, and learning rate significantly reduces misclassifications. Further refinement with HyperOpt led to the best results, achieving 93.12% accuracy, with improved precision and recall, proving its superiority due to Bayesian optimization and Tree-structured Parzen Estimators (TPE). This research demonstrates significant advantages over previous studies in phishing detection. Compared to Sonowal and Kuppusamy (2020), the LSTM with HyperOpt model achieves a slightly higher accuracy 93.12% compared to their 92.72% without relying on complex multi-layered filters or accessibility-based features. While Anwekar and Agrawal (2022) report higher accuracy with Random Forest (97.14%), the deep learning approach in this research offers superior scalability and adaptability, especially with systematic optimization through PSO and HyperOpt. Most notably, the model vastly outperforms simpler models like the logistic regression used by Palaniappan et al. (2020), achieving over 30% higher accuracy. Additionally, the balanced performance across precision, recall, and F1-score underscores the

robustness and practical utility of our method in detecting phishing websites. The findings emphasize that proper hyperparameter tuning enhances model robustness, reduces false positives, and strengthens cybersecurity defenses, making it a crucial step in developing effective phishing detection systems.

REFERENCES

Ali, W. (2017). Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection. International Journal of Advanced Computer Science and Applications(IJACSA), 8(9), 72-76. http://dx.doi.org/10.14569/IJACSA.2017.080910

Kulkarni, A., & Brown, L. L. (2019). Phishing Websites Detection using Machine Learning. International Journal of Advanced Computer Science and Applications, 10(7), 8-13. http://dx.doi.org/10.14569/IJACSA.2019.0100702

Mahajan, R., & Siddavatam, I. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications, 181(23), 45-47. https://doi.org/10.5120/ijca2018918026

Ndabula, J. N., Olanrewaju, O. M., & Echobu, F. O. (2023). Detection of Hate Speech Code Mix Involving English and Other Nigerian Languages. Journal of Information Systems and Informatics, 5(4), 1416-1431. doi: https://doi.org/10.51519/journalisi.v5i4.595

Ajik, E. D., Obunadike, G. N., & Echobu, F. O. (2023). Fake News Detection Using Optimized CNN and LSTM Techniques. FUDMA Journal of Science, 5(3), 1044-1057. https://doi.org/10.51519/journalisi.v5i3.548

Kulkarni, A. D. (2023). Convolution Neural Networks for Phishing Detection. International Journal of Advanced Computer Science and Applications, 14(4), 15-19. https://doi.org/10.14569/ijacsa.2023.0140403

Sonowal, G., & Kuppusamy, K. S. (2020). PhiDMA– A phishing detection model with multi-filter approach. Journal of King Saud University - Computer and Information Sciences, 32(1), 99-112. https://doi.org/10.1016/j.jksuci.2017.07.005

Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. In Journal of King Saud University - Computer and Information Sciences, 35(2), 590–611. https://doi.org/10.1016/j.jksuci.2023.01.004

Palaniappan, G., Sangeetha, S., Rajendran, B., Sanjay, Goyal, S., & Bindhumadhava, S. B. (2020). Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features. In Procedia Computer Science, 654(661), 654–661. https://doi.org/10.1016/j.procs.2020.04.071

Anwekar, S. A., & Agrawal, V. (2022). Phishing Website Detection Using Machine Learning Algorithms. International Research Journal of Modernization in Engineering Technology and Science, 4(5), 2664-2668.

Garje, A., Tanwani, N., Kandale, S., Zope, T., & Gore, S. (2021). Detecting Phishing Websites Using Machine Learning. International Journal of Creative Research Thoughts (IJCRT), 9(11), 243-246.

Priya, P. H. (2023). Detection of Phishing Website Using Machine Learning. International Journal of Research Publication and Reviews, 4(12), 4990-4995.

Renursee, B. (2021). Detecting Phishing Website Using Machine Learning. Journal of Engineeering Sciences, 12(8), 126-132.

Prasad, A., & Chandra, S. (2023). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 103545. https://doi.org/10.1016/j.cose.2023.103545

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.