This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3543738

IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Digital Object Identifier

Enhancing Phishing Detection: A Machine Learning Approach with Feature Selection and Deep Learning Models

GANESH S NAYAK[1], BALACHANDRA MUNIYAL[2], (Member, IEEE), MANJULA C BELAVAGI[3]
Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India
(e-mail: ganeshsnayak29@gmail.com, bala.chandra@manipal.edu, manjula.cb@manipal.edu)

Corresponding author: Balachandra Muniyal (e-mail: bala.chandra@manipal.edu) and Manjula C Belavagi (e-mail: manjula.cb@manipal.edu).

ABSTRACT With the rise in cybercrime, phishing remains a significant concern as it targets individuals with fake websites, causing victims to disclose their private information. The effective implementation of phishing detection relies on cost efficiency, with the increased feature extraction factor contributing to these costs. This research analyzes a dataset containing 58,645 URLs, examining 111 features of the latest phishing websites dataset to identify the differences between phishing sites and legitimate sites. Astonishingly, using only 14 characteristics, the feedforward model achieved a remarkable accuracy of 94.46%, confirming the efficiency of Machine Learning in phishing detection. Through the exploitation of a multiple classifier collection, including Deep Neural Network (DNN), Wide and Deep, and TabNet, this research advances ongoing efforts to improve the accuracy and efficiency of phishing detection mechanisms and enhance cybersecurity defenses against malicious activities. The methodology introduces a new metric called the ’anti-phishing score,’ which evaluates performance based on false positives and negatives, beyond traditional model accuracy. The model was trained through a robust design of extensive experimentation and hyperparameter-sensitive grid search, ensuring an optimized configuration for phishing detection. Furthermore, the trained model was validated on a new dataset to evaluate its generalizability, enhancing its practical applicability. Through the integration of feature selection principles, advanced algorithmic techniques, and comprehensive evaluation approaches, this research offers a robust approach to phishing detection, considering the evolving nature of cyber threats. The findings provide a beneficial framework for cybersecurity specialists and researchers, enabling more effective preventive measures against phishing attacks.

INDEX TERMS Phishing Detection, Cybersecurity, Deep Learning, Neural Networks, Feature Selection, Hyperparameter Optimization, Real-time Detection, TabNet, Wide and Deep Model.

I. INTRODUCTION

In the 21st century, cyberspace is highly networked, making cybersecurity a critical concern due to phishing attacks, which lead to scams affecting individuals, businesses, and organizations globally. Deceptive tactics like phishing, used by cybercriminals to steal personal information, pose a primary threat to cybersecurity, privacy, and the stability of financial institutions. The term ’phishing’ is derived from ’fishing,’ in which an attacker attempts to lure individuals into submitting sensitive information. These attacks often occur when cybercriminals impersonate well-known financial institutions, social media sites, or government entities to deceive victims. The victims then unknowingly hand over their personal information, passwords, bank details, and more.

The presence of phishing attacks in almost all digital spaces and services indicates their pervasiveness: email, social media, and instant messaging are common forms of phishing, yet other forms still exist for scammers to exploit. In contrast to the phishing reported by [1] and [2], email service impersonation and communication trust attacks exploit the fact that many people use these platforms for both personal and business purposes. These tactics are a form of social engineering, designed to trick users into compromising their security. Initially perceived as simple email-based scams, these attacks have evolved into complex chains of actions capable of deceiving even experienced users and bypassing robust security systems.

A phishing attack can lead to serious consequences such as financial loss, identity theft, damaged reputation, or even legal action. This form of attack is often associated with data breaches, ransomware infections, and other criminal cyber activities. There is concern that these threats may expand into other security areas, affecting individuals and organizations. Security researchers and cybersecurity specialists continue to develop and implement measures to detect, counteract, and minimize phishing attacks.

The study of phishing has led to numerous research efforts that explore various aspects and detection methods. Chiew et al. [3] conducted a comprehensive survey on phishing attacks, categorizing them by type, channel, and technical tactic. Their work underscores the need for sensitive detection mechanisms to identify and prevent phishing attempts. Yahya et al. [4] discuss the use of machine learning techniques for phishing site identification, emphasizing that feature selection is crucial for achieving high detection accuracy.

Baykara and Gürel [2] investigate a heuristic analysis approach for detecting phishing attacks. Their findings suggest that combining heuristic algorithms with machine learning techniques can enhance detection. Alswailem et al. [5] demonstrate the effectiveness of machine learning algorithms in classifying malicious web content. Zuhair et al. [6] review feature selection methods for phishing detection, highlighting the importance of selecting appropriate features for precision accuracy. These studies form the foundation for developing models to detect phishing scams in email correspondence, outlining the techniques and measures commonly employed for this purpose.

This research aims to explore various methods for detecting phishing emails using machine learning techniques. By systematically evaluating and comparing different models, including Feed Forward, DNN, Wide and Deep, and the latest architecture – TabNet, the study examines their strengths, weaknesses, and practical applications in phishing detection. The evaluation of each model strengthens the argument regarding their effectiveness in addressing phishing attacks. This highlights the pressing need to advance effective cybersecurity measures to protect users from the growing threat of phishing.

Additionally, the study investigates a new evaluation metric called the "anti-phishing score," which addresses both false positives and false negatives. Data-driven approaches and optimization techniques, such as grid search for hyperparameter tuning, enable researchers to identify the most suitable models for phishing detection. A script was developed to automate the model generation process with new datasets, enhancing the productivity of academic research in this area. A holistic approach to phishing detection is proposed, utilizing classical feature selection procedures, modern machine learning algorithms, and comprehensive evaluation methods to address evolving cyber risks.

This research intends to develop as well as test machine learning systems which are particularly made for tracking phishing e-mails to empower cybersecurity by weakening the evolving cyber threats. The main contributions are summarized as follows:

- The model chosen was a feed-forward one and was trained with a dataset from PhishTank comprising 111 features, forming the basis of effective phishing detection.
- The feed-forward approach was identified as the most effective for the field project through careful evaluation.
- Grid search optimization was applied to fine-tune the hyperparameters of the feed-forward model, achieving target accuracy and efficiency.
- Extensive evaluation on a new dataset demonstrated the model's effectiveness in detecting phishing elements across various contexts.

II. LITERATURE SURVEY

This section provides an overview of the existing research in phishing detection using machine learning techniques, emphasizing feature selection and deep learning models. Table 1 summarizes key contributions from selected works.

III. MATERIALS AND PROPOSED METHODS

A. DATASETS AND TOOLS

In the initial stage of the research design, the first step was to obtain various datasets to support the experiments. One of the pivotal datasets utilized was sourced from Mendeley 2020 [7]. This dataset, referred to as Dataset 1, is primarily used in subsections V-A, V-B, and V-C for training and testing the model. The label feature in Dataset 1 is 'phishing', with the values 0 and 1, where 0 represents legitimate and 1 represents phishing. Dataset 1 proved to be indispensable to the research process. Through meticulous analysis, the structure of URLs was examined, identifying five key components: Domain, File, Directory, Parameters, and the complete URL. The contributors to Dataset 1 carefully parsed various URLs, and the features of these URLs were analyzed in detail. These features included a comprehensive set of signs for each stage, providing valuable information for the research.

Furthermore, another dataset from 2021, referred to as Dataset 2 [8], was incorporated into the experimental process. The performance of models during the hyperparameter phase mentioned in V-C can be validated and tested with Dataset 2. The label feature in Dataset 2 is 'status', with values 'legitimate' and 'phishing'. The primary goal of using these datasets was to enhance the robustness and reliability of the research findings.

