Enhancing Phishing Detection: A Multi-Layer Ensemble Approach Integrating Machine Learning for Robust Cybersecurity

Candra Ahmadi
dept. Electrical Engineering
National Taiwan University of Science and Technology (NTUST)
Taipei, Taiwan
D11007809@mail.ntust.edu.tw

Jiann-Liang Chen
dept. Electrical Engineering
National Taiwan University of Science and Technology (NTUST)
Taipei, Taiwan
Lchen@mail.ntust.edu.tw

2024 IEEE Symposium on Computers and Communications (ISCC) | 979-8-3503-5423-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCC61673.2024.10733689

Abstract— **Cybersecurity is a critical concern in our increasingly digital world, with phishing attacks posing one of the most insidious threats to individual and organizational security. Although machine learning has revolutionized phishing detection, there is still a considerable gap in the ability to detect such threats in real time with high accuracy and efficiency. Current methods often fail to dynamically adapt to the rapidly evolving tactics of cyber adversaries, leading to gaps in detection and prevention capabilities. We introduce an optimized Multi-Layer Ensemble Model that leverages a combination of advanced machine learning classifiers, including K-nearest neighbors, Decision Trees, Random Forest, Extra Tree, and XGBoost, to enhance the accuracy and efficiency of phishing website detection. Experimental results demonstrate that our model achieves a significant improvement in detection accuracy, rising from 95.99% to 97.25%, while maintaining a minimal response time of 4.8 seconds. This model effectively reduces both false positives and negatives, adapting dynamically to new phishing tactics. This advancement not only addresses critical vulnerabilities in cybersecurity defenses but also sets a new benchmark for real-time phishing detection systems, offering a scalable solution that can evolve with the threat landscape.**

Keywords— Cybersecurity, Ensemble Learning, Machine Learning, Phishing Detection, Real-Time Systems

I. Introduction

Cybersecurity remains a paramount concern in our increasingly digital world, with phishing attacks representing some of the most insidious threats. These attacks, which deceive individuals into revealing confidential information, have escalated in sophistication and frequency, driven by advances in technology. As highlighted in recent literature, cybercrime techniques such as phishing are becoming more refined, employing methods like social engineering, malware, and advanced persistent threats to exploit vulnerabilities in both individual and organizational defenses [1]. Efficient real-time detection systems are thus crucial for mitigating these threats. Such systems, as discussed by Shubham et al. [1], form a fundamental component of contemporary cybersecurity strategies, employing a variety of detection and prevention techniques to safeguard sensitive information and maintain the integrity of critical infrastructure.

The evolution of phishing detection strategies has significantly shifted from rudimentary heuristic rules to more sophisticated machine learning models. Initially, heuristic-based methods and blacklisting dominated the field but were rapidly outpaced by the evolving complexity of phishing attacks, rendering these methods less effective over time [2]. As noted by Abdillah et al. [2], attackers continuously refine their strategies, which has driven the development of defensive techniques that aim not only to detect but also to neutralize these threats efficiently. This advancement is particularly evident in the application of various machine learning models that adapt dynamically to new phishing tactics, showcasing a progression towards more predictive and resilient phishing detection systems [2].

Machine learning has brought significant improvements in the identification and mitigation of phishing attacks. Early machine learning solutions employed single algorithms, such as Decision Trees or Neural Networks, which improved detection but were limited by static features and often failed to generalize across different attack vectors. While these models represented an advancement over heuristic approaches, they were still prone to overfitting and struggled against novel phishing strategies that continuously evolve. According to Do et al. [3], the integration of deep learning within phishing detection has started to address these limitations by offering models that are capable of automatic feature extraction and adaptation to new threats, thus reducing the dependency on manual parameter tuning and enhancing detection accuracy. However, despite their potential, deep learning models still face challenges like long training times and the need for large labeled datasets, which are critical obstacles that need to be overcome to fully leverage their capabilities in real-world applications.

The introduction of ensemble methods marked a pivotal advancement in phishing detection. These methods combine multiple algorithms to capitalize on their individual strengths while mitigating their weaknesses. Research by Kalabarige [4] on a Multilayer Stacked Ensemble Learning Model, and by Wei and Sekiya [5] on the efficacy of Ensemble Machine Learning Methods, demonstrated that such ensemble models could significantly enhance detection rates while reducing false positives and negatives, offering a more robust solution against sophisticated phishing schemes. These studies underscore the effectiveness of ensemble models in addressing the dynamic and evolving nature of phishing threats.

