Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 15, Issue. 3, March 2026, pg.167 – 177

DETECTION OF PHISHING WEBSITES USING NATURAL LANGUAGE PROCESSING AND SUPERVISED MACHINE LEARNING TECHNIQUES

I.H. Ezeh[1]; A.A. Galadima[2]; Ifeanyi.C. Emeto[3]; U. I. Ismail[4]; Emmanuel.C. Ochuba[5]; S. Kwaghbee[6]

[1,2,3,5,6]Department of Cybersecurity, Federal University of Technology, Owerri, Imo State, Nigeria
[4]Federal University, Kashere, Gombe State, Nigeria
[1] harrison.ezeh@futo.edu.ng; [2] abdullahagaladima@gmail.com; [3] ifeanyi.emeto@futo.edu.ng;
[4] usmanidris@fukshere.edu.ng; [5] emmanuel.ochuba@futo.edu.ng; [6] kwaghbee.sever@futo.edu.ng
Corresponding Author: I.H. Ezeh, harrison.ezeh@futo.edu.ng

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i03.018

ABSTRACT: Phishing websites remain a pervasive and evolving cyber threat, causing significant financial losses and eroding trust in online services. Traditional detection methods like blacklists are often reactive and struggle to identify new, zero-day phishing attacks. This research investigates the application of Natural Language Processing (NLP) combined with supervised machine learning to create a more proactive and intelligent phishing detection system. The study utilized a large, publicly available dataset of approximately 549,000 URLs, labeled as phishing or legitimate. The core methodology involved transforming raw URL strings into a numerical format using a character-level Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. This approach allowed the model to learn deceptive linguistic and structural patterns within the URLs themselves. Two supervised learning algorithms, Multinomial Naive Bayes and a Stochastic Gradient Descent (SGD) Classifier, were trained and evaluated on this processed data. The results demonstrated the effectiveness of the NLP-driven approach. A comparative analysis revealed a distinct performance trade-off: the Naive Bayes model achieved a higher recall of 90.55%, making it adept at catching more phishing attempts, while the SGD Classifier excelled in precision, reaching 97.19%, thereby minimizing false alarms against legitimate sites. Prioritizing user trust and reliable warnings, the SGD Classifier was selected as the final model. It was subsequently integrated into a functional web application, "Phishing Shield AI," which provides real-time URL analysis with a confidence score.

Keywords: Phishing Detection, Natural Language Processing (NLP), Supervised Machine Learning, TF-IDF Vectorization, SGD Classifier, URL Analysis, Cybersecurity, CRISP-DM.

I. INTRODUCTION

Phishing attacks have become one of the most persistent and damaging cybersecurity threats, targeting individuals and organizations through fraudulent websites designed to steal sensitive information. These websites often imitate trusted brands, exploit user psychology, and employ subtle modifications in domain names and textual structures to evade detection. According to recent industry reports, phishing remains a primary attack vector responsible for credential theft, financial loss, and identity compromise.

Existing defense mechanisms, such as URL blacklists and rule-based heuristics, suffer from fundamental limitations. They depend on previously reported phishing instances and therefore fail to detect zero-day phishing websites that appear and disappear within short time frames. As attackers continuously modify domain structures and content, static detection techniques struggle to keep pace.

Recent advances in machine learning have enabled the development of adaptive phishing detection systems capable of learning patterns directly from data. In particular, Natural Language Processing (NLP) provides a powerful mechanism for extracting linguistic and structural cues from URLs and website content. When combined with supervised learning, NLP-based models can distinguish between phishing and legitimate websites with high accuracy.

This paper proposes a phishing website detection framework that relies solely on URL strings, eliminating the need for external metadata such as WHOIS records or page rendering. By leveraging character-level TF–IDF features and efficient supervised classifiers, the system achieves high accuracy while remaining computationally LIGHTWEIGHT AND SUITABLE FOR REAL-TIME DEPLOYMENT.

II. RELATED WORK

Phishing website detection has attracted sustained research interest due to the increasing sophistication of social engineering attacks and the short lifespan of malicious websites. Existing approaches can broadly be categorized into blacklist-based techniques, heuristic and rule-based methods, and machine learning-driven solutions.

