A Comparative Analysis of Machine Learning Techniques for URL Phishing Detection

Adel Ataih Albishri
Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Saudi Arabia
adelatyh@gmail.com

Mohamed M. Dessouky
Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Saudi Arabia | Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Egypt
mmdessouky@uj.edu.sa (corresponding author)

Received: 5 September 2024 | Revised: 9 October 2024 and 14 October 2024 | Accepted: 16 October 2024

 DOI: https://doi.org/10.48084/etasr.8920

ABSTRACT

The growing threat of URL phishing attacks raises the need for advanced detection systems to protect digital environments. This paper explores the effectiveness of various machine learning models in classifying URLs as phishing or benign, focusing on the random forest model. Using ensemble learning, the random forest demonstrated superior accuracy and reliability compared to traditional methods, achieving consistent performance with accuracy rates between 99.93% and 99.98%. The model's performance was evaluated daily over eight days, highlighting its robustness in handling real-world scenarios. This study utilized GridSearchCV to optimize model hyperparameters, enhancing model robustness and minimizing overfitting. Future research directions include advanced feature engineering, deep learning techniques, and multimodal data integration to further improve phishing detection systems.

Keywords-phishing attacks; phishing detection; ensemble learning; random forest

I. INTRODUCTION

With the rapid expansion of digital communication, the prevalence of cyber threats has surged, posing significant risks to individuals, organizations, and society. Among these threats, URL phishing is a formidable adversary that exploits human vulnerability and technological loopholes to mislead users into disclosing sensitive details such as passwords, financial data, and personal information. As cyber criminals continually refine their tactics and techniques, the sophistication and frequency of URL phishing attacks continue to escalate, threatening the integrity and security of digital ecosystems worldwide. Consequently, the need for robust and effective URL phishing detection mechanisms has become paramount to protecting against malicious cyber activities. Without adequate defenses, individuals and organizations remain vulnerable to exploitation, risking financial loss, reputational damage, and compromised privacy. Therefore, proactive measures must be taken to develop and deploy advanced detection solutions capable of thwarting evolving phishing threats and preserving the trust and safety of online interactions.

URL phishing involves the creation of deceptive web links that mimic legitimate URLs to trick users into believing that they are accessing a trusted website or service. These fraudulent URLs are often distributed through emails, social media messages, or other online platforms, accompanied by convincing pretexts designed to lure victims to click on malicious links [1]. Once clicked, unsuspecting users may inadvertently disclose confidential information or download malware onto their devices, compromising their security and privacy. The sophistication of URL phishing attacks continues to evolve, with perpetrators employing increasingly sophisticated tactics to avoid detection and exploit vulnerabilities in users browsing habits and online behaviors [1].

Recent phishing techniques include email phishing, spear phishing, whaling, and smishing. Email phishing involves creating and sending malicious emails intended to impersonate legitimate sources, deceiving the victims into providing sensitive financial information used for subsequent attacks [2]. Spear phishing adopts a targeted approach, focusing on specific groups or individuals within an organization [3]. Attackers in this category are disguised as colleagues, making it challenging to identify the perpetrators effectively. Whaling also offers a targeted approach, focusing on senior employees within an organization [4]. These targeted attacks are intended to obtain high-valued corporate information, leading to massive financial implications for the victims. Smishing leverages Short Messaging Services (SMS) to mislead the victims into

providing sensitive information [5]. In this attack, the attacker masquerades as a legitimate source, misleading the victim into giving the required information. These attacks are illustrated in Figure 1.

Google Safe Browsing is a pioneering web security initiative that offers a comprehensive framework to protect users from malicious online content. Launched by Google in 2007, this service utilizes a constantly updated URL blocker to host phishing, malware, and other malicious content [6]. This service operates by continuously scanning billions of URLs across the Internet, promptly identifying malicious content, and issuing warnings to users before interacting with potentially harmful websites. By integrating Safe Browsing into web browsers and other online platforms, users receive real-time warnings and alerts when accessing potentially dangerous websites [6]. In addition, Safe Browsing employs advanced algorithms to identify and flag suspicious URLs, leveraging data from user reports, automated crawling, and machine learning techniques. Its seamless integration with popular web browsers such as Google Chrome, Mozilla Firefox, and Safari ensures widespread adoption and adequate protection for millions of users around the world [6]. Additionally, Safe Browsing offers an API for developers to incorporate its security features into their applications, further extending its reach and impact across the digital landscape.