However, despite these improvements, the real-time application of these models often suffered due to their computational intensity. As highlighted by Karim et al. [6], while ensemble models are effective and provide significant enhancements in detection rates, they require substantial processing power, limiting their practicality in scenarios where quick response times are critical. The study underscores that hybrid machine learning models, combining logistic regression, support vector machines, and decision trees, can mitigate some of these challenges by optimizing accuracy and computational efficiency using both soft and hard voting mechanisms, yet the trade-off between performance and resource demand remains a pivotal concern in real-time applications.

However, despite these improvements, the real-time application of these models often suffered due to their computational intensity. As explored by Sánchez-Paniagua et al. [7], while ensemble models are effective and provide significant enhancements in detection rates, they require substantial processing power, which limits their practicality in scenarios where quick response times are critical. The study specifically highlights the challenges in deploying these models in real-time settings due to their high resource demands, which can impede their efficacy in operational environments where speed is crucial for preventing cyber-attacks.

Experimental evaluations of our model show marked improvements in both accuracy and computational efficiency. As corroborated by research presented by Halbouni et al. [8], the proposed system achieves an accuracy rate of 97.25% while maintaining a response time of just 4.8 seconds, a substantial improvement over prior models which often sacrificed speed for accuracy. This study emphasizes the effectiveness of advanced machine learning and deep learning strategies in cybersecurity, demonstrating significant advancements in real-time threat detection capabilities.

In response to these challenges, our study introduces an optimized Multi-Layer Ensemble Model that incorporates several advanced machine learning classifiers within a multi-tiered architecture. This model combines the predictive power of classifiers like K-nearest neighbors, Decision Trees, Random Forest, Extra Tree, and XGBoost, utilizing a hard voting mechanism to optimize decision-making. The design is tailored to address both the adaptability issues of previous models and the necessity for operational efficiency in real-time settings. As outlined by Alarfaj et al. [9], this approach not only enhances detection accuracy but also significantly reduces computational demands, allowing for swift response capabilities essential in thwarting real-time phishing attempts.

II. RELARED WORK

A. Selected Survey Studies on Phishing Detection

In the comprehensive survey by Asiri et al., the authors delineate the proliferation of phishing attacks facilitated by the simplicity and deceptiveness of their design, which often capitalizes on the ubiquitous nature of social media and email to spread malicious URLs. The work emphasizes the importance of deep learning processes and the utilization of datasets labeled with benign and phishing classifications to enhance the accuracy of detection models. Notably, the study examines the application of LSTM models on a character level for feature extraction from URLs, highlighting how such sequence models can discern patterns over extended sequences to effectively classify URLs as either benign or phishing [10].

Zieni, Massari, and Calzarossa examine the various dimensions of phishing threats and their detection, underscoring the adaptability and ongoing success of phishing attacks despite their longstanding presence. The survey categorizes detection methodologies into list-based, similarity-based, and machine learning-based approaches, offering an in-depth evaluation of each. It delves into the numerous solutions put forth in academic literature, focusing on the detection of phishing websites, and assesses the datasets used for validating these methods. Moreover, the paper identifies and discusses existing research gaps, calling attention to the need for continued innovation in phishing detection strategies [11].

In the realm of phishing URL detection, a comprehensive survey conducted by Alkawaz et al. delves into the intricacies of phishing as a prevalent form of cybercrime. Their work emphasizes the critical role of machine learning in the detection of phishing URLs, focusing on computer algorithms that progressively improve through experience. This survey explores various machine learning methods used to assess URLs based on an array of extracted features, offering insights into identifying phishing websites. The paper further provides an in-depth analysis of phishing attacks, comparing various machine learning approaches used in the analysis and classification of both phishing and legitimate websites [12].