Python was selected as the main programming language for model development and implementation due to its versatility and ease of use. Python's adaptability made it well-suited for implementing deep learning and machine learning algorithms. Additionally, TensorFlow and Keras were

TABLE 1. Summary of Literature Survey

Author [Year, Ref.]	Work	Remarks
Salahdine et al. [2016, [1]]	Proposed a machine learning-based phishing detection technique using a dataset of 4000 phishing emails with 10 selected features. Achieved 94.5% accuracy using an ANN.	Reinforced the importance of feature selection and neural networks for high accuracy. Influenced the focus on robust evaluation metrics.
Baykara and Gürel [2018, [2]]	Developed the "Anti Phishing Simulator," detecting phishing and spam emails with a Bayesian algorithm and URL-based control.	Highlighted Bayesian classification and URL-based analysis. Informed feature selection and keyword expansion strategies.
Chiew et al. [2018, [3]]	Provided a systematic review of phishing attack approaches, mediums, and vectors. Explored gaps in anti-phishing efforts.	Guided development of feature-based detection mechanisms and cybersecurity defenses.
Yahya et al. [2019, [4]]	Discussed supervised learning algorithms for detecting phishing websites, achieving 97.6% accuracy with KNN and low false-negative rates with Random Forest.	Showed effectiveness of ML algorithms for phishing detection and informed model selection strategies.
Chinnasamy et al. [2020, [9]]	Proposed heuristic-based classification of phishing URLs using features like web traffic and URL structure, achieving 94.73% accuracy.	Highlighted integration of heuristic techniques with ML and potential of hybrid approaches.
T. R. N and Gupta [2020, [10]]	Reviewed feature selection methods, emphasizing their impact on computational efficiency, accuracy, and model predictability.	Provided foundational understanding for refining feature engineering in phishing detection.
Ramachandran et al. [2020, [11]]	Reviewed dimensionality reduction techniques like PCA, LDA, and ICA for handling big data.	Identified optimal dimensionality reduction methods for phishing datasets.
Dangwal and Moldovan [2021, [12]]	Demonstrated effectiveness of feature selection tools in phishing detection, achieving 98.11% accuracy with Random Forest.	Highlighted importance of feature selection for efficient detection models.
Alswailem et al. [2019, [5]]	Presented a browser extension using Random Forest for phishing detection, achieving 98.8% accuracy with 26 selected features.	Demonstrated practical application of ML for real-time phishing detection in browsers.
Almseidin et al. [2019, [24]]	Phishing Detection Based on Machine Learning and Feature Selection Methods. Employs machine learning and feature selection to detect phishing websites, achieving 98.11% accuracy with Random Forest using 20 selected features.	Highlights the role of feature selection in improving phishing detection efficiency.
Zuhair et al. [2016, [6]]	Surveyed feature selection methods, discussing their impact on classification accuracy and efficiency.	Offered insights into hybrid detection methods and robust feature selection.
Wei and Sekiya [2022, [13]]	Proposed a framework for reducing phishing dataset features, achieving 97% accuracy using only 14 features.	Provided practical optimization for phishing detection models with reduced feature sets.
Rajeswary et al. [2023, [15]]	Proposed an LSTM-driven model for phishing detection in the Tor network, addressing URL-based, cloning, and network metric attacks.	Demonstrated application of LSTM for real-time detection in dynamic networks.
Yu et al. [2022, [19]]	Proposed a multi-feature neural network for phishing detection, combining MLP, CNN, and RNN. Achieved 97.75% accuracy and 99.01% recall.	Demonstrated robust detection with multi-feature neural networks.
Jayaraj et al. [2024, [25]]	Introduced Hybrid Ensemble Feature Selection (HEFS) for phishing URL detection. Focused on URL features and addressing unauthorized attacks.	Provided a novel feature selection approach for ML-based phishing detection systems.
Tanimu and Shiaeles [2022, [22]]	Proposed phishing detection using image visualization of website code and feature extraction from malicious URLs.	Highlighted combination of image-based techniques and feature elimination for detection.
Borisov et al. [2021, [26]]	Explored state-of-the-art methods in deep learning for tabular data, categorized into transformations, architectures, and regularization. Addressed challenges in applying DNNs to tabular data.	Provided guidance for using DNNs with tabular datasets, focusing on challenges like overfitting and interpretability.
Ye et al. [2024, [27]]	Proposed a benchmark of 300 tabular datasets to evaluate and compare DNN-based methods with tree-based methods. Identified key factors influencing DNN success.	Provided insights into the performance of DNNs on tabular data and offered tools for phishing detection research.
Cheng et al. [2023, [28]]	Introduced SHAPNN, a deep tabular model leveraging Shapley value regularization for interpretability and robustness. Improved AUROC, transparency, and continual learning.	Highlighted innovative use of Shapley values for interpretability, relevant for phishing detection research.
Bondarenko [2021, [29]]	Proposed an end-to-end regression algorithm for tabular data using a deep ensemble of self-normalizing neural networks. Achieved top results in a competition.	Demonstrated adaptability of DNNs for structured data, offering insights into addressing generalization and uncertainty challenges.

instrumental for machine learning model development and optimization, significantly enhancing the efficiency and effectiveness of the research.

TABLE 2. Summary of datasets: total instances, legitimate and phishing URLs, and features.

Dataset	Total Instances	Legitimate	Phishing	Features
Dataset 1	58,645	27,998	30,647	111
Dataset 2	11,430	5,715	5,715	87

B. PROPOSED DEEP LEARNING METHODS

This research involved the implementation of four specific Deep Learning(DL) models targeting phishing detection tasks. These models included an FNN, a DNN, a Wide and Deep Model, and a TabNet Model.

The FNN and DNN designs consisted of multiple fully connected layers with ReLU activation functions and a sigmoid unit at the output layer. Additionally, the Wide and Deep Model combined wide and deep components using TensorFlow Keras's functional API. The wide component was effective at memorizing intricate details, while the deep component captured complex patterns. The Wide and Deep Model leveraged these strengths for robust classification.

Furthermore, the TabNet Model, a deep learning model featuring a novel sequential attention mechanism for tabular data, was employed. The TabNet model was initially trained using a TabNet Pretrainer, followed by fine-tuning for the classification tasks.

The hyperparameters for the DL models were optimized to achieve maximum performance.These parameters are selected based on experimentaion. Key hyperparameters included dropout rates, epochs, batch size, and early stopping criteria. The grid search optimization method was applied to determine the best hyperparameters for each model structure.

TABLE 3. DL Hyperparameters for Training the Datasets.

Parameter	Value
Epochs	10
Loss Function	Binary Crossentropy
Batch Size	64
Optimizer	Adam
Hidden Activation	ReLU
Output Activation	Sigmoid
Metrics	Accuracy

1) The Feedforward Neural Network (FNN) Model

The Feedforward Neural Network (FNN) is a vital domain of deep learning, as the architecture appears to be flexible and capable of handling classification problems across diverse areas. Unlike RNNs or CNNs that process inputs with temporal or spatial dependencies, FNNs process input data sequentially, but without any cyclic dependency, making them particularly suitable for tasks that do not require attention to order or sequence.

The architecture of the FNN model consists of three interconnected layers: an input layer, a hidden layer, and an output layer (as shown in Figure 1). The input layer receives a feature vector representing preprocessed email data, which typically includes basic information relevant to phishing detection. The number of neurons in the input layer corresponds to the dimensions of the input features, which depend on the type and representation of the selected features. The key computational operations of the FNN model take place in its hidden layer(s) (see Figure 1), where it extracts significant features from the input data through weighted connections and non-linear activation functions. In this implementation, Rectified Linear Unit (ReLU) activation functions are applied in all hidden layers, enabling the model to classify complex data patterns in non-linear ways, thereby enhancing its effectiveness.

