IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 August 2024, accepted 16 October 2024, date of publication 25 October 2024, date of current version 20 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3486462

RESEARCH ARTICLE

Phishing Website Detection Using Deep Learning Models

UME ZARA[1], KASHIF AYYUB[1], HIKMAT ULLAH KHAN[2], ALI DAUD[3], TARIQ ALSAHFI[4], AND SAIMA GULZAR AHMAD[1]

[1]Department of Computer Science, COMSATS University Islamabad, Wah Campus, Islamabad 47040, Pakistan
[2]Department of Information Technology, University of Sargodha, Sargodha 40100, Pakistan
[3]Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates
[4]Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia

Corresponding authors: Ali Daud (alimsdb@gmail.com), Hikmat Ullah Khan (hikmatemail@gmail.com), and Saima Gulzar Ahmad (saimagulzarahmad@ciitwah.edu.pk)

ABSTRACT This research addresses the imperative need for advanced detection mechanisms for the identification of phishing websites. For this purpose, we explore state-of-the-art machine learning, ensemble learning, and deep learning algorithms. Cybersecurity is essential for protecting data and networks from threats. Detecting phishing websites helps prevent fraud and safeguard personal information. To evaluate the efficacy of our proposed method, the top features using information gain, gain ratio, and PCA are used to predict and identify a website as phishing or non-phishing. The proposed system is trained using a dataset that covers 11,055 websites. The ensemble learning model applied achieved an impressive 99% accuracy in predicting phishing websites, surpassing previous models, and setting a new benchmark in the field. The findings highlight the effectiveness of combining deep learning architectures with ensemble learning, offering not only improved accuracy but also adaptability to emerging phishing techniques.

INDEX TERMS Deep learning, ensemble learning, feature selection, GRU, LSTM, machine learning, phishing detection, RNN, RF, XGBoost.

I. INTRODUCTION

The recent advancements in the digital world have revolutionized our lives and brought technological developments in every type of business such as banking, marketing, service delivery, networking and communication, etc. This digital revolution has resulted in an unprecedented rise in the number of individuals utilizing Internet services for several diverse purposes. Communication technology has been the major contender in this development, continually altering to meet the consumers' ever-changing needs, giving real-time interactions, information access, and a worldwide feeling of connectedness. Simultaneously, opponents who are as adept at adapting to exploiting flaws in this interconnected society have developed in the world of digital media. These adversaries employ sophisticated tactics to disrupt communication, frequently employing malware and phishing techniques, intending to steal sensitive information, emphasizing the significance of strong cybersecurity measures and increased user awareness to guarantee the internet's ongoing benefits while reducing the associated risks. These adversaries obtain critical information by tricking users with malware or phishing sites. Phishing is one of the fake techniques used in the online world. The phisher sends out bait that mimics the genuine website and watches for victims. When a user falls for the phisher's scam and believes the mimicked page, the phisher wins. FIGURE 1 Shows you the whole life cycle of phishing and how the attacker targets the user to steal their data.

Phishing on websites happens when hackers build perfect replicas of trustworthy websites and advertise other websites or tech giants like Facebook, Twitter, Google, and so forth. Additionally, some phishing websites take advantage of security indicators like Hypertext Transfer Protocol Secure (HTTPS) [1] and a green lock, making the situation challenging for users to distinguish between reputable and fraudulent websites. To safeguard innocent Internet users, scientists have

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Yuan Chen.

FIGURE 1. Life cycle of phishing attack.

recently focused their attention on phishing attempts [2]. Many organizations, notably NSfocus and the Anti-Phishing Working Group (APWG), conducted surveys of attacks.

BitDefender, Symantec, McAfee, VeriSign, and other security product, service-oriented, law enforcement, trade, and international treaty organizations are among the members of the non-profit, international APWG group that examines phishing attacks that are stated by its affiliates. Weekly or semi-annual data reports on phishing trends in cyberspace are produced by

The most current APWG[1] [3] (2023) report states that there were 1,077,501 scams in the fourth quarter of 2023. APWG recorded over five million phishing attempts in 2023, the worst year on record. assaults against social media platforms increased dramatically in late 2023, accounting for 42.8% of all phishing attempts. Every quarter, there is a rise in phone phishing, often known as voice phishing or ''vishing''. The number of electronic funds transfer BEC assaults in Q4 grew by 24% over the previous quarter. While the total number of attacks like this increased, the average monetary amount per attempt dropped to $56,195. Attacks may come via websites, emails, or malicious software.