Several factors come into play when comparing Google Safe Browsing with similar security frameworks. One notable advantage is its extensive database of known malicious URLs, constantly updated in real-time, providing users with up-to-date protection against emerging threats [1]. Its seamless integration with popular web browsers ensures wide coverage and ease of use for the end user. However, Safe Browsing has faced criticism for its reliance on blocklisting, which may lead to false positives and inadvertent blocking of legitimate websites. Furthermore, while Safe Browsing excels in detecting known threats, its effectiveness against zero-day attacks and sophisticated phishing attempts may be limited [1]. In contrast, some alternative frameworks, such as Microsoft's SmartScreen Filter, employ a combination of blocklisting and heuristic analysis to provide more comprehensive protection.

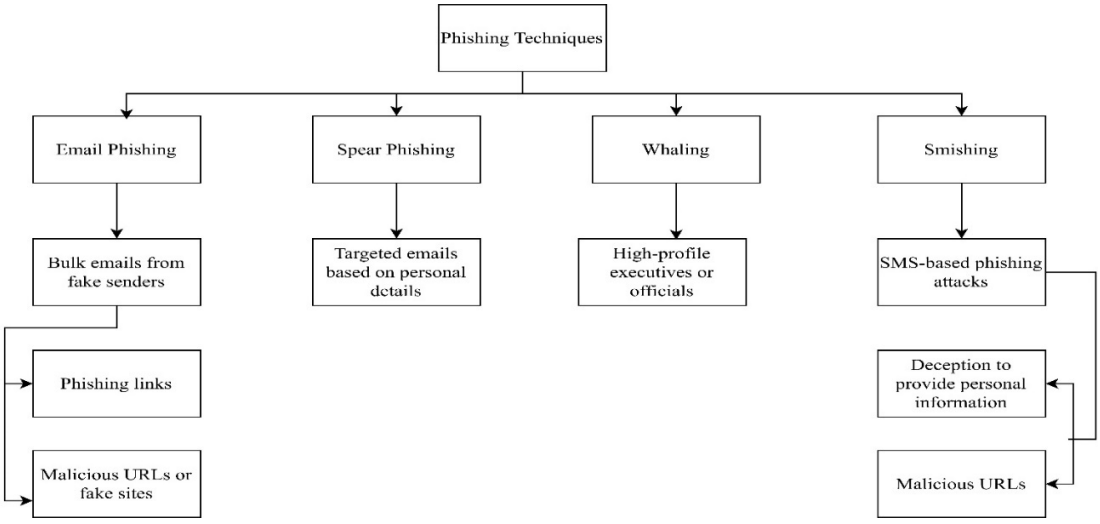

Fig. 1. Overview of phishing attacks.

This study sought to develop a novel advanced phishing detection engine using the strengths of supervised machine learning approaches. It makes two significant novel contributions to handling phishing attacks. First, it introduces a robust hyperparameter tuning process using GridSearchCV. This optimization technique systematically explores the parameter space for each model, ensuring that they are fine-tuned for optimal performance. This approach enhances accuracy and minimizes overfitting, allowing the models to better generalize to unseen data. The successful application of this tuning method significantly contributes to the field by improving detection capabilities. In addition, the study incorporates an advanced feature selection strategy combining traditional characteristics, URL length, and HTTPS presence with more sophisticated indicators such as WHOIS information and JavaScript code analysis. This comprehensive feature set enhances the model's ability to differentiate between phishing and benign URLs, improving detection accuracy. The emphasis on feature engineering underscores its critical role in building effective phishing detection systems.

II. METHODOLOGY

The experimental environment and machine learning algorithms were meticulously selected and configured to ensure robustness and reproducibility. Modeling was carried out using Python, specifically utilizing the Scikit-learn library for various machine learning algorithms, including Random Forest (RF), Decision Trees (DT), and Support Vector Machines (SVM). Each model was systematically trained and tested using the training dataset, allowing a thorough evaluation of its performance metrics. The experiments were carried out on a 16 GB RAM computer to ensure computational efficiency and resource availability. Python's versatility allowed for the seamless integration of various algorithms and data preprocessing techniques, facilitating the exploration of different modeling approaches. Using Scikit-learn's extensive

functionalities streamlined the implementation and evaluation processes, enabling efficient experimentation and comparison of multiple algorithms.

