Adaptive Phishing Website Detection Using Incremental Machine Learning: A Dynamic Approach to Cybersecurity Threats

Ajla Kulaglic, Mutaz A. B. Al-Tarawneh
College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait

***Abstract*—The rapid expansion of internet services and cloud-based platforms has increased cybersecurity threats, particularly phishing attacks that deceive users into disclosing sensitive information. Traditional phishing detection methods, including blacklists and batch-learning models, often struggle to adapt to the continuously evolving nature of these attacks. In order to address this challenge, this study proposes an adaptive phishing detection framework based on incremental machine learning techniques that enable real-time learning and dynamic adjustment to new attack patterns. A comprehensive evaluation of multiple incremental algorithms was performed using the River-ML framework and a publicly available phishing website dataset. The models were assessed based on accuracy, precision, recall, F1 score, Cohen's kappa, and memory efficiency. Evaluation results demonstrate that models such as Aggregated Mondrian Forest, Extremely Fast Decision Trees, and Logistic Regression achieved strong classification performance, with the best accuracy reaching 90.15%, precision up to 91.05%, recall up to 89.42%, F1 score up to 88.75%, and Cohen's kappa up to 79.99%, while lightweight models like ALMA maintained extreme memory efficiency, requiring as little as 1.81 KB. In general, the proposed incremental learning framework significantly improves the effectiveness of phishing detection and computational efficiency, providing a scalable and adaptive defense mechanism against evolving cyber threats.**

***Keywords*—Phishing detection; incremental learning; online machine learning; cybersecurity threats; real-time classification**

I. Introduction

Phishing is a deceptive cyber attack that exploits both social engineering and technical manipulation to unlawfully obtain user credentials and financial information. Attackers craft fraudulent emails, messages, or website links that closely resemble legitimate communications from reputable organizations, luring users to access counterfeit websites where they unknowingly disclose sensitive details such as usernames and passwords. Beyond deceptive messaging, hackers further amplify these attacks by deploying malicious software designed to extract login credentials and intercept user data from compromised systems. Various techniques are used, including email spoofing, manipulated URLs, instant messaging, forum postings, phone calls, and SMS phishing (smishing) to trick users into providing personal information. The authenticity of phishing content often mirrors that of genuine platforms, making it difficult for users to distinguish between legitimate and fraudulent interactions. The primary objective of phishing attacks is to obtain personal data for financial gain or identity theft, causing significant economic damage on a global scale [1].

Cybercriminals acquire confidential data by replicating legitimate websites and emails, often impersonating financial institutions or organizations that handle sensitive financial transactions [2], [3], [4]. These fraudulent emails are meticulously designed using authentic company logos, slogans, and branding elements to appear genuine. By exploiting HTML structures, attackers replicate entire websites or images, making them visually indistinguishable from their legitimate counterparts [5]. This deceptive practice has been facilitated by the widespread use of the Internet as a primary communication medium, allowing threat actors to exploit well-known brands, trademarks, and corporate identities that users trust for verification. To maximize their reach, phishers distribute large volumes of deceptive emails, commonly referred to as "spooled" emails, targeting unsuspecting recipients. Once these emails are opened, victims are typically redirected from an authentic-looking platform to a fraudulent website, where they unknowingly submit personal credentials [6], [7], [8]. The exploitation of user information through phishing has become a critical cybersecurity challenge in modern society [9], [10], [11]. Extensive research has been conducted to analyze various phishing indicators, including website URLs, webpage content, source code structures, and visual similarities, in an effort to mitigate these threats [12], [13], [14].

However, a significant gap remains in the availability of adaptive and efficient anti-phishing tools that can proactively detect and counteract evolving phishing tactics. Machine Learning (ML) techniques have been proven to be highly effective in identifying malicious URLs in real-time, offering a more dynamic alternative to conventional detection methods [15], [16], [17], [18], [19], [20], [21]. Traditional blacklist-based URL detection relies on manually curated databases of known phishing links, which require frequent updates and struggle to keep pace with the continuous emergence of new phishing domains. Attackers further evade detection through automated techniques such as Domain Generation Algorithms (DGA), rendering blacklist-based defenses inadequate [22]. The complexity of maintaining an exhaustive blacklist highlights the need for more intelligent and adaptive detection mechanisms. Researchers have increasingly advocated for ML-driven approaches, which leverage pattern recognition and predictive modeling to classify URLs as malicious or benign. By framing phishing detection as a binary classification problem, ML models can efficiently map high-dimensional feature spaces to corresponding threat labels, offering superior adaptability and generalization compared to conventional blacklist-based methods [23]. Despite growing interest in ap-

plying machine learning techniques for phishing detection, most existing approaches rely on batch or offline learning, which may be insufficient to handle evolving and dynamic attack strategies. Batch learning models present several limitations, particularly in the context of phishing detection. First, the training process is computationally intensive, requiring substantial processing power and memory resources. Second, the performance of the model is highly dependent on the size and quality of the training data, making generalization to new threats a challenge. Third, batch learning operates under the assumption that training data remain static over time, preventing the model from adapting to new samples or changes in data distribution (concept drift), necessitating frequent retraining and model redevelopment. Given the continuous nature of security threats and the dynamic landscape of web activity, incremental machine learning algorithms offer a more effective alternative to batch-based approaches. These algorithms are designed to process unbounded data streams, allowing continuous learning and real-time model updates. Additionally, they are well-suited for real-time phishing detection, addressing challenges that traditional batch models struggle with, such as handling rapidly evolving threats. In addition, incremental learning excels at detecting concept drift, enabling models to dynamically adjust to changing attack patterns and browsing behaviors. This adaptability makes incremental machine learning a promising solution for phishing detection in environments where threat patterns change unpredictably over time [24], [25], [26]. Several incremental machine learning algorithms have been proposed in the literature [27], [28], [29], [26], [30], [31], [32], [33], yet their applicability to phishing detection and the trade-offs between performance metrics and memory efficiency remain underexplored. This study aims to systematically evaluate various incremental learning algorithms to determine their effectiveness in detecting phishing attempts. In addition, an in-depth analysis of key performance trade-offs, including accuracy, precision, recall, kappa statistic, and memory consumption, is conducted to identify the most suitable models for real-time phishing detection.

The remainder of this study is structured as follows: Section II reviews related research efforts in phishing detection and incremental learning. Section III outlines the research methodology and details the experimental setup and evaluation framework. Section IV presents the results of the online machine learning evaluation and examines the effectiveness of online model selection using multi-armed bandits. Finally, Section V summarizes the findings and concludes the study.

II. Literature Review

Phishing is a cyberattack in which malicious actors impersonate reputable websites or businesses to deceive users into revealing sensitive personal information, such as passwords, usernames, and bank account details. These attacks exploit human trust and employ deceptive strategies, including social engineering, email spoofing, and URL manipulation, to mislead unsuspecting users. Phishing incidents can have severe consequences, including financial loss, identity theft, and compromised cybersecurity. Various detection techniques have been proposed in the literature to counteract these threats, including List-Based, Visual Similarity, Heuristic, Machine Learning, and Deep Learning approaches [34].

List-based detection techniques are widely employed in web browsers such as Google Chrome, Microsoft Edge, and Firefox to identify phishing websites [35]. These approaches rely on whitelisting and blacklisting methods. A whitelist contains a list of legitimate URLs that browsers are allowed to access, ensuring that only approved websites can be loaded. In contrast, a blacklist maintains a database of known phishing or fraudulent URLs, preventing browsers from loading potentially malicious web pages. Although blacklisting can effectively block previously identified phishing websites, it requires frequent updates to remain effective. Even minor modifications to a phishing URL can bypass blacklist-based defenses, making this method less reliable against newly emerging threats.

Visual similarity-based detection methods aim to identify phishing websites by analyzing their similarities to legitimate ones. This approach compares various visual elements, including the logo of the website, text layout, CSS, source code, and screenshots, to determine potential fraud. Since phishing websites often mimic the appearance of legitimate platforms, this technique is useful for detecting impersonation attempts. However, it has limitations in identifying zero-hour phishing attacks, as it relies on previously recorded or stored websites for comparison. Without prior knowledge of new phishing attempts, visual similarity-based detection may fail to recognize novel threats [36]. Heuristic-based detection methods analyze various characteristics of a website to distinguish between phishing and legitimate web pages. This approach examines several features, including URL structures, domain registration details, SSL certificate status, website traffic, and DNS records. Using predefined rules and classification algorithms, heuristic detection can identify phishing attempts even when the fraudulent website is not included in a blacklist. The effectiveness of this method depends on the quality of the selected set of features, training samples, and classification techniques. One significant advantage of heuristic-based detection is its ability to identify zero-hour phishing attacks, making it a more dynamic approach compared to static detection mechanisms.

Machine learning techniques have gained widespread attention for phishing detection due to their ability to learn patterns from large datasets and identify emerging threats.With the increasing availability of large-scale datasets, machine learning has proven to be an efficient tool in identifying phishing websites, especially in high-speed and high-volume data environments [37]. Machine learning models are trained using features extracted from phishing URLs, website structures, and JavaScript behavior. These extracted attributes help classifiers, such as Support Vector Machines (SVM), Random Forest, and Neural Networks, differentiate between phishing and benign websites. By analyzing patterns in big data environments, machine learning-based detection methods provide improved accuracy and adaptability over traditional list-based approaches. However, the effectiveness of machine learning models depends on the quality of the data and the selection of features [38]. Deep learning techniques further enhance phishing detection by leveraging advanced neural networks to automatically extract complex features from web pages. Recent studies suggest that deep learning models can outperform traditional machine learning classifiers in detecting phishing websites. Popular deep learning architectures for phishing detection include deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN),

feed-forward deep neural networks, restricted Boltzmann machines, deep belief networks and deep autoencoders [39]. These models can analyze web content, user behavior, and URL characteristics to identify phishing attempts with higher accuracy. The ability of deep learning models to generalize patterns and adapt to new phishing techniques makes them a promising solution to combat evolving cybersecurity threats. Despite the effectiveness of existing detection methods, phishing attacks continue to evolve, requiring more adaptive and real-time detection mechanisms. The use of incremental learning in phishing detection remains an underexplored area, highlighting a critical gap in research. Addressing this limitation by incorporating incremental learning techniques could improve the robustness and efficiency of phishing detection systems in dynamic environments. The remainder of this section provides a more detailed discussion of machine learning and deep learning-based techniques, as they play a crucial role in phishing detection, particularly in the approach proposed in this study.