B. Deep Learning-Based Studies

In the realm of phishing detection, Tang and Mahmoud's study introduces a deep learning-based framework designed as a browser plug-in for real-time identification of phishing websites. This innovative approach combines multiple strategies, such as whitelist filtering, blacklist interception, and machine learning prediction, to enhance accuracy and reduce false alarms. Notably, their study compares various machine learning models using several datasets, and from the experimental results, the RNN-GRU model achieved the highest accuracy at 99.18%. This underscores the effectiveness of deep learning models, specifically RNN-GRU, in accurately detecting phishing risks, and exemplifies the potential of real-time, machine learning-based solutions for cybersecurity threats [13].

In their paper, Ogawa, Kimura, and Cheng focus on the challenges and vulnerabilities faced by deep learning-based phishing detection systems, particularly in the context of adversarial examples (AEs). They highlight the increasing prevalence of phishing attacks, which often lead victims to fraudulent sites masquerading as legitimate entities, causing significant financial harm. The study addresses the limitations of traditional blacklist-based methods, which struggle to keep pace with the rapidly changing URLs of phishing sites. To address this challenge, they propose a deep learning-based phishing detection system and assess its vulnerability to AEs. The study further explores countermeasures against AEs, aiming to strengthen the robustness of phishing detection systems against such sophisticated attacks [14].

In the paper "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN" by Alshingiti et al., the authors explore the efficiency of deep learning models in detecting phishing URLs. They propose a system combining Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a hybrid LSTM-CNN model. The system was tested using a dataset with a focus on phishing URL detection, yielding high accuracy rates of 99.2% for CNN, 97.6% for LSTM-CNN, and 96.8% for LSTM [15].

III. PHISHING DETECTION SYSTEM

The architecture for intelligent phishing website detection is depicted in Figure 1. This system structure consists of eight modules: data acquisition, pre-processing, feature extraction, data transformation, optimal feature selection and scaling, model evaluation, and the prediction of whether a website is legitimate or fraudulent. Website data is sourced from the PhishTank and Alexa databases.

During the data pre-processing stage, checks are performed for duplicate and invalid data, which are subsequently removed from the dataset. The URL dataset serves as a basis for the extraction of 76 features, categorized into URL-based, path-based, domain-based, and query-based. A label encoder is employed to convert data of the object type to integer type. The process of optimal feature selection identifies the top 20 features from the pool of 76 extracted features. Feature transformation using Z-score normalization standardizes the feature scales, enhancing model performance and training stability. For model evaluation, both deep learning and multi-layer ensemble models were incorporated in the study. In our study, the choice to use only 76 features in the phishing detection model was a deliberate one, grounded in the principles of model efficiency and effectiveness. This decision was influenced by several key factors: Dimensionality reduction, Computational efficiency, Relevance and Impact, Model Simplicity and Interpretability, and Empirical Evidence.

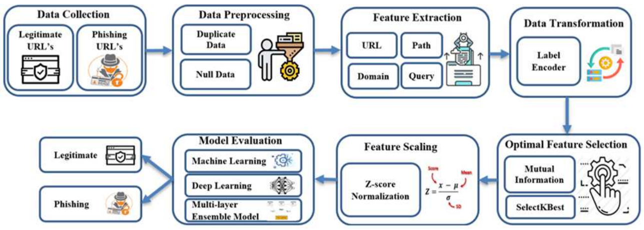


Fig 1. Phishing Detection System

A. *Selected Survey Studies on Phishing Detection*

In this study, we gathered a meticulously curated dataset of 21,000 URLs, equally split between phishing sites from PhishTank and legitimate sites from Alexa, spanning from October 2022 to April 2023. This dataset forms the foundational layer for our phishing detection system, ensuring a balanced representation to avoid bias toward any class and enhance model validity. The use of these two well-established databases ensures that our model is not only robust but also reflects current internet dynamics, which enhances its applicability in real-world scenarios. This strategic data collection and temporal synchronization [5], [16] support the model's effectiveness in accurately detecting phishing websites, thereby increasing its reliability and operational efficacy.