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classification model at various classification thresholds. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across different threshold values. The TPR, also known as sensitivity or recall, measures the proportion of actual positive cases the model correctly identifies. On the other hand, the FPR measures the proportion of actual negative cases incorrectly classified as positive. The ROC curve provides valuable insights into the model's trade-off between sensitivity and specificity across different threshold settings. A perfect classifier would have an ROC curve that passes through the top-left corner of the plot (TPR=1, FPR=0), indicating high sensitivity (no false negatives) and high specificity (no false positives). On the other hand, a random classifier would produce an ROC curve that is a diagonal line from the bottom-left to the top-right corner.

Integration of the ROC with the confusion matrix graph offers a comprehensive understanding of the model performance. While the ROC curve helps visualize the trade-off between sensitivity and specificity across different threshold values, the confusion matrix provides detailed information on the distribution of correct and incorrect predictions. Collectively, they offer insights into the strengths and weaknesses of the model and help identify areas for improvement. Furthermore, the AUC-ROC value is a quantitative metric to assess the overall performance of a model. This holistic approach to evaluation allows researchers and practitioners to gain a nuanced understanding of the model's behavior and effectiveness in classifying data. Thus, this robust visualization of the ROC and confusion matrix enhances knowledge of the model's performance, effectively determining its suitability for phishing detection in production networks.

The models that displayed exceptional performance were carefully selected for further refinement through hyperparameter tuning and feature selection. This critical step was intended to optimize their predictive capabilities and enhance their effectiveness. This process is illustrated in the diagram shown in Figure 2.

A. Evaluation Metrics

The evaluation metrics were calculated using standard formulas and techniques to assess the rigorous performance of the machine learning models. Accuracy was calculated as the ratio of accurately grouped instances to the total cases in the dataset. Precision was computed by dividing the number of true positive predictions by the total predicted positives. At the same time, recall was determined by dividing the number of true positive predictions by the total actual positives. The F1 score was then derived as the harmonic mean of precision and recall. These metrics provided a comprehensive overview of the models' performance across different aspects, including their ability to correctly classify instances and avoid false positives and negatives. By employing these metrics, researchers could gain insight into each model's strengths and limitations and make informed decisions regarding model selection and refinement. Table I shows the formulas for these metrics.

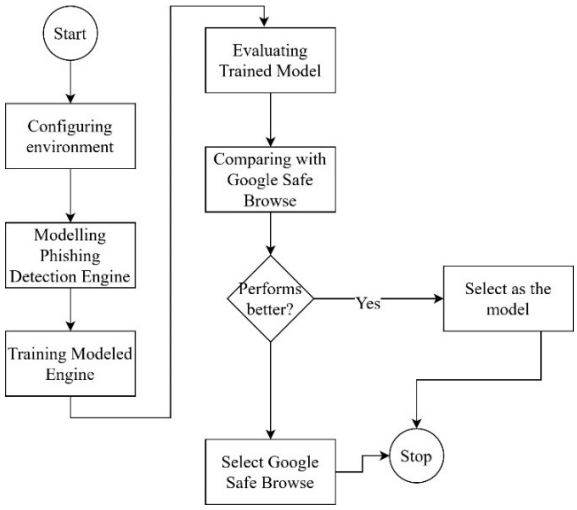

Fig. 2. Method for developing a phishing detection engine.

TABLE I. EVALUATION METRICS

Metric	Formula
Accuracy	(TP + TN)/(TP + TN + FP + FN)
Precision	(TP)/(TP + FP)
Recall	(TP)/ (TP + FN)
F1	2TP/ (2TP + FP + FN)

B. Dataset and Selection

The dataset used for phishing detection consists of 1,561,932 entries, which includes a balanced distribution of phishing and benign URLs [7]. This dataset is publicly available and serves as a comprehensive resource for researchers and practitioners in cybersecurity. The dataset encompasses various features extracted from URLs, including URL length, presence of IP addresses, geographic location of the server, and Top-Level Domains (TLDs) [7]. This extensive collection of labeled data enables practical training and evaluation of machine learning models aimed at detecting phishing attempts. Twelve thousand samples were selected from this dataset through random sampling. This strategy provided a relatively large dataset representation while considering the computing resources needed to effectively train the model.