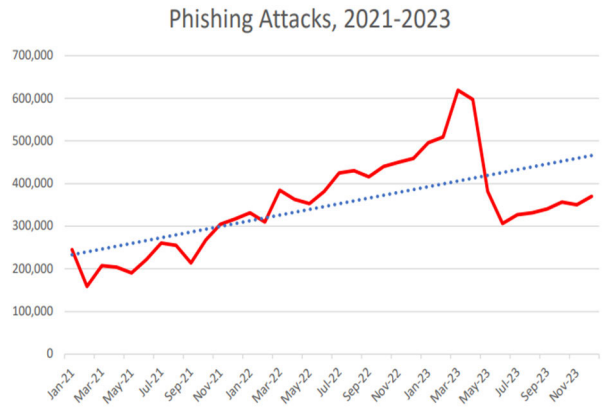


FIGURE 2. Number of phishing attacks (2021-2023).

FIGURE 2 shows that 2020 was the most well-known year for phishers. This information became available after around half a million phishing URLs were found during the specified year.

[1]https://apwg.org/trendsreports Accessed date March 15, 2024

A. BACKGROUND INFORMATION

One website that is shown in FIGURE 3 may be found using the Uniform Resource Locators. It consists of the following seven components: Protocol, Top-level domain, malicious Domain name [4], Path, Parameter, child domain, and Query. Communication between a web server and the web browser is governed by a protocol. Web Transfer Protocol (WTP), Post Office Protocol (POP), Some of the most extensively used protocols are Simple Mail Transfer Protocol (SMTP), HTTPS, and Internet Message Access Protocol (IMAP). Furthermore, a website's domain name serves as a distinctive online reference to identify it. In a web server, the path designates a specific place, such as /home/address/image.jpeg, where a particular directory or file lives. Within the primary domain name is a branch domain. For instance, cs.istqb.ac.in besides mail.oxford.edu are child domain of oxford.edu and istqb.ac.in. The top-level domain (TLD) is always included in a domain name; in the example of stanford.edu, the TLD remains edu. Dynamic web pages contain queries. A question mark is always placed after an inquiry. A client uses a query string to execute the programmer when it requests a page from a server. For instance, https://example.com/completed/track/there?name = alexa. In this URL, name = alexa is a query [5].

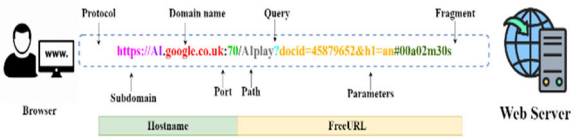


FIGURE 3. Anatomy of URL.

According to Cybersecurity Ventures, Cybercrime is expected to price the world economy $8 trillion annually by 2023. This is made worse by the growing expense of cybercrime-related losses, which is predicted to reach a total of $10.5 trillion by 2025. Phishing schemes accounted for about 30% of the 791,790 cyberattack complaints admitted by the Internet Crime Complaint Centre (IC3), making the greatest complained-about kind of cybercrime, and consequential in damages exceeding USD 54 million.[2] For those who use the Internet to browse, being able to differentiate between legitimate and fraudulent websites is essential. To distinguish phishing websites, visitors require visual aids.

As is well known, Phishing is a type of internet deceit in which a perpetrator impersonates a reliable organization or individual to deceive targets into disclosing sensitive data, such as bank account information or login passwords. An attacker must go through five phases before stealing money from an individual's profile or using the information for subsequent attacks.

This study achieves the following contributions:

- Applying feature selection algorithm including information gain (IG), gain ratio (GR), and principal component

[2]https://www.esentire.com/resources/library/2023-official-cybercrime-report. Accessed date March 20, 2024

analysis (PCA) to optimize the model's ability to recognize relevant patterns for effective phishing website detection.
- Comparison of various algorithms including AdaBoost, XGBoost, random forest (RF), support vector machine (SVM), Decision Tree (DT), K-Nearest Neighbors (KNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN), and on a different feature sets.
- To enhance classification accuracy, employ top-performing algorithms from diverse models.
- Performance evaluation measurements, for instance, accuracy, precision, recall, and f-measure are used to evaluate the effectiveness of all models.