A. Blacklist and Rule-Based Approaches

Early phishing detection systems relied primarily on blacklist-based mechanisms, where URLs reported as malicious are stored and compared against user requests [1]. Browser-integrated solutions and community-driven platforms such as PhishTank follow this approach [2]. While blacklist-based systems are computationally efficient, they are inherently reactive and ineffective against zero-day phishing websites, which often remain active for only a few hours [3].

Rule-based and heuristic approaches attempt to overcome this limitation by identifying suspicious URL patterns, including abnormal length, excessive use of special characters, or the presence of IP addresses in place of domain names [4]. Although these methods improve coverage, they require continuous manual updates and are highly vulnerable to evasion as attackers adapt their strategies [5]. Moreover, handcrafted rules often fail to generalize across different datasets and attack campaigns.

B. Traditional Machine Learning-Based Methods

To address the limitations of static rules, researchers introduced supervised machine learning techniques for phishing detection. Early studies extracted manually engineered features from URLs, webpage source code, and metadata, and employed classifiers such as Naive Bayes, Decision Trees, Random Forests, and Support Vector Machines [6], [7].

These approaches demonstrated improved detection accuracy compared to blacklist-based systems, particularly when trained on large labeled datasets [8]. However, their reliance on handcrafted features limits scalability and robustness. In addition, many models depend on external information such as WHOIS records or DNS lookups, which increases detection latency and reduces suitability for real-time deployment [9].

Despite these challenges, traditional machine learning models remain attractive due to their interpretability and low computational requirements. Studies have shown that probabilistic and linear classifiers can achieve competitive results when paired with appropriate text-based feature representations [10].

C. NLP-Based Phishing Detection

Recent research has increasingly adopted Natural Language Processing (NLP) techniques to automatically extract discriminative patterns from URLs and website content. Instead of relying on manually defined features, NLP-based approaches treat URLs as character or token sequences, enabling models to learn lexical and structural characteristics directly from data [11].

Character-level n-gram models combined with TF–IDF vectorization have proven particularly effective in capturing obfuscation techniques such as misspellings, deceptive keyword insertion, and homograph attacks [12], [13]. These approaches offer a strong balance between performance and efficiency, making them suitable for large-scale phishing detection systems.

Some studies extended NLP-based methods to webpage content analysis, extracting textual features from HTML documents and form fields [14]. While this improves detection accuracy, it introduces additional overhead and increases exposure to adversarial manipulation during content retrieval.

D. Deep Learning Approaches

With the advancement of deep learning, researchers proposed convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for phishing detection tasks [15]. Long short-term memory (LSTM) models have been used to capture sequential dependencies in URLs, achieving high classification accuracy [16].

More recently, transformer-based architectures have been explored for URL and webpage classification [17]. Although deep learning models often outperform traditional methods in controlled experiments, they require substantial computational resources, large labeled datasets, and longer inference times. Additionally, their limited interpretability poses challenges in security-sensitive environments where decision transparency is critical [18].

E. Comparative and Hybrid Approaches

Several comparative studies evaluated traditional machine learning models against deep learning-based approaches for phishing detection [19]. Interestingly, these studies found that when character-level NLP features are carefully designed; classical classifiers can achieve performance comparable to deep learning models, particularly in terms of precision and false-positive reduction.

Hybrid systems that combine URL-based NLP features with network-level or content-based attributes have also been proposed [20]. While these systems achieve high detection accuracy, their increased complexity often limits scalability and real-time usability.