The study in [40] extensively examined the application of machine learning techniques to URL-based phishing detection. The authors evaluated multiple algorithms, including Naïve Bayes, Random Forest, K-Nearest Neighbor, AdaBoost, K-star, Support Vector Machines, and Decision Trees, using a self-constructed phishing dataset. Their findings indicated that the Random Forest algorithm, which is based solely on NLP-based features, exhibited the highest performance, achieving an accuracy rate of 97.98%. These results underscore the effectiveness of feature engineering to improve phishing detection. Similarly, the research conducted in [41] assessed the classification capabilities of Support Vector Machines, Decision Trees, and Logistic Regression using the PhishTank and DMOZ phishing datasets. Their findings revealed that among the models tested, the Support Vector Machines achieved the highest accuracy of 89.3%, demonstrating their robustness in distinguishing phishing from legitimate websites.

Further expanding on machine learning models, the study in [42] explored the performance of Random Forest, Support Vector Machines, Naïve Bayes, C4.5, JRip and PART classifiers in several phishing datasets, including PhishTank, OpenPhish, Alexa and Common Crawl. The results demonstrated that the Random Forest classifier outperformed other models, achieving an accuracy of 96.17%. This reinforces the previously established effectiveness of Random Forest in phishing detection. Based on these findings, [43] investigated the classification of legitimate and illegitimate websites using machine learning models such as Random Forest, Decision Trees, K-Nearest Neighbor (KNN), Iterative Dichotomiser-3 (ID3), and Naïve Bayes, trained on the UCI phishing dataset. Their study highlighted the importance of feature selection, as Genetic Algorithms (GAs) were shown to improve detection accuracy when used for feature optimization. More specifically, the combination of Iterative Dichotomiser-3 (ID3) with Yet Another Generating Genetic Algorithm (YAGGA) led to a significant increase in detection accuracy, reaching up to 95%. In [44], a novel approach was introduced, in which the authors developed the Hybrid Feature-Based Phishing Classifier (PHFBC), which integrates the Decision Tree and Naïve Bayes models with a statistical measure called the phishing ratio. Furthermore, the study proposed a robust feature selection technique using the Recursive Feature Subset Selection Algorithm (RFSSA). The proposed methodology was tested on the PhishTank, Chinese e-Business and DMOZ phishing datasets, achieving a True Positive Rate (TPR) of 0.984. This further emphasizes the importance of feature selection in improving detection performance. In another study, [45] focused on optimizing machine learning-based phishing detection through feature selection and hyperparameter tuning. The research explored multiple optimization strategies, including the Tree-structured Parzen Estimator (TPE) and Genetic Algorithms (GA) for hyperparameter optimization, as well as Moth Flame Optimization (MFO) and Particle Swarm Optimization (PSO) for feature selection. The experiments conducted on the PhishTank and Alexa datasets indicated that the optimal combination was Random Forest with PSO for feature selection and TPE for hyperparameter tuning, resulting in an impressive accuracy of 99.33%. These findings highlight the critical role of optimization techniques in improving the accuracy of phishing detection. In a related effort, [46] proposed a phishing detection system that uses URL analysis in eight different machine learning algorithms, including Logistic Regression, K-Nearest Neighbor, Decision Tree, Support Vector Machines, Naïve Bayes, XGBoost, Random Forest, and Artificial Neural Networks. Their comparative analysis on three different datasets—PhishTank, Alexa, and Common Crawl—showed that Random Forest applied to the PhishTank dataset achieved the highest accuracy of 94.59%. This reinforces the dominance of Random Forest in phishing detection applications. Another significant study was conducted by [47], who explored the use of domain name features for phishing detection by incorporating DNS data, blacklists, and lexical attributes. Their classifier, based on logistic regression, was trained using four datasets: PhishTank, Alexa, ICANN, and DNS-BH. However, while their approach demonstrated effectiveness in distinguishing between benign and malicious domains, overall accuracy remained relatively low at approximately 60%, indicating the limitations of purely lexical-based phishing detection. In contrast, [48] introduced a data augmentation method to improve machine learning-based phishing detection. Their study leveraged an Adversarial Autoencoder (AAE) to generate synthetic phishing samples, enhancing training datasets. The proposed model was tested on multiple classifiers, including support vector machines, decision trees, gradient booster, K-nearest neighbor, and random forest, using the UCI and Mendeley phishing datasets. The results demonstrated that models trained with AAE-generated data exhibited greater robustness and accuracy. Specifically, the Gradient Boosting classifier achieved the highest accuracy of 95.47%, validating the benefits of augmenting phishing datasets with adversarial examples. Expanding on previous research, the study in [49] examined the structural characteristics of URLs from phishing websites by extracting 12 different types of information. The authors trained four machine learning algorithms, namely Logistic Regression, Support Vector Machine, Naïve Bayes, and Decision Tree, using the PhishTank and DMOZ datasets. Their findings indicated that Logistic Regression outperformed the other classifiers, achieving an optimal accuracy of 95.12%. These results highlight the importance of URL-based feature extraction in enhancing the accuracy of phishing detection. Further advancing the field, the authors in [50] tackled the challenge of phishing detection within e-commerce platforms by proposing a framework for constructing reproducible and extensible datasets. Their methodology involved categorizing and selecting 87 widely

recognized phishing features, assessing their relevance and runtime performance, and leveraging them to build a robust dataset. Through conceptual replication and evaluation across the PhishTank, Alexa, and OpenPhish datasets, their results demonstrated that Random Forest was the most effective classifier, particularly when utilizing features from external services. Furthermore, their analysis showed that hybrid features contributed to the highest classification accuracy of 96.61%, while filter-based feature selection methods outperformed wrapper techniques, reaching an accuracy of 96.83%. Similarly, the work in [51] introduced a novel approach to detect real-time phishing attacks by evaluating deep hybrid features using the Light Gradient Boosted Machine model. Their method involved normalizing web page request features and applying Sparse Autoencoder and Principal Component Analysis for dimensionality reduction. The experimental evaluation, conducted on the ISCXURL-2016 dataset, demonstrated that their approach achieved a high classification accuracy of 99.6%, effectively minimizing false positives before processing web requests.

Extending the discussion on phishing datasets, the study in [52] highlighted the limitations of existing datasets, particularly their exclusion of login pages, which are critical for phishing detection. To address this gap, the authors introduced PILU-60K, a novel dataset that incorporates login URLs. Their experimental analysis revealed that while the support vector machines exhibited resilience against evolving phishing tactics, Random Forest achieved superior performance with PILU-60K, achieving an accuracy of 94.59%. These findings underscore the need to include login page data in phishing datasets to enhance detection accuracy. Furthermore, the research in [53] examined the security risks associated with browsing the Internet, particularly due to the low awareness of users of cyber threats such as phishing attacks. Their study introduced a supervised machine learning approach that utilizes Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine, and Multi-Layer Perceptron classifiers. By extracting novel features solely from URLs and testing them on the PhishTank dataset, the study assessed the effectiveness of their approach compared to Google Safe Browsing (GSB), the default security control in major web browsers. Their results demonstrated that the proposed method consistently outperformed GSB, even when tested against phishing URLs that remained active for more than a year after model training. In particular, their optimized Random Forest model achieved the highest accuracy level of 99.29%. Building on the effectiveness of ensemble learning, [54] introduced PDCFS, a novel phishing detection model that uses hybrid cumulative feature selection. Their model partitioned datasets into multiple subsets based on diverse feature selection techniques, such as Chi-Square and Principal Component Analysis, and employed classifiers like Support Vector Machine and Random Forest. By implementing a five-fold cross-validation strategy and aggregating the results through majority voting, their findings indicated that Random Forest, when applied to a reduced feature set of 32, achieved the highest accuracy of 98.36%. Meanwhile, the PDCFS model itself achieved a close accuracy of 98.24%, demonstrating its competitive performance against other hybrid models.

The study in [55] explored the importance of phishing attacks and the ongoing challenge of developing robust detection mechanisms, particularly against zero-day attacks. To address these issues, the authors proposed a novel approach that integrates convolutional operations to model character-level URL features with a deep convolutional autoencoder (CAE). Their extensive experiments, conducted on three real-world datasets—PhishTank, PhishStorm, and ISCX-URL-2016—spanning a total of 222,541 URLs, demonstrated that the proposed method outperformed the latest deep learning models. In particular, receiver operating characteristic (ROC) curve analysis and 10-fold cross-validation revealed a 3.98% improvement in sensitivity over existing state-of-the-art models. These results underscore the effectiveness of using both sustainability and intelligibility in phishing detection, reinforcing the importance of deep learning-driven feature representation. Similarly, the research presented in [56] examined the surge in cyber-attacks, particularly phishing, that intensified due to increased digital activity during the pandemic. The study implemented a phishing detection system based on Convolutional Neural Networks (CNNs), where n-gram features were extracted from URLs using the PhishTank dataset. Experimental findings indicated that unigram features, particularly those focused on 70 specified characters irrespective of case sensitivity, achieved the highest accuracy rate of 88.90%. Additionally, the system demonstrated efficient URL classification, processing each URL in approximately 0.008 seconds. This research highlights the potential of deep learning models in achieving fast and effective phishing detection. Expanding on hybrid approaches, [57] proposed an innovative spam detection model that integrates dataset acquisition, feature extraction, optimal feature selection, and detection phases. The study collected a benchmark dataset containing both text and image data, extracting features using Term Frequency-Inverse Document Frequency (TF-IDF) for text analysis and the color correlogram and Gray-Level Co-occurrence Matrix (GLCM) for image analysis. To enhance classification accuracy, the authors employed a metaheuristic optimization algorithm called the Fitness Oriented Levy Improvement-based Dragonfly Algorithm (FLI-DA) for feature selection. The detection process combined recurrent neural networks (RNNs) and convolutional neural networks (CNNs), with FLI-DA further optimizing the number of hidden neurons. Experimental evaluations demonstrated significant improvements in spam email classification performance, reinforcing the effectiveness of hybrid deep learning approaches in phishing detection.