The remaining portion of the study is structured as well. Section II examines prior research and strategies for detecting Web phishing. Section III defines the methodologies used in this study. Section IV examines the findings, while Section V provides the study's conclusions.

II. RELATED WORK

This section provides a summary of prior studies on phishing attempts in general, with an emphasis on the classification approaches used to identify web phishing.

A. LIST-BASED PHISHING DETECTION SYSTEMS

Whitelists and blacklists are two distinct lists applied by phishing detection (PD) systems to identify and categorize authentic and fake URLs. Whitelist-based Phishing detection systems build stable and trustworthy sites that deliver relevant knowledge. Doubtful sites must coincide with the whitelisted domains; the user has judged it suspicious and unsafe if it does not.

In [6]. to create a whitelist-based system that creates a whitelist by keeping track of the IP address of each site with a login page Where people may enter their details. The Windows 2008 system alerts the user to the inconsistency of registered information details when they utilize this login interface. For this reason, when consumers visit reputable websites for the first time, this system immediately mistrusts them. Another study [7] created a system that automatically updates and maintains the whitelist regularly, alerting users when they come across a phishing website. The taking out of properties concealed the connection among the source code and the component that corresponds to the domain's internet address determines how well this system performs. The study's preliminary findings indicate that the TP score was 86.02%, with a FN score of 1.48%.

Reference [8] based on the data of URLs referred to be phishing websites, blacklists were compiled. Record entries are gathered for list generation from a variety of sources, including user notifications, spam system detection, and third-party authorities. Systems can stop attackers from logging their IP addresses and URLs to the blacklist. Because the blacklist-based system recognizes the attackers' prior URLs or IP addresses, they must use a new one the following time. By detecting malicious URLs or IP addresses, System security administrators may continually update the blacklist regularly, repelling new attackers. As an alternative, people who want to upgrade their security system can obtain these lists. Blacklist-based systems are primarily vulnerable to zero-day attacks because they are unable to recognize fresh or first-day assaults. When compared to machine learning-based systems, these interference-finding methods have a reduced false-positive score. Based on the blacklist, the accuracy of detecting intrusions or assaults on these systems is quite high, with a success rate of about 20%. Thus, this demonstrates that several firms' identity systems that rely on blacklist techniques, including Phish Net and Google Safe Browsing API,[3] are trustworthy in identifying phishing attempts by using blacklists. Such safety measures use approximate algorithms for matching to pair dangerous URLs with blacklist URLs. Blacklists that make use of these technologies must be updated often.

This study [9] has a 97% correctness rate for phishing URL detection using browser extension techniques numerous automatic PD methods have been presented recently. This study's 92% detection accuracy was achieved by using abbreviated URL characteristics. Phish-Safe detection technique finds despicable sites. Obtained 90% accuracy in phishing detection using supervised-based ML techniques, namely SVM and naive Bayes. In this study, [10] The email phishing assault was detected using the ensemble learning approach. To attain 91% accuracy with only 11 features, feature selection approaches that are no longer connected with accuracy are employed to shift such features.

In [11], The whitelist was utilized by researchers to detect phishing websites. Website access is restricted to the study to those whose URLs are on the whitelist. Another technique is the blacklist method. In addition to programmers like Phish Net and Google Safe Browsing API, there is research employing blacklists in the literature [12]. Blacklist-based systems verify the address against the list and deny access to URLs that aren't on the list. The primary drawback of these methods is that even a slight alteration to the URL might preclude matching in the list. Furthermore, current security measures are insufficient for avoiding the most recent attacks, frequently referred to as zero days.

B. RULE-BASED PHISHING DETECTION SYSTEMS

Such approaches use relational rule mining to acquire features. The idea's purpose is to recognize features that more frequent in phishing URLs [13]. The purpose of research using this type of technology is to more actively categorize data by using useful qualities. These systems follow specified rules. When these rules are employed to train the system, the accuracy ratio improves. CANTINA [14] the study, To identify phishing assaults, rules, and the Term Frequency - Inverse Document Frequency (TF-IDF) were

[3]https://safebrowsing.google.com/

employed. Furthermore, some traits and guidelines were used to develop models in related investigations.