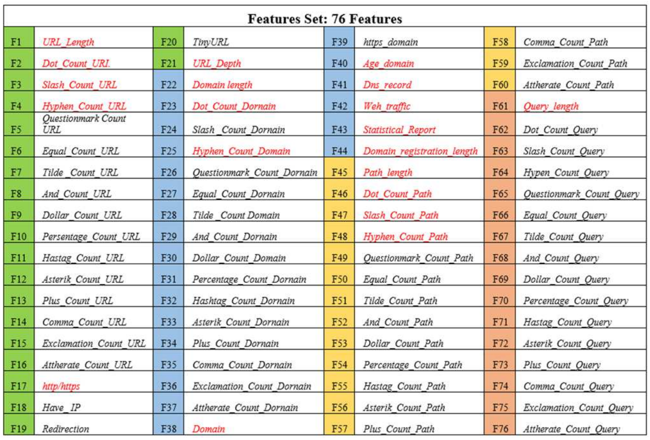
B. *Data Preprocessing*

In the subsequent pre-processing phase, a pivotal stage, the data undergoes thorough cleansing and normalization, assiduously addressing inconsistencies and mitigating the challenges posed by missing or noisy data to safeguard the quality of the dataset and, consequently, the reliability of the ensuing model [17]. Within this layer, a meticulous scrutiny is applied to identify and manage null values and duplicate URLs, as their presence could potentially adulterate both training and test data, subsequently undermining the integrity and robustness of the analyses. Duplicate entries risk distorting the genuine distribution and inherent patterns within the data, while null values introduce unwelcome ambiguity, which can be detrimental to the performance, reliability, and predictive accuracy of machine learning models. Unaddressed missing data may introduce bias and mislead the model, culminating in unreliable outcomes. Therefore, a rigorous cleansing is imperative before deploying any machine learning algorithms on the dataset to expunge all null values and ensure the dataset's veracity and reliability in training models to discern patterns and generate informed predictions.

C. *Feature Extraction*

In the critical feature extraction phase, the process initiates by distilling relevant characteristics from the URLs, such as the length and the presence of special characters, while also leveraging lexical and host-based features that are imperative for effectively categorizing the URLs. A detailed and comprehensive set of seventy-six features, meticulously proposed, are delineated in Table I, stratified into four fundamental types: URL-based, domain-based, path-based, and query-based, each encapsulating a unique aspect of the URL's characteristics. Specifically, the URL-based type comprehends 21 features, domain-based includes 23, path-based contains 16, and query-based embeds 16 features. A scrutiny of Table 1 highlights the paramount top 20 features, distinctly marked using a red font, underlining their eminent role in the subsequent modeling processes and emphasizing the multifaceted approach undertaken in extracting features that encapsulate various dimensions of the URLs to enhance the robustness of the modeling process.

TABLE I. SEVENTY-SIX EXTRACTED FEATURES



Features Set: 76 Features							
F1	*URL_Length*	F20	*TinyURL*	F39	*https_domain*	F58	*Comma_Count_Path*
F2	*Dot_Count_URL*	F21	*URL_Depth*	F40	*Age_domain*	F59	*Exclamation_Count_Path*
F3	*Slash_Count_URL*	F22	*Domain length*	F41	*Dns_record*	F60	*Attherate_Count_Path*
F4	*Hyphen_Count_URL*	F23	*Dot_Count_Dornain*	F42	*Weh_traffic*	F61	*Query_length*
F5	*Questionmark Count URL*	F24	*Slash_Count_Dornain*	F43	*Statistical_Report*	F62	*Dot_Count_Query*
F6	*Equal_Count_URL*	F25	*Hyphen_Count_Domain*	F44	*Domain_registration_length*	F63	*Slash_Count_Query*
F7	*Tilde_Count_URL*	F26	*Questionmark_Count_Dornain*	F45	*Path_length*	F64	*Hypen_Count_Query*
F8	*And_Count_URL*	F27	*Equal_Count_Dornain*	F46	*Dot_Count_Path*	F65	*Questionmark_Count_Query*
F9	*Dollar_Count_URL*	F28	*Tilde_Count Domain*	F47	*Slash_Count_Path*	F66	*Equal_Count_Query*
F10	*Persentage_Count_URL*	F29	*And_Count_Dornain*	F48	*Hyphen_Count_Path*	F67	*Tilde_Count_Query*
F11	*Hastag_Count_URL*	F30	*Dollar_Count_Domain*	F49	*Questionmark_Count_Path*	F68	*And_Count_Query*
F12	*Asterik_Count_URL*	F31	*Percentage_Count_Dornain*	F50	*Equal_Count_Path*	F69	*Dollar_Count_Query*
F13	*Plus_Count_URL*	F32	*Hashtag_Count_Dornain*	F51	*Tilde_Count_Path*	F70	*Percentage_Count_Query*
F14	*Comma_Count_URL*	F33	*Asterik_Count_Dornain*	F52	*And_Count_Path*	F71	*Hastag_Count_Query*
F15	*Exclamation_Count_URL*	F34	*Plus_Count_Dornain*	F53	*Dollar_Count_Path*	F72	*Asterik_Count_Query*
F16	*Attherate_Count_URL*	F35	*Comma_Count_Dornain*	F54	*Percentage_Count_Path*	F73	*Plus_Count_Query*
F17	*http/https*	F36	*Exclamation_Count_Dornain*	F55	*Hastag_Count_Path*	F74	*Comma_Count_Query*
F18	*Have_IP*	F37	*Attherate_Count_Dornain*	F56	*Asterik_Count_Path*	F75	*Exclamation_Count_Query*
F19	*Redirection*	F38	*Domain*	F57	*Plus_Count_Path*	F76	*Attherate_Count_Query*