The dataset's distribution of phishing and benign cases is well balanced, ensuring that the machine learning models trained on these data can effectively differentiate between legitimate and malicious URLs. This balance is crucial to preventing biases and ensuring that the detection algorithms generalize well across various scenarios and domains. The dataset includes real-world examples of phishing websites sourced from diverse geographic locations and TLDs. By incorporating authentic instances of phishing attacks, the dataset provides a realistic representation of the types of threats

encountered in the wild. This feature allows for more accurate and reliable training of phishing detection systems, as they can learn to recognize patterns and characteristics commonly associated with malicious URLs. Ten features were selected from the dataset, with each feature presented in its column. The first feature is the website's Uniform Resource Locator (URL), serving as its unique online address. These features are summarized in Table II.

TABLE II. SUMMARY OF THE FEATURES USED IN THE DATASET

Component	Description
URL	The URL of the website, indicating its web address
url_len	The length of the URL in characters
ip_add	The IP address of the website's server
geo_loc	The geographical location of the website server
TLD	The website's top-level domain, such as .com, .org, .net, etc.
who_is	WHOIS information about the website, which provides details about the domain registration
HTTPS	A binary indicator (yes/no) indicating whether the website uses HTTPS protocol
js_len	The length of JavaScript code present on the webpage
js_obf_len	The length of obfuscated JavaScript code on the webpage, if any
Label	Indicates the website's classification, such as "good" or "bad."

Domain variation techniques are fundamental strategies employed by hackers to craft deceptive domain names that closely resemble legitimate ones. These techniques introduce subtle alterations to the original domain, making it difficult for users to distinguish between genuine and malicious domains. The first method is addition, where extra characters are appended to the original domain. For instance, transforming "sample.com" into "samplea.com" introduces a minimal change that may go unnoticed by unsuspecting users. The second method is bitsquatting, which involves flipping a single character in the domain name. This strategy results in domains such as "scmple.com," which can easily be mistaken for the original. These alterations exploit human error and cognitive biases, which make it imperative for cybersecurity professionals to remain vigilant against such tactics. The third approach is homoglyph substitution, where visually similar characters replace legitimate ones. For example, substituting the letter 'a' with its visually similar counterpart 'ã' in "sãmple.com" creates a domain that closely resembles the original.

Hyphenation introduces hyphens into the domain name, as seen in "sam-ple.com," adding a layer of deception. These alterations capitalize on users' tendency to overlook subtle differences, increasing the effectiveness of phishing and other malicious activities. Other techniques, such as subdomain manipulation, involve modifying or adding subdomains to the original domain. For instance, "sa.mple.com" adds the subdomain 'sa' to create a deceptive variant. The inclusion of a TLD alters the domain name's TLD, as demonstrated by "samplecom.com," where the TLD 'com' is included in the domain itself. These variations exploit users' familiarity with domain structures, leveraging their trust in recognizable domain patterns to facilitate cyberattacks. Also, domain variation techniques, such as transposition and vowel swap, introduce subtle changes by swapping characters or replacing vowels. For instance, "sapmle.com" transposes the positions of 'p' and 'm,' while "semple.com" swaps 'a with 'e.' These alterations aim to evade detection by relying on users' limited attention spans and cognitive processing, further emphasizing the importance of user education and awareness in mitigating cyber threats.

Statistical analysis of the overall dataset reveals critical insights into URL length across benign and phishing categories. The mean URL length is 35.80, with benign entries averaging slightly higher at 35.81 than phishing entries at 35.63. Both categories share a median URL length of 32.00, indicating symmetry in distribution. At the 90th percentile, the overall value is 55.00, consistent with benign entries, while phishing entries have a lower value of 51.50. The 95th percentile is also higher in the overall dataset at 63.00, compared to 58.50 for phishing. Finally, the 99th percentile shows a significant difference, with an overall value of 86.00, aligning with benign entries (86.48) and surpassing the phishing category (72.00). This analysis highlights the distinctions in URL lengths, illustrating the higher values associated with benign entries. Examining the percentiles provides further insight into the spread and distribution of the data. In the "Benign" category, the values at these percentiles are consistently higher than their counterparts in the "Phishing" category. For instance, at the 90th percentile, the "Benign" category has a value of 55.00 compared to 51.50 in the "Phishing" category. This phenomenon suggests that more data points in the "Benign" category are higher than those in the "Phishing" category. Similarly, the 95th percentile corroborates this trend, with both categories presenting values of 63.00 and 58.50, respectively, further affirming the broader data spread within "Benign." Furthermore, the 99th percentile, representing extreme values, highlights notable differences between the categories. The "Benign" category exhibits a significantly higher value of 86.00 compared to 72.00 in the "Phishing" category. This feature indicates that extreme values are more prevalent in the "Benign" category, potentially indicating more significant variability or outliers within this category. These results are summarized in Table III.

