

EnLeM: An Ensemble Learning-based Model for Detecting Phishing Websites

MOST NILUFA YEASMIN¹, MD ABU RUMMAN REFAT², BIKASH CHANDRA SINGH^{1,5}, ZULFIKAR ALOM³, ZEYAR AUNG⁴, (Senior Member, IEEE), and MOHAMMAD ABDUL AZIM³, (Member, IEEE)

¹Department of Information and Communication Technology, Islamic University, Kushtia-7003, Bangladesh (e-mails: nilufa.yeasmin5284@gmail.com; bikash.singh@ice.iu.ac.bd)

²Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka-1216, Bangladesh (e-mail: refatiuice@gmail.com)

³Department of Computer Science, Asian University for Women (AUW), Chattogram-4000, Bangladesh (e-mails: zulfikar.alom@auw.edu.bd, azim@ieee.org)

⁴Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, UAE (e-mail: zeyar.aung@ku.ac.ae)

⁵School of Cybersecurity, Old Dominion University, Norfolk, VA 23508, USA

Corresponding author: Bikash Chandra Singh (e-mail: bikash.singh@ice.iu.ac.bd).

ABSTRACT

Phishing, an unlawful practice, involves tricking individuals into revealing private data like user IDs, bank details, and passwords. The surge in fraud is tied to increased deception, impersonation, and advanced online attacks. A more potent phishing detection approach is crucial due to escalating global phishing threats. While methods like Heuristics, Signatures, and Visual similarity attempt to detect phishing sites, Machine Learning (ML) and Deep Learning (DL) shine in cybersecurity for learning from data, offering insights, and forecasting. Still, solo ML algorithms face real-world limitations with complex data, while DL surpasses traditional ML in performance but requires more data and time. This paper introduces “EnLeM,” an Ensemble Learning Model that excels in precision compared to individual ML/DL. To tackle computational efficiency, we employ Univariate feature selection, yielding promising results in comparison to DL models. Furthermore, we assess our model against five conventional ML-based classifiers and two DL-based counterparts, evaluating performance metrics pre/post feature selection, along with execution times. The experimental outcomes underscore the efficacy of EnLeM, showcasing outstanding, scalable, and consistent performance.

INDEX TERMS Phishing, Univariate feature selection, Machine learning, Ensemble algorithm, Deep learning.

I. INTRODUCTION

WITH widespread internet access and the advent of innovative services, a majority of individuals have transitioned to online platforms for tasks such as shopping, banking, and various services, thereby minimizing the need for physical queues. However, in contrast, cybercriminals exploit this trend to target victims and illicitly obtain funds. The prevalence of phishing kits further simplifies this process, enabling even those with limited technical expertise to initiate phishing campaigns. Phishing, a social engineering attack, involves scammers masquerading as trustworthy sources to extract personal information. They aim to connect through emails, calls, or texts, accessing victims’ financial accounts. Phishing attacks are on the rise, with attackers observing victims’ online actions and bypassing security measures [1]. The FBI cites phishing as the primary cybercrime in 2020,

and instances of repeated attacks have increased. Monthly phishing attacks surged by 65%, spawning 1.5 million sites in 2018 [2]. A notable breach exposed 80 million Anthem Healthcare customers’ data in 2014 [3].

Amid the COVID-19 pandemic, remote work has strained cybersecurity resources for many. Exploiting this, attackers launch phishing attempts via emails and bogus websites. Google’s Threat Analysis Group [4] has pinpointed attacker groups employing COVID-19 themes for malware and phishing schemes. Attackers employ diverse tools to replicate websites for phishing purposes. Distinguishing phishing sites can be challenging due to elements like copyright, anchor links, logos, or genuine website keywords. Various approaches [5] exist to detect phishing sites, each with its advantages and drawbacks.

The preeminent method for identifying phishing is re-

ferred to as *Heuristics-based approach* and *Signature-based method*. The heuristics-based approach serves as a conventional means of identifying fraudulent web pages. Its primary use lies in uncovering unauthorized sites [6]. Despite its accuracy-enhancing role, this technique is intricate and financially demanding to implement [7]. Similarly, the signature-based method is a critical tool for spotting phishing sites. This method relies on a database containing blacklisted phishing sites. A limitation of this approach is that it fails to identify newly generated phishing sites not yet in the database [8].

An additional method for detecting phishing websites involves the *Visual similarity-based approach*, which relies on an assortment of images, logos, text formatting, and HTML tags from a database. However, a notable drawback of this technique is its demand for substantial computational resources for effective image processing [7].

The employment of a *Machine Learning (ML) approach* is another avenue for identifying phishing websites. Despite the various ML techniques employed to distinguish between phished and legitimate sites, conventional ML models often fall short in detecting phishing webpages as effectively as browser security indicators. A mere 23% of these models rely on webpage content to ascertain legitimacy [9]. Furthermore, the general population may struggle to differentiate between the padlock icon in a browser and the actual webpage content. Additionally, ML-based detection methods come with computational burdens, and their sluggish processing can contribute to increased latency in identifying phishing websites [7]. Certain researchers have highlighted limitations in current ML models, including increased false positives, decreased detection rates, and classifier inefficiencies [10], [11]. Furthermore, many traditional ML models struggle with unstructured and streaming data due to the diverse nature of real-world data. Such data can contain noise, irrelevant attributes, redundant features, and anomalies [12].

Other researchers have explored the automatic identification of malicious webpages using *Deep Learning (DL)-based models*. Performance varies across different DL-based classification algorithms, influenced by training data and features. While numerous features can enhance DL models' performance, their slower speed occasionally hinders efficient phishing website detection. Yet, training DL models necessitates substantial data and resources like GPUs and parallel processing to expedite learning and classification [13].

Due to the constraints of current models (including Heuristics-based approaches, Signature-based methods, Visual similarity-based approaches, as well as ML-based and DL-based approaches) in effectively identifying phishing websites, expecting users to accurately discern between phishing and legitimate sites in real-time becomes impractical.

We introduce a novel model, "EnLeM," based on ensemble learning, which surpasses individual machine learning and deep learning algorithms in effectively distinguishing between phishing and legitimate websites. EnLeM involves training multiple machine learning models and combining

their outcomes to address specific issues. In this context, our EnLeM model improves accuracy and maintains consistency before and after feature selection. To mitigate model computation time concerns, we utilize the univariate feature selection method, yielding promising outcomes. Additionally, we compare the accuracy of the EnLeM model with individual ML-based and DL-based models.

The main contributions of our work can be summarized as follows:

- 1) EnLeM, our new ensemble learning architecture, offers a cutting-edge approach to detecting emerging phishing websites. Addressing efficiency concerns, we utilized the publicly available UCI dataset to train and evaluate the model. This allowed us to determine the top-performing model based on accuracy, training, and testing times.
- 2) Creating a fast and accurate predictive model is challenged by the curse of dimensionality. Overcoming this, we employed univariate feature selection to eliminate irrelevant and noisy features, yielding positive results across all classifiers.
- 3) Our model combines multiple individual models using univariate feature selection to detect phishing and non-phishing URLs, resulting in minimal error rates and exceptional, scalable, and consistent performance metrics.
- 4) Additionally, we compare the experimental outcomes of the EnLeM model against five individual ML-based models and two DL-based models. This evaluation helps determine the most accurate model for the UCI dataset on phishing website detection.

The rest of the paper is organized as follows. A short outline of the related literature is presented in section II. Section III proposes an overall system framework for phishing website detection. In it, the proposed EnLeM classification model is emphasized in Section III-E. Section IV covers the performance metrics and experimental design that is used in our work and illustrates the experimental results. Finally, Section V concludes this paper and shows the future work for this project.

II. LITERATURE REVIEW

Phishers attempt to snatch personally identifiable information, banking, credit card details, and user passwords. Phishing websites are look-alike websites, and attackers gather users' personal information. Cybercriminals successfully gain access to personal data via emails, text messages, direct messages on social media, or video games.