Further advancing phishing detection in the email domain, the study in [58] highlighted the growing vulnerability of organizations to phishing and spam emails, emphasizing the need for efficient detection strategies. To this end, the authors introduced Phish Responder, a hybrid machine learning framework that integrates long-short-term memory (LSTM) networks for text-based datasets and Multilayer Perceptron (MLP) for numerical-based datasets in natural language processing. Their experimental evaluation demonstrated that Phish Responder achieved an impressive accuracy of 99% using the LSTM model and 94% with the MLP model. Comparisons with existing solutions revealed the superiority of their approach, particularly in numerical-based phishing detection, thereby showcasing the potential of hybrid machine learning models in combating sophisticated phishing schemes. Based on the need for improved phishing detection systems, [59] developed a hybrid phishing detection model that combined URL and content analysis based on deep learning. The study

addressed the growing prevalence of phishing attacks, particularly those targeting victims through email communications. The proposed system aimed to improve detection accuracy while minimizing false positives by incorporating both URL-based and content-based features. Experimental evaluations, conducted on a high-risk URL and content-based phishing detection data set sourced from PhishTank, demonstrated a significant improvement in accuracy, reaching 98.37%. These results further highlight the advantages of integrating multiple detection mechanisms to enhance the accuracy of phishing identification while reducing erroneous classifications. The study in [60] explored the increasing threat of phishing attacks and proposed a deep learning-based detection model to improve the identification of phishing. By providing a comprehensive overview of phishing techniques, the authors emphasized the urgent need for more effective detection technologies. Their approach involved training and validating a model using split dataset, that analyzed email text and other relevant features to classify emails as phishing or non-phishing. The results demonstrated that the boosted decision tree algorithm achieved impressive accuracy rates of 88%, 100%, and 97% across different datasets, reinforcing the robustness and adaptability of the model to various phishing patterns. Expanding on ensemble learning strategies, the work in [61] introduced HELPHED, a phishing email detection framework that integrates hybrid features with Ensemble Learning methods. The methodology utilized two distinct approaches: Stacking Ensemble Learning and Soft Voting Ensemble Learning, each using multiple machine learning algorithms to improve classification accuracy. A rigorous evaluation process, which incorporates innovative guidelines and extensive experimentation, validated the effectiveness of the model. Conducted on a highly imbalanced dataset comprising 32,051 benign and 3,460 phishing email samples, the study revealed that the Soft Voting Ensemble Learning approach outperformed competing deep learning and machine learning models, achieving an outstanding F1 score of 0.9942. These findings highlight the potential of hybrid ensemble techniques in improving phishing detection. Similarly, [62] introduced the LBPS model, a hybrid deep neural network designed for the detection of phishing scam accounts. Using LSTM-FCN and BP neural networks, the model analyzed transaction records to establish implicit relationships between the extracted features. Additionally, the LSTM-FCN network effectively captured temporal patterns from all transaction records associated with a target account. The experimental evaluation demonstrated the effectiveness of the selected features, as the LBPS model consistently outperformed existing detection methods, achieving a remarkable F1-score of 97.86% on Ethereum transaction datasets. This research underscores the importance of hybrid deep learning approaches in identifying fraudulent accounts with high accuracy. Further advancing phishing detection methodologies, the study in [63] introduced Hybrid Feature Detection based on Hybrid Features (PDHF), a framework that integrates optimal artificial and automatic deep learning features. Artificial phishing features were optimized by eliminating redundancies using a novel evaluation index and an enhanced bidirectional search algorithm, while deep features were extracted from URLs through a one-dimensional CNN and a disorderly quantized attention mechanism. The experimental findings confirmed the superiority of PDHF's over existing phishing detection methods, achieving an exceptional accuracy of 99.65%, along with a precision of 99.42%, recall of 99.40%, and an F1-score of 99.41%. These results further validate the advantages of hybrid feature-based approaches in refining phishing detection accuracy. Building on these advances, [64] introduced RNT, a deep learning-based phishing detection technique that combines the ResNeXt architecture with an embedded GRU model. Addressing the class imbalance issues, the study incorporated the SMOTE technique, while autoencoders and ResNet (EARN) were used for enhanced feature engineering. Furthermore, hyperparameter optimization was performed using the Jaya optimization method (RNT-J), ensuring improved model performance. Experimental results demonstrated that RNT consistently outperformed state-of-the-art phishing detection models, achieving an impressive accuracy of 98% while maintaining low false positive and false negative rates. These findings highlight the effectiveness of combining deep learning architectures with optimized feature engineering to develop high-precision real-time phishing detection solutions.

In summary, this section highlights the widespread and continuously evolving nature of phishing attacks, which pose significant threats to individuals and organizations worldwide. Over the years, various detection methods have been proposed, including List-Based, Visual Similarity, Heuristic, Machine Learning, and Deep Learning techniques, each offering distinct advantages in identifying phishing websites. However, despite their effectiveness, most of these approaches rely on batch or offline machine learning models, limiting their adaptability to emerging and dynamic phishing strategies. Numerous studies have evaluated machine learning algorithms such as Random Forest, Support Vector Machine, Naïve Bayes, and Decision Tree across diverse datasets, demonstrating high accuracy in phishing detection. Furthermore, deep learning techniques, including Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN), have shown significant promise in improving detection capabilities through automated feature learning. Although these advances have contributed to improved accuracy, they remain largely dependent on static training models that require frequent retraining to accommodate new threats. Despite these progresses, a critical research gap persists in the development and application of incremental machine learning techniques specifically tailored for phishing detection. Given the dynamic and continuously evolving nature of phishing attacks, detection systems must incorporate adaptive learning mechanisms capable of updating models in real time. Addressing this limitation is crucial for enhancing the robustness and efficiency of phishing detection systems, ultimately leading to more resilient cybersecurity solutions. Hence, further research should focus on integrating incremental learning methodologies to ensure adaptability and sustained performance in real-world phishing detection scenarios.

III. Methodology

This section describes the key procedures implemented to evaluate the effectiveness of the selected incremental machine learning algorithms using the phishing URL dataset provided in [65], [66]. The evaluation process, illustrated in Fig. 1, is carried out using the River Incremental Machine Learning Platform [67]. Each incremental machine learning algorithm is assessed following the same evaluation procedure to ensure consistency and comparability. As depicted in Fig. 1, the evaluation process begins with loading the phishing URL

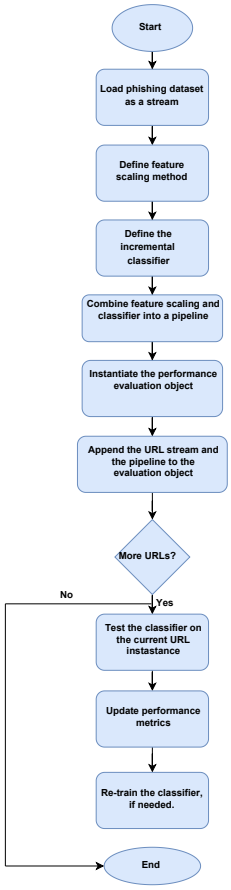

Fig. 1. Evaluation flowchart.

dataset as a continuous data stream in the River environment. Next, an incremental machine learning pipeline is constructed that incorporates a feature scaling method along with an incremental classifier. Once the pipeline is established, the phishing URL stream and the configured pipeline are integrated into the performance evaluation module, preparing the system for training and assessment. This setup ensures that the evaluation process accurately reflects the real-time adaptability of incremental learning models in phishing detection.

In this study, the performance of incremental machine learning classification algorithms is evaluated using the prequential evaluation method, also known as the interleaved test-then-train approach. This evaluation technique is specifically designed for incremental machine learning scenarios, where each incoming data sample, called an instance, plays a dual role in both testing and training the model. As instances are processed sequentially in their order of arrival, they become inaccessible after processing, ensuring that the model continuously adapts to new data without retaining past instances. In the context of phishing URL detection, the prequential evaluation process begins by using each incoming URL sample to test the incremental machine learning algorithm, generating a prediction. Immediately afterward, the same sample is utilized for training, allowing the model to refine its learning dynamically. Performance metrics are incrementally updated with each observed instance, allowing for a continuous assessment of the effectiveness of the model and its adaptability to previously unseen data. This ensures that the incremental machine learning algorithm undergoes a real-time evaluation, with its performance metrics continuously reflecting its ability to generalize to new phishing attempts. To comprehensively assess the effectiveness of the algorithms tested, several widely recognized performance metrics are used. These include accuracy, precision, recall, F score, kappa statistic, and memory footprint of the machine learning model during online learning. In the context of phishing URL detection, these metrics are formally defined as follows:

- Classification Accuracy: Represents the percentage of correctly classified URL samples [see Eq. (1)].

$$Accuracy = \frac{TN + TP}{TP + FP + FN + TN} \times 100\% \quad (1)$$

where, TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. TP represents the number of correctly classified phishing URLs, while TN corresponds to correctly classified benign URLs. In contrast, FP denotes benign URLs misclassified as phishing, whereas FN refers to phishing URLs misclassified as benign.