TABLE III. STATISTICAL ANALYSIS OF THE URL LENGTH

	Overall	Benign	Phishing
Mean	35.80	35.81	35.63
Median	32.00	32.00	32.00
90^{th} percentile	55.00	55.00	51.50
95^{th} percentile	63.00	63.00	58.50
99^{th} percentile	86.00	86.48	72.00

III. MODEL SELECTION AND TUNING

Eight classification algorithms were considered for training, namely the Naive Bayes (NB), Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Multi-Layer Perceptron's (MLP), Gradient Boosting, and Linear Regression (LR). Each model started with default configurations, allowing for a standardized baseline performance assessment. The NB model utilized the MultinomialNB() classifier, focusing on probabilistic

classification based on Bayes' Theorem. DT adopts a tree-based structure that improves interpretability, while the RF model combines multiple DTs to mitigate overfitting. Gradient Boosting (GB) operates iteratively, refining the model to capture complex patterns indicative of phishing attacks. SVM employed the Support Vector Classifier (SVC), constructing hyperplanes to classify data points with critical parameters, including the regularization parameter *C* and the kernel type. KNN classified websites based on the majority class of nearest neighbors, using *k*=5. MLP utilized a neural network architecture to capture nonlinear relationships in the data, with default parameters for hidden layers, activation functions, and regularization. LR served as a simpler baseline model, predicting the likelihood of phishing based on linear combinations of features. The dataset was divided into 80% for training and 20% for testing for all models. This approach ensured that there were 9,600 training samples and 2,400 testing samples. This segmentation ensures that most of the data was used to train the models while leaving a portion for evaluating their performance on unseen examples. The 80:20 ratio facilitated a robust assessment of each model's generalization capabilities, providing insights into their efficiency in detecting phishing websites based on the selected features. Tables IV and V summarize the training and testing results, illustrating the performance metrics of each model. Figures 3 and 4 show the test results for the RF model.

TABLE IV. ACCURACY, PRECISION, AND RECALL FOR MODEL TRAINING

Classifier	Accuracy	Precision	Recall	F1 score
NB	75.21%	99.94%	74.73%	85.52
DT	100.00%	100.00%	100.00%	100.00
RF	100.00%	100.00%	100.00%	100.00
GB	100.00%	100.00%	100.00%	100.00
SVM	98.45%	98.44%	100.00%	99.21
KNN	98.97%	98.98%	99.98%	99.48
MLP	99.51%	99.90%	99.60%	99.75
LR	99.82%	99.82%	100.00%	99.91

TABLE V. CLASSIFICATION RESULTS FOR MODEL TESTING

Classifier	Accuracy	Precision	Recall	F1 score
NB	76.63%	100.00%	76.14%	86.45%
DT	99.96%	100.00%	99.96%	99.98%
RF	99.92%	99.92%	100.00%	99.96%
GB	99.96%	100.00%	99.96%	99.98%
SVM	98.46%	98.45%	100.00%	99.22%
KNN	98.54%	98.53%	100.00%	99.26%
MLP	99.46%	99.79%	99.66%	99.72%
LR	99.79%	99.79%	100.00%	99.89%

NB achieved an accuracy of 76.63%, indicating that it correctly classified about three-quarters of the instances in the dataset. Its precision of 100% signifies that when it predicts a positive class, it is always correct, which is an impressive feature. However, its recall of 76.14% suggests that it misses identifying a significant portion of positive cases. The F1 score is 86.45%, indicating a reasonably balanced performance regarding false positives and negatives. Despite its relatively lower accuracy than other classifiers, the NB classifier's perfect precision makes it a suitable choice for tasks where minimizing false positives is critical. DT demonstrated exceptional performance across all metrics, with an accuracy of 99.96%, a precision of 100%, a recall of 99.96%, and an F1 score of 99.98%. These results show that it accurately identified almost all instances in the dataset without false positives or negatives. Its ability to learn complex decision boundaries and its interpretability make it a valuable tool for classification tasks, especially in scenarios where understanding the decision-making process is as vital as predictive accuracy. Additionally, its robustness to outliers and missing values further enhances its applicability across various domains.