There are many approaches, namely, the Signature-based method, the Visual-based approach, the machine learning-based approach, and the deep learning-based approach. Some approaches have been proposed and exhibited for detecting phishing websites.

Salihovic et al. [14] used models detecting phishing websites named Artificial Neural Networks, Logistic Regression, Random Forest (RF), Support Vector Machine

(SVM), k -Nearest Neighbor, and Naive Bayes. First, they used the UCI phishing websites dataset, with 31 attributes. Then, they applied two feature selection methods. The first one is BestFirst+CfsSubsEvaluation, and the one is the Ranker+Principal Components feature selection optimizer. In the first experiment, they achieved the highest accuracy (97.33%) in RF. The first optimizer reduced ten attributes in the phishing dataset with a 1.53% decreased accuracy on average. In the second experiment, the authors reduced only one feature, and accuracy increased in RF, whereas the accuracy decreased by 0.09% in SVM. In addition, the authors used the spam emails dataset, where the RF provides the best performance.

Yuan et al. [15] proposed a method to detect phishing websites based on features from URLs and web page links. The authors applied the Deep Forest model, resulting in a positive rate of 98.3% and a false alarm rate of 2.6%. Primarily, they used an effective strategy based on search operators via search engines to find the phishing targets' accuracy of 93.98%. Jain et al. [16] shortlisted 19 most important features and analyzed several features of phishing web pages. They used SVM, RF, neural network, logistic regression, and Naive Bayes (NB) to classify phishing webpages and obtained approximately 99% accuracy.

Gu et al. [17] introduced a brilliant and automatic system for detecting phishing web pages. They used a Naive-Bayes classifier to detect phishing websites and analyzed the feature URL. They reclassified using SVM for ambiguous web pages. They proved that their system provides high accuracy with less time for phishing site detection. Moghimi and Varjani [18] proposed a model that used a combination of SVM and decision tree (DT) models where the two models serve two different purposes. For example, SVM is used for training purposes. The DT generates the rules for detecting phishing websites targeting the banking domain.

In addition, many researchers have proposed and developed deep learning-based models for phishing website detection. To illustrate, Yerima et al. [19] proposed a DL approach named Convolutional Neural Networks (CNN) for detecting phishing webpages. They used the UCI phishing websites dataset of 6,157 genuine and 4,898 phishing websites. The authors compared their proposed DL-based CNN model with traditional ML models. They proved that their proposed model achieved the highest accuracy compared to conventional ML models.

Zhang et al. [20] proposed multilayer perceptron neural networks for detecting phishing emails and evaluated the effectiveness and efficiency of this model. They compared their proposed model with different classification algorithms such as NN, SVM, and DT, Naïve Bayes. Their proposed model gave the highest accuracy and recall value, around 95%, and demonstrated that their model was one of the best for detecting phishing emails. Rao and Pais [21] proposed a novel classification model based on a heuristic dataset. They trained eight ML models, including Random forest J48, logistic regression, Bayes network, multilayer perceptron,

minimal sequential optimization, AdaBoostM1, and SVM, and proved that the RF algorithm outperformed with an accuracy of 99.31%. The authors attempt to find the best classifier for phishing detection, consequently using different (orthogonal and oblique) random forest classifiers. The Principal component analysis Random Forest (PCA-RF) provided the highest accuracy, around 99.55%.

Aksu et al. [22] studied comparatively among traditional ML approaches and DL approaches (Neural networks, SVM, DT, DNN, Stacked auto-encoder technique (SAE)) at phishing detection. They proved that the DL technique is the best for phishing detection with an accuracy of 80%.

Zhang et al. [23] proposed a model for detecting Chinese phishing e-business websites based on URL and website content. The authors included fifteen domain-specific features for detecting phishing attacks on Chinese e-business websites. They evaluated them with four traditional ML models: SMO, Naïve-Bayes, RF, and Logistic Regression. Their experiment said the Sequential minimal optimization (SMO) algorithm best detects phishing sites with about 95.83% accuracy and 95.58% f_1 -score. However, their experiment had some limitations, such that it targeted only one domain of phishing sites, i.e., it was not working efficiently with non-Chinese websites.

Whittaker et al. [24] proposed a blacklisting technique based on known phishing websites; subsequently, they extracted standard features from their dataset for detecting phishing sites. The authors used an RF classifier in their detection. They proved their classifier achieved the best accuracy with noisy live phishing data and obtained less than 0.1% false positive rate. Nevertheless, their approach works when the websites have been published or are visible; otherwise, it does not work well.

Ubung et al. [25] proposed an ensemble learning (EL) model based on majority voting for detecting phishing websites and used feature selection techniques and comparison with traditional ML models. They proved that their proposed model provided a better accuracy rate than the current technologies for detecting phishing websites, and the model accuracy was around 95.4% and f_1 -score was about 94.7%. However, their analysis had some limitations, like that the authors of this paper focused only on improving accuracy; others needed to be considered more balanced.

Toolan et al. [26] used the C5.0 decision tree algorithm and an ensemble of other classifiers (k -Nearest Neighbour, SVMs, Naive Bayes, Linear Regression) for classifying emails into Phishing and non-Phishing. They used only 8000 emails with five features, and their systems classify that around 50% of emails were phishing and the existing 50% were legitimate. The average accuracy and precision of the C5.0 model are about 97.15% and 98.56%, as well as average Ensemble accuracy is 93.68%. The paper also discussed the advantages of recall boosting classifiers which provided better accuracy than C5.0 and Ensemble classifiers.

Table 1 summarizes all references in terms of phishing website detection. However, this research work proposed a

novel approach based on ML-based techniques. Our proposed approach is very efficient for detecting phishing URLs compared to the others.

TABLE 1. Summary of existing works on phishing website detection and our proposed method.

Researcher	Methodology
Salihovic et al. (2019) [14]	ML with feature selection
Yuan et al. (2018) [15]	Deep forest
Jain et al. (2018) [16]	ML with feature selection
Gu et al. (2013) [17]	ML
Moghim and Varjani (2016) [18]	ML
Yerima et al. (2020) [19]	CNN (DL)
Zhang et al. (2012) [20]	MLP neural networks
Aksu et al. (2019) [22]	ML and DL
Zhang et al. (2014) [23]	ML with feature selection
Whittaker et al. (2010) [24]	Blacklist technique
Ubung et al. (2019) [25]	Ensemble learning (EL) with feature selection
Toolan et al. (2009) [26]	C5.0 and EL
Proposed approach (EnLeM)	Majority voting based-EL on ML methods with univariate feature selection

III. PROPOSED SYSTEM FRAMEWORK

The overall system architecture of the proposed EnLeM model is shown in **Figure 1**. This research uses the UCI dataset, which has phishing and legitimate URLs. To improve performance metrics, we need data pre-processing.

A. DATASET DESCRIPTION

This study used the UCI Phishing website dataset [27] to evaluate the performance metrics of the proposed phishing website detection model and other models, including six ML-based models and two DL-based algorithms. This dataset is explained as follows;

The feature description can be found in [28] and [27]. The dataset comprises 30 features with 11055 instances extracted from 6157 legitimate web pages and 4898 phishing web pages. The rationale behind choosing the dataset is recent and available in the public domain. The dataset's attributes and corresponding values are summarized in **Table 2**.