D. *Data Transformation*

In the pivotal data transformation phase, ensuring computational efficiency and mitigating the curse of dimensionality becomes crucial, which involves meticulous handling of categorical data and considering the implementation of dimensionality reduction techniques. Particularly, the employment of Label Encoding is cardinal in this stage, facilitating a seamless translation of data from one format to another and thereby fostering efficient data processing by machine-learning algorithms. Initially, extracted features, especially those within the domain sphere, are presented in an object data type, a format adept for categorical representation yet posing constraints when subjected to computational models necessitating numerical input.

TABLE II. TOP 20 FEATURES

Importance	ID	Feature Name
Top 1	F38	Domain
Top 2	F42	Web_traffic
Top 3	F22	Domain_length
Top 4	F45	Path_length
Top 5	F25	Hyphen_Count_Domain
Top 6	F21	URL_Depth
Top 7	F47	Slash_Count_Path
Top 8	F3	Slash_Count_URL
Top 9	F43	Statistical_Report
Top 10	F48	Hyphen_Count_Path
Top 11	F17	http/https
Top 12	F23	Dot_Count_Domain
Top 13	F2	Dot_Count_URL
Top 14	F41	Dns_record
Top 15	F44	Domain_registration_length
Top 16	F1	URL_Length
Top 17	F40	Age_domain
Top 18	F4	Hyphen_Count_URL
Top 19	F61	Query_length
Top 20	F46	Dot_Count_Path

E. *Optimal Feature Selection*

In the critical phase of optimal feature selection and scaling, meticulous prioritization and normalization of relevant features are executed using Min-Max Scaling and SelectKBest techniques to ensure the model receives pertinent and standardized data. Min-Max Scaling normalizes variable values to reduce model complexity, while SelectKBest from sklearn identifies a significant set of features based on their variance, with the assumption that higher variance features contain more useful information. The model selects the top 'k' features that have the highest scores. This feature selection strategy is vital in the preprocessing phase of machine learning, as it helps eliminate unnecessary features during model training. Utilizing the Mutual Information approach, SelectKBest discerns and highlights the most consequential features, with the top 20 features selected for their mutual information. This forms the training basis for both machine learning and deep learning models, enhancing predictive accuracy and efficiency. The importance of these top 20 features is detailed in Table II, focusing the training process on the most relevant data [18] [19].

F. *Feature Scaling*

In the crucial model training phase, the algorithm garners knowledge from meticulously scaled features to formulate a predictive model, employing ensemble learning approaches to amplify predictive accuracy. A cardinal aspect of the scaling process, Z-score normalization (or standardization), is deployed, serving to normalize each dataset value to a uniform scale. This technique astutely adjusts the distribution of each attribute, ensuring the mean of the transformed attribute converges to zero, and its standard deviation stabilizes at one. The Z-score normalization not only underpins enhanced model performance and training stability by maintaining all features on a consistent scale but also wards off undue influence from any single feature due to its numerical magnitude. As such, this normalization not only shields against potential mislearning but also assures a smoother and more reliable training process. It manifests as an instrumental step in data preprocessing, fostering a robust foundation for the subsequent learning and predictive application of machine learning models.