F. Summary of Related Works

Table I: Summary of Related Work

Authors & Year	Focus / Approach	Methodology	Key Results	Limitations / Gaps
Benavides-Astudillo et al. (2023)	Content-based phishing detection	NLP + deep learning (LSTM, GRU variants) over page text embeddings	BiGRU achieved 97.39% accuracy	No multilingual testing or cross-validation reported
Çolhak et al. (2024)	HTML & NLP-based phishing website detection	MLP + CANINE (titles) + RoBERTa (content) fusion model	F1 = 96.80%, Accuracy = 97.18%	Needs testing across multilingual/dynamic pages
Boulieris et al. (2023)	Fraud detection via NLP and datasets	NLP for transactions, benchmarks on FraudNLP dataset	Surpassed benchmarks in fraud detectin domain	Focuses on banking fraud, not phishing website detection
Owen (2024)	NLP-based URL feature extraction + ML	Lexical, semantic, sentiment NLP features with ML models	Achieved >95% detection accuracy	Preprint status; lacks peer validation
Xu (2021)	Transformer model on URLs	Transformer architecture vs six classical models	97.3% accuracy (best among all)	Ignores content-based clues; no obfuscated URL tests
Rashid & Abdullah (2022)	Cloud ML via SageMaker	XGBoost, Linear Learner, k-NN on AWS dataset	XGBoost: 96.4% accuracy, 0.0005 min response	Uses generic tools; lacks phishing-specific features
Kalla & Kuraku (2023)	Comparative ML on URL features	Evaluated multiple ML models on URL/NLP features	SVC and RF performed best	No real-time or dynamic testing
Patra & Giri (2023)	Persuasion-based NLP email detection	TF-IDF + persuasion principles + classifiers	AdaBoost = 99% accuracy	Needs real-world testing; linguistic adaptability unknown
Omari (2023)	Comparative ML on phishing datasets	Seven ML models on UCI phishing domains	Gradient Boost: 97.2%, Random Forest: 97.1%	Single dataset; lacks real-time/multilingual testing
Elumalai & Bose (2024)	Enhanced feature extraction for phishing URLs	Supervised ML using advanced URL features	Promising feature-driven approach	No performance metrics in abstract; full text missing
Faheem & Ahmad (2024)	Deep learning-based phishing URL detection	1D CNN model on PhishTank dataset	99.7% accuracy	Narrow focus on URL features; unable to detect text-based/social engineering phishing

G. Research Gap and Motivation

Despite extensive research on phishing detection, a gap remains between high-performing academic models and practically deployable solutions. Many existing approaches prioritize accuracy while overlooking false-positive rates, computational overhead and real-time constraints [21].

This work addresses these challenges by proposing a lightweight phishing detection framework based solely on URL strings. By combining character-level TF–IDF representations with efficient supervised classifiers, the proposed approach achieves a balance between detection performance and deployability. Unlike deep learning-based systems, the model maintains low latency, transparency, and ease of integration, making it suitable for real-world cybersecurity applications.

III. METHODOLOGY

This study adopts a supervised machine learning approach combined with Natural Language Processing (NLP) to detect phishing websites using URL strings. The methodology follows a structured pipeline consisting of data acquisition, preprocessing, feature extraction, model training, and performance evaluation.

A. System Overview

The proposed system accepts a URL as input and classifies it as either phishing or legitimate. The overall process involves preprocessing raw URLs, transforming them into numerical features using TF–IDF vectorization, and applying supervised classifiers for prediction.

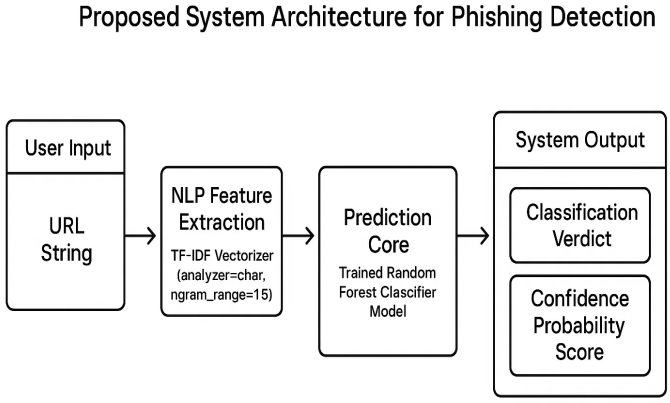

Fig. 1. Overall architecture of the phishing detection system

B. Dataset Description

The dataset consists of approximately **549,000 labeled URLs** collected from publicly available phishing and legitimate website repositories. Each instance contains a raw URL and a corresponding binary class label. The large dataset size improves model generalization and supports robust supervised learning.