The output layer, positioned at the end of the model, provides the final predictions or classifications (refer to Figure 1). For phishing detection, the output layer contains a single neuron with a sigmoid activation function that outputs a score representing the probability of phishing. The sigmoid function maps these probabilities to the interval [0,1], facilitating interpretability. The Adam optimizer is employed to train the FNN model, chosen for its robustness and reliability in handling gradients, ensuring efficient optimization. Binary cross-entropy is used as the loss function, which is particularly suitable for binary classification tasks like phishing detection. This function minimizes the difference between the predicted probability and the true class labels, thereby optimizing the model's parameters.

The performance of the FNN model is assessed using the accuracy metric, defined as the ratio of correctly classified samples to the total number of samples. By carefully selecting hyperparameters and refining the model's architecture, the goal is to develop a robust and efficient phishing detection system based on the FNN, capable of accurately distinguishing between phishing emails and legitimate messages, thus achieving an optimal balance between precision and performance.

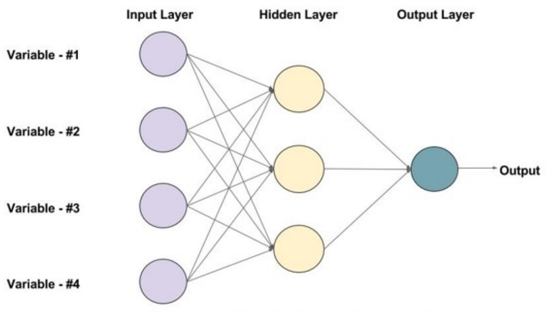

FIGURE 1. Architecture of the Feedforward Neural Network (FNN) model [33]

.

2) The Deep Neural Network (DNN) Model

This model undoubtedly is the most influential, as it stands out as a basic block in the world of deep learning methodologies. Known for its flexibility together with its outstanding tendency to be applied in the field of classification tasks, DNNs can prove to be indeed efficacious in detecting phishing. As shown in the Figure 2, DNNs are designed with a multilayered architecture, which is utilized to extract complex patterns and features progressively.

The DNN model is meticulously designed with a chain of densely interconnected layers refer to Figure 2; hence, from management elements to financial outcomes to the competitive landscape, the model will scrutinize them all. The hidden layers of these networks, each one containing a dense network of neurons, function as the engine room of the model by carefully handling complex calculations and ultimately identifying consistent and meaningful trends in the data. This network is constructed with many hidden layers, in which Rectified Linear Units (ReLUs) are applied at every hidden layer. The introduction of a non-linear characteristic in the model helps it learn the essential data representations. The operation expressed as $f(x) = \max(0, x)$, allows the ReLU activation function to preserve only non-negative values, leading to a more accurate prediction of complex relationships within the dataset.

At the top layer of the DNN structure, depicted in Figure 2, is the output layer or node, which discloses the final results or classifications. Regarding phishing detection, this layer consists of a lone neuron utilizing the sigmoid activation function, which produces an output probability in the range from 0 to 1. This score represents the probability of a sample being a phishing attack. The sigmoid function creates the necessary atmosphere by providing an output probability range that is within this range and a CAP number that can be easily interpreted.

Training of the DNN model is controlled by the Adam optimizer, which is popular and of great value for gradient-based optimization. Such an optimizer is crucial for guiding the model to the optimal parameter values and enhancing prediction accuracy. The binary cross-entropy is accepted as the proper cost function during training, as it is most suitable for binary classification tasks, such as phishing detection. Binary cross-entropy evaluates the dissimilarity between the expected probabilities and the actual class labels and serves as a critical input to the model while optimization takes place. The DNN model is mostly evaluated according to accuracy, a major quality criterion of how the model works. Accuracy is the ratio of correctly classified samples to all samples in a dataset and demonstrates the extent to which the model correctly predicts. The core aim here is to scrutinize and calibrate every parameter sufficiently and also optimize the network to create a robust and reliable phishing detection system, while applying the efficiency and flexibility of the DNN model.

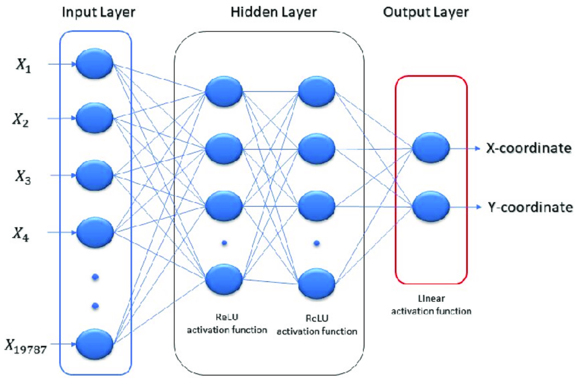

FIGURE 2. Architecture of the DNN model [32]

3) The Wide and Deep Model (WDM)

This model synthesizes the best of two learning techniques, shallow learning, and deep learning, providing the most wide-ranging solution for classification tasks such as phishing detection. A functional API design is used in the WDM architecture, which includes deep and wide components to fully utilize their complementary strengths (see Figure 3). Both wide and deep models are effective at storing detailed information from the datasets, but deep models excel at finding complex patterns. In essence, the WDM takes advantage of these strengths to enhance the accuracy of the classification.

The wide model consists of layers that are very densely connected and functions as a memorization engine that is good at capturing high-dimensional features from sparse inputs. The deep section, through its multilayered architecture, excels at learning hierarchical data representations and understanding complex relationships between data.

The wide model has two dense layers with ReLU activation functions, which enable feature extraction and transformation that are computationally efficient. The deep model, however, consists of four dense layers with the same ReLU non-linear activation functions. To combine these components, the Concatenate layer merges their outputs, which facilitates information interchange between the branches. The combined architecture culminates in the final layer containing one neuron that uses a sigmoid function to generate a probability score for phishing attempts.

Training of the WDM model is implemented using the Adam optimizer, a gradient-based optimization method. Binary cross-entropy, the chosen loss function, measures the deviation between predicted probabilities and true class labels during the training process. The evaluation metric used is accuracy, a fundamental measure of the model's efficiency. Through accurate adjustment of hyperparameters and careful architectural tuning, the system is ensured to be robust and dependable. This works in synergy with the WDM.

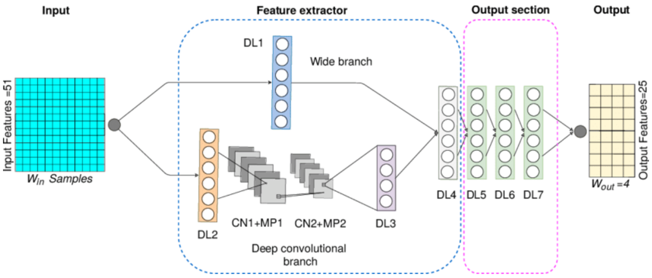

FIGURE 3. Architecture of the Wide and Deep model [30].

4) The TabNet Model

The novelty of this model lies in the deployment of a deep learning architecture focused on the analysis of structured data. As a result, it has demonstrated high competence in numerous categorization processes, such as phishing detection (as shown in Figure 4). In contrast to traditional methods, this model uses the concept of sequential attention mechanisms, where highly informative features are identified and incorporated into the overall classification process of the dataset. This enhances the model's capability to classify data based on identified and processed features.

The principal phase of the TabNet model implementation is the pre-training phase using the TabNetPretrainer on standardized input data. This enables the model to capture the key features during training, which favors deeper learning and improved data interpretation in subsequent training stages. Afterward, the TabNet network is designed for classification. The main hyperparameters are carefully selected (as Figure 4 indicates) to balance model complexity and generalization capability.

Training involves iterative epochs, with early stopping employed to combat overfitting and promote faster convergence. The Adam optimizer, a well-established gradient-based optimization algorithm, is applied. The model's performance is measured using an independent testing set, with accuracy as the main evaluation metric for determining predictive performance. The goal of iterative training with optimal solutions is to minimize overfitting. Once convergence is achieved, the model is prepared for deployment on phishing detection, with its reliability grounded in its robust prediction architecture.