- Precision: Measures the proportion of URLs classified as phishing that actually belong to the phishing class [see Eq. (2)].

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

- Recall: Evaluates the proportion of phishing URLs correctly identified out of all phishing URLs in the observed stream [see Eq. (3)].

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

- F-score: Computes the harmonic mean of precision and recall, providing a balanced measure of a model's classification performance [see Eq. (4)].

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

- Cohen's Kappa (κ): A robust measure of classification accuracy that accounts for agreement by chance. Unlike traditional accuracy metrics, Cohen's Kappa highlights improvements over a majority-class classifier that predicts all instances as the most frequent class [68]. This is particularly useful when evaluating the classification performance in unbalanced data streams [see Eq. (5)].

$$\kappa = \frac{p_0 - p_c}{1 - p_c} \quad (5)$$

where, p_0 represents the prequential accuracy of the classifier, while p_c denotes the probability that a chance classifier

correctly predicts labels [69]. A value of $\kappa = 1$ indicates perfect classification performance.

- Model Size: Represents the memory space occupied by the incremental machine learning model during the prequential evaluation process, reflecting its computational efficiency.

To comprehensively evaluate incremental machine learning algorithms, an extensive set of experiments was conducted using various models, as detailed in Table I. This study explores a diverse range of incremental machine learning approaches, including ensemble-based methods, forest models, linear classifiers, Naïve Bayes, nearest neighbor models, and tree-based classifiers. Additionally, each algorithm was tested under three different feature scaling settings: no scaling, min-max scaling, and standard scaling.

The Min-Max scaling, defined in Eq. (6), transforms the data to a fixed range between 0 and 1, preserving the relative features relationships. In contrast, standard scaling, as described in Eq. (7), adjusts the data to have zero mean and unit variance, ensuring that the features contribute equally to model training. In this study, the impact of these scaling techniques on the performance of incremental learning models is systematically analyzed.

TABLE I. UTILIZED INCREMENTAL MACHINE LEARNING ALGORITHMS

Category	Algorithm Name
Ensemble Models	ADWINBagging Classifier ADWINBoosting Classifier AdaBoost Classifier BOLE Classifier Bagging Classifier Leveraging Bagging Classifier Streaming Random Patches (SRP) Classifier Stacking Classifier Voting Classifier
Forest Models	Aggregated Mondrian Forest (AMF) Classifier Adaptive Random Forest (ARF) Classifier
Linear Models	Logistic Regression (LR) Classifier Approximate Large Margin Algorithm (ALMA) Classifier Passive-aggressive (PA) Classifier Perceptron Classifier
Naïve Bayes Models	Gaussian (NB) Classifier
Neighbours Models	K-Nearest Neighbors (KNN) Classifier
Tree Models	Hoeffding Tree (HT) Classifier Hoeffding Adaptive Tree (HAT) Classifier Extremely Fast Decision Tree (EFDT) Classifier Stochastic Gradient Tree (SGT) Classifier

$$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{6}$$

where,

- x denotes the original feature value.
- x_{scaled} represents the scaled feature value.
- x_{min} refers to the running minimum value of the feature in the data stream.
- x_{max} refers to the maximum running value of the feature in the data stream.

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma} \tag{7}$$

where,

- x represents the original feature value.
- x_{scaled} denotes the scaled value of the feature.
- μ corresponds to the running mean of the feature in the data stream.
- σ denotes the running standard deviation of the feature in the data stream.

IV. RESULTS

This section presents the results obtained by implementing the methodology described in Fig. 1 on the incremental machine learning algorithms summarized in Table I, evaluated using different feature scaling techniques. The objective is to analyze the impact of feature scaling on model performance and memory efficiency.

Fig. 2 illustrate the performance results for linear models in standard, min-max, and no feature scaling settings, respectively. The results show that the scaling of the features plays a crucial role in influencing the accuracy, precision, recall, F1 score, Cohen's kappa and model size.

As shown, the application of standard feature scaling significantly enhances the performance of the logistic regression model, which outperforms its linear counterparts. Specifically, logistic regression achieves an accuracy of 89.20%, with precision, recall, F1 score, and Cohen's kappa values of 86.42%, 89.42%, 87.89% and 78.15%, respectively. This improvement suggests that standard scaling effectively normalizes feature distributions, allowing the model to learn more efficiently and generalize better. On the other hand, the ALMA (Adaptive Linear Margin Algorithm) model exhibits the lowest memory consumption, occupying only 4.69 KB, making it the most resource-efficient among the linear models. This result highlights a critical trade-off between accuracy and memory efficiency, where ALMA prioritizes compactness at the cost of predictive performance.

A different performance pattern emerges under Min-Max scaling. The Perceptron model achieves the highest accuracy of 83.60%, indicating that the min-max scaling may provide a more favorable feature distribution for this model. Meanwhile, logistic regression maintains the highest precision value of 90.70%, suggesting that min-max scaling helps reduce false positive classifications in phishing detection. The Passive-Aggressive (PA) model, in contrast, attains the highest recall (81.57%), F1 score (81.27%), and Cohen's kappa (66.56%), indicating that it is particularly adept at capturing phishing instances under this scaling technique. Consistent with the previous setting, the ALMA model retains the lowest memory footprint, occupying only 6.56 KB.

When no feature scaling is applied, as shown in Fig. 2, the performance of the model remains relatively similar to the Min-Max scaling scenario, but with slight variations in key metrics. The Perceptron model continues to exhibit the highest accuracy (83.60%), while logistic regression achieves the highest precision (91.05%). The PA model, once again, records the highest recall (81.57%), F1 score (81.27%), and Cohen's kappa (66.56%), reinforcing its strong capability to identify phishing instances. In particular, the ALMA model

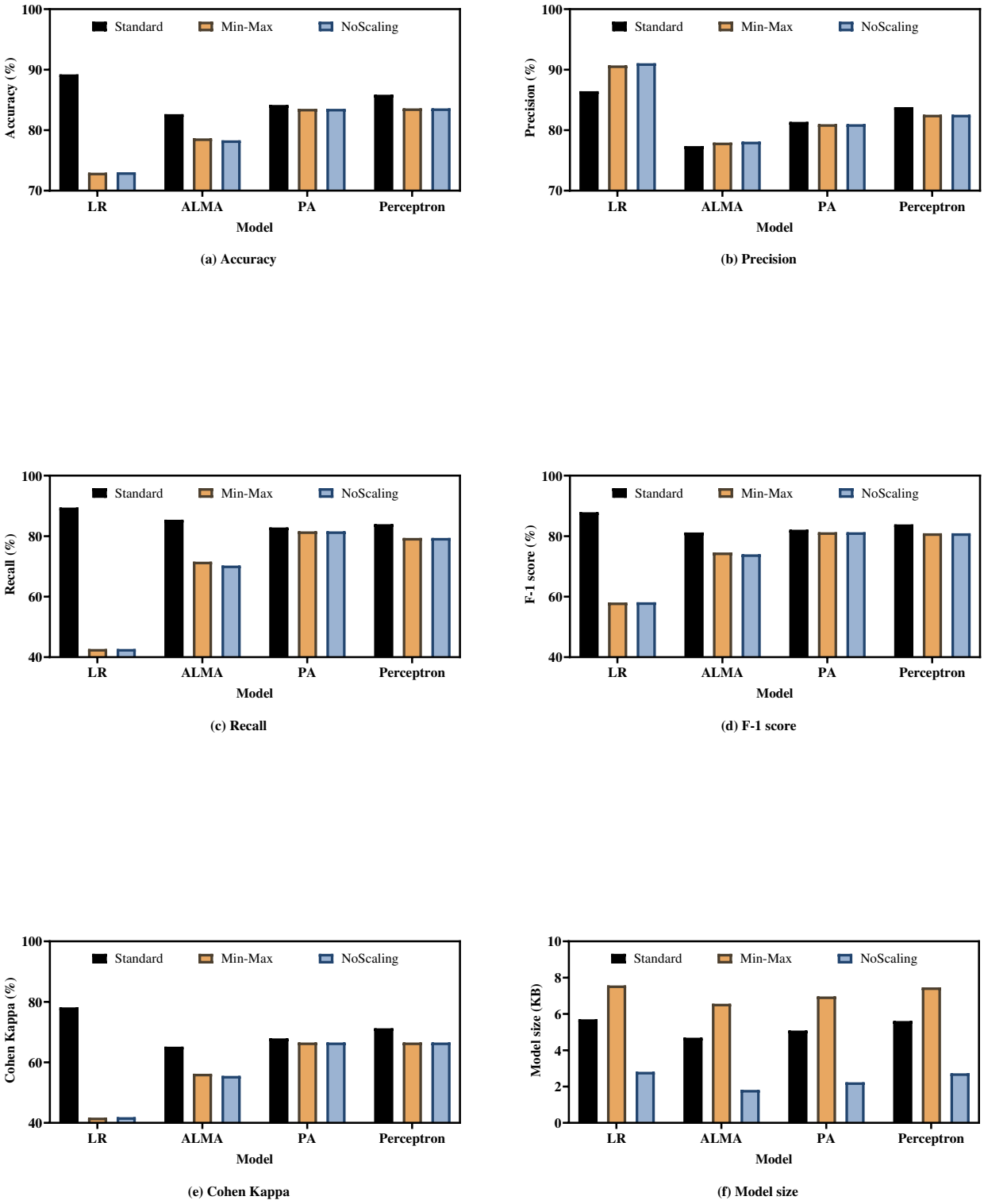

Fig. 2. Linear models' performance.

maintains its efficiency in terms of memory consumption, with a significantly reduced model size of just 1.81 KB, demonstrating its adaptability to unscaled feature spaces.