C. Data Preprocessing

Preprocessing was applied to standardize URL inputs and improve data quality. All URLs were converted to lowercase, invalid entries were removed, and duplicate URLs were eliminated to prevent bias. Special characters were retained because they carry discriminative information useful for phishing detection.

D. Feature Extraction

To transform textual URL data into numerical form, **character-level TF–IDF vectorization** was employed. This approach captures lexical and structural patterns commonly used in phishing URLs, such as deceptive keywords, misspellings, and irregular domain structures. The feature space was limited to the most informative n-grams to ensure computational efficiency.

E. Classification Models

Two supervised classifiers were evaluated:

- **Multinomial Naive Bayes:** Selected for its effectiveness in high-dimensional text classification and fast training performance.
- **Stochastic Gradient Descent (SGD) Classifier:** A scalable linear model trained using logistic loss, offering better control over false positives.

Both models were trained using identical TF–IDF features to allow fair comparison.

F. TRAINING AND EVALUATION

The dataset was split into 80% training and 20% testing while preserving class distribution. Model performance was evaluated using accuracy, precision, recall, and F1-score. Confusion matrices were also analyzed to assess classification behavior.

G. MODEL SELECTION

Although the Naive Bayes classifier achieved higher recall, it produced more false positives. The SGD classifier demonstrated higher precision and was therefore selected as the final model due to its suitability for real-world phishing detection applications.

IV. RESULTS

This section presents the experimental results obtained from the proposed phishing detection system and analyzes both the classification performance and the practical implementation of the system. In addition to quantitative evaluation, the physical realization of the system through a user interface is presented to demonstrate real-world applicability.

A. Overall Classification Performance

The performance of the Multinomial Naive Bayes and Stochastic Gradient Descent (SGD) classifiers was evaluated using accuracy, precision, recall, and F1-score which will be below in tabular form:

Table 1: Comparative Performance Metrics of All Trained Models

Metric	**Naive Bayes**	**SGD Classifier**
Accuracy	0.9571 (95.71%)	0.9272 (92.72%)
Precision	0.9418 (94.18%)	0.9719 (97.19%)
Recall	0.9055 (90.55%)	0.7666 (76.66%)
F1-Score	0.9233 (92.33%)	0.8571 (85.71%)

The Naive Bayes classifier achieved higher recall, indicating its effectiveness in detecting a larger proportion of phishing URLs. However, this advantage is accompanied by a higher false-positive rate. In contrast, the SGD classifier produced fewer false alarms and demonstrated higher precision, making it more reliable for practical use where excessive warnings can reduce user trust.

B. Confusion Matrix Analysis

Further insight into model behavior was obtained through confusion matrix analysis as shown below in figure 3 and 4 respectively.

The Naive Bayes confusion matrix shows a higher number of true positives but also a noticeable number of legitimate URLs misclassified as phishing. This behavior reflects the model’s tendency to prioritize detection coverage.

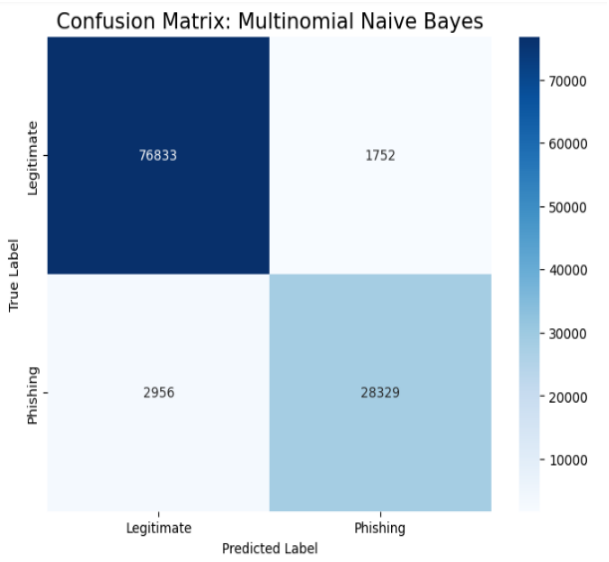

Fig. 2. Heatmap of Confusion Matrix for Naive Bayes

The SGD classifier shows a more balanced classification pattern, with significantly fewer false positives. Although a small number of phishing URLs were misclassified as legitimate, the overall trade-off favors usability and reliability.