The objective is to leverage TabNet's attention-based mechanisms and feature-extraction abilities to develop a high-performance phishing detection system that efficiently distinguishes phishing emails from legitimate communications.

IV. METHODOLOGY

The methodology used to enhance the effectiveness of phishing attack detection via machine learning in cybersecurity follows a step-by-step plan. Access to sufficient and up-to-date datasets presents one of the major challenges, as phishing websites continue to evolve. On the path to a solution, scientists utilize features extracted from PhishTank, a phishing dataset, which proves helpful in enabling similar performance indicators to be utilized across various models. The roadmap for the research approach is illustrated in Figure 5, which is constructed to provide a logical and ordered approach to the research methodology.

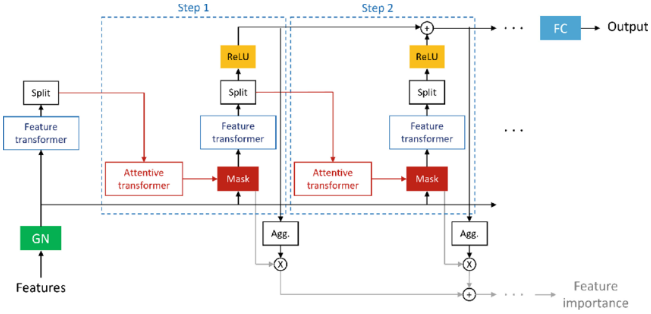

FIGURE 4. Architecture of the TabNet model [31].

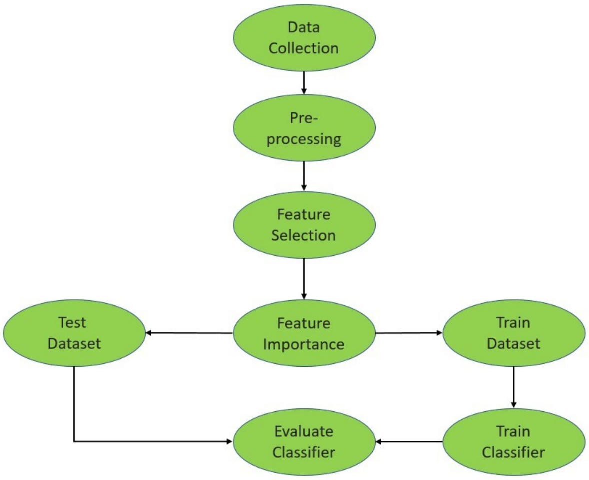

FIGURE 5. Flow diagram illustrating the proposed classification system for phishing detection based on machine learning.

A. DATA COLLECTION

The first phase of the experimentation involved data acquisition for training and supervision of the machine learning approach to phishing detection. One of the first observations was the basic structure of URLs, which are essential for accessing web resources. Figure 6 depicts the five distinct parts that comprise a URL: Directory, File, Domain, the complete URL itself, and Parameters.

Feature extraction depends on mastering URL structure, a vital part of SEO. The researchers who compiled the 2020 Mendeley dataset also utilized this knowledge by disassembling URLs into their basic elements, as shown in Figure 6. During this process, 17 features were generated, with each extracted feature corresponding to the count of specific characters in a particular URL component (e.g., folder name

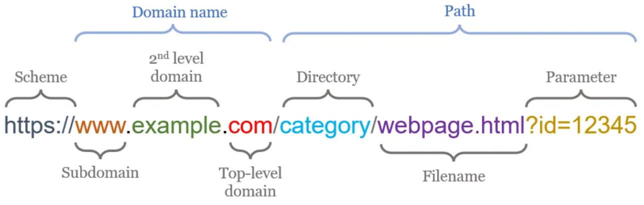

FIGURE 6. An Example of URL Structure [34].

separators, number of dots in the domain, etc.). Extracting essential features from a diverse set of URLs provided valuable information for model training and subsequent analysis.

The data collection effort extended beyond the Mendeley dataset for 2020. Extensive effort was made to explore additional sources of datasets covering a wide range of phishing modalities, ensuring that the models could adapt to real-world scenarios. The main objective of this data collection was to establish a solid foundation from which to train machine learning models capable of effectively detecting phishing in various scenarios.

B. PREPROCESSING

The preprocessing stage is essential for ensuring the quality and consistency of the dataset. Key responsibilities during this stage include handling single-valued columns, addressing columns containing only two values where one is -1 (which renders the column defective), and imputing missing values. Additionally, Min-Max normalization is used to rescale the data, thereby ensuring the reliability of data quality. Overall, this preprocessing helps make the data more explanatory and suitable for later analysis and model training.

C. FEATURE SELECTION

Among all the stages of training that improve the accuracy of machine learning models, selecting a relevant feature set is the most critical. Correlation analysis, a principal approach to feature selection, is widely used to identify features with a correlation coefficient greater than 0.8, indicating a strong linear relationship between variables. This analysis is crucial for setting up the training and testing processes by emphasizing the most powerful variables, thus enhancing model performance and achieving the best mix of features for accurate predictions. To ensure the effectiveness of the feature selection process, a comparison was made between the permutation importance method and three other widely used techniques: Information Gain, Chi-Square, and Fisher's Score. The findings highlighted that permutation importance provides a more comprehensive understanding of feature relevance across models, justifying its use in this study. This thoughtful screening methodology improves dataset efficiency, facilitates better model training, enhances detection model accuracy, and strengthens phishing attack detection.

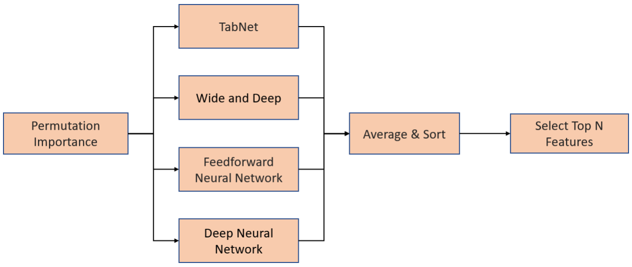

FIGURE 7. Flowchart illustrating the Permutation Importance based Feature Selection process in the machine learning pipeline.

D. FEATURE IMPORTANCE

Recognizing feature importance is essential for developing high-performing models. In this research, the permutation test will be used to evaluate the contribution of each feature to the Deep Neural Network (DNN), Feedforward Neural Network (FNN), TabNet, and Wide and Deep model. In permutation importance analysis, features are randomly scrambled to examine changes in model performance when each feature is perturbed individually. Features that cause significant reductions in model accuracy upon permutation are deemed crucial, as they significantly affect the model's predictions. Once the permutation importance analysis is completed for all models, the resulting scores will be averaged and ordered, indicating the top features that contribute most meaningfully to model performance. This feature prioritization process is central to developing more effective models, leading to stronger and more precise machine-learning solutions for phishing attack detection.

E. TRAIN AND TEST MODEL

The model development phase involves the implementation, training, and testing of the DNN, TabNet, FNN, and Wide and Deep models. This sequential procedure starts with training each model using the top N features selected through the feature selection process, which are considered the most relevant. To ensure the reliability of the research results, the rigorous 10-fold cross-validation methodology will be applied. This approach provides more opportunities to evaluate each model's performance and reduces the likelihood of selecting an overfit model. It refines the models and serves as a strong basis for future analyses and optimization processes.