In general, these findings highlight the substantial impact of feature scaling on the performance of linear incremental learning models. Standard scaling generally enhances classification accuracy and stability, particularly benefiting logistic regression, while min-max scaling provides a more favorable

environment for models such as the Perceptron and PA. The absence of scaling appears to produce results comparable to min-max scaling, though with minor fluctuations in performance metrics. Additionally, ALMA consistently maintains the lowest memory usage in all settings, making it the most suitable choice for resource-constrained environments. These results emphasize the importance of selecting appropriate feature scaling techniques based on the specific trade-offs between accuracy and computational efficiency required for phishing detection tasks.

Fig. 3 illustrates the performance metrics obtained for the incremental machine learning models of Naive Bayes (NB) and K-Nearest Neighbors (KNN) under different feature scaling techniques. Interestingly, both NB and KNN models demonstrate a high degree of consistency in their performance across all feature scaling settings, indicating their relative robustness to variations in feature distributions.

As depicted, the KNN model achieves superior performance in terms of accuracy, precision, F1-score, and Cohen's kappa, attaining values of 88.87%, 86.04%, 87.51% and 77.48%, respectively. These results suggest that KNN benefits from its ability to classify new data points based on proximity in the feature space, which remains stable even after standard scaling is applied. In contrast, the NB model excels in recall, achieving a value of 89.40%, indicating its strength to correctly identify phishing instances. Additionally, the NB model maintains a smaller memory footprint, occupying 61.45 KB, which is notably more efficient compared to KNN.

This performance pattern remains largely consistent under min-max and no-scaling cases, where NB and KNN models continue to exhibit comparable trends, albeit with slight variations in individual performance metrics. The persistence of these trends across different scaling methods suggests that both NB and KNN models inherently adapt well to diverse feature distributions, making them relatively insensitive to the choice of feature scaling.

In general, these findings highlight the resilience of the NB and KNN models to feature scaling, maintaining consistent classification performance regardless of the applied normalization technique. This robustness makes them suitable candidates for phishing detection tasks, where feature distributions may vary dynamically. Furthermore, the trade-off between model size and performance suggests that, while KNN offers higher predictive accuracy, NB remains a computationally efficient alternative, making it particularly useful in scenarios with resource constraints.

Fig. 4 illustrates the performance of several tree-based models under different feature scaling techniques. The results indicate that tree-based models demonstrate relatively stable performance across various scaling settings, with minimal variations in classification accuracy and other key metrics.

As shown, under standard feature scaling, the Extremely Fast Decision Tree (EFDT) emerges as the best performing model, achieving the highest accuracy (88.47%), recall (88.85%), F1-score (87.10%) and Cohen's kappa (76.68%). These results suggest that EFDT effectively learns from streaming data while maintaining strong generalization capabilities. In contrast, the Hoeffding Tree (HT) algorithm attains the highest precision (85.51%) and maintains the lowest memory footprint, with a model size of 139.03 KB, demonstrating its computational efficiency.

Moving to Min-Max scaling, EFDT continues to achieve the highest accuracy (88.63%), F1-score (87.16%) and Cohen's kappa (76.96%), reinforcing its effectiveness across different scaling techniques. However, performance differences become more pronounced among other tree-based models. The Hoeffding Adaptive Tree (HAT) attains the highest precision (87.10%), while the Stochastic Gradient Tree (SGT) model achieves the highest recall (88.32%). Despite these variations, the HT algorithm retains its advantage in memory efficiency, maintaining a compact model size of 140.81 KB.

When no feature scaling is applied, EFDT continues to lead in accuracy (88.79%), F1-score (87.34%), and Cohen's kappa (77.29%), highlighting its robustness in handling raw feature distributions. Meanwhile, the HT model exhibits the highest precision (87.52%), further supporting its tendency to reduce false positives. Furthermore, the SGT model records the highest recall (88.32%), strengthening its ability to detect phishing instances effectively. The HT model maintains its efficiency in memory consumption, achieving the smallest model size of 131.73 KB.

In general, tree-based models exhibit minimal performance variations in different feature scaling settings, suggesting their inherent ability to adapt to varying feature distributions without requiring extensive normalization. The EFDT model consistently outperforms its counterparts in accuracy and generalization metrics, making it a strong candidate for phishing detection tasks. Meanwhile, the HT model remains the most memory-efficient, balancing classification performance with low computational overhead. These findings suggest that, while feature scaling can have a significant impact on certain machine learning models, tree-based algorithms remain relatively resilient, ensuring consistent performance across different preprocessing techniques.

Fig. 5 illustrates the performance of the Aggregated Mondrian Forest (AMF) and Adaptive Random Forest (ARF) models under different feature scaling techniques. The results indicate that both AMF and ARF exhibit consistently strong classification performance across all scaling settings, highlighting their robustness in handling varying feature distributions.

As shown, under standard feature scaling, the AMF model achieves the highest performance across multiple metrics, including accuracy (90.15%), precision (88.83%), F1-score (88.75%), Cohen's kappa (79.99%), and model size (3051.52 KB). These results suggest that AMF effectively leverages feature standardization to enhance classification performance while maintaining a relatively compact memory footprint. Meanwhile, the ARF model demonstrates a marginally higher recall value of 88.85%, indicating its strength in correctly identifying phishing instances while slightly compromising precision. A similar performance trend is observed when the Min-Max feature scaling is applied. Both AMF and ARF models continue to maintain high classification performance, with minimal deviations in accuracy, precision, and recall. This consistency reinforces their adaptability to different feature normalization techniques, making them well-suited for real-world phishing detection scenarios.

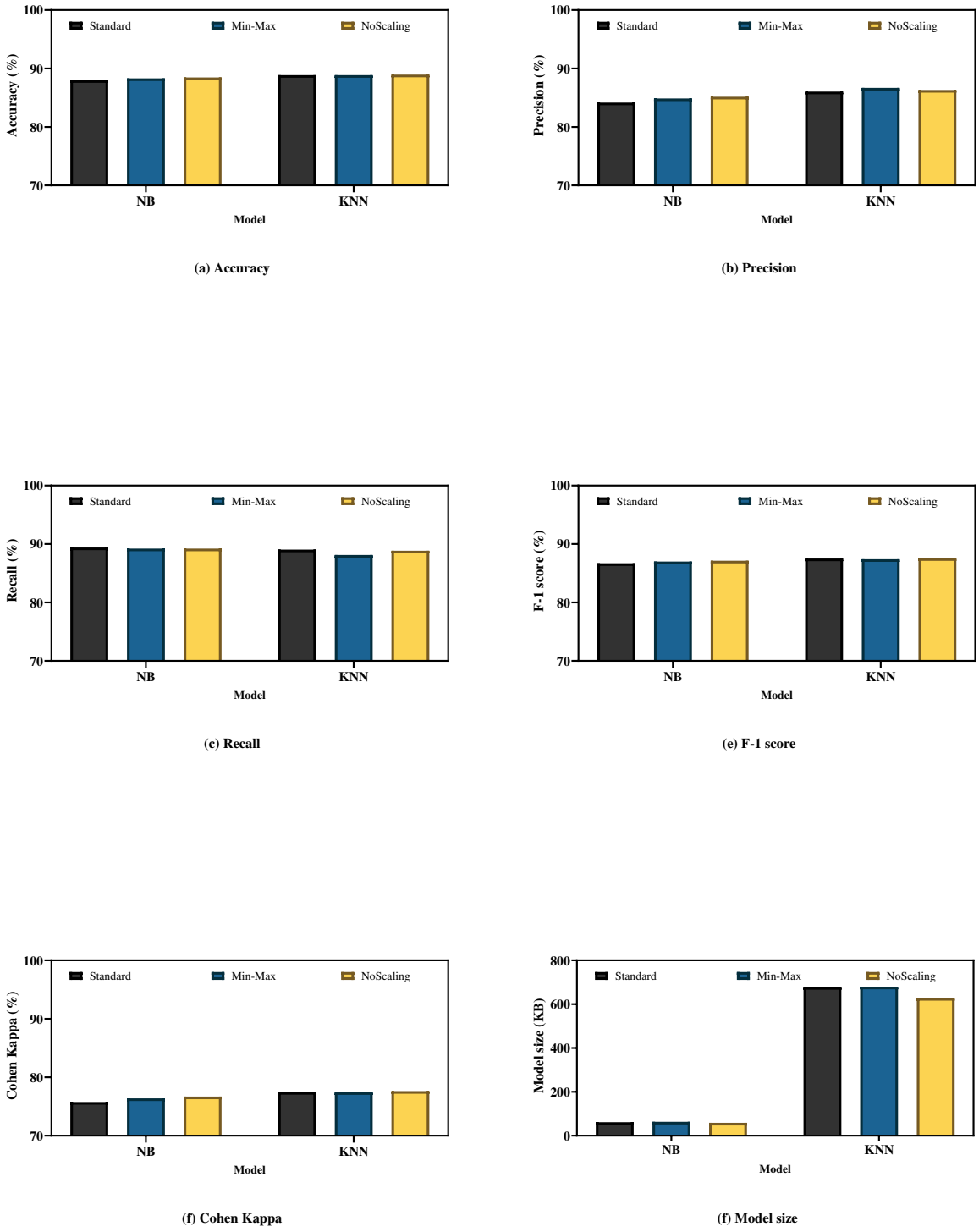

Fig. 3. Naive and Neighbour's models.