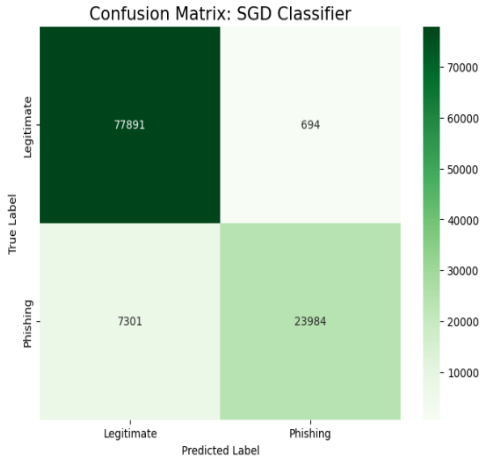

Fig. 3. Heatmap of Confusion Matrix for SGD Classifier

C. Comparative Performance Discussion

The comparative analysis highlights a clear precision–recall trade-off between the two models. While Naive Bayes is more aggressive in flagging phishing URLs, the SGD classifier offers better precision, which is desirable for deployment in user-facing systems. Based on this observation, the SGD classifier was selected as the final model.
Below is a graphical representation of the performance of the two models.

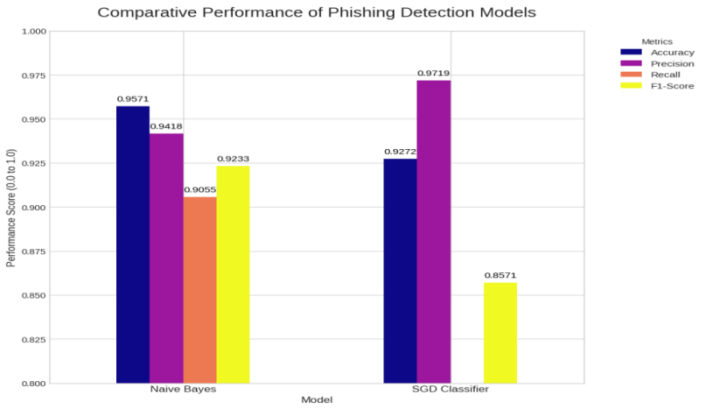

Fig. 4. Comparative Performance Bar Chart

D. Physical Design and User Interface Results

Beyond algorithmic performance, the effectiveness of a phishing detection system also depends on how results are presented to users. To evaluate this aspect, a simple and intuitive user interface was designed and implemented as part of the physical system output.

1. User Interface Design

The user interface provides a platform through which users can input a URL and receive an immediate phishing prediction. The interface was designed with simplicity and clarity in mind to ensure accessibility for non-technical users. It consists of a URL input field, a prediction button, and a result display area.

When a URL is submitted, the system processes the input using the trained SGD classifier and displays the classification result along with an indication of whether the URL is phishing or legitimate. This interaction demonstrates the practical feasibility of integrating the proposed model into real-world applications such as browser tools or security dashboards.

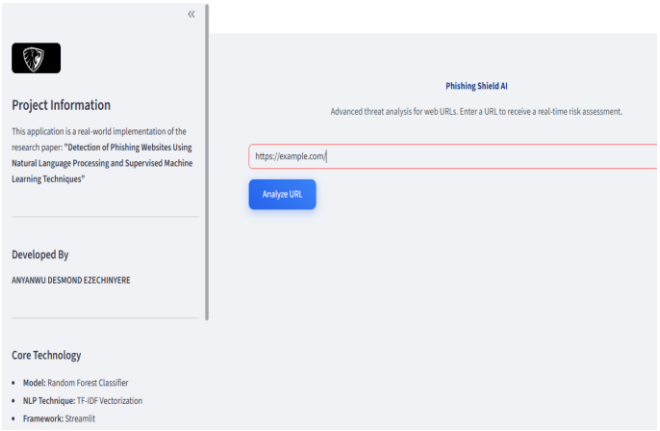

Fig. 6. Simple and Clean Input Screen

A. Analysis of Phishing Prediction Output

The phishing prediction interface confirms that the trained model performs effectively outside the experimental environment. Test URLs submitted through the interface were classified consistently with offline evaluation results. Phishing URLs were correctly flagged, while legitimate URLs were identified with minimal false alarms.