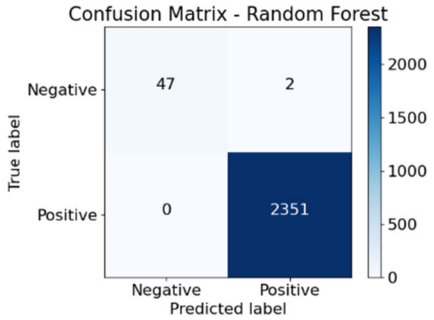

Fig. 3. Confusion matrix for Random Forrest (RF).

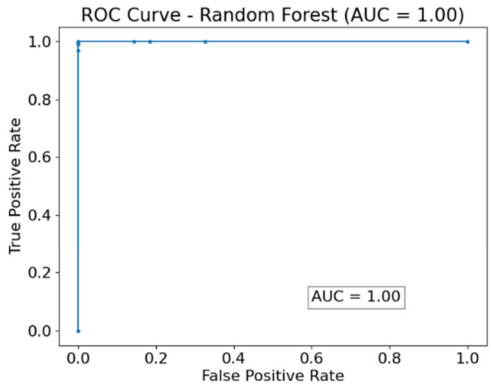

Fig. 4. ROC for Random Forrest (RF).

RF and GB exhibited outstanding performance, close to DT, with near-perfect scores across all metrics. These ensemble learning methods take advantage of the strengths of multiple DTs to make predictions, resulting in robust and reliable classifiers. By aggregating the predictions of individual trees, both classifiers reduce overfitting and improve generalization to unseen data, thereby enhancing predictive accuracy. Furthermore, their ability to capture complex interactions between features and handle high-dimensional data makes them well-suited for challenging classification tasks. SVM achieved strong performance on most metrics, with an accuracy of 98.46%, precision of 98.45%, recall of 100%, and F1 score of 99.22%. These results show that SVM effectively distinguished different classes in the feature space, maximizing the margin between classes, and promoting generalization to unseen data. SVM's ability to handle high-dimensional data and nonlinear decision boundaries makes it suitable for various

classification tasks, particularly in scenarios with well-separated classes. Additionally, its robustness to overfitting and ability to handle binary and multiclass classification further enhance its versatility and applicability. KNN achieved high accuracy (98.54%), precision (98.53%), recall (100%), and F1 score (99.26%), showing its effectiveness in capturing the underlying patterns in the dataset and making accurate predictions. KNN's simplicity and ease of implementation make it an attractive choice for classification tasks, especially in scenarios where the decision boundary is irregular or complex to define analytically. However, KNN's performance heavily relies on the choice of the number of neighbors (k) and the distance metric used, necessitating careful parameter tuning for optimal results. However, its ability to handle binary and multiclass classification problems and its resilience to noisy data make KNN a versatile and valuable tool in machine learning.

The MLP and LR classifiers demonstrated strong performance across all metrics, with high accuracy, precision, recall, and F1 score. MLP, with an accuracy of 99.46% and an F1 score of 99.72%, showcases its ability to capture complex nonlinear relationships in the data and make accurate predictions. Its adaptability and learning ability make it well-suited for classification tasks involving intricate patterns and structures. On the other hand, LR, with an accuracy of 99.79% and an F1 score of 99.89%, provides a baseline performance against which more complex classifiers can be compared. Despite its simplicity, LR effectively captures linear relationships between features and target variables, making it useful for tasks where interpretability and computational efficiency are priorities. MLP and LR classifiers offer reliable and interpretable solutions for classification tasks across various domains.

Based on the results above, it is evident that three models excelled above others: DT, RF, and GB. These models were selected for optimization in the testing dataset. The selected model was improved by leveraging GridSearchCV, a technique that systematically searches a predefined parameter grid to identify the optimal combination of hyperparameters for the model. GridSearchCV facilitates fine-tuning the model's hyperparameters, optimizing its performance, and enhancing its accuracy and robustness. GridSearchCV identifies the configuration that maximizes the chosen performance metric by exhaustively evaluating different hyperparameter combinations through cross-validation. This process ensures that the model is fine-tuned to achieve the best possible performance on unseen data, effectively mitigating the risk of overfitting. This evaluation process provides valuable insights into how well the models generalize to unseen data and their ability to classify URLs into phishing and benign categories accurately.