When no feature scaling is applied, both models experience a slight decrease in classification performance. However, in this setting, ARF marginally outperforms AMF, achieving an accuracy of 89.83%, precision of 88.04%, recall of 88.85%, F1-score of 88.44%, and Cohen's kappa of 79.37%. Despite this, the AMF model maintains an advantage in memory efficiency, requiring only 3000.32 KB compared to the larger footprint of the ARF. This trade-off suggests that while ARF can slightly improve predictive performance in unscaled data, AMF remains the more memory-efficient option.

In general, these findings indicate that AMF and ARF mod-

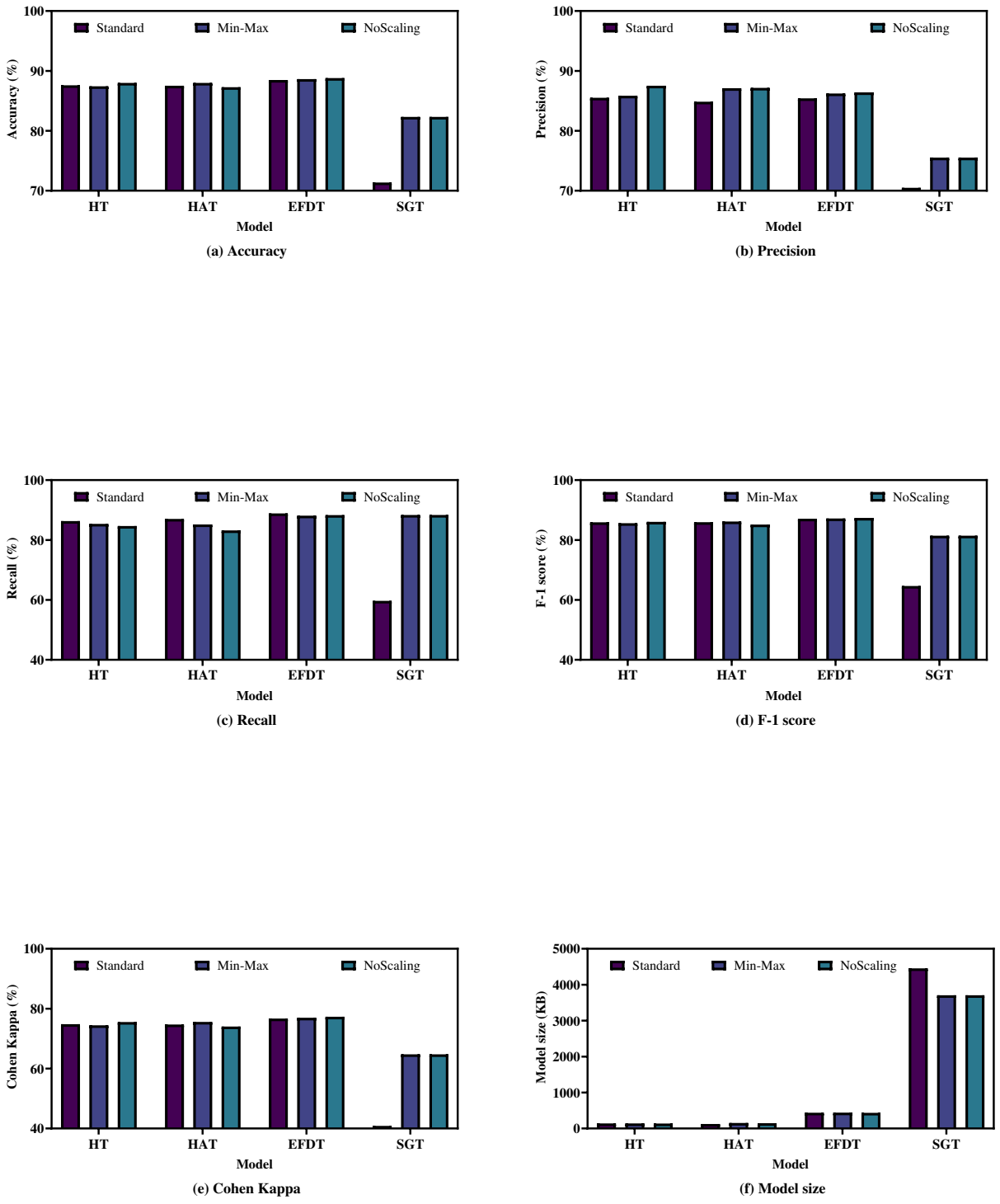

Fig. 4. Tree-based models' performance.

els exhibit resilience to feature scaling techniques, consistently achieving high classification accuracy regardless of the applied normalization approach. The AMF model excels in terms of memory efficiency while maintaining strong classification performance, while the ARF model demonstrates slightly superior recall and generalization capabilities. These insights suggest that both models are well suited for phishing detection tasks, particularly in environments with varying data preprocessing constraints.

Fig. 6 presents the performance evaluation of ensemble-based incremental machine learning models under different

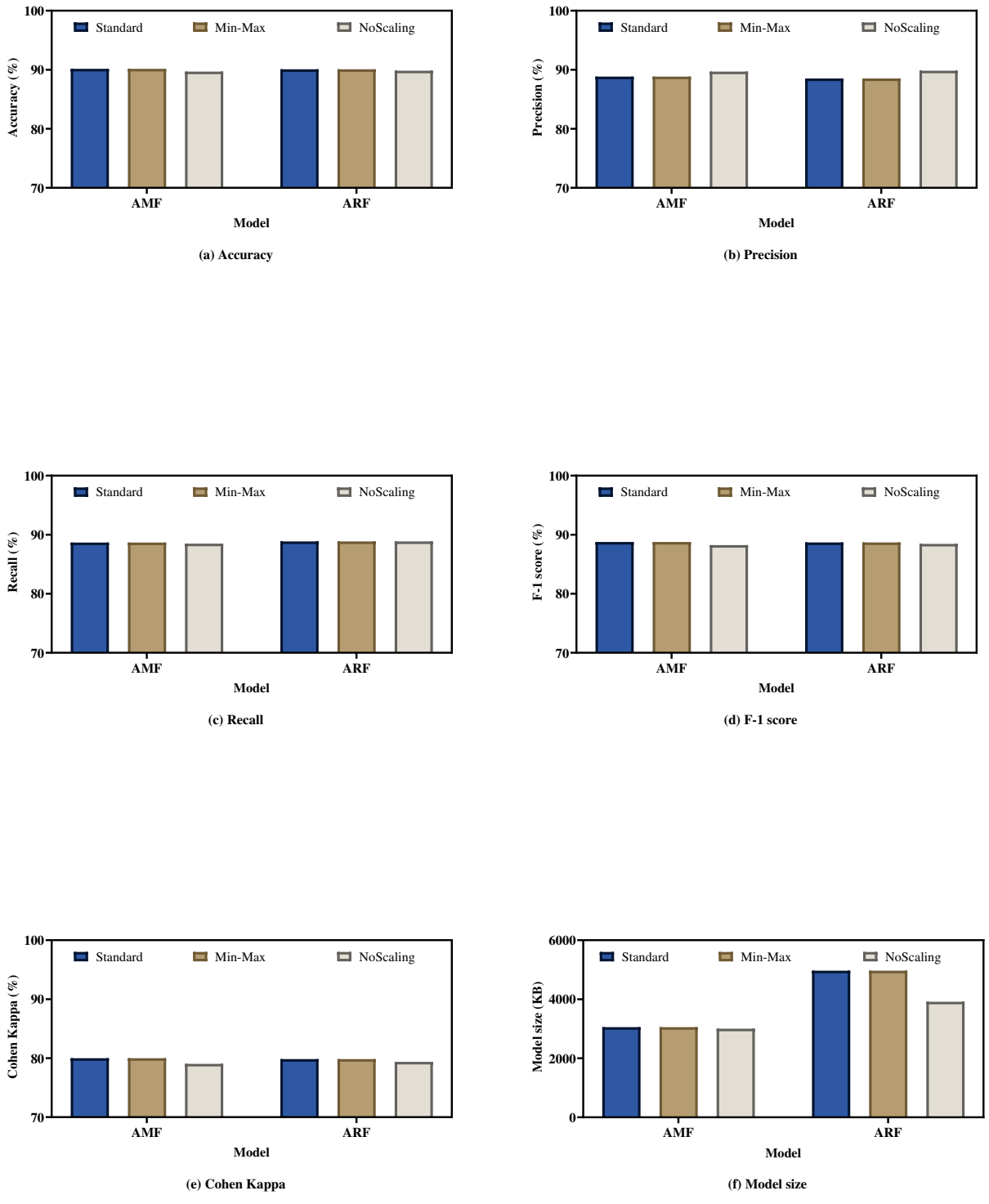

Fig. 5. Forest models' performance.

feature scaling techniques. Unlike other model categories, the ensemble methods exhibit noticeable variations in performance metrics depending on the applied feature scaling strategy.