This result validates that the proposed model is not only accurate in controlled experiments but also stable when deployed in a user-interactive setting. The real-time

response and clear output presentation enhance the system's usability and demonstrate its readiness for practical deployment.

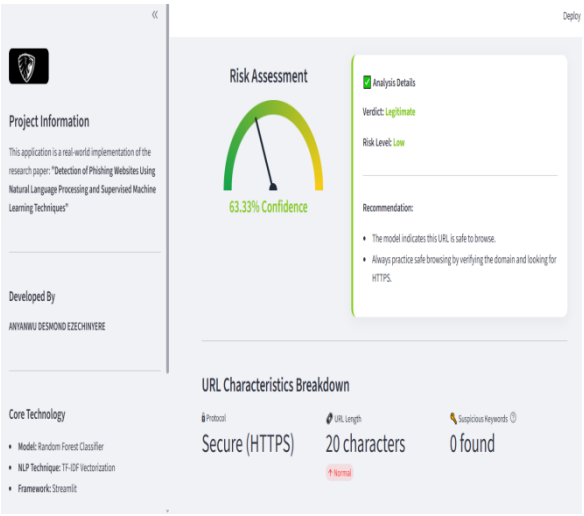

Fig. 7. Phishing Prediction

IV. SYSTEM DEPLOYMENT

The trained model was integrated into a lightweight web application that allows users to input a URL and receive an immediate phishing risk assessment. The system outputs:

- Classification verdict (Phishing or Legitimate)
- Confidence score
- User-friendly security recommendations

This demonstrates the feasibility of deploying NLP-driven phishing detection systems in real-world environments without relying on complex infrastructure.

V. SUMMARY AND CONCLUSION

SUMMARY

This research was undertaken to address the growing and evolving threat of phishing websites, which continue to pose significant risks to individuals and organizations. The study set out to design and implements a detection system that leverages the strengths of Natural Language Processing (NLP) for feature extraction and supervised machine learning for classification. The primary aim was to create a model capable of distinguishing between phishing and legitimate websites with high reliability.

The research followed the CRISP-DM methodology, providing a structured framework that guided the project from business understanding through to deployment. A substantial dataset of approximately 549,000 URLs was acquired and processed. The core innovation in the methodology was the use of character-level TF-IDF vectorization, which transformed raw URL strings into a rich numerical representation capturing subtle linguistic and structural patterns indicative of phishing.

Two distinct machine learning models, the Multinomial Naive Bayes and the Stochastic Gradient Descent (SGD) Classifier, were developed and rigorously evaluated. The analysis revealed a critical trade-off between the two models: Naive Bayes demonstrated a higher overall recall, making it adept at catching more phishing threats, while the SGD Classifier achieved superior precision, meaning its positive classifications were more reliable and resulted in fewer false alarms.

Based on the project's goal of building a trustworthy user application, the SGD Classifier was selected for the final system due to its high precision of 97.19%. This model

was successfully integrated into a functional and user-friendly web application called "Phishing Shield AI," built using the Streamlit framework. The application provides real-time analysis, delivering a clear verdict alongside a confidence score and actionable recommendations to the end-user.

CONCLUSION

In conclusion, this project successfully demonstrates the viability of combining NLP techniques with supervised machine learning to create an effective solution for the problem of phishing website detection. The findings confirm that the textual and structural patterns within a URL itself contain profound signals that can be computationally learned and used for accurate classification.