F. EVALUATION

Evaluation is the final stage used to quantify the models with parameters such as accuracy, TPR, FPR, testing time, and anti-phishing score. This detailed analysis offers valuable insights into the efficiency of machine learning algorithms in detecting phishing attacks. The evaluation procedure includes both quantitative and qualitative metrics that describe the models' performance. Accuracy assesses the general correctness of the model's predictions, while precision determines the model's ability to accurately identify positive predictions. The 'Anti-Phishing Score' is a comprehensive metric that considers various factors for accurate phishing attack detection. This thorough evaluation is crucial for validating the practical usefulness of machine learning models in real-world scenarios. It will also guide further improvements and updates to ensure the models effectively address the complexity of phishing attacks.

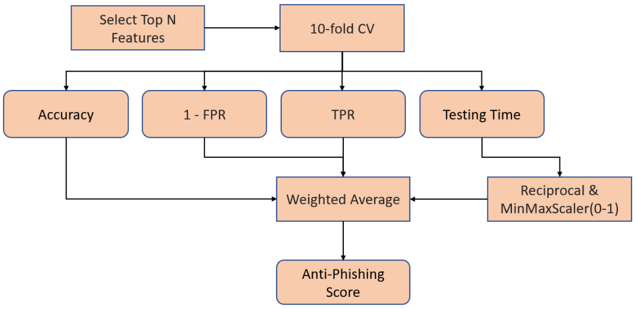

FIGURE 8. Evaluation Workflow for the Feature Selection Approach.

V. RESULTS AND DISCUSSION

The purpose of using Google Colab, which provided an edge in the phishing detection field, was to utilize its computing resources. By leveraging codes from various Python libraries such as TensorFlow and Keras, an in-depth study of phishing networks was conducted. This approach provided insightful analysis and a strong foundation for developing detection models.

A. FEATURE SELECTION

Feature selection is a critical step in developing phishing detection systems as it reduces computational complexity, mitigates overfitting, and enhances interpretability while maintaining high model performance. In this study, feature selection was conducted using Permutation Importance, a model-agnostic technique that evaluates the significance of features by measuring the decrease in model performance when a specific feature's values are randomly shuffled. This method captures each feature's unique contribution within the model's prediction process and was applied across four models: Deep Neural Network (DNN), Feedforward Neural Network (FNN), TabNet, and Wide and Deep model. The dataset was initially reduced to 73 features through preprocessing and feature selection steps, and the top 20 most relevant features were identified, including 'qty_slash_url', 'time_domain_activation', 'length_url', 'qty_mx_servers', and 'tls_ssl_certificate'. These features consistently ranked highly across all models, underscoring their critical role in distinguishing phishing websites from legitimate ones.

The decision to focus on the top 20 features from the initial 111 was justified by several factors. Selecting a smaller feature subset reduces latency and computational demands during both training and testing, enabling real-time phishing detection. Empirical studies supported this approach; for instance, Almseidin et al. demonstrated that using 20 features out of 48 with a Random Forest model achieved an accuracy of 98.11%. Additionally, emphasizing a smaller set of features minimizes redundancy by prioritizing the unique contributions of each attribute, which is vital for computational efficiency and robustness.

To validate the effectiveness of Permutation Importance, its results were compared with those obtained using Information Gain, Chi-Square, and Fisher's Score. These traditional methods rely on statistical properties, with Information Gain measuring the mutual information between features and the target variable, Chi-Square evaluating feature independence, and Fisher's Score using a ratio of inter-class variance to intra-class variance. However, such methods are inherently limited by their univariate nature, which does not account for interactions between features or their joint contributions to predictions. In contrast, Permutation Importance provides a holistic evaluation of features in the context of model performance, making it more suitable for complex machine learning models. For example, features like 'qty_slash_url' and 'time_domain_activation' demonstrated significant combined influence on phishing detection, effectively captured by Permutation Importance but overlooked by univariate methods.

Features selected through Permutation Importance demonstrated remarkable consistency across all models. Attributes such as 'length_url', 'qty_mx_servers', and 'tls_ssl_certificate' were consistently ranked among the top-performing features, whereas features identified by statistical methods varied significantly depending on dataset-specific properties. Furthermore, traditional statistical methods often selected redundant features, such as 'qty_dot_directory' and 'qty_dot_file', which provide overlapping information. Permutation Importance mitigated this redundancy by evaluating each feature's unique impact on model performance.

From a practical perspective, the use of Permutation Importance not only improved the accuracy of the phishing detection system but also ensured computational efficiency by reducing feature dimensionality without compromising performance. This reduction aligns with the goal of maintaining low latency and efficient resource utilization. Features identified, such as 'directory_length' and 'time_domain_expiration', directly enhance the system's ability to identify phishing websites while adapting to evolving threats. By accounting for interactions between features and their combined contributions to model predictions, Permutation Importance demonstrated clear advantages over traditional statistical methods, establishing itself as an essential tool for feature selection in phishing detection systems.

B. MODEL EVALUATION

The model evaluation process was conducted using ten-fold cross-validation, which ensured a comprehensive assessment of the models in terms of phishing detection. To address potential sample bias, stratified sampling was employed, ensuring that each fold in the cross-validation maintained a

TABLE 4. Features Selected by Different Feature Selection Methods

Information Gain	Chi-Square	Fisher’s Score
qty_slash_url	qty_mx_servers	qty_slash_url
directory_length	qty_hyphen_domain	time_domain_activation
length_url	qty_tilde_params	qty_dot_file
file_length	server_client_domain	length_url
asn_ip	qty_hyphen_file	qty_dot_directory
time_domain_activation	ttl_hostname	qty_dot_domain
qty_dot_directory	qty_tilde_url	qty_equal_url
qty_dot_file	tld_present_params	directory_length
ttl_hostname	qty_underline_domain	qty_tld_url
params_length	qty_dollar_url	tld_present_params
time_domain_expiration	url_google_index	qty_at_params
qty_dot_domain	qty_underline_url	qty_underline_url
qty_equal_url	url_shortened	qty_vowels_domain
qty_equal_params	qty_underline_directory	time_domain_expiration
qty_hyphen_url	domain_spf	qty_hyphen_url
qty_dot_params	qty_underline_params	qty_dot_url
qty_tld_url	qty_plus_params	qty_nameservers
qty_dot_url	qty_space_url	domain_spf
qty_underline_url	tls_ssl_certificate	qty_underline_params
tld_present_params	domain_in_ip	ttl_hostname

TABLE 5. Top Features Selected through Permutation Importance Analysis

Feedforward Neural Network	Deep Neural Network	TabNet	Wide and Deep
time_domain_activation	qty_slash_url	qty_slash_url	qty_slash_url
qty_slash_url	time_domain_activation	length_url	length_url
length_url	length_url	time_domain_activation	time_domain_activation
url_shortened	qty_mx_servers	qty_dot_directory	file_length
domain_spf	tld_present_params	url_shortened	directory_length
tls_ssl_certificate	qty_at_params	qty_underline_url	qty_dot_domain
tld_present_params	tls_ssl_certificate	qty_dot_domain	qty_mx_servers
qty_dot_directory	url_shortened	qty_hyphen_domain	qty_at_url
qty_dot_domain	qty_nameservers	qty_ip_resolved	time_domain_expiration
qty_nameservers	directory_length	qty_hyphen_params	qty_nameservers
qty_hyphen_domain	asn_ip	qty_hyphen_url	qty_vowels_domain
qty_ip_resolved	qty_hyphen_domain	qty_equal_directory	tls_ssl_certificate
qty_mx_servers	qty_dot_url	qty_slash_params	qty_at_params
domain_in_ip	qty_ip_resolved	qty_percent_url	qty_ip_resolved
qty_underline_url	qty_dot_directory	time_domain_expiration	url_shortened
qty_hyphen_url	qty_dot_domain	qty_exclamation_file	qty_dot_url
qty_vowels_domain	qty_hyphen_url	qty_vowels_domain	qty_dot_file
qty_exclamation_file	qty_equal_directory	qty_underline_directory	qty_equal_params
qty_tld_url	qty_vowels_domain	qty_and_file	qty_hyphen_url
ttl_hostname	domain_in_ip	qty_equal_params	qty_equal_directory

representative distribution of both legitimate and phishing instances. This approach mitigated the risk of any fold being disproportionately skewed towards one class, thereby enhancing the reliability of the evaluation. A variety of metrics were analyzed to assess the effectiveness of the models, including accuracy, false positive rates (FPR), true positive rates (TPR), and testing time. An anti-phishing score, a comprehensive weighted measurement of performance, was used to provide a holistic view of the model’s effectiveness. The ’anti-phishing score’ functions as a multi-dimensional metric, reviewing performance on each aspect separately. This approach considers not only critical metrics such as accuracy, FPR, and TPR but also testing time, which is crucial in real-world applications where prompt detection is key.