The optimized classifiers demonstrated high performance with minimal variations. High accuracy scores, equal to 99.98%, indicate that the classifiers successfully and correctly classify most of the instances in the dataset. This phenomenon implies a robust ability to distinguish between positive and negative cases, showcasing their effectiveness in making accurate predictions. Furthermore, precision, which represents the ratio of correctly predicted positive observations to the total predicted positive observations, was consistently high, with values of 100.00% for DT and GB and 99.96% for RF. This indicates a low rate of false positives, demonstrating the classifiers' precision in identifying positive instances.

Furthermore, the recall scores, reflecting the ability of the classifiers to correctly identify positive instances out of all actual positive instances, were also high across the board. While DT and GB achieved recall scores of 99.96%, RF achieved a perfect score of 100.00%. This feature implies that the classifiers exhibit high sensitivity in detecting positive instances, minimizing the number of false negatives. Additionally, the F1 score, the harmonic mean of precision and recall, is a comprehensive measure of a classifier's performance, balancing precision and sensitivity. All three classifiers achieved F1 scores of 99.98%, highlighting their robust performance in achieving a balance between precision and recall. These results are shown in Table VI.

TABLE VI. OPTIMIZED MODEL RESULTS

Metric	DT	RF	GB
Accuracy	99.96	99.96	99.96
Precision	100.00	99.96	100.00
Recall	99.96	100.00	99.96
F1 Score	99.98	99.98	99.98

A comparative assessment of the F1 scores of DT, RF, and GB before and after optimization shows the implications of model optimization on its overall performance. All classifiers demonstrated exceptionally high F1 scores before and after optimization, indicating their effectiveness in achieving a balance between precision and recall. The DT and GB classifiers maintain their F1 scores at 99.98% after optimization, with delta values of 0.00, signifying that the optimization process did not affect their performance. Conversely, after optimization, the RF classifier slightly improved its F1 score from 99.96% to 99.98%, resulting in a delta of 0.02. This suggests that the optimization process led to a marginal enhancement in its ability to classify instances accurately. These results are shown in Table VII.

TABLE VII. PERFORMANCE COMPARISON OF OPTIMIZED MODELS

Classifier	F1 score	F1 score (Optimized)	Delta
DT	99.98	99.98	0.00
RF	99.96	99.98	0.02
GB	99.98	99.98	0.00

IV. DISCUSSION AND ANALYSIS

RF was selected for classification tasks in the URL Phishing model, as it emerged as the most effective model for classifying URLs into phishing and benign after optimization. Leveraging the power of ensemble learning, RF aggregates predictions from multiple decision trees, resulting in robust and reliable classification models. Its ability to handle high-dimensional data, mitigate overfitting, and capture complex feature interactions makes it well-suited for diverse classification tasks. RF consistently outperformed other

algorithms, demonstrating its superiority in accurately classifying URLs.

The models were optimized using GridSearchCV, which facilitates fine-tuning their hyperparameters, optimizing their performance, and enhancing their accuracy and robustness. GridSearchCV identifies the configuration that maximizes the chosen performance metric by exhaustively evaluating different hyperparameter combinations through cross-validation. This process ensures that the model is fine-tuned to achieve the best possible performance on unseen data, effectively mitigating the risk of overfitting. The evaluation metrics for these models were calculated based on the predictions made by the models on the label column that contained the ground truth labels indicating whether each URL was phishing or benign. This evaluation process provides valuable insights into how well the models generalize to unseen data and their ability to classify URLs into phishing and benign categories accurately.

The proposed model was evaluated by classifying phishing URLs daily for eight consecutive days to assess its stability and reliability over time, ensuring its effectiveness in real-world scenarios. By continually monitoring the model's performance and comparing it against ground truth labels, the study gained insights into its ability to accurately classify URLs into phishing and benign categories under varying conditions. This daily evaluation approach comprehensively evaluates the model's performance and suitability for practical deployment in cybersecurity applications. In this evaluation, the average accuracy remained consistently high, ranging between 99.93% and 99.97%, indicating its robustness in accurately classifying URLs. The stability and reliability of the model's performance over time highlight its potential for real-world deployment in cybersecurity applications. Despite minor fluctuations in accuracy scores, the developed model consistently maintained a high level of accuracy throughout the observation period. This stability is essential for ensuring the reliability of URL classification systems in practical scenarios, where consistent and accurate detection of malicious URLs is critical to protecting users from cyber threats. The findings suggest that the proposed model is promising for enhancing URL classification accuracy and improving overall cybersecurity measures.