Feng et al. [15] created the client-side anti-phishing program BogusBiter, which finds a suspected phishing site with a blatantly high number of bogus certificates. By concealing the victim's actual credentials amid the false ones, BogusBiter makes it possible for the authentic website to quickly detect the stolen credentials. Smadi et al. [16] recommended a method for detecting phishing that uses URLs and extracts 17 characteristics from the websites. The retrieved data, which include domain name, length of URL, and unique URL symbols, among others, offer a benchmark for determining if a connection is fraudulent. To identify newly emerging phishing URLs, data mining techniques are utilized to find new hidden rules.

C. PHISHING DETECTION SYSTEMS BASED ON VISUAL SIMILARITY

Programs compare the visual similarities across online pages. By examining websites from the server-side perspective, phishing, and non-phishing sites are categorized. Compare these two sets of data using image processing methods. Often, fake websites have designs that are quite like the real ones. However, there aren't many visible variations between them. Finding simpler to discover variations using Image processing techniques, which are not immediately seen by consumers. The degree of resemblance found determines whether a website is phishing. Within the literature, as in the study [17], Certain investigations identify distinctions based on fundamental commonalities.

Chen et al. [18] suggested a technique for recognizing phishing websites. Websites are initially categorized according to their degree of similarity, and different techniques are used for each group. They utilize a color histogram in conjunction with wavelet hashing (wHash) for webpages that bear a striking resemblance to the original. Using the k-NN classifier and the scale-invariant feature transform (SIFT) approach, the locally comparable websites are assessed. A self-collected database from Phish Tank, comprising over 1200 websites that mimicked the appearance of Microsoft, Dropbox, and Bank of America websites, was used for the trials. The findings show that the present perceptual hashing approach is not as accurate as the suggested wash method with color histogram. Additionally, the accuracy of the SIFT approach was 92.93%, 93.61%, and 95.95% for the datasets. A hue-based descriptor has been proposed as an auto-update approach for phishing databases to identify visual similarities between a fake website and a genuine website [19]. The Earth mover's distance (EMD) metric is used to match the descriptors. The testing was conducted using a self-collected collection of 2943 phishing websites that imitated the websites of PayPal, Bank of America, and Facebook. When the auto-updating technique is used, the findings demonstrate a 30% improvement. The technique to identify phishing sites is to make an effort to impersonate based on visual resemblance. They have put up many MPEG-7-based compact visual descriptors to express edge and color information. SVM and RF techniques were used to build the picture categorization, and these descriptors were employed in patch-based and holistic situations. The studies were conducted using a self-assembled database that included 2852 samples from 14 different websites. The suggested method's best outcome was an F1 score of 90%.

D. PHISHING DETECTION SYSTEMS-BASED MACHINE LEARNING

ML-based phishing detection systems use artificial intelligence approaches to classify the necessary characteristics to detect phishing websites. Features are produced by compiling data under many headings, including URL, web address, features, content, and so on. It is widely used in user security due to its dynamic nature, notably in identifying irregularities in websites.

There are a few studies on this kind of detecting technique in literature. The mentioned CANTINA initiative [20] was completed by the application of machine learning. TF-IDF and heuristic methods indicated that they found 90% accuracy rates. In [21], By categorizing URL data including its length, Amount of special characters, web address, directory, and file name may be used to detect fraudulent websites. Transport layer security characteristics are used in conjunction with URL-based measures. They used the instructions that the apriority algorithm produced to find a 93% accuracy percentage. In, [22] To ascertain whether a website is phishing, a nonlinear regression technique is employed. SVM and harmony search techniques were used to run the system. They made use of 20 features and 11055 websites. Rather than utilizing the cover, the DT technique was used to choose the features. Data to find an accuracy percentage of 92.80%. In another study [23], It was suggested to use some Natural Language Processing (NLP) based features and 209 word-vector characteristics to create a phishing detection system. After a comparison of the SMO, and NB algorithms, the RF method produced the best results, with an accuracy rate of 89.9%.