B. DATA PRE-PROCESSING

The success of learning algorithms depends on the data quality we used to solve our classification problem. The pre-processing of data is an inevitable step for achieving better performance metrics. Although there are several data pre-processing steps [29], [30], we use data normalization, data de-noising, and data extraction. Using data normalization, we reduce the bias in those attributes in our dataset with higher numerical contributions and fetch all features in a typical range. Due to data normalization, we can ensure all features have an equal numerical contribution, but all feature is not equally important for classification. Some features are highly relevant, while others are irrelevant and redundant, increasing the complexity of learning algorithms. However, the vast amount of irrelevant features leads to the curse of

TABLE 2. Features description of collected dataset

Attribute	Type	Possible Values
Having IP Address	Numeric	-1, 1
URL Length	Numeric	1,0,-1
Shortening Service	Numeric	1, -1
Having At Symbol	Numeric	1,-1
Double slash redirecting	Numeric	-1,1
Prefix Suffix	numeric	-1,1
Having Sub Domain	Numeric	-1,0,1
SSLfinal State	Numeric	-1,1,0
Domain registration length	Numeric	-1,1
Favicon	Numeric	1,-1
Port	Numeric	1,-1
HTTPS token	Numeric	-1,1
Request URL	Numeric	-1,1
URL of Anchor	Numeric	-1,0,1
Links in tags	Numeric	1,-1,0
SFH (server form handler)	Numeric	-1,1,0
Submitting to email	Numeric	-1,1
Abnormal URL	Numeric	-1,1
Redirect page	Numeric	0,1
On Mouse Over (using to hide link)	Numeric	1,-1
Right Click	Numeric	1,-1
Using pop-up window	Numeric	1,-1
Iframe	Numeric	1,-1
Age of domain	Numeric	-1,1
DNS Record	Numeric	-1,1
Web traffic	Numeric	-1,0,1
Page Rank	Numeric	-1,1
Google Index	Numeric	-1,1
Links pointing to page	Numeric	1,0,-1
Statistical report	Numeric	-1,1

dimensionality; we solve the problem by the feature selection method that is discussed later in Section III-C.

We extract one feature and provide this name as a label. We assign a label *legitimate* when the result value is 1, and the label is called *phishing* if the result value is 0. The extracted features of legitimate and phishing URL datasets in the UCI dataset are concatenated without shuffling. We must shuffle our dataset in training and testing datasets.

C. FEATURE SELECTION

Feature selection is the process of selecting a subset of relevant features or attributes for various reasons, such as easier to interpret by users, short training time, avoiding unnecessary attributes, improving data compatibility, etc. Researchers used many feature selection methods to improve their model compatibility, including univariate selection, recursive feature elimination, principal-component analysis, etc.

In this work, we have used the univariate features selection method, where univariate feature selection selects the best essential features based on univariate statistical tests. We use Select-*k*-Best, which selects features according to the *k* highest scores, and different statistical tests can be used in this selection method. The univariate test uses the Mutual Information (MI) approach ("*mutual_info_classif*" in scikit-learn) and selects the top 20 features from our data set.

MI between two random variables is a non-negative value, which measures the dependency between the variables. The

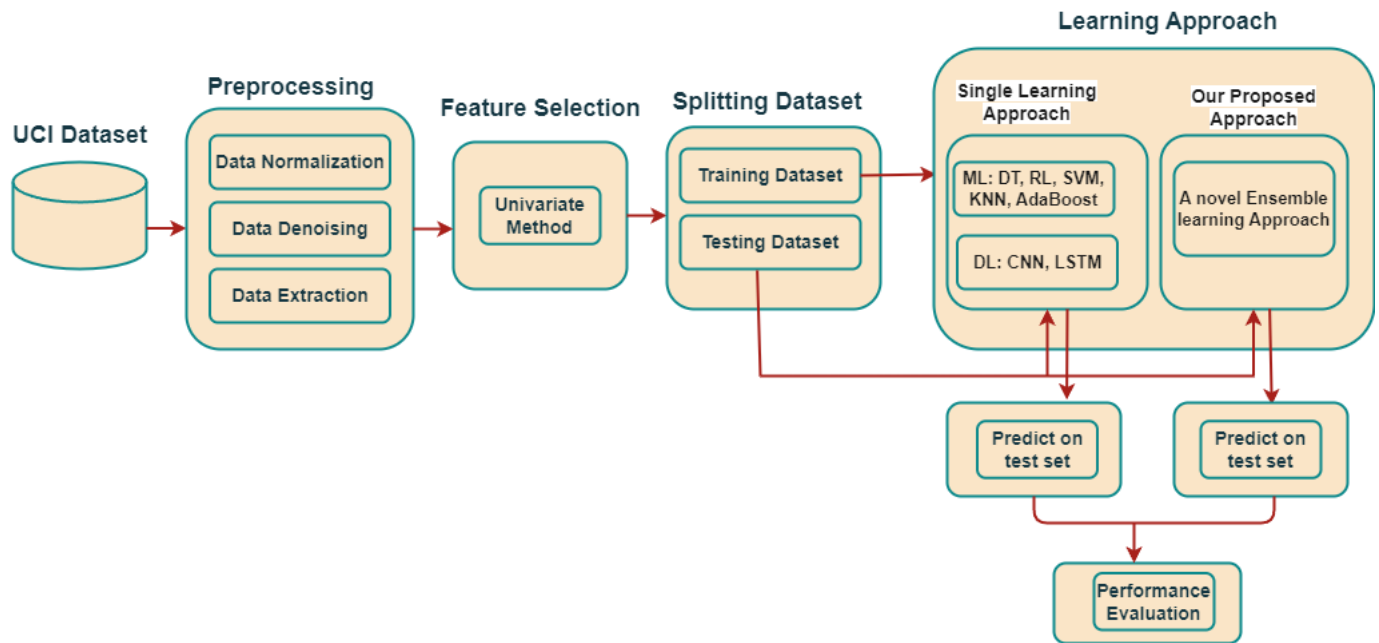


FIGURE 1. Overall system framework of the proposed method (EnLeM).

value of MI is zero if and only if the two random variables are independent. Otherwise, the value of MI is equal to one. If X and Y denote the two random variables, then:

$$I(X; Y) = H(X) - H(X|Y) \quad (1)$$

where, $I(X; Y)$ is the mutual information of X and Y , $H(X)$ denotes the entropy of X , $H(X|Y)$ is the conditional entropy for X given Y .

For our dataset, the MI approach is used to choose the top 20 features among the 30. **Figure 2** shows that the top 20 features are more important for classification purposes.

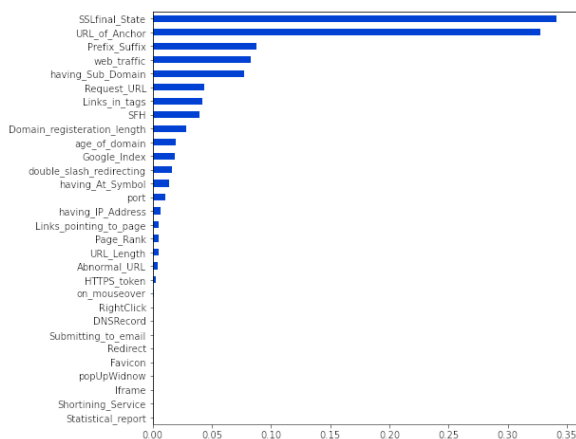


FIGURE 2. Graphical representation of mutual information of attributes in feature selection.

D. DATASET SPLITTING

In this stage, we use the k -fold cross-validation technique, which is a strategy for enhancing the holdout approach to split the UCI dataset. Instead of selecting a single dataset for training and testing, it divides it into k subsets and applies the holdout method to each subset k times. The steps for k -fold cross-validation are given below.

- 1) It divides our dataset into k folds or k chunks.
- 2) It creates k folds from our dataset, and for each fold, build a model using the other $k - 1$ folds. Then, it assesses the model's performance on the k -th fold by predicting the labels (i.e., phishing vs. legitimate) and then evaluating the accuracy.
- 3) This procedure is repeated until all k folds have been put to the test using separate test sets.
- 4) The findings are noted, and the anticipated accuracy average is then calculated and given as the test measure for the model that is currently being evaluated or used.

For each of our classifiers, the k -fold cross-validation has been taken into account. We experimented with several k values and decided that $k = 10$ was the best choice for our experiment.

E. PROPOSED APPROACH: ENLEM

This article introduces an approach based on hard voting ensemble learning that predicts the class labels with the highest sum of the votes from learning models. Our proposed method, **EnLeM**, is an extended voting-based ensemble learning (EL) approach.