The implemented system, Phishing Shield AI, meets its core objectives by providing a proactive, learning-based alternative to traditional blacklist methods. It effectively classifies URLs with high precision, ensuring that when a phishing warning is issued, it is highly trustworthy. This capability is crucial for fostering user confidence and reducing the fatigue associated with false alarms. The journey from data acquisition and preprocessing to model training, evaluation, and final deployment underscores the practical applicability of the research. It stands as a testament to the power of a methodical, data-driven approach in tackling contemporary cybersecurity challenges.

REFERENCES

[1]. Abawajy, J. H. (2023). Phishing attacks and countermeasures: A survey. *Journal of Information Security and Applications, 73*, 103488.

[2]. Aleroud, A., & Zhou, L. (2017). Phishing environments, techniques, and countermeasures: A survey. *Computers & Security, 68*, 160–196.

[3]. Anti-Phishing Working Group. (2024). *Phishing activity trends report*. Retrieved from https://apwg.org

[4]. Basit, A., Zafar, M., & Karim, A. (2021). Detection of phishing attacks using natural languageprocessing and machine learning. *IEEE Access, 9*, 116108–116122.

[5]. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., & Rodríguez Galán, G. (2023). A phishing-attack-detection model using natural language processing and deep learning. Applied Sciences, 13(9), 5275.

[6]. Boulieris, P., Pavlopoulos, J., Xenos, A., & Vassalos, V. (2023). Fraud detection with natural language processing. Machine Learning, 113, 5087–5108.

[7]. Chiew, K. L., Yong, K. S. C., & Tan, C. L. (2018). A survey of phishing attacks: Their types, vectors, and technical approaches. *Expert Systems with Applications, 106*, 1–20.

[8]. Çolhak, F., Ecevit, M. İ., Uçar, B. E., Creutzburg, R., & Dağ, H. (2024). Phishing Website Detection through Multi-Model Analysis of HTML Content. arXiv preprint arXiv:2401.04820.

[9]. (Eds.), Proceedings of International Conference on Network Security and BlockchainTechnology (pp. 15–24). Lecture Notes in Networks and Systems, 738. Springer, Singapore.

[10]. Elumalai, K., & Bose, D. (2024). Advancement of Phishing Attack Detection Using Machine Learning. Journal of Educational Society, 76(1).

[11]. Jain, A. K., & Gupta, B. B. (2022). Phishing detection: Analysis of visual similarity-based approaches. *Journal of Ambient Intelligence and Humanized Computing, 13*(1), 225 241.

[12]. Kalla, D., & Kuraku, S. (2023). Phishing website URL's detection using NLP and machine learning techniques. Journal on Artificial Intelligence, 5(1), 145–162.

[13]. Nguyen, H. T., Nguyen, T. T., & Pham, H. D. (2023). Hybrid machine learning model for phishing website detection using natural language processing features. *Computers, Materials & Continua, 75*(3), 5231–5246.

[14]. Omari, K. (2023). Comparative Study of Machine Learning Algorithms for Phishing Website Detection. International Journal of Advanced Computer Science and Applications, 14(9), 417–424.

[15]. Owen, J. (2024). Phishing website URL's detection using natural language processing and machine learning techniques. EasyChair Preprint 14123.

[16]. Patra, C., & Giri, D. (2024). Machine Learning-Based Phishing E-mail Detection Using Persuasion Principle and NLP Techniques. In J. K. Mandal, B. Jana, T.-C. Lu, & D. De

[17]. Rao, R. S., & Pais, A. R. (2019). Detection of phishing websites using machine learning algorithms. *Procedia Computer Science, 165*, 631–641.

[18]. Rashid, S. H., & Abdullah, W. D. (2022). Cloud-based machine learning approach for accurate detection of website phishing. International Journal of Information Sciences and Applications,11, 2870.

[19]. Verma, R., & Das, A. (2017). What's in a URL: Fast feature extraction and malicious URL detection. *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, 55–63.

[20]. Xu, P. (2021). A transformer-based model to detect phishing URLs. arXiv preprint arXiv:2109.02138.

[21]. Zhang, Y., Hong, J. I., & Cranor, L. F. (2020). CANTINA: A content-based approach to detecting phishing websites. *Proceedings of the 16th World Wide Web Conference*, 639–648.