In [24], Three distinct machine learning methods were contrasted based on the accuracy values of their NLP vector counts. After comparing the SMO, and NB algorithms, the hybrid strategy using the RF algorithm produced the best results with an accuracy percentage of 97.2%. Researchers put in place a mechanism for detecting phishing [25] by classifying data using neural networks that are accomplished by self-adaptation. The study uses seventeen different characteristics that are also used by third-party providers. As a consequence, it was reported that real-world implementation would take far longer.

In [26], the method and the danger decrease idea are employed in a neural network-based classification system designed to detect phishing websites and focused on how functions are trained to impact neural networks to increase

the efficacy of implications. Reference [27], Email headers, content URLs, and HTML content, are recorded. Fifty characteristics from each of these groups were used in the ML classification process. The outcomes displayed an accuracy of 96.6%. In this [28], Features taken from address, source code, and outside other services are compared with ML methods. Analyzing the principal components Random Forest was able to recognize zero-day phishing assaults with 94.55% accuracy. In research using NLP [29], Email text was examined and categorized, Thirty-five features and TF-IDF, as well as hand-crafted features, were used in the classification process. The study examined the recognition rates of phishing attacks using six distinct algorithms. RF algorithm generated the greatest results, with 92.55% accuracy.

This study [30] looks at how computers can read text in languages with limited resources, such as Vietnamese or Urdu. They accomplished this by reviewing several prior publications. According to the survey, the majority of research is currently focused on these low-resource languages. Previous reviews did not compare approaches effectively. This article provides an excellent review of the studies on interpreting text in several languages. It also offers improvements to this sort of study. The purpose is to contribute to the development of better computer systems for language interpretation with minimal resources.

E. PHISHING DETECTION SYSTEMS BASED ON DEEP LEARNING

DL is a sort of ML that learns via deeply structured architectures. Deep learning memory networks that are often used include CNN, LSTM, and RNN. Many deep learning-based phishing detection tools are being released in response to the instant expansion of NLP and DL algorithms [31].

Ahmed and Ali created a DNN and genetic algorithm (GA) intelligence phishing detection solution [32]. To fit the DNN model, the classification part used the features given like feature parameters and data set from the UCI as the training set. In contrast, The GA-DNN model achieved a low 89% accuracy. Parameters like quantity data used for training have a substantial impact on DL model accuracy [33].

Aljofey et al. published a CNN architecture for the detection of phishing constructed on URLs in 2020 [34]. They extracted character-level data from real URLs received from both phishing and non-phishing sites. The algorithm has a 95.02% rate accuracy on its dataset of 318,642 occurrences, according to the study results. Wang et al. developed the PDRCNN model, which used URL content as input, retrieved characteristics using an RNN and CNN, and categorized using the Sigmoid methods [35].

Alexa.com[4] instances and PhishTank.com[5] and obtained semantic features by encoding the URL string to a tensor, which was then used as input to the RNN, using the word embedding technique. The RNN architecture was created by employing a bidirectional LSTM network strategy to take out global features, this data was subsequently loaded into the CNN. The resultant one-dimensional tensor was a set of features generated by many convolutional and maximum-pooling layers. Finally, the one-dimensional vector was passed into a fully connected layer that used a sigmoid function to detect if the first input URL was phishing or not. The experiments' findings suggested that they were 92.97% right [36].

Table 1 below compares significant cutting-edge solutions. Although accuracy diverse across datasets, Alternative models used the random forest approach. The UCI dataset is frequently employed in machine learning, especially for beginners and researchers who might not know much about security. But when we want to use it in real-time systems, we must get information from a website address (URL). This process uses security standards and might need help from other services. To make things work better and faster, some smart people came up with models that mix different ways of choosing which information is important (feature selection) with a regular computer method that makes decisions (classifier). Interestingly, deep learning, which is powerful but sometimes not very accurate for this task, has an advantage here. It works almost in real-time, and it doesn't need experts in cybersecurity or those extra services. Once the model is trained, it reacts faster compared to traditional systems that depend on the usual features.