The concept of ensemble learning combines multiple models that solve a particular computational intelligence problem

using combined decisions that provide multiple classifiers. Essentially, EL is used to improve the performance of (classification, prediction, function approximation, etc.). Numerous applications of EL include decision-making, optimal feature selection, data fusion, incremental learning, non-stationary learning, and error-correcting [46]. Dasarathy et al. [47] were among the first to propose the EL approach in a divide-and-conquer fashion, partitioning the feature space using two or more classifiers.

For example, someone planning to purchase a new computer would like to go to a computer showroom and buy one the salesperson shows him. Most people say the answer no because people are likely to ask their friends or colleagues or go to online portals about showing the reviews, then they decide which one is best for them. Instead of directly choosing, people try to know information before purchasing one computer. Ensemble learning resembles people who want to buy a new computer.

The voting ensemble is a distinct EL combining multiple classifiers. More than two models are created, and then the preceding models are wrapped using a voting mechanism to combine with all models' predictions. The majority voting effort combines multiple ML models' outcomes and may be used for classification and regression. While used in a classification problem, it sums the predictions of each label. It then selects the model, a majority vote known as *Classification Voting Ensemble*.

The EnLeM method introduced a way that gradually helps detect phishing websites with promising performance metrics. As a version of the voting-based EL model, it also aggregates more than one classifier, and every model makes a prediction; this model chooses the one that receives the highest votes. Thus, this approach helps predict the class labels (e.g., phishing/non-phishing). Suppose *Classifier₁* predicts an unlabeled URL as a phishing URL, whereas *Classifier₂* predicts the same URL as a benign website, and *Classifier₃* predicts the URL as a phishing URL. Generally, it can be very complex to conclude which label is appropriate for this URL. This situation will increase the error rate, providing low-performance metrics with high variance. The details of the proposed approach are illustrated as follows.

In the EnLeM approach, three different classifiers with the training dataset (i.e., labeled URLs) are created, and every individual classifier provides its prediction label on the testing dataset (i.e., unlabeled URLs). Afterward, every classifier provides a label, whether phishing or benign. However, this step works to find the label that provides most classifiers and make the final decision.

1) Voting Algorithm

As illustrated in **Algorithm 1**, we set features ($f_1, f_2, f_3, \dots, f_{30}$) to our training and testing dataset. Initially, it has 30 features, including 6157 legitimate web pages and 4898 phishing web pages (line 1). Moreover, we diminish a subset of irrelevant features or attributes for various reasons, as we can see in Section III-C. To build our proposed model, three

base classifiers, namely, *Decision Tree*, *Random Forest*, and *k-Nearest Neighbour* are constructed using the labeled URLs (for instance, L_{web}). Theoretically, we could use any number of constituent classifiers (i.e., more than three) in our EL approach. But, here, we use only three as a proof of concept.

Additionally, the classifier predicts the labels of the unlabeled URLs (for example, U_{web}) that show from the line (6-15). Lastly, a label is a voting base designed by aggregating each class's label (i.e., phishing, non-phishing). In **Algorithm 1**, the label returned for U_{web} is associated with the maximum votes that provide the three different classifiers.

Suppose there is a webpage, $w = \text{"https://fmovies.ps"}$. It has 30 different attributes such as "Having IP Address = -1", "URL Length = 1" "Shortening Service = 1", "Having At Symbol = 1", "Double slash redirecting = -1", "Prefix Suffix = -1", "Having Sub Domain = -1", "SSLfinal State = -1", "Domain registration length = -1", "Favicon = 1", "Port = 1", "HTTPS token = -1", "Request URL = 1", "URL of Anchor = -1", "Links in tags = 1", "SFH (server form handler) = -1", "Submitting to email = -1", "Abnormal URL = -1", "Redirect page = 0", "On Mouse Over (using to hide link) = 1", "Right Click = 1" "Using pop-up widow = 1", "Iframe = 1", "Age of domain = -1", "DNS Record = -1", "Web traffic = -1", "Page Rank = 1", " Google Index = 1", "Links pointing to page = 1", and "Statistical report = -1". Assume that the *Decision Tree* predicts that this webpage is phishing, another classifier *Random Forest* predicts that it is a legitimate webpage, and the *k-Nearest Neighbor* one predicts that as a phishing webpage. Consequently, the EnLeM model predicts that it is a phishing webpage according to the **Algorithm 1**.

2) Base Classifiers

Here, we briefly describe the three constituent base classifiers (DT, RF, and *k*-NN) in our EnLeM model.

(i) Decision Tree (DT)

Decision Tree (DT) [53] is a flowchart constructed through an algorithmic approach that identifies the ways to split a dataset based on different conditions. It is easy to handle categorical and continuous data. The model is easy to read and interpret. The generated rules are human-understandable. Nonetheless, its unstable nature and high computational costs are the drawbacks.

(ii) Random Forest (RF)

Bagging (bootstrap aggregation) is an intuitive, straightforward ensemble-based technique, providing surprisingly good performance [48]. In this approach, multiple instances of the base classifier model are used for classification. Firstly, it creates multiple bootstrap samples from the original dataset, so each bootstrap sample works as a new independent dataset. The individual classifiers are trained with the bootstrapped samples. Finally, the outputs of the individual models are combined by averaging or voting and create one output with minor variance.

Algorithm 1: Proposed ensemble learning approach.

Input: L_{web} the set of labeled in training data and U_{web} the set of unlabeled in testing data from the UCI Phishing websites dataset.

Output: 0 if a website is phishing; 1 if a website is legitimate.

- 1 Let (f_1, f_2, f_3, \dots) be the features set of L_{web} and U_{web}
- 2 Let N be the set of ML classifiers used to build the proposed ensemble approach. Here we set $N = 3$
- 3 $Classifier_1 =$
 $Decision_Tree(L_{web}(f_1, f_2, f_3, \dots))$
- 4 $Classifier_2 =$
 $Random_Forest(L_{web}(f_1, f_2, f_3, \dots))$
- 5 $Classifier_3 =$
 $K_Nearest_Neighbour(L_{web}(f_1, f_2, f_3, \dots))$
- 6 **for** $F \in U_{web}$ **do**
- 7 Let $X_L(Phishing)$, $X_L(Legitimate)$ be initialized to zero.
- 8 **for** $j \in \{1, \dots, N\}$ **do**
- 9 $Predicted_Label_for_F_j =$
 $Classifier_j(F(f_1, f_2, f_3, \dots))$
- 10 Let $X_L^j(Phishing)$ be the label of F being predicted as a phishing website.
- 11 Let $X_L^j(Legitimate)$ be the label of F being predicted as a legitimate website.
- 12 $X_L(Phishing) =$
 $X_L(Phishing) + X_L^j(Phishing)$
- 13 $X_L(Legitimate) =$
 $X_L(Legitimate) + X_L^j(Legitimate)$
- 14 Return: The label corresponding to
 $Max(X_L(Phishing), X_L(Legitimate))$

Random Forest (RF) [56], [57] is an ensemble of decision trees that uses a greedy method to get the best split. It decreases the variance and overfitting of the model. It also provides information about significant features from the dataset. However, it is difficult for humans to interrupt. The method also exhibits poor performance on imbalanced data.

(iii) k -Nearest Neighbor (k -NN)

k -NN [54], [55] selects the k -closest examples from the dataset. It is intuitive and straightforward to understand. New data can be added seamlessly. Unfortunately, this method does not work well with large datasets with high dimensions. It needs feature scaling.

F. BENCHMARKING

Following the spirit of various research articles, we present five machine learning models, including DT, RF, AdaBoost, SVM, and k -NN, and two deep learning-based models, namely, deep learning-based CNN model and long short-term memory (LSTM) models for comparison (benchmarking)

against the proposed classifier (EnLeM). If the ML-based and DL-based classifiers individually predict the labels as either phishing or legitimate, then each classifier is called a single-view learning model. However, individual learning approaches can only make up some types of applications. However, multiple classifiers take multiple decisions and can predict unlabeled data for predicting class.