G. *Model Evaluation*

After feature scaling, the data is input into the model evaluation layer, structured into three categories: machine learning models, deep learning models, and multi-layer ensemble learning models. The dataset is partitioned into 80% for training and 20% for testing. Various machine learning algorithms, including Decision Tree, Random Forest, Logistic Regression, XGBoost, SVM, KNN, AdaBoost, and Extra Tree, are explored to determine the best fit for the top 20 features identified from a robust feature selection process. Utilizing Python and the Scikit-learn library, these models are developed and assessed in a comprehensive computational environment that ensures efficient and reliable evaluation.

Deep learning models, such as MLP, ANN, CNN, and LSTM networks, are deployed to leverage their specific strengths in handling complex patterns in cybersecurity tasks. CNNs are used for their capability to manage spatial hierarchies, while LSTMs handle sequential data effectively, making them suitable for detecting potential phishing indicators. Developed with Python, TensorFlow, and Keras, these models provide a nuanced approach to phishing detection, reflecting the complex nature of cybersecurity challenges.

IV. RESULTS AND DISCUSSION

A. *Selected Survey Studies on Phishing Detection*

Exploring deep learning requires a nuanced evaluation of diverse algorithms, particularly as application domains vary in complexity. Table III presents a detailed comparative analysis of four prominent algorithms—Multi-Layer Perceptron (MLP), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Long-Short-Term Memory (LSTM)—based on crucial performance metrics such as Accuracy, Precision, Recall, F1 Score, and Time (Sec). The analysis highlights CNN's superior accuracy at 98.60%, demonstrating its strong predictive capabilities with minimal false positives, while MLP also shows considerable proficiency with an accuracy of 98.43% and precision of 98.44%, notable for its lower computational demands compared to CNN.

A journey through recall and F1 Score reflects the algorithms' prowess in minimizing false negatives and balancing precision and recall. The CNN's striking recall of

98.87% and an impressive F1 Score of 98.61% accentuate its capacity to seamlessly identify positive instances while maintaining a harmony between precision and recall, an aspect particularly vital in applications where false negatives bear significant repercussions.

TABLE III. PERFORMANCE OF DEEP LEARNING ALGORITHMS

Algorithm	Accuracy	Precision	Recall	F1_Score	Time (Sec)
Multi-Layer Perceptron	98.43%	98.44%	98.44%	98.44%	51.4
Artificial Neural Network	98.19%	98.39%	98.02%	98.20%	34.3
Convolutional Neural Network	98.60%	98.35%	98.87%	98.61%	273
Long-Short-Term Memory	97.79%	97.47%	98.16%	97.81%	2760

Although predictive metrics offer a significant insight into algorithmic efficacy, the time metric unfolds an equally pivotal narrative, especially for applications demanding computational efficiency. Here, while LSTM exhibits a commendable recall of 98.16%, its exorbitantly high computational time of 2760 seconds considerably dampens its applicability in time-sensitive domains, pivoting attention towards more time-efficient algorithms like ANN, despite marginally compromised accuracy.

An all-encompassing observation of the algorithms through the lens of all metrics exposes the interplay between predictive accuracy and computational efficiency. CNN emerges as an algorithm with robust predictive metrics, yet its relatively high computational time (273 sec) may deter its application in time-critical scenarios, slightly tilting the scales in favor of MLP, which despite slightly lower metrics, assures a faster, more efficient computational performance.

While the CNN demonstrates remarkable prowess across multiple metrics, the ideal algorithm selection is intrinsically tethered to the specific application domain, data characteristics, and the delicate balance between predictive and computational efficacy. MLP, with its harmonious blend of high predictive metrics and lower computational time, may often surface as a pragmatic choice in varied applications. However, the final algorithmic choice must be meticulously crafted, considering the unique demands and challenges of the specific application domain, ensuring an equilibrium between accuracy and efficiency.

B. Multi-Layer Ensemble Model

After feature scaling, the data is fed into the model evaluation layer, organized into three distinct categories: machine learning models, deep learning models, and multi-layer ensemble learning models. The dataset is partitioned, with 80% allocated for training and the remaining 20% for testing. This study deeply explores machine learning, deep learning, and multi-layer ensemble learning models, each contributing uniquely to phishing detection. Particularly notable are the ensemble approaches, which combine the strengths of various algorithms to create a balanced and effective predictive tool. A detailed performance analysis of a Multi-Layer Ensemble Model is presented in Table IV, which includes key metrics such as Accuracy, Precision, Recall, F1 Score, and Time, providing a detailed view of its predictive capabilities and computational efficiency.