V. CONCLUSION

This study addressed a significant gap in URL phishing detection by developing a machine learning-based detection engine that enhances accuracy and real-time detection. Traditional phishing detection methods struggle with zero-day attacks. The proposed model achieved up to 99.97% accuracy using RF and GB classifiers. This improvement is attributed to the use of GridSearchCV for hyperparameter tuning and an advanced feature selection process that incorporates both standard and novel URL features. This study introduces enhanced feature engineering and optimization techniques, addressing the limitations of existing machine learning models for phishing detection. Integrating traditional machine learning techniques with optimization strategies makes the phishing detection model more adaptable to real-world threats.

Future research in URL phishing detection and classification is needed to explore potential approaches to enhance the effectiveness and robustness of detection systems. An area of interest could be the development of more advanced feature engineering techniques explicitly tailored for URL attributes. This involves identifying and extracting new features from URLs that can provide valuable insights into their phishing potential, such as linguistic patterns, syntactic structures, or semantic meanings embedded in URL strings. Additionally, investigating the integration of advanced machine learning techniques, such as deep learning architectures, such as convolutional or recurrent neural networks, could lead to the development of more sophisticated and adaptive models capable of capturing complex patterns and relationships in URL data. Research focusing on integrating multimodal data sources, such as textual content, webpage structure, and network traffic patterns, could offer a holistic approach to phishing detection by leveraging diverse data sources to improve model performance. Similarly, exploring adversarial machine learning techniques to enhance model robustness against adversarial attacks and evasion strategies employed by sophisticated phishing campaigns could be another promising direction. In addition, the proposed model can be tested in IoT environments and with different datasets [8, 9]. These future research areas have the potential to advance URL phishing detection and classification, contributing to the development of more effective cybersecurity solutions.

REFERENCES

[1] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards Lightweight URL-Based Phishing Detection," *Future Internet*, vol. 13, no. 6, Jun. 2021, Art. no. 154, https://doi.org/10.3390/fi13060154.

[2] S. Srivastava and S. K. Gupta, "Phishing Detection Techniques: A Comparative Study," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, Sep. 2021, pp. 1–6, https://doi.org/10.1109/ICRITO51393.2021.9596093.

[3] F. Tchakounte, V. S. Nyassi, D. E. H. Danga, K. P. Udagepola, and M. Atemkeng, "A Game Theoretical Model for Anticipating Email Spear-Phishing Strategies," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 30, 2021, https://doi.org/10.4108/eai.26-5-2020.166354.

[4] M. F. Alghenaim, N. A. A. Bakar, F. Abdul Rahim, V. Z. Vanduhe, and G. Alkawsi, "Phishing Attack Types and Mitigation: A Survey," in *Data Science and Emerging Technologies*, Khulna, Bangladesh, 2023, pp. 131–153, https://doi.org/10.1007/978-981-99-0741-0_10.

[5] C. Balim and E. S. Gunal, "Automatic Detection of Smishing Attacks by Machine Learning Methods," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, Ankara, Turkey, Nov. 2019, pp. 1–3, https://doi.org/10.1109/UBMYK48245.2019.8965429.

[6] S. Bell and P. Komisarczuk, "An Analysis of Phishing Blacklists: Google Safe Browsing, OpenPhish, and PhishTank," in *Proceedings of the Australasian Computer Science Week Multiconference*, Melbourne, Australia, Feb. 2020, pp. 1–11, https://doi.org/10.1145/3373017.3373020.

[7] A. K. Singh, "Malicious and Benign Webpages Dataset," *Data in Brief*, vol. 32, Oct. 2020, Art. no. 106304, https://doi.org/10.1016/j.dib.2020.106304.

[8] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, May 2015, https://doi.org/10.4236/jcc.2015.35021.

[9] A. Hannousse, "Web page phishing detection." Mendeley, Jun. 25, 2021, https://doi.org/10.17632/C2GW7FY2J4.3.