Bagging-based ensemble learning allows more than two weak models, aggregates their predictions, and provides exemplary accuracy. However, it often results in high bias and underfitting when not modeled correctly [31]. Moreover, boosting-based ensemble learning is challenging because of its high complexity.

1) Tradition Machine Learning (ML) Methods

A comprehensive list of machine learning approaches was devised and used to predict phishing emails or websites [32], [33]. Nevertheless, the outcomes of these machine-learning approaches depend on the dataset's characteristics. Therefore, no perfect model performs best for all problems [34].

For instance, Yadav et al. [35] employed decision trees (J48), random forest, and logistic regression while about 99% precision with random forest algorithm best differentiates between phishing and ham emails.

In another work, Kolla et al. [36] used several machine learning algorithms for predicting phishing URLs, such as Decision trees, Random forests, Multilayer Perceptions, Support Vector Machines, and XGBoost. In this study, the random forest provided 90% accuracy, which was the highest. Based on phishing detection, for instance, in this paper [37], they only trained the decision tree model and got approximately 90% accuracy, SVM with 95.66% accuracy in [38]. However, Ojewumi et al. used three ML algorithms while they got k -NN at 93.39%, SVM at 91.74%, and Random Forest at 98.35% accuracy [39]. All of these findings represent variability in the scores of performance metrics across several studies.

Therefore, in our study, we experimented with five popular machine learning algorithms that have been mostly used for phishing website detection, including Decision Tree (DT), Random Forest (RF), AdaBoost Classifier, Support Vector Machine (SVM), and k -Nearest Neighbour (k -NN). Their concepts, advantages, and disadvantages are briefly described below.

(i) Decision Tree (DT)

Decision Tree (DT) was described above in Section III-E2(i).

(ii) Random Forest (RF)

Random Forest (RF) was detailed in Section III-E2(ii).

(iii) AdaBoost

Boosting [49] is a meta-algorithm that can be used for model averaging. It is mainly used for classification problems and sometimes for regression. Boosting creates weak

classifiers, and the sequence of models is created iteratively. Each model is trained on an entire data set, boosting attempts to add higher weights that need to be misclassified or better estimated by the previous model. When their accuracy scores and the weight of the output, all the sequences of models are combined using voting (for classification) or averaging (for regression) to create a final estimation.

AdaBoost [58] is an ensemble of decision trees with only one split (one level). It is easy to program with high speed and flexible to combine with other models. However, the base classifier is very weak and vulnerable to uniform noise.

(iv) Support Vector Machine (SVM)

SVM [54] selects the best hyper-plane that maximizes separability between classes. It is robust against overfitting. The model can solve any complex problem with a perfect kernel function. However, it is memory intensive and needs a longer training time.

(v) k -Nearest Neighbor (k -NN)

k -NN was briefly explained in Section III-E2(iii).

2) Deep Learning (DL) Methods

In addition to the traditional machine learning methods, deep learning (DL) methods become widely used recently. Here, we briefly describe two of the DL methods that we use for benchmarking, namely 1D-CNN and LSTM.

(i) Convolutional Neural Network (CNN)

Artificial Neural Networks (ANNs) are dynamically developing in every field. CNN evolved from artificial neural networks. The first CNN to be similar to today's structure was the Neocognitron network [40]. The first real CNN network, named the LeNet 5 network, was widely used [41] and specially solved the problem of recognizing handwritten numbers. The basic architecture of the CNN classifier can be learned from the paper [42].

The CNN model also used a one-dimensional structure for a processed data set named 1D-CNN [19]. A particular segment may derive an exciting feature from the overall data set that time 1D-CNN models work efficiently. Some significant applications of 1D-CNN automatic speech recognition are real-time electrocardiogram (ECG) monitoring, vibration-based structural damage detection in civil infrastructure, and high-power multilevel converters [43]. We used the 1D-CNN model, which contains a convolutional layer, a pooling layer, and one final fully connected layer with an activation function. We classified the UCI data set into two final classes using 1D-CNN: phishing or legitimate. We also used the ReLu activation function in this CNN model to reduce vanishing and exploding gradient issues. We used another classical non-linear activation function, such as sigmoid, which is more efficient for training massive data.

(ii) Long Short Term Memory (LSTM)

LSTM is a particular type of recurrent neural network (RNN) used in DL proposed by S. Hochreiter et al. [44] deals with the vanishing gradient problem for introducing Constant Error Carousel (CEC) units during the time between 1995 and 1997. Unlike standard feed-forward neural networks, it has feedback connections. Some applications of LSTM are language modeling, handwriting recognition, speech recognition, image processing, music generation, and anomaly detection. LSTM has several units like a cell, an input gate, an output gate, and a forget gate. For more details about the basic architecture of the LSTM model [45].

IV. EXPERIMENTAL EVALUATION

A. EXPERIMENTAL DESIGN

We used various ML and DL libraries throughout the experiments, including Scikit-learn, Keras, Numpy, and Pandas. As described above in Section III-F, five individual and ensemble ML models were used: DT (individual), RF (ensemble), Adaboost (ensemble), SVM (individual), and k -NN (individual), as well as two DL algorithms: 1D-CNN and LSTM. We have used the binary cross-entropy [50] function as a loss function represented in Equation 2 and employ gradient-based Adam optimization [51] to minimize the loss for CNN and LSTM.

$$Loss = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log y'_i + (1 - y_i) \cdot \log(1 - y'_i) \quad (2)$$

where y_i acts as desired output and y'_i considers as predicted output of i^{th} data.

All training is carried out using the default settings of the defined libraries for Machine Learning, an initial learning rate of 0.001, and a mini-batch size of 100 for DL. Our proposed techniques were developed at Google Colab using a 12 GB NVIDIA Tesla K80 GPU and a 2.3 GHz Xeon hyper-threaded processor.

B. PERFORMANCE METRICS

After cross-validating our proposed approaches, we entail the tools to evaluate the performance of our model. This research evaluates the performance of the models using a set of commonly used evaluation metrics for classification problems. The metrics are precision, recall, F1-score, and accuracy [52]. Table 3 defines the metric used to evaluate the performance of the classifiers. Precision is defined as the number of true positives (TP) divided by the sum of true positives and false positives (FP). The recall is defined as the number of true positives divided by the sum of true positives and false negatives (FN). At the same time, the F1-score is defined as the harmonic mean of precision and recall. Finally, accuracy is defined as the number of correct predictions divided by the total number of predictions.

C. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, a comparative experimental investigation is carried out to determine the classification approach's

TABLE 3. Evaluation metrics definitions. (TP = true positives; FP = false positives; TN = true negatives; FN = false negatives.)

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ \text{F1-score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ \text{Accuracy} &= \frac{TP+TN}{TP+FP+TN+FN} \end{aligned}$$

performance. We use the most widely used cross-validation technique, k -fold (where $k = 10$ is set for the experimentation). Cross-validation on a k -fold scale divides the dataset into k -folds, with each fold serving as a performance test for the model. To complete the operation, k repetitions of the training and testing data are required. We divided the data into ten folds using k -folds cross-validation, each roughly identical to the other folds in the dataset. Each iteration of the trained learning scheme is repeated nine times, and the performance of the learning approaches is evaluated using the remaining one fold, termed the testing set. The learning scheme is repeated ten times on the UCI phishing dataset. The prediction accuracies of the ten repetitions are averaged to obtain an overall prediction result in terms of precision, recall, F1-score, and accuracy.

1) Performance Before Feature Selection

To evaluate the performance of the proposed ensemble classifier among several ML and DL classifiers, we evaluated 30 features before applying any feature selection techniques. **Table 2** provides the weighted average precision, recall, F1-score, and accuracy values for all machine learning, DL, and proposed algorithms before feature selection with their execution times. By the conducted research, we have found our proposed approach has the highest accuracy of 97.21%, followed by 1-D CNN, DT, RF, Adaboost, k -NN, SVM, and LSTM, with an accuracy of 96.75%, 92.37%, 93.21%, 92.79%, 93.21%, 91.71%, 94.74%, 92.79%, and 90.84%, respectively. The graphical representation of accuracy is depicted in **Figure 3**.