The method for calculating the anti-phishing score is as follows:

$$\begin{aligned}\text{Anti-Phishing Score} &= 0.3 \times \text{Accuracy} + 0.25 \times (1 - \text{FPR}) \\ &\quad + 0.25 \times \text{TPR} + 0.2 \times \text{Testing Time}\end{aligned}$$

Each component of the anti-phishing score formula contributes uniquely to the overall assessment:

- **Accuracy (0.3):** Accuracy indicates how well the model classifies the instances based on the given classes, which is crucial for classification tasks. By giving greater weight to the accuracy score, the anti-phishing evaluation emphasizes the model’s classification performance with the aim of accurate discrimination of phishing attempts.
- **False Positive Rate (FPR) (0.25):** FPR indicates the proportion of legitimate cases incorrectly classified as phishing. This is a critical factor in correctly identifying harmful entities without mistakenly marking legitimate

TABLE 6. Top Features Selected through Permutation Importance Analysis

Feature	Description
qty_slash_url	Quantity of slashes in URL
time_domain_activation	Time since domain activation
length_url	Length of URL
qty_mx_servers	Quantity of MX servers
qty_dot_directory	Quantity of dots in directory
qty_dot_domain	Quantity of dots in domain
url_shortened	Presence of URL shortening
directory_length	Length of directory
file_length	Length of file
tls_ssl_certificate	Presence of TLS/SSL certificate
qty_nameservers	Quantity of nameservers
qty_at_params	Quantity of '@' in parameters
qty_ip_resolved	Quantity of resolved IP addresses
tld_present_params	TLD presence in parameters
qty_hyphen_domain	Quantity of hyphens in domain
qty_at_url	Quantity of '@' in URL
qty_vowels_domain	Quantity of vowels in domain
qty_hyphen_url	Quantity of hyphens in URL
time_domain_expiration	Time until domain expiration
domain_spf	Presence of SPF in domain

ones as malicious. The anti-phishing score is based on 1 minus FPR to normalize all metrics on the same scale, so lower FPR values lead to better model performance and thus a higher anti-phishing score.

- **True Positive Rate (TPR) (0.25):** TPR represents the ratio of actual phishing instances correctly identified, reflecting the algorithm's ability to detect phishing attacks. Together with FPR and precision, the score is calculated with equal weightage. Both TPR and FPR are considered to minimize the risk of incorrect classification, which could result in either misidentifying a benign instance as malicious or missing an actual threat.
- **Testing Time (0.2):** Operational efficiency, reflected in testing time, is crucial for real-time phishing detection systems. By incorporating the inverse of the test time before scaling, the anti-phishing score accounts for computational efficiency, favoring models with shorter processing times and thus optimizing operational performance.

This weighted performance metric considers both classification accuracy and operational efficiency, applying it to real-time phishing detection scenarios. Table 7 shows the performance measures of the models, highlighting their anti-phishing scores to emphasize their effectiveness in combating phishing attacks.

TABLE 7. Model Performance Metrics

Model	Accuracy	FPR	TPR	Testing Time	Anti-Phishing Score
DNN	0.9392	0.0456	0.9103	0.6492	0.7479
FNN	0.9427	0.0479	0.9250	1.7470	0.9521
TabNet	0.9382	0.0494	0.9147	1.3685	0.7420
Wide & Deep	0.9139	0.0854	0.9126	0.7100	0.8788

The visualized model performance metrics, such as ROC and precision-recall curves, enable an in-depth understanding of the efficacy of the phishing detection models. Figures 9 and 10 illustrate ROC and precision-recall curves, respectively, for the four different models: Feedforward, DNN, Wide & Deep, and TabNet, as well as a random classifier for comparability. ROC curves show the balance between TPR and FPR, whereas precision-recall curves demonstrate the trade-off between precision and recall. These graphical elements provide crucial information about the models, facilitating informed decision-making by elucidating how each model achieves its true positives-false positives balance and precision-recall, which are essential factors in real-world phishing detection.

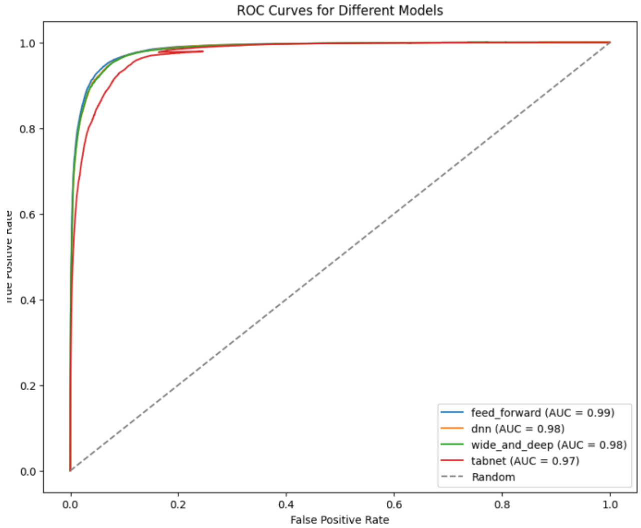

FIGURE 9. ROC Curves

Notably, all models demonstrate ROC curves positioned favorably, indicating high true positive rates (TPR) and low false positive rates (FPR). The Area Under the Curve (AUC) is a measure of the model's performance, with a higher AUC indicating better results. Among the different models utilized, the Feedforward model has the best AUC of 0.99, while the DNN and Wide & Deep models are tied for second with AUCs of 0.98. TabNet has an overall AUC of 0.97. Nevertheless, the precision-recall curves imply that all models are moving toward higher precision while maintaining a decreasing recall rate as precision increases. This indicates an accuracy-recall trade-off, necessitating further research and parameter tuning for the best outcome.

Considering the different models mentioned, the FNN (Feedforward Neural Network) stands out. The system is notably impressive because it achieved a 0.9521 anti-phishing score, thereby leading the way in phishing detection.

C. OPTIMIZING MODEL PERFORMANCE: GRID SEARCH HYPERPARAMETER TUNING

The goal of hyperparameter grid search was to achieve the appropriate level of effectiveness and efficiency, meaning that

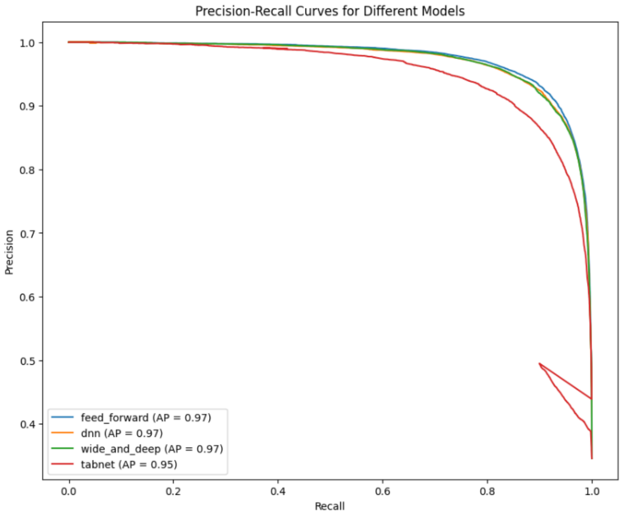

FIGURE 10. Precision Recall Curves

phishing detection will be done correctly but the speed of the whole process will not be compromised. Stratified sampling was employed during the grid search process to maintain the class distribution in both the training and test sets, ensuring reliable evaluation across different hyperparameter configurations. In Table 8, the result of this search is displayed, highlighting the best-fitting model as well as an alternative one with slightly different features.