III. RESEARCH METHODOLOGY

The proposed framework for phishing website detection is explained in this section, this process begins with the selection of a phishing dataset, different classifiers are applied, python is used to extract feature selection strategies, and the Performance Evaluation Measures are implemented to conduct the tests. In the first stage, select the phishing website dataset. The top features will be evaluated using feature selection techniques like IG, GR, and PCA. After determining the significance of characteristics, data will be split into testing and training sets 80% and 20%. Classifiers' performance for detecting phishing websites is assessed using performance evaluation techniques. For models 300 iterations and 200 hidden layers and adaptive learning rate are used. Figure 4 shows the suggested framework.

A. APPLIED ALGORITHMS

These are the applied algorithms that have been applied from three different categories of shallow ML, EL, and DL and the main aim was to achieve improved results.

1) SHALLOW MACHINE LEARNING

ML algorithms play a dynamic role in accurately classifying data across various industries, including marketing, telecommunications, and information technology. In this study, three commonly used classification techniques are SVM, Decision Tree, and K-Nearest Neighbors. The SVM algorithm is widely recognized for its ability to find the best classifier for

[4]https://www.amazon.com/b?node=21576558011
[5]https://www.phishtank.com/index.php

TABLE 1. Comparison of major state-of-the-art solution.

Ref	Years	Type	Model	Dataset	Accuracy
[24]	2018	Hybrid	NLP+RF	Phishing websites	97.2%
[35]	2019	Deep learning	RNN + CNN	Examined Alexa, Phish Tank; 490,408 entries, half phishing, half legitimate. Utilized word embedding for content analysis.	95.79%
[37]	2020	Hybrid	LBET (LR + extra tree)	Phishing Website Dataset	97.57%
[36]	2020	Deep learning	Convolutional auto encoder + DNN	Website (Clients' daily requests, PhishTank) 6116 instances of rule-based features.	89.00%
[38]	2020	Single	CNN+LSTM	Phishing website dataset	93.28%
[20]	2020	Single	SVM+ CANTINA	Dataset collected from real phishing cases	96.0%
[39]	2021	Single	Adaptive Boosting	Phish Tank, Google Search; unspecified data quantity; 30 details per information.	98.30%
[40]	2021	Hybrid	Grey wolf optimizer + SVM	Websites (PhishTank) 1353 instances: 548 legitimate URLs; 805 phishing URLs	90.38%
[41]	2021	Hybrid	Bagging + LMT	Phishing Website Dataset	97.42%
[7]	2021	Single	automated white list	Six different Phishing URLs dataset	95.0%
[21]	2021	Single	NB+DT+RF+SVM	Experiments consist of active phishing attacks + Google Safe Browsing	97.21%
[42]	2022	Hybrid	ISHO + SVM	Phishing Website Dataset	98.64%
[43]	2022	Single	SVM + NB + RF	Phishing Dataset	92.0%
[32]	2023	Deep learning	Genetic algorithm (GA) + DNN	Phishing Website Dataset	89.50%
[34]	2023	Deep learning	LSTM+CNN	phishTank URL features, with 20,000 records of 80 features,	97.6%

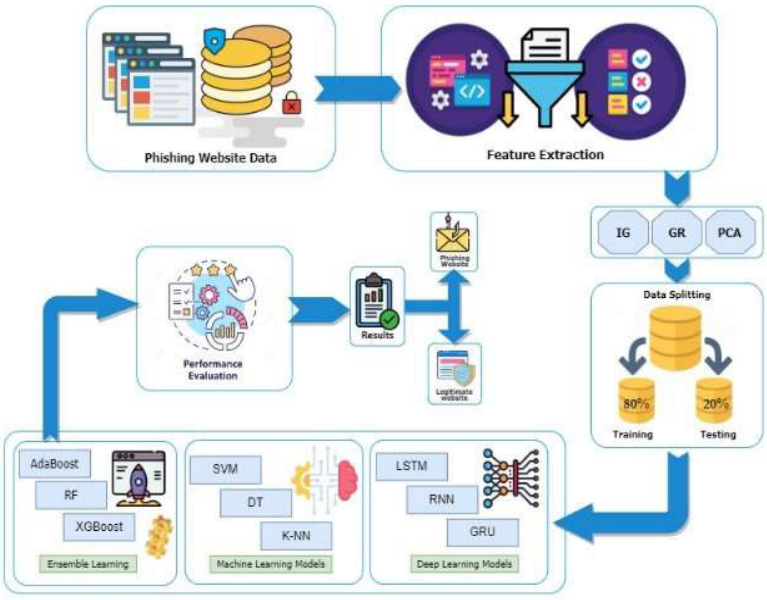


FIGURE 4. Steps of the proposed research framework.

given data, achieving excellent generalization performance. On the other hand, Decision Tree utilizes tree-like structures to classify decisions and output the class of the given data. Lastly, K-Nearest Neighbors is a proximity-based algorithm that classifies data points based on their neighbors. In the discipline of machine learning, classification algorithms are critical for properly categorizing data.