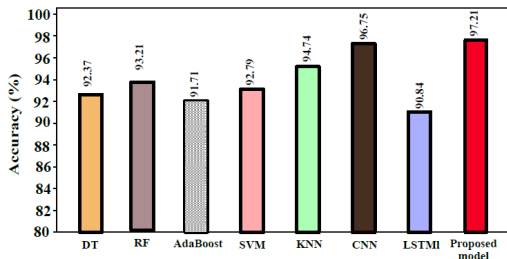


FIGURE 3. Accuracy obtained from different classifiers before feature selection.

Also, our proposed algorithm provides a maximum recall value of 97.97% and a second maximum of 97.09% by the DT classifier. However, LSTM is the least accurate classifier,

with a 90.84% accuracy and a recall value of 91.85%. **Figure 4** compares several classifiers based on their recall values before feature selection.

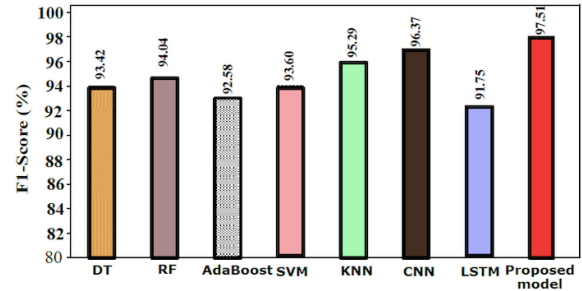


FIGURE 4. F1-score obtained from different classifiers before feature selection.

2) Performance After Feature Selection

Following selection, the top 20 features using the univariate feature selection technique, which uses mutual information to select essential features, are employed to classify phishing websites. Table 4 shows the weighted average values of precision, recall, F1-score, and accuracy for all seven ML and DL algorithms with the proposed approach and execution time. All algorithms employed the same top 20 features selected using the univariate feature selection approach. Moreover, it again shows that the proposed ensemble learning approach exhibits the highest accuracy of 96.26% followed by k -NN, 1-D CNN, DT, RF, Adaboost, SVM, and LSTM with an accuracy of 94.35%, 78.06%, 92.27%, 92.96%, 91.68%, 92.79%, and 91.23%, respectively. Once again, the minimum accuracy provided by 1-D CNN is 78.06%. **Figure 5** compares the accuracy of several classifiers after feature selection.

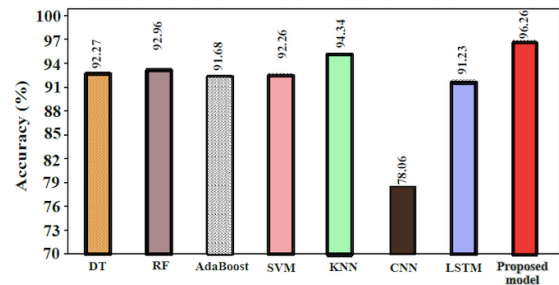


FIGURE 5. Accuracy obtained from different classifiers after features selection.

In contrast, our proposed algorithm also provides the maximum recall value of 97.96%. Besides, 1-D CNN is the least accurate classifier, with a 78.06% accuracy and a recall value of 97.66%. 1-D CNN confronted the regarding accuracy with fewer features at the time of training. **Figure 4** compares the graphical representation of several classifiers based on their recall values after feature selection.

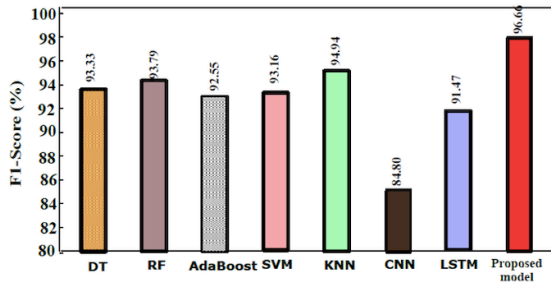


FIGURE 6. F1-score obtained from different classifiers for after feature selection.

3) Comparison between Before and After Feature Selection

The comparison of classifiers based on the accuracy before and after feature selection is depicted in Table 4. According to our research findings, the accuracy of almost all proposed algorithms remains constant or statistically insignificant before and after feature selection, except for 1-D CNN. This shows that the features chosen are the most important for classifying phishing websites, as determined by this study's univariate feature selection method.

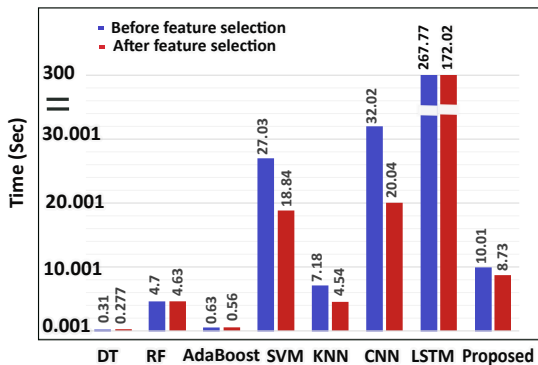


FIGURE 7. Comparison of model execution time before and after feature selection.

Figure 7 shows the execution time of classification models, as well as a comparison of execution times before and after feature selection. In general, feature selection helps reduce the execution time. Figure 7 also shows how feature selection reduces the time required to execute the models. As lazy learners, DT models have very fast execution times compared to other techniques, whereas LSTM takes longer to train due to memory-bandwidth-bound computing.

For our proposed EnLeM method, although the performance metrics after the feature selection are slightly lower than those before the feature selection, the differences are minuscule. The paired t-test [59] result on the four performance metrics (accuracy, precision, recall, and F1-score) before and after the feature selection gives us a p-value of 0.1353, which is higher than the threshold of 0.05, and hence not statistically significant. However, in terms of the

running time, the difference is 1.29 sec (which means a 13% speedup). This can be useful in mission-critical applications with high traffic volume and high throughput.

Compared to other ML and DL techniques, our proposed EnLeM method is the best technique to predict phishing websites to achieve the highest accuracy, precision, recall, and F1-score, with a relatively short execution time.

V. CONCLUSION AND FUTURE WORK

This article describes a novel and effective approach named EnLeM for detecting and classifying phishing websites. This approach is an extended approach of the voting-based EL approach. It also enables us to give approximately the same performance metrics when we deduct some features using the *univariate feature selection method* for selecting a set of relevant attributes from our UCI dataset. Furthermore, this research compared ML-based approaches and DL-based approaches, but our proposed approach, EnLeM, has shown that it is more accurate for detecting than previous studies. More specifically, the approach can detect more accurately which one is benign websites and which one is not, with the highest performance metrics compared to others, including 97.51% accuracy, 96.26% precision, and 97.51% F1-score. Although the DL-based 1d-CNN model provides good accuracy before feature selection, it needs more execution time. When we require more time to detect phishing websites, it will be detrimental because the opportunities for cybercrime will remain the same. However, it cannot provide better performance metrics after feature selection.

In this direction, no research is available that achieved 100% accuracy, so more investigation is needed to detect phishing URLs. In the future, the main objective is to execute a more compelling feature engineering method to increase the performance metrics of the specific problem using the approach.