The best result was obtained when an architecture with 20 columns of interest was trained 40 times and achieved 95.53% accuracy when compared to the chosen model, which also underwent 14 columns of interest but only trained 50 times, resulting in an accuracy of 94.46%. Nevertheless, although there are minor errors, this model has the advantage of reduced computational costs. This puts Feedforward Neural Network at the top in both accuracy and anti-phishing scores by showing superiority over other models such as DNN, TabNet, and Wide & Deep. The obvious superiority was seen by the performance metrics which are represented in Table 7. Although the others achieved great results, the FNN demonstrated the best effectiveness and accuracy.

Furthermore, it is relevant to underline that the new dataset applied for validation and model selection also played a role. Referring to Dataset 2 described in III-A, it was crucial for accurately benchmarking the model's performance and finding the best hyperparameters. Dataset 2 provided the 'URL' feature; from these URLs, the same features selected from permutation importance were extracted. The number of features selected depended on the grid search. The values of those features were extracted from these URLs using the scripts provided on the Dataset 2 website. Dataset 2 is the updated 2021 version, so the precision rate of 80% is an efficient indicator of improvement. Its accessibility and suitability for verifying model accuracy eventually resulted in the selection of the Feedforward Neural Network as the best approach.

The chosen model represents a compromise between accuracy and computational efficiency, achieved by reducing the number of relevant columns and only slightly increasing the amount of training data. Such an approach greatly saves computation time while retaining high accuracy, making it an optimal solution for real-time phishing detection, where velocity is a key factor. In addition, both models use a test size of 0.2 for rigorous validation and verification of correct performance, leading to acceptable results. The variables of learning rate, batch size, and optimizer are consistent between the two approaches, which makes them crucial for model training and accuracy. After extracting the values from the URLs using the scripts from the Dataset 2 website, the model was validated with these values. After applying the chosen model to Dataset 2, it achieved 80% accuracy on new data, suggesting its applicability and competency in practice. Although the accuracy level may be slightly lower compared to the best result, this model is the best choice for practical advantages such as speed and efficiency, making it preferred for deployment in real-time phishing detection systems.

TABLE 8. Hyperparameter Grid Search Results

Parameters	Best Result	Chosen Model
Columns of Interest	20	14
Epochs	40	50
Test Size	0.2	0.2
Batch Size	128	64
First Hidden Layer	64	128
Learning Rate	0.01	0.001
Second Hidden Layer	64	64
Optimizer	Adam	Adam
Accuracy on Data	0.9553	0.9446
Accuracy on New Data	81.05%	80%

D. AUTOMATED SCRIPT

Feature selection and hyperparameter tuning scripts will be presented as an innovative way of combining phishing tactics to strengthen them. Another significant step is minimal human intervention as a definite feature, thus, putting the research on the threshold of automated detection and monitoring of new phishing attacks. This article focuses on a new technology that has a transformative nature for the framework of phishing detection making it more robust. Furthermore, these scripts could be developed to cope with growing phishing attacks.

VI. CONCLUSION AND FUTURE WORK

The research identified Feed Forward Neural Networks (FNN) as the most effective model for phishing detection due to its strong performance and generalizability. This underscores the importance of rigorous experimentation and evaluation in selecting effective models for real-world applications. Additionally, a phishing prevention browser extension was developed, which utilizes web crawling techniques to extract URLs from web pages. These URLs are sent to a Flask server, which employs an advanced learning algorithm to identify harmful URLs, allowing users to quickly assess their safety. The development of this extension demonstrates the practical application of the research findings in enhancing cybersecurity measures.

Future research could explore integrating phishing detection systems with other cybersecurity measures. For example, the study by Jayaraj et al. [25] discusses intrusion detection based on phishing detection with machine learning and introduces the Hybrid Ensemble Feature Selection (HEFS) method. This method employs a Cumulative Distribution Function gradient (CDF-g) algorithm to refine feature subsets, thereby enhancing the accuracy and efficiency of phishing detection systems. Integrating phishing detection with Network Intrusion Detection Systems (NIDS) and endpoint protection solutions could create a multi-layered defense strategy. This integration would allow systems to dynamically update NIDS to block malicious traffic and improve endpoint protection mechanisms. Another promising direction is the incorporation of real-time threat intelligence sharing mechanisms, which would improve the system's responsiveness to emerging phishing tactics by ensuring all components of the cybersecurity framework are updated with the latest threat information. Future research should also consider the impact of adversarial machine learning attacks on phishing detection systems, including those using HEFS. Understanding potential vulnerabilities could lead to the development of more robust algorithms capable of maintaining efficacy under adversarial conditions.

Future work will focus on fine-tuning and enhancing the phishing prevention browser extension for broader use and integrating additional functionalities to improve its effectiveness in protecting users from phishing threats. Besides deep learning (DL) optimization, there is a need to improve the accuracy of phishing detection models by examining different model structures, fine-tuning hyperparameters, and utilizing larger datasets for training and evaluation. Efforts will also aim to expand the automated script's capabilities for phishing detection by incorporating additional features such as real-time URL analysis and behavior-based detection mechanisms. Furthermore, the practical deployment of the phishing detection framework in real-world scenarios, such as web browsers, email clients, and network security systems, will be explored to ensure consistent protection against phishing threats by integrating it into existing cybersecurity infrastructure.

REFERENCES

[1] F. Salahdine, Z. El Mrabet, and N. Kaabouch, "Phishing Attacks Detection: A Machine Learning-Based Approach," in 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2021, pp. 250–255.

[2] M. Baykara and Z. Z. Gürel, "Detection of phishing attacks," in 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey, 2018, pp. 1–5.

[3] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," Expert Systems with Applications, vol. 106, pp. 1–20, 2018.

[4] F. Yahya et al., "Detection of Phishing Websites using Machine Learning Approaches," in 2021 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2021, pp. 40–47.

[5] A. Alswailem, B. Alabdullah, N. Alrumayh, and A. Alsedrani, "Detecting Phishing Websites Using Machine Learning," in 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1–6.

[6] H. Zuhair, A. Selamat, and M. Salleh, "Feature selection for phishing detection: A review of research," International Journal of Intelligent Systems Technologies and Applications, vol. 15, pp. 147, 2016.

[7] Phishing Websites Dataset, Version 1, Mendeley Data, Sep. 2020. [Online]. Available: https://data.mendeley.com/datasets/72ptz43s9v/1. DOI: 10.17632/72ptz43s9v.1

[8] Web Page Phishing Detection Dataset, Version 3, Mendeley Data, Jun. 2021. [Online]. Available: https://data.mendeley.com/datasets/c2gw7fy2j4/3. DOI: 10.17632/c2gw7fy2j4.3

[9] P. Chinnasamy et al., "An Efficient Phishing Attack Detection using Machine Learning Algorithms," in 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2022, pp. 1–6.

[10] T. R. N and R. Gupta, "Feature Selection Techniques and its Importance in Machine Learning: A Survey," in 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2020, pp. 1–6.

[11] R. Ramachandran, G. Ravichandran, and A. Raveendran, "Evaluation of Dimensionality Reduction Techniques for Big data," in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 226–231.