As depicted, when standard feature scaling is applied, no single ensemble model dominates across all performance metrics. The Leveraging Bagging (LB) technique achieves the highest accuracy at 89.68%, demonstrating its robustness in classifying phishing URLs correctly. Meanwhile, the stacking ensemble outperforms other models in terms of precision, attaining a value of 88.38%, suggesting its effectiveness in

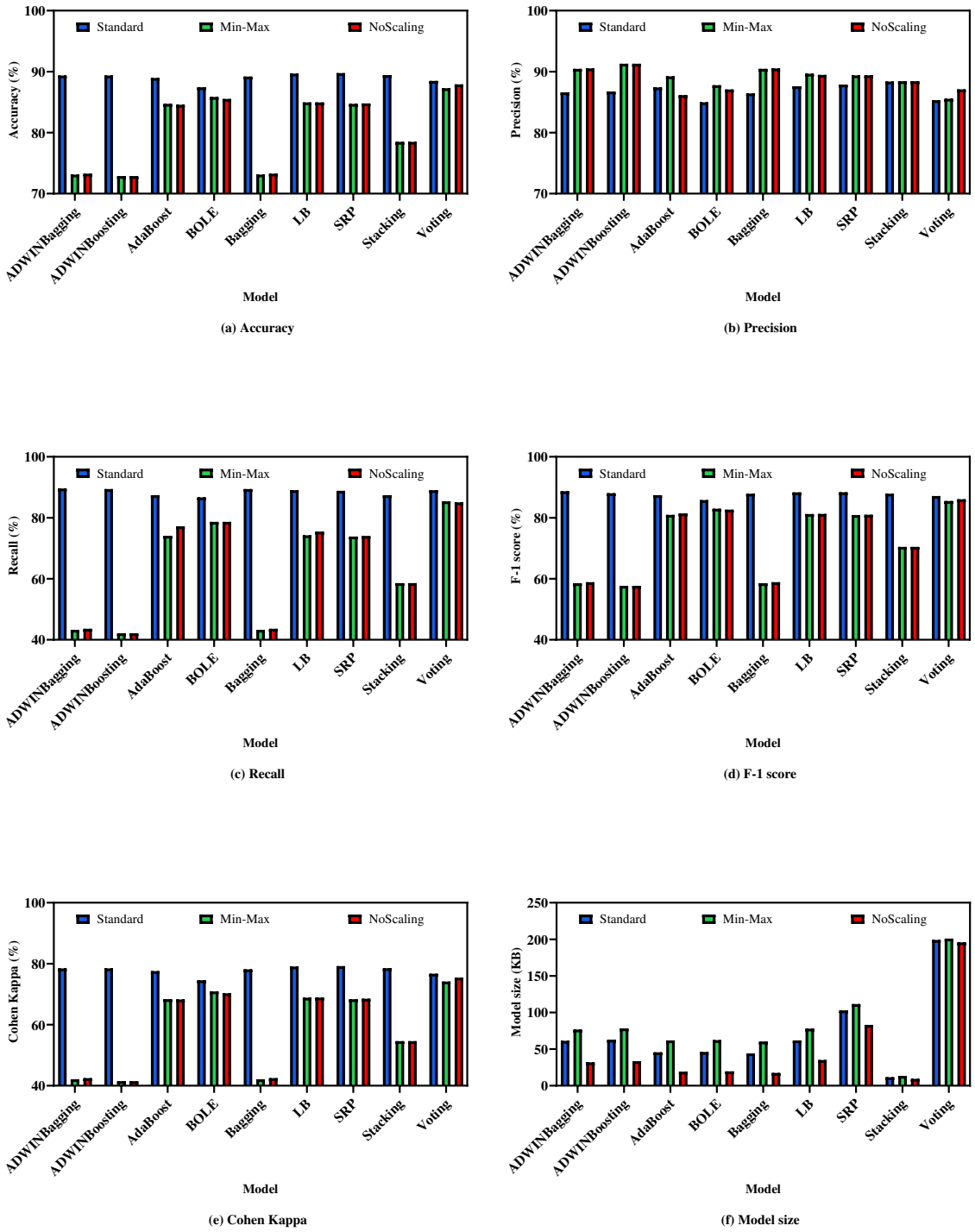

Fig. 6. Ensemble models' performance.

minimizing false positives. The ADWINBagging ensemble stands out in recall (89.60%) and F1-score (88.70%), indicating its strength in correctly identifying phishing instances. Additionally, the Streaming Random Patches (SRP) ensemble secures the highest Cohen's kappa value (79.21%), reflecting its strong overall classification reliability. Regarding model size, the Stacking ensemble maintains the smallest memory footprint, requiring only 11.55 KB, making it the most efficient in terms of storage.

On the other hand, when Min-Max feature scaling is applied, a decline in most performance metrics is observed, except for precision, which sees slight improvements. The Voting ensemble model achieves the highest accuracy (87.28%), recall

(85.40%), F1-score (85.48%), and Cohen's kappa (74.16%), indicating that it adapts well to min-max scaling despite the overall performance reduction. The ADWINBoosting ensemble exceeds others in precision, reaching 91.30%, suggesting that it maintains strong decision boundaries even when the feature values are normalized to a fixed range. The Stacking ensemble continues to be the most memory-efficient, requiring 13.43 KB of storage.

A similar trend is observed in case of no-scaling, which presents the performance of ensemble models when no feature scaling is applied. The Voting ensemble maintains its advantage in accuracy (87.92%), recall (85.04%), F1-score (86.06%), and Cohen's kappa (75.40%), reinforcing its stability across different feature preprocessing conditions. The ADWINBoosting model, once again, records the highest precision (91.30%), indicating its consistent ability to minimize false positives. The Stacking ensemble remains the most compact model, occupying only 9.35 KB of memory, further highlighting its efficiency in resource-constrained environments.

In general, these results underscore the sensitivity of ensemble-based models to feature scaling techniques. While certain models, such as Voting and ADWINBoosting, maintain strong classification performance across multiple scaling settings, others, such as Stacking, optimize for memory efficiency. The variation in performance metrics suggests that selecting an appropriate ensemble method should be guided by the specific trade-offs between accuracy, precision, and computational efficiency required for phishing detection tasks.

V. Conclusion

This study has provided a comprehensive examination of the evolving cybersecurity landscape, particularly in the context of phishing detection. By systematically analyzing incremental machine learning methods and online model selection strategies leveraging multi-armed bandits, valuable insights have been gained that have broad implications for enhancing cybersecurity defenses.

The findings underscore the intricate performance variations exhibited by different machine learning models under various scaling conditions of features. From linear models to tree-based and ensemble methods, each algorithm demonstrated distinct sensitivities to feature scaling adjustments, emphasizing the necessity of tailoring detection methodologies to specific operational contexts. In particular, while some models exhibited resilience to scaling changes in the feature, others displayed significant performance fluctuations, highlighting the critical role of pre-processing techniques in optimizing detection accuracy.

In general, this study contributes to the broader discourse on cybersecurity by offering practical insights into optimizing incremental machine learning approaches and evaluating model selection driven by reinforcement learning for phishing detection. By elucidating the nuanced performance characteristics of various models and selection strategies, this work lays the foundation for the development of more resilient, adaptive, and intelligent cybersecurity frameworks. Ultimately, these advances will strengthen real-time defenses against evolving cyber threats, reinforcing the robustness of modern cybersecurity systems.

Acknowledgment

Machine learning training and evaluation have been performed using the Phoenix High Performance Computing facility at the American University of the Middle East, Kuwait.

References

[1] R. Goenka, M. Chawla, and N. Tiwari, "A comprehensive survey of phishing: mediums, intended targets, attack and defence techniques and a novel taxonomy," *International Journal of Information Security*, vol. 23, no. 2, pp. 819–848, Apr 2024. [Online]. Available: https://doi.org/10.1007/s10207-023-00768-x

[2] T. Stojnic, D. Vatsalan, and N. A. G. Arachchilage, "Phishing email strategies: Understanding cybercriminals' strategies of crafting phishing emails," *SECURITY AND PRIVACY*, vol. 4, no. 5, p. e165, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spy2.165

[3] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 21, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/21/4545

[4] L. Gallo, D. Gentile, S. Ruggiero, A. Botta, and G. Ventre, "The human factor in phishing: Collecting and analyzing user behavior when reading emails," *Computers and Security*, vol. 139, p. 103671, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823005813

[5] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or not phishing? a survey on the detection of phishing websites," *IEEE Access*, vol. 11, pp. 18 499–18 519, 2023.

[6] S. R. Borra, B. Gayathri, B. Rekha, B. Akshitha, and B. Hafeeza, "K-nearest neighbour classifier for url-based phishing detection mechanism," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 14, no. 03, pp. 34–40, 2023.

[7] C. Pires and J. Borges, "Detecting targeted phishing websites for brand protection and cyber defence using computer vision," in *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, 2023, pp. 1–6.

[8] C. C. L. Tan, K. L. Chiew, K. S. Yong, Y. Sebastian, J. C. M. Than, and W. K. Tiong, "Hybrid phishing detection using joint visual and textual identity," *Expert Systems with Applications*, vol. 220, p. 119723, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423002245

[9] A. Garg, B. Sharma, A. Gupta, and R. Khan, "Security of modern networks and its challenges," in *Cyber Security Using Modern Technologies*. CRC Press, 2023, pp. 57–71.

[10] S. A. Afaq, M. S. Husain, A. Bello, and H. Sadia, "A critical analysis of cyber threats and their global impact," in *Computational Intelligent Security in Wireless Communications*. CRC Press, 2023, pp. 201–220.

[11] M. S. Kheruddin, M. A. E. M. Zuber, and M. M. M. Radzai, "Phishing attacks: Unraveling tactics, threats, and defenses in the cybersecurity landscape," *Authorea Preprints*, 2024.

[12] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on url," *IEEE Access*, vol. 11, pp. 36 805–36 822, 2023.

[13] P. Pandey and N. Mishra, "Phish-sight: A new approach for phishing detection using dominant colors on web pages and machine learning," *International Journal of Information Security*, vol. 22, no. 4, pp. 881–891, 2023.

[14] V. Ganganwar, S. I. Hussain, A. Powar, S. Gaur, and A. Kumar, "Using website content for detecting phishing urls: A novel approach," in *Advances in Data-Driven Computing and Intelligent Systems*, S. Das, S. Saha, C. A. Coello Coello, and J. C. Bansal, Eds. Singapore: Springer Nature Singapore, 2024, pp. 87–102.

[15] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672–694, 2021.

[16] E. Gandotra and D. Gupta, "An efficient approach for phishing detection using machine learning," *Multimedia Security: Algorithm Development, Analysis and Applications*, pp. 239–253, 2021.