REFERENCES

- [1] Ramzan, Z. Phishing attacks and countermeasures. *Handbook Of Information And Communication Security*. pp. 433-448 (2010).
- [2] Retruster., 2020. [Online]. Available: <https://retruster.com>.
- [3] Los Angeles Times., 2020. [Online]. Available: <https://www.latimes.com/business/la-fi-mh-anthem-is-warning-consumers-20150306-column.html>.
- [4] Threat Analysis Group, Findings on COVID-19 and online security threats., 2020. [Online]. Available: <https://blog.google/technology/safety-security/threat-analysis-group/findings-covid-19-and-online-security-threats/>.
- [5] Varshney, G., Misra, M. & Atrey, P. A survey and classification of web phishing detection schemes. *Security And Communication Networks*. **9**, 6266-6284 (2016).
- [6] Nguyen, L., To, B., Nguyen, H. & Nguyen, M. A novel approach for phishing detection using URL-based heuristic. 2014 International Conference On Computing, Management And Telecommunications (ComManTel). pp. 298-303 (2014).
- [7] Suleman, T. A Survey on Web Phishing Detection Techniques. *International Journal For Electronic Crime Investigation*. **5**, 25-36 (2021).
- [8] Revathy, S., Sathya Priya, S., Rafi, S., Pranideep, P. & Rajesh, D. Detection of Active Attacks Using Ensemble Machine Learning Approach. *Ambient Communications And Computer Systems: Proceedings Of RACCCS 2021*. pp. 507-519 (2022).

TABLE 4. Performance metrics comparison of existing ML-based models and DL-based approaches with our proposed EnLeM approach. (The highest value for each evaluation criterion, except execution time, is highlighted in bold.)

Classifiers	Before feature selection					After feature selection				
	Accuracy	Precision	Recall	F1-score	Ex.time (s)	Accuracy	Precision	Recall	F1-score	Ex.time (s)
Decision Tree (DT)	92.37	90.03	97.09	93.42	0.31	92.27	89.82	97.14	93.33	0.27
Random Forest (RF)	93.21	92.04	96.13	94.04	4.70	92.96	92.28	96.13	93.79	4.63
AdaBoost	91.71	92.37	92.80	92.58	0.63	91.68	92.31	92.81	92.55	0.56
Support Vector Machine (SVM)	92.79	92.56	94.67	93.60	27.03	92.26	91.72	94.64	93.16	18.84
k-Nearest Neighbor (k-NN)	94.74	95.12	95.47	95.29	7.18	94.34	94.56	95.32	94.94	4.54
1D-Convolutional Neural Network (1D-CNN)	96.75	96.53	96.24	96.37	32.02	78.06	76.93	97.66	84.80	20.04
Long Short Term Memory (LSTM)	90.84	91.90	91.85	91.75	267.77	91.23	90.69	92.40	91.47	172.02
Proposed Approach (EnLeM)	97.21	97.06	97.97	97.51	10.01	96.26	96.36	97.96	96.66	8.72

- [9] Dhamija, R., Tygar, J. & Hearst, M. Why phishing works. Proceedings Of The SIGCHI Conference On Human Factors In Computing Systems. pp. 581-590 (2006).
- [10] Wei, W., Ke, Q., Nowak, J., Korytkowski, M., Scherer, R. & Woźniak, M. Accurate and fast URL phishing detector: a convolutional neural network approach. Computer Networks. **178** pp. 107275 (2020).
- [11] Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A. & Liang, Z. Phishing page detection via learning classifiers from page layout feature. EURASIP Journal On Wireless Communications And Networking. **2019**, 1-14 (2019).
- [12] Masoudi-Sobhanzadeh, Y., Motieghader, H. & Masoudi-Nejad, A. FeatureSelect: a software for feature selection based on machine learning approaches. BMC Bioinformatics. **20**, 1-17 (2019).
- [13] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I., Chen, S., Liu, D. & Li, J. Performance comparison and current challenges of using machine learning techniques in cybersecurity. Energies. **13**, 2509 (2020).
- [14] Salihovic, I., Serdarevic, H. & Kevric, J. The role of feature selection in machine learning for the detection of spam and phishing attacks. Advanced Technologies, Systems, And Applications III: Proceedings Of The International Symposium On Innovative And Interdisciplinary Applications Of Advanced Technologies (IAT), Volume 2. pp. 476-483 (2019).
- [15] Yuan, H., Chen, X., Li, Y., Yang, Z. & Liu, W. Detecting phishing websites and targets based on URLs and webpage links. 2018 24th International Conference On Pattern Recognition (ICPR). pp. 3669-3674 (2018).
- [16] Jain, A. & Gupta, B. Towards detection of phishing websites on client-side using machine learning based approach. Telecommunication Systems. **68**, 687-700 (2018).
- [17] Gu, X., Wang, H. & Ni, T. An efficient approach to detecting phishing web. Journal Of Computational Information Systems. **9**, 5553-5560 (2013).
- [18] Moghimi, M. & Varjani, A. New rule-based phishing detection method. Expert Systems With Applications. **53** pp. 231-242 (2016).
- [19] Yerima, S. & Alzaylaee, M. High accuracy phishing detection based on convolutional neural networks. 2020 3rd International Conference On Computer Applications & Information Security (ICCAIS). pp. 1-6 (2020).
- [20] Zhang, N. & Yuan, Y. Phishing detection using neural network. CS229 Lecture Notes. (2012).
- [21] Rao, R. & Pais, A. Detection of phishing websites using an efficient feature-based machine learning framework. Neural Computing And Applications. **31**, 3851-3873 (2019).
- [22] Aksu, D., Turgut, Z., Üstebay, S. & Aydin, M. Phishing analysis of websites using classification techniques. International Telecommunications Conference. pp. 251-258 (2019).
- [23] Zhang, D., Yan, Z., Jiang, H. & Kim, T. A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites. Information & Management. **51**, 845-853 (2014).
- [24] Whittaker, C., Ryner, B. & Nazif, M. Large-scale automatic classification of phishing pages. (2010).
- [25] Ubung, A., Jasmi, S., Abdullah, A., Jhanjhi, N. & Supramaniam, M. Phishing website detection: An improved accuracy through feature selection and ensemble learning. International Journal Of Advanced Computer Science And Applications. **10**, 252-257 (2019).
- [26] Toolan, F. & Carthy, J. Phishing detection using classifier ensembles. 2009 ECrime Researchers Summit. pp. 1-9 (2009).
- [27] Mohammad, R., McCluskey, L. & Thabtah, F. UCI machine learning repository: phishing websites data set (2015). (University of California, Irvine, School of Information, 2016), <http://archive.ics.uci.edu/ml/datasets/phishing+websites>.
- [28] Mohammad, R., Thabtah, F. & McCluskey, L. Phishing websites features. School Of Computing And Engineering, University Of Huddersfield. (2015).
- [29] Garcí'a, S., Luengo, J. & Herrera, F. Data preprocessing in data mining. (Springer, 2015).
- [30] Dougherty, G. Pattern recognition and classification: an introduction. (Springer Science Business Media, 2012).
- [31] Alokandana Ghoshal, "Bagging and Boosting," March 2023. [Online]. Available: <https://www.educba.com/bagging-and-boosting>.
- [32] Shahrivari, V., Darabi, M. & Izadi, M. Phishing detection using machine learning techniques. ArXiv Preprint ArXiv:2009.11116. (2020).
- [33] Abu-Nimeh, S., Nappa, D., Wang, X. & Nair, S. A comparison of machine learning techniques for phishing detection. Proceedings Of The Anti-phishing Working Groups 2nd Annual ECrime Researchers Summit. pp. 60-69 (2007).
- [34] Fernández-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. Do we need hundreds of classifiers to solve real world classification problems?. The Journal Of Machine Learning Research. **15**, 3133-3181 (2014).
- [35] Yadav, N. & Panda, S. Feature selection for email phishing detection using machine learning. International Conference On Innovative Computing And Communications. pp. 365-378 (2022).
- [36] Kolla, J., Praneeth, S., Baig, M. & Karri, G. A comparison study of machine learning techniques for phishing detection. Journal Of Business And Information Systems (e-ISSN: 2685-2543). **4**, 21-33 (2022).
- [37] Pathak, D., Ammar, M. & Bhandari, M. Phishing Detection Approach Using Machine Learning. International Research Journal of Modernization in Engineering Technology and Science. **4**, 4233-4238 (2023).
- [38] Mohammed, M., Prasanth, K. & Subhash, S. Phishing Detection Using Machine Learning Algorithms. 2022 4th International Conference On Smart Systems And Inventive Technology (ICSSIT). pp. 921-924 (2022).
- [39] Ojewumi, T., Ogunleye, G., Oguntunde, B., Folorunsho, O., Fashoto, S. & Ogbu, N. Performance evaluation of machine learning tools for detection of phishing attacks on web pages. Scientific African. **16** pp. e01165 (2022).
- [40] Fukushima, K. & Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. Competition And Cooperation In Neural Nets. pp. 267-285 (1982).
- [41] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. Proceedings Of The IEEE. **86**, 2278-2324 (1998).
- [42] LeCun, Y., Bengio, Y., Hinton, G. & Others. Deep learning. Nature. **512**, 436-444 (2015).
- [43] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. & Inman, D. 1D convolutional neural networks and applications: A survey. Mechanical Systems And Signal Processing. **151** pp. 107398 (2021).
- [44] Hochreiter, S. & Schmidhuber, J. Long short-term memory. Neural Computation. **9**, 1735-1780 (1997).
- [45] Rojas-Barahona, L. Deep learning for sentiment analysis. Language And Linguistics Compass. **10**, 701-719 (2016).
- [46] Polikar, R. Ensemble learning. Scholarpedia. **4**, 2776 (2009), revision 186077.
- [47] Dasarathy, B. & Sheela, B. A composite classifier system design: Concepts and methodology. Proceedings Of The IEEE. **67**, 708-713 (1979).
- [48] Breiman, L. Bagging predictors. Machine Learning. **24**, 123-140 (1996).
- [49] Schapire, R. The strength of weak learnability. Machine Learning. **5**, 197-227 (1990).
- [50] Kingma, D. & Ba, J. Adam: A Method for Stochastic Optimization. CoRR. **abs/1412.6980** (2015).