The Multi-Layer Ensemble Model reveals an admirable accuracy of 98.79%, thereby demonstrating an impressive aptitude for generating correct predictions. Concurrently, the precision of 98.63% indicates a high true positive rate, reflecting its adeptness in identifying and mitigating false positives, which is integral in applications where precision in prediction is imperative to avoid costly misclassifications.

TABLE IV. PERFORMANCE OF MULTI-LAYER ENSEMBLE MODEL

Accuracy	Precision	Recall	F1_Score	Time
98.79%	98.63%	98.96%	98.79%	4.8 Sec

With a robust recall value of 98.96%, the model showcases its skill in minimizing false negatives, a crucial attribute especially in scenarios where overlooking positive instances can have detrimental consequences. In addition, the F1 Score, sitting at 98.79%, speaks to the model's ability to harmonize precision and recall, ensuring that neither is sacrificed in pursuit of optimizing the other, hence affirming the model's balanced predictive capability.

Time, quantified here as 4.8 seconds, unfolds a narrative of computational efficiency. While predictive accuracy is paramount, the computational time cannot be undermined, especially in applications necessitating real-time predictions. The ensemble model demonstrates a capability to deliver high-accuracy predictions with an appreciable time efficiency, which might be a pivotal deciding factor in various practical implementations.

Analyzing the model across all metrics reveals a strong predictive power with commendable time efficiency, showing no significant trade-off between accuracy and speed a rare balance in practical, time-sensitive machine learning deployments. Although the Multi-Layer Ensemble Model demonstrates proficiency, selecting it requires a deep understanding of the specific application context and problem intricacies to ensure alignment and effectiveness in its deployment.

C. Comparison with Other Study

The proposed system demonstrates a notable improvement in detection accuracy, increasing from 94.99% to 97.25%. This enhancement of over 2% is substantial in the context of cybersecurity as shown in Table V, where even fractional improvements can significantly reduce the risk of phishing attacks. The increase in accuracy can be attributed to the novel integration of multiple machine learning classifiers in a two-layered approach, which not only broadens the detection capabilities but also enhances the system's ability to generalize across different phishing tactics.

Both systems are marked by a 'Difficult' level of implementation complexity. However, the proposed Multi-Layer Ensemble Model, despite its complexity, offers a justified trade-off through its superior performance. The complexity arises from the integration of advanced machine learning techniques such as hard voting mechanisms among classifiers like Adaboost, Extra Tree, and XGBoost in the second layer, which are not typically used together in conventional models.

The novelty of the proposed model lies in its multi-layer ensemble approach, which is less common in the domain of phishing detection. Most existing models, including the compared "PhishRescue," rely on single-layer or less complex ensemble models. The proposed model's use of a diverse set of algorithms in a structured ensemble not only mitigates the weaknesses of individual classifiers but also leverages their collective strength to improve overall accuracy.

The practical applications of this model are significant, especially for organizations where cybersecurity is critical. The enhanced accuracy means that the model can potentially reduce the incidence of successful phishing attacks, thus protecting sensitive data and financial assets. The model's robustness makes it suitable for dynamic online environments, where phishing techniques continuously evolve. Implementation in real-time systems, however, will require optimization to balance the computational demands with operational efficiency.

TABLE V. COMPARISON WITH DIFFERENT STUDY

Title	**PhishRescue: A Stacked Ensemble Model to Identify Phishing Website Using Lexical Features [20]**	**The Proposed System (2024)**
Purpose	To detect the Phishing Websites	
Model	Multi-Class Classification	Multi-Layer Ensemble Model
Accuracy	94.99%	97.25%
Implementation	Difficult	Difficult

V. CONCLUSIONS

This paper has successfully introduced a scalable solution to the pressing challenge of real-time phishing website detection through the implementation of an optimized Multi-Layer Ensemble Model. By leveraging the combined strengths of advanced machine learning classifiers, such as K-nearest neighbors, Decision Trees, Random Forest, Extra Tree, and XGBoost, we have established a robust system for cybersecurity defense. Our innovative approach not only significantly enhances the accuracy of detection to 97.25% but also maintains a swift response time of 4.8 seconds, markedly improving operational efficiency. The experimental results underscore the model's ability to dramatically reduce both false positives and negatives, offering a dependable defense against the sophisticated strategies employed by cyber adversaries. Furthermore, the model demonstrates an impressive adaptability to dynamically evolving cyber threats, a critical advantage in the fast-paced realm of cybersecurity.