2) SHALLOW ENSEMBLE LEARNING

AdaBoost, XGBoost, and Random Forest are examples of Ensemble Learning (EL) algorithms that have achieved popularity in a variety of domains due to their competence in enhancing model prediction accuracy. These algorithms combine numerous weak learners or foundational models to create a stronger learner. AdaBoost focuses on iteratively

training ineffective classifiers and providing greater weight to misclassified points of data in every iteration of training, resulting in ongoing improvement of overall model performance. XGBoost, which stands for Extreme Gradient Boosting, is a more advanced form of gradient boosting that uses a more regularised model formalization to reduce overfitting and increase generalization. Random Forest, on the other hand, produces a maximum amount of decision trees throughout training and uses the approach of the classes (classification) or the mean prediction (regression) for each of the trees to make the final prediction. It uses crowd intelligence to make reliable predictions by averaging the outputs of numerous decision trees. Each of these ensemble approaches has distinct features and can be adapted to specific problem domains, making them useful tools for machine learning research and applications.

3) SHALLOW DEEP LEARNING

Deep learning(DL) methods, such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU), have transformed sequential data processing by accurately capturing temporal connections. These algorithms specialise at handling time-series data and repetitive patterns. RNNs are designed to handle series of inputs by storing hidden states that contain information from previous time steps, enabling them to reflect temporal connections. However, conventional RNNs suffer the vanishing gradient problem, which limits their ability to detect long-range correlations. DL approaches like LSTM, RNN, and GRU have transformed serial data processing by accurately capturing temporal links. These algorithms excel at processing time-series data and sequential patterns. RNNs are designed to handle input sequences by maintaining hidden states including information from prior time steps, allowing them to represent temporal relationships. However, conventional RNNs suffer from the vanishing gradient problem, which bounds their ability to obtain long-term associations.

4) FEATURE REDUCTION METHODS

The feature reduction strategies described below are used to reduce features in phishing data sets.

a: INFORMATION GAIN(IG)

IG is often referred to as mutual information. It reduces the bias towards multi-valued features by taking into account the quantity and size of branches when picking an attribute.ML algorithms typically use IG prediction class IG to compute results in bits. IG is often used to extract useful information from data. IG is derived by reducing the overall entropy value and assessing the impact of the feature addition. In this equation, E represents entropy.

$$info(F) = \sum_{j=1}^{m} (P_j \log_2 P_j) \tag{1}$$

Equation (1) Where m defines the class number and p_i defines the probability of any item. C_j contain as $|C_j, F|/|F|$. log_2 is the encoding information in the form of bits. For attributes, A { $a_1 \cdots a_v$ } F would be in v partitions { $F_1 \cdot F_v$. } Eq (2) is used to calculate entropy information.

$$\text{valueinfoA(F)} = -\frac{\sum_{j=1}^{v} |Fj|}{|F|} Xinfo(Fj) \tag{2}$$

Equation (2) Where $|F_j| / |F|$ is the weight of the j_{th} partition and entropy of F_j is defined as Info (F_j). IG by separate on A is:

$$IG(F) = info(F) - infoA(F) \tag{3}$$

Equation (3) Attributes having a top value of IG are used to categorize the document into the specified class.

b: GAIN RATIO (GR)

GR employs a repeating procedure to pick a minimum feature set based on the GR score. GR is commonly used to reduce the dimension. The GR algorithm calculates feature differences. The highest GR score determines the feature's usefulness. Normalization value represents a divided value of information. The training document is divided into v divisions based on the number of outputs for a feature.