[17] A. Almomani, M. Alauthman, M. T. Shatnawi, M. Alweshah, A. Alrosan, W. Alomoush, and B. B. Gupta, "Phishing website detection with semantic features based on machine learning classifiers: a comparative study," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 18, no. 1, pp. 1–24, 2022.

[18] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn," *Electronics*, vol. 12, no. 1, p. 232, 2023.

[19] S. Jalil, M. Usman, and A. Fong, "Highly accurate phishing url detection based on machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 9233–9251, 2023.

[20] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, vol. 11, no. 1, pp. 217–242, 2024.

[21] S. Pallaiyah, M. Amir, I. S. Mathew, S. S. Varma, S. Shakeer, M. Rajendran, and J. Govindaraj, "Url phishing detection using machine learning," in *AIP Conference Proceedings*, vol. 3042, no. 1. AIP Publishing, 2024.

[22] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting phishing attacks," *Computers and Security*, vol. 108, p. 102328, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821001528

[23] J. Bharadiya, "Machine learning in cybersecurity: Techniques and challenges," *European Journal of Technology*, vol. 7, no. 2, pp. 1–14, 2023.

[24] X. Li, Y. Zhou, Z. Jin, P. Yu, and S. Zhou, "A classification and novel class detection algorithm for concept drift data stream based on the cohesiveness and separation index of mahalanobis distance," *Journal of Electrical and Computer Engineering*, vol. 2020, p. 4027423, Mar 2020. [Online]. Available: https://doi.org/10.1155/2020/4027423

[25] L. Rutkowski, M. Jaworski, and P. Duda, *Basic Concepts of Data Stream Mining*. Cham: Springer International Publishing, 2020, pp. 13–33.

[26] M. Al-Tarawneh, "Data stream classification algorithms for workload orchestration in vehicular edge computing: A comparative evaluation," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 21, no. 2, pp. 101–122, Jun 2021. [Online]. Available: http://www.ijfis.org/journal/view.html?doi=10.5391/IJFIS.2021.21.2.101

[27] U. Adhikari, T. H. Morris, and S. Pan, "Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4049–4060, 2018.

[28] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[29] J. S. Rojas, A. Pekar, A. Rendón, and J. C. Corrales, "Smart user consumption profiling: Incremental learning-based ott service degradation," *IEEE Access*, vol. 8, pp. 207 426–207 442, 2020.

[30] A. Alkhresheh, M. A. B. Al-Tarawneh, and M. Alnawayseh, "Evaluation of online machine learning algorithms for electricity theft detection in smart grids," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.0131096

[31] D. Jin, S. Chen, H. He, X. Jiang, S. Cheng, and J. Yang, "Federated incremental learning based evolvable intrusion detection system for zero-day attacks," *IEEE Network*, vol. 37, no. 1, pp. 125–132, 2023.

[32] A. GLAVAN and V. CROITORU, "Incremental learning for edge network intrusion detection," *REVUE ROUMAINE DES SCIENCES TECHNIQUES—SÉRIE ÉLECTROTECHNIQUE ET ÉNERGÉTIQUE*, vol. 68, no. 3, pp. 301–306, 2023.

[33] M. A. Shyaa, Z. Zainol, R. Abdullah, M. Anbar, L. Alzubaidi, and J. Santamaría, "Enhanced intrusion detection with data stream classification and concept drift guided by the incremental learning genetic programming combiner," *Sensors*, vol. 23, no. 7, p. 3736, 2023.

[34] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157823000034

[35] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, "An improved elm-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Systems with Applications*, vol. 165, p. 113863, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420306734

[36] A. K. Jain and B. B. Gupta, "Two-level authentication approach to protect from phishing attacks in real time," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1783–1796, Nov 2018. [Online]. Available: https://doi.org/10.1007/s12652-017-0616-z

[37] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen, and R. Ramli, "A comprehensive survey on identification and analysis of phishing website based on machine learning methods," in *2021 IEEE 11th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, 2021, pp. 82–87.

[38] H. F. Atlam and O. Oluwatimilehin, "Business email compromise phishing detection based on machine learning: A systematic literature review," *Electronics*, vol. 12, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/1/42

[39] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: a systematic literature review," *Knowledge and Information Systems*, vol. 64, no. 6, pp. 1457–1500, Jun 2022. [Online]. Available: https://doi.org/10.1007/s10115-022-01672-x

[40] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418306067

[41] C.-Y. Wu, C.-C. Kuo, and C.-S. Yang, "A phishing detection system based on machine learning," in *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*. IEEE, 2019, pp. 28–32.

[42] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153–166, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519300763

[43] M. T. Suleman and S. M. Awan, "Optimization of url-based phishing websites detection through genetic algorithms," *Automatic Control and Computer Sciences*, vol. 53, no. 4, pp. 333–341, Jul 2019. [Online]. Available: https://doi.org/10.3103/S0146411619040102

[44] H. Zuhair and A. Selamat, "Phishing hybrid feature-based classifier by using recursive features subset selection and machine learning algorithms," in *Recent Trends in Data Science and Soft Computing*, F. Saeed, N. Gazem, F. Mohammed, and A. Busalim, Eds. Cham: Springer International Publishing, 2019, pp. 267–277.

[45] J. Stobbs, B. Issac, and S. M. Jacob, "Phishing web page detection using optimised machine learning," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 483–490.

[46] M. Korkmaz, O. K. Sahingoz, and B. Diri, "Detection of phishing websites by using machine learning-based url analysis," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–7.

[47] G. Palaniappan, S. S, B. Rajendran, Sanjay, S. Goyal, and B. B S, "Malicious domain detection using machine learning on domain name features, host-based features and web-based features," *Procedia Computer Science*, vol. 171, pp. 654–661, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920310383

[48] H. Shirazi, S. R. Muramudalige, I. Ray, and A. P. Jayasumana, "Improved phishing detection algorithms using adversarial autoencoder synthesized data," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020, pp. 24–32.

[49] W. Bai, "Phishing website detection based on machine learning algorithm," in *2020 International Conference on Computing and Data Science (CDS)*, 2020, pp. 293–298.

[50] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Engineering Applications of Artificial*

Intelligence, vol. 104, p. 104347, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197621001950

[51] Ö. Kasim, "Automatic detection of phishing pages with event-based request processing, deep-hybrid feature extraction and light gradient boosted machine model," *Telecommunication Systems*, vol. 78, no. 1, pp. 103–115, Sep 2021. [Online]. Available: https://doi.org/10.1007/s11235-021-00799-6

[52] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro, and E. Alegre, "Impact of current phishing strategies in machine learning models for phishing detection," in *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, Á. Herrero, C. Cambra, D. Urda, J. Sedano, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2021, pp. 87–96.

[53] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight url-based phishing detection," *Future Internet*, vol. 13, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/6/154

[54] M. S. Munir Prince, A. Hasan, and F. Muhammad Shah, "A new ensemble model for phishing detection based on hybrid cumulative feature selection," in *2021 IEEE 11th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, 2021, pp. 7–12.

[55] S.-J. Bu and S.-B. Cho, "Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing url detection," *Electronics*, vol. 10, no. 12, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/12/1492

[56] M. Korkmaz, E. Kocyigit, O. K. Sahingoz, and B. Diri, "Phishing web page detection using n-gram features extracted from urls," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–6.

[57] K. V. Samarthrao and V. M. Rohokale, "Enhancement of email spam detection using improved deep learning algorithms for cyber security," *Journal of Computer Security*, vol. 30, pp. 231–264, 2022, 2. [Online]. Available: https://doi.org/10.3233/JCS-200111

[58] M. Dewis and T. Viana, "Phish responder: A hybrid machine learning approach to detect phishing and spam emails," *Applied System Innovation*, vol. 5, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2571-5577/5/4/73

[59] M. Korkmaz, E. Kocyigit, O. K. Sahingoz, and B. Diri, "A hybrid phishing detection system using deep learning-based url and content analysis," *Elektronika ir Elektrotechnika*, vol. 28, no. 5, pp. 80–89, Oct. 2022. [Online]. Available: https://eejournal.ktu.lt/index.php/elt/article/view/31197

[60] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Computing*, vol. 25, no. 6, pp. 3819–3828, Dec 2022. [Online]. Available: https://doi.org/10.1007/s10586-022-03604-4

[61] P. Bountakas and C. Xenakis, "Helphed: Hybrid ensemble learning phishing email detection," *Journal of Network and Computer Applications*, vol. 210, p. 103545, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804522001862

[62] T. Wen, Y. Xiao, A. Wang, and H. Wang, "A novel hybrid feature fusion model for detecting phishing scam on ethereum using deep neural network," *Expert Systems with Applications*, vol. 211, p. 118463, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742201555X

[63] E. Zhu, K. Cheng, Z. Zhang, and H. Wang, "Pdhf: Effective phishing detection model combining optimal artificial and automatic deep features," *Computers & Security*, vol. 136, p. 103561, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823004716

[64] F. S. Alsubaei, A. A. Almazroi, and N. Ayub, "Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics," *IEEE Access*, vol. 12, pp. 8373–8389, 2024.

[65] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414001481

[66] "UCI Machine Learning Repository — archive.ics.uci.edu," http://archive.ics.uci.edu/dataset/379/website+phishing, [Accessed 03-04-2024].

[67] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem *et al.*, "River: machine learning for streaming data in python," 2021.

[68] T. Vasiloudis, F. Beligianni, and G. De Francisci Morales, "Boostvht: Boosting distributed streaming decision trees," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 899–908. [Online]. Available: https://doi.org/10.1145/3132847.3132974

[69] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, and B. Pfahringer, "Efficient online evaluation of big data stream classifiers," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 59–68. [Online]. Available: https://doi.org/10.1145/2783258.2783372