[12] S. Dangwal and A.-N. Moldovan, "Feature Selection for Machine Learning-based Phishing Websites Detection," in 2021 International Conference on Cyber Situational Awareness, Data Analytics, and Assessment (CyberSA), Dublin, Ireland, 2021, pp. 1–6.

[13] Y. Wei and Y. Sekiya, "Feature Selection Approach for Phishing Detection Based on Machine Learning," in Proceedings of the International Conference on Applied CyberSecurity (ACS) 2021, H. Ragab Hassen and H. Batatia, Eds., Cham, Springer International Publishing, 2022, pp. 61–70.

[14] M. R. Chinguwo and R. Dhanalakshmi, "Detecting Cloud Based Phishing Attacks Using Stacking Ensemble Machine Learning Technique," International Journal For Science Technology And Engineering, vol. 11, no. 3, pp. 360-367, 2023. DOI: 10.22214/ijraset.2023.49422

[15] C. Rajeswary and M. Thirumaran, "The LSTM-based automated phishing detection driven model for detecting multiple attacks on Tor hidden services," J. Intell. Fuzzy Syst., vol. 44, no. 6, pp. 8889–8903, Jan. 2023. DOI: 10.3233/JIFS-200912

[16] B. Subba, "A heterogeneous stacking ensemble-based security framework for detecting phishing attacks," in 2023 National Conference on Communications (NCC), 2023, pp. 1-6. DOI: 10.1109/NCC56989.2023.10068026

[17] K. R. Nataraj, D. K. Yashaswini, R. Hema, N. S. Pawar, and S. Yashaswi, "Phishing Attack Detection Using Machine Learning," in Lecture Notes in Electrical Engineering: Proceedings of the 4th International Conference on Data Science, Machine Learning and Applications, Springer Nature Singapore, 2023, pp. 355-370. DOI: 10.1007/978-981-99-2058-7_33

[18] D. T. Mosa et al., "Machine Learning Techniques for Detecting Phishing URL Attacks," Computers, Materials & Continua, vol. 75, no. 1, pp. 1271-1290, 2023. DOI: 10.32604/cmc.2023.036422

[19] S. Yu et al., "Phishing Detection Based on Multi-Feature Neural Network," in 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), 2022, pp. 73-79. DOI: 10.1109/IPCCC55026.2022.9894337

[20] J. Novakovic and S. Marković, "Detection of URL-based Phishing Attacks Using Neural Networks," in 2022 International Conference on Theoretical and Applied Computer Science and Engineering (ICTASCE), 2022, pp. 132-136.

[21] S. A. Salihu et al., "Detection of Phishing URLs Using Heuristics-Based Approach," in 2022 5th Information Technology for Education and Development (ITED), 2022, pp. 1-7. DOI: 10.1109/ITED56637.2022.10051199

[22] J. Tanimu and S. Shiaeles, "Phishing Detection Using Machine Learning Algorithm," in 2022 IEEE International Conference on Cyber Security and Resilience (CSR), 2022, pp. 317-322. DOI: 10.1109/CSR54599.2022.9850316

[23] M. Sánchez-Paniagua et al., "Phishing websites detection using a novel multipurpose dataset and web technologies features," Expert Systems with Applications, vol. 207, p. 118010, 2022. DOI: 10.1016/j.eswa.2022.118010

[24] M. Almseidin, A. Abu Zuraiq, M. Al-kasassbeh, and N. Alnidami, "Phishing Detection Based on Machine Learning and Feature Selection Methods," International Association of Online Engi-

neering, Dec. 2019. Accessed on: Jun. 8, 2024. [Online]. Available: https://www.learntechlib.org/p/216410

[25] R. Jayaraj, A. Pushpalatha, K. Sangeetha, T. Kamaleshwar, S. Udhaya Shree, and D. Damodaran, "Intrusion detection based on phishing detection with machine learning," Measurement: Sensors, vol. 31, p. 101003, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917423003392.

[26] V. Borisov, A. Leethorp, A. Sepliarskaia, A. L. Molchanov, and M. L. Prokhorenkova, "Deep neural networks and tabular data: A survey," arXiv preprint arXiv:2110.01889, 2021. [Online]. Available: https://arxiv.org/abs/2110.01889.

[27] H. Ye, J. Fan, Q. Zhang, and J. Tang, "A closer look at deep learning on tabular data," arXiv preprint arXiv:2407.00956, 2024. [Online]. Available: https://arxiv.org/abs/2407.00956.

[28] L. Cheng, Y. Liu, and J. Lin, "SHAPNN: Shapley value regularized tabular neural network," arXiv preprint arXiv:2309.08799, 2023. [Online]. Available: https://arxiv.org/abs/2309.08799.

[29] A. Bondarenko, "More layers! End-to-end regression and uncertainty on tabular data with deep learning," arXiv preprint arXiv:2112.03566, 2021. [Online]. Available: https://arxiv.org/abs/2112.03566.

[30] R. D. Corin, "Architecture of the wide and deep neural network," Accessed on 2024-05-09. [Online]. Available: https://www.researchgate.net/figure/Architecture-of-the-wide-and-deep-neural-network-The-prediction-of-the-sensors-states-is_fig1_337945667

[31] Creative Commons Attribution 4.0 International, "TabNet architecture consisting of the encoder for classification," Accessed on 2024-05-09. [Online]. Available: https://www.researchgate.net/figure/TabNet-architecture-consisting-of-the-encoder-for-classification-This-is-composed-of_fig1_371301234

[32] Creative Commons Attribution-NonCommercial 4.0 International, "Deep Neural Network (DNN) model architecture," Accessed on 2024-05-09. [Online]. Available: https://www.researchgate.net/figure/Deep-Neural-Network-DNN-model-architecture_fig1_370641687

[33] learnopencv, "An example of FNN with one hidden layer," Accessed on 2024-05-09. [Online]. Available: https://learnopencv.com/understanding-feedforward-neural-networks/

[34] A. Yadav, "What is Web Application," Apr. 22, 2023. Accessed on 2024-06-08. [Online]. Available: https://medium.com/@akashyadav1452/what-is-web-application-5f912f18f50f

GANESH S NAYAK has completed his B.Tech in Information Technology from Manipal Institute of Technology, Udupi, Karnataka, India, and is currently working as a Software Development Engineer (SDE) at Boeing India Pvt Ltd, Bangalore. He has previously interned at Boeing India and Telenetix Pvt Ltd, Udupi, gaining experience in software engineering and Python development. Additionally, he contributed to the PRISM project at Samsung Research India for six months, focusing on advanced research and development. His research interests include machine learning, cybersecurity, and software development.

DR. BALACHANDRA MUNIYAL received his B.E degree in Computer Science and Engineering from Mysore University and M.Tech and Ph.D in Computer Science and Engineering from Manipal Academy of Higher Education, Manipal, India. His research area is Cyber Security. He has more than 80 publications in national and international conferences/journals. Currently he is working as a Professor in the Dept. of Information & Communication Technology, Manipal Institute of Technology, Manipal. He is also coordinating Centre of Excellence for Cybersecurity, MAHE, Manipal. He was Head of the Department from 2017 to 2020. He has 30 years of teaching experience in various Institutes. Under his supervision five research students have completed their PhD and currently he is guiding eight students.

DR. MANJULA C BELAVAGI received the B.E. degree in computer science and engineering from Karnatak University, Dharwad, India, the master's degree in network and internet engineering from JNNCE, Shivamogga, VTU, Belgaum, India, and the Ph.D. degree from the Manipal Academy of Higher Education, Manipal, India. She is currently working as an Assistant Professor-Selection Grade with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal. She has published research papers in national and international conference proceedings and journals. Her research interests include machine learning, game theory, and wireless sensor networks security.