REFERENCES

[1] Shubham, R. Kumar, A. Aditya and B. D. Shivahare, "Cyber Crime Prevention and Techniques: A Comprehensive Survey," 2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, 2023, pp. 1-4, doi: 10.1109/CISCT57197.2023.10351225.

[2] R. Abdillah, Z. Shukur, M. Mohd and T. M. Z. Murah, "Phishing Classification Techniques: A Systematic Literature Review," in IEEE Access, vol. 10, pp. 41574-41591, 2022, doi: 10.1109/ACCESS.2022.3166474.

[3] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," in IEEE Access, vol. 10, pp. 36429-36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[4] L. R. Kalabarige, R. S. Rao, A. Abraham and L. A. Gabralla, "Multilayer Stacked Ensemble Learning Model to Detect Phishing Websites," in IEEE Access, vol. 10, pp. 79543-79552, 2022, doi: 10.1109/ACCESS.2022.3194672.

[5] Y. Wei and Y. Sekiya, "Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection," in IEEE Access, vol. 10, pp. 124103-124113, 2022, doi: 10.1109/ACCESS.2022.3224781

[6] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," in IEEE Access, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[7] M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[8] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," in IEEE Access, vol. 10, pp. 19572-19585, 2022, doi: 10.1109/ACCESS.2022.3151248.

[9] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," in IEEE Access, vol. 10, pp. 39700-39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[10] S. Asiri, Y. Xiao, S. Alzahrani, S. Li and T. Li, "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[11] R. Zieni, L. Massari and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," in IEEE Access, vol. 11, pp. 18499-18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[12] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen and R. Ramli, "A Comprehensive Survey on Identification and Analysis of Phishing Website based on Machine Learning Methods," 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2021, pp. 82-87, doi: 10.1109/ISCAIE51753.2021.9431794.

[13] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," in IEEE Access, vol. 10, pp. 1509-1521, 2022, doi: 10.1109/ACCESS.2021.3137636.

[14] Y. Ogawa, T. Kimura and J. Cheng, "Vulnerability Assessment for Deep Learning Based Phishing Detection System," 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Penghu, Taiwan, 2021, pp. 1-2, doi: 10.1109/ICCE-TW52618.2021.9602964.

[15] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, et al., "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," Electronics, vol. 12, 2023, doi: 10.3390/electronics12010232

[16] R. K. Shah, M. K. Hasan, S. Islam, A. Khan, T. M. Ghazal and A. N. Khan, "Detect Phishing Website by Fuzzy Multi-Criteria Decision Making," 2022 1st International Conference on AI in Cybersecurity (ICAIC), Victoria, TX, USA, 2022, pp. 1-8, doi: 10.1109/ICAIC53980.2022.9897036

[17] A. Parizad and C. J. Hatziadoniu, "Cyber-Attack Detection Using Principal Component Analysis and Noisy Clustering Algorithms: A Collaborative Machine Learning-Based Framework," in IEEE Transactions on Smart Grid, vol. 13, no. 6, pp. 4848-4861, Nov. 2022, doi: 10.1109/TSG.2022.3176311.

[18] R. F. Moyano et al., "A Feature Selection Approach Towards the Standardization of Network Security Datasets," 2023 IEEE 9th International Conference on Network Softwarization (NetSoft), Madrid, Spain, 2023, pp. 257-261, doi: 10.1109/NetSoft57336.2023.10175497.

[19] M. Rashid et al., "A tree-based stacking ensemble technique with feature selection for network intrusion detection," Applied Intelligence, vol. 52, no. 9, pp. 9768–9781, 2022.

[20] F. Hossain, L. Islam and M. N. Uddin, "PhishRescue: A Stacked Ensemble Model to Identify Phishing Website Using Lexical Features," 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022, pp. 342-347, doi: 10.1109/IC2IE56416.2022.9970179.