- [51] Zhang, Z. & Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. 32nd Conference On Neural Information Processing Systems (NeurIPS). (2018).
- [52] Powers, D. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. Journal Of Machine Learning Technologies. **2**, 37-63 (2011).
- [53] Kumar, N., Hemanth, N., Premnath, S., Kumar, N. & Uma, S. Detection of Phishing Websites using an Efficient Machine Learning Framework.
- [54] Altaher, A. Phishing websites classification using hybrid SVM and k -NN approach. International Journal Of Advanced Computer Science And Applications. **8**, 90-95 (2017)
- [55] Tavallaei, M., Bagheri, E., Lu, W. & Ghorbani, A. A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium On Computational Intelligence For Security And Defense Applications. pp. 1-6 (2009)
- [56] Subasi, A., Molah, E., Almkallawi, F. & Chaudhery, T. Intelligent phishing website detection using random forest classifier. 2017 International Conference On Electrical And Computing Technologies And Applications (ICECTA). pp. 1-5 (2017).
- [57] Vallepu, R. & Karunakaran, M. An innovative method to improve performance analysis in classification with accuracy of phishing websites using random forest algorithm by comparing with support vector machine algorithm. AIP Conference Proceedings. **2655** (2023)
- [58] Alsharaiah, M., Abu-Shareha, A., Abualhaj, M., Baniata, L., Adwan, O., Al-saaidah, A. & Oraiqt, M. A new phishing-website detection framework using ensemble classification and clustering. International Journal Of Data And Network Science. **7**, 857-864 (2023)
- [59] Mowery, B.D., 2011. The paired t-test. Pediatric Nursing. **37**, 320-322 (2011)



big data analysis, and blockchain systems.

DR. BIKASH CHANDRA SINGH is a postdoctoral fellow at ODU, Virginia. He was a postdoc fellow in the Department of Electronics and Information Engineering, Hong Kong Polytechnic University, and a faculty member at the Department of Information and Communication Technology at Islamic University, Bangladesh. He obtained Ph.D. from the Department of Computer Science, Insubria University, Italy. His research covers all data privacy and security areas, machine learning,



of New Brunswick, Canada. Moreover, He received the ICT Ministry Fellowship award from the ICT Division-Government of the People's Republic of Bangladesh. Dr. Alom has published many research papers in prestigious journals and conferences. His research interests are Data security and privacy, Machine Learning, Deep Learning, Graph Neural Networks, Reinforcement Learning, Social Networks Analysis, Malware Detection, and Bioinformatics/Biological Data Processing.

DR. ZULFIKAR ALOM, Assistant Professor (on leave) of Computer Science at the Asian University for Women (AUW), Chattagram, Bangladesh. Currently, he is working as a post-doctoral research fellow at the University of Manitoba, Canada. He received a Doctor of Philosophy (Ph.D.) in Computer Science and Computational Mathematics from the University of Insubria, Italy, in 2019. After that, he was awarded a Post-Doctoral Research fellowship from the University



MOST NILUFA YEASMIN received a B.S. degree in Information and Communication Technology from Islamic University, Kushtia, Bangladesh, in 2020. She is currently pursuing her M.S. degree in Information and Communication Technology from Islamic University, Kushtia, Bangladesh

Currently, she is a Research Assistant in Data Privacy and Security Lab at Islamic University, Bangladesh, since 2020. She is the author of two journal papers and four internal conference papers.

Her research interest includes Machine Learning, Deep Learning, Graph Neural Network, Computer Vision, NLP, Data Mining, IoT, Bioinformatics, and Data privacy and security.



MD ABU RUMMAN REFAT received his B.Sc. and M.Sc. degrees in Information and Communication Technology from the Islamic University (IU), Kushtia, Bangladesh, in 2017 and 2019, respectively. In 2019, Refat was awarded the prestigious ICT Graduate Research Scholarship from the Information and Communication Technology Division, Dhaka-1207, Bangladesh.

He is currently a Lecturer in the Department of Computer Science and Engineering at the Green University of Bangladesh. His research interests include machine Learning, Deep Learning, Medical Image Analysis, NLP, and Data Privacy and Security.



ZEYAR AUNG (M'11-SM'16) received his Ph.D. degree in computer science from the National University of Singapore in 2006. From 2006 to 2010, he worked as a Research Fellow at the Institute for Infocomm Research (I2R), Agency for Science, Technology, and Research (A*STAR), Singapore. In 2010, he joined Masdar Institute, which later became a part of Khalifa University, UAE, as an Assistant Professor. He is currently serving as an Associate Professor with the Department of

Electrical Engineering and Computer Science at Khalifa University. His past research interests include bioinformatics and chemoinformatics. His current research interests include data analytics, machine learning, and their applications in various domains like cyber security, social media, financial systems, renewable energy, environmental science, etc.



MOHAMMAD ABDUL AZIM is an assistant professor in computer science at Asian University for Women. He received his Ph.D. in electrical and information engineering at the University of Sydney, Australia. He received his B.S. and M.S. in electrical & electronic engineering and computer & telecommunications engineering from the Chittagong University of Engineering and Technology and the University of Wollongong, respectively.

After completion of his Ph.D. he has worked in Malaysian Institute of Microelectronic Systems (MIMOS), Malaysia, Institut national de recherche en informatique et en automatique (INRIA), France, Memorial University of Newfoundland (MUN), Canada, Khalifa University of Science and Technology (Masdar campus), UAE and Gyeongsang National University (GNU), South Korea as researchers. His research interest includes artificial intelligence and machine learning, network privacy and security, wireless sensor networks, flying ad hoc networks, delay-tolerant networking, localization, resource allocation, energy efficiency, routing, MAC, aggregation and clustering, network security and cooperative communications, and network coding. He is a member of the technical program committee of various international conferences such as IEEE Globecom, WCNC, ICC, PIMRC, etc., and a regular reviewer of various journals and conferences in wireless networking and protocols. He is on the editorial board of the International Journal of advanced computer research, Springer Journal of Ambient Intelligence and Humanized Computing, and Frontiers in Artificial Intelligence.

• • •