$$\text{equalnfoA(E)} = -\frac{\sum_{j=1}^{v} |Ej|}{|E|} Xlog2|Ej|/|E| \tag{4}$$

Equation (4) high *splitinfo* information is sparse and consistent. Few partitions maintain peak values. GR is computed as follows:

$$GR(F) = \frac{IG}{splitting(F)} \tag{5}$$

c: PRINCIPLE COMPONENT ANALYSIS (PCA)

PCA is a technique for lowering dimensionality approach in machine learning and statistical analysis that converts high-dimensional information into a lower-dimensional representation while maintaining as much variability as feasible. It identifies the major elements, which are the linear combinations of the initial characteristics, and arranges them in order of decreasing variance. The transformation of the data is given by:

$$Z = X \cdot W \tag{6}$$

Z is the matrix of transformed data (principal components), X is the matrix of original data, and W is the matrix of eigenvectors (principal components) obtained from the covariance matrix.

The covariance matrix Σ is calculated as:

$$the \sum = \frac{1}{n-1} \cdot \left(X - \bar{X}\right)^T \cdot \left(X - \bar{X}\right) \tag{7}$$

where n is the sample number, X is the matrix of original data, and X is the mean of each feature across samples.

d: EXPERIMENTAL SETUP

The experimental design includes selecting a phishing website dataset, followed by feature selection by applying methods such as Information Gain (IG), Gain Ratio (GR), and Principal Component Analysis (PCA). The dataset is divided between 80 percent training and 20 percent testing sets. Multiple classifiers are trained and evaluated across 300 iterations, with 200 hidden layers and an adjustable training rate, and performance is determined using conventional evaluation criteria. Results are computed on an i7 12th generation computer with 8GB RAM and 512 SSD Gen-3.

Dataset: The phishing website data collection is publicly accessible for research purposes Kaggle.[6] The data set covers 11,055 websites and 32 attributes.

Performance Evaluation Measures: Several Performance Evaluation Measures from the phishing site data set are utilized to predict classifier accuracy. The parameters being measured are accuracy, recall, precision, and F1 measure, as shown in equations 8, 9, 10, and 11, separately.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{9}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{10}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{11}$$

In the above equations, TP stands for true positive results that the model correctly identified as a positive class, TN for true negative results that the model correctly indicated as a negative class, FP for false positive results that the model predicted incorrectly as a positive class, and FN for false negative outcomes that the model predicted incorrectly as a negative class.

[6]www.kaggle.com/akashkr/phishing-website-dataset#dataset.csv Accessed 3 March 2024.

IV. RESULTS AND DISCUSSIONS

This section describes the feature analysis, data analysis, and experimental results generated from the phishing detection dataset.

A. FEATURE ENGINEERING

Selecting the strong features is a significant and difficult process, particularly when it comes to creating precise projections and identifications in a fresh dataset. The method requires precisely assigning values to various qualities, and the results have a considerable impact on determining a website's rating. In this context, websites are classified according to their nature: -1 indicates a phishing website, 0 suggests a suspicious website, and 1 represents a trustworthy website. This systematic approach to feature selection and value

TABLE 2. Features in the data and their description.

Features Name	Feature Description
SFH (Server Form Handler)	Server Form Handler (SFH). The string "About: blank" is used. Typically, demands a name & password, and once entered, the page is forwarded to the suspected website. Its key features are as follows: -1,0,1
popupWindow	It is a web browser window that is used to change the size of the screen. It shows choices for allowing the website's menu bar to appear.
SSLfinal_State	SSL establishes a secure communication connection between a client and a server. The SSL final state is the cert state that is stored in your computer's cache whenever you visit or transact on a website. Its value qualities include: -1,0,1
Request_URL	Images, movies, and other content are loaded onto a web page from another URL. However, most significant websites use the same domain to store data.
URL_of_Anchor	Same properties as request_URL. However, if the site reveals changing domain names, it might be a fraudulent site. Its value characteristics are as follows: -1,0,1
Web_traffic	Web traffic defines the users who visit the website. Web traffic measures the number of visitors to a website. Overall, it is intended to drive traffic to an online shopping or private website.
URL_Length	The lengthy address suggests phishing. If the size is fewer than 54 characters, it is considered a genuine website; otherwise, it is considered a phishing or suspicious website. Its key features are as follows: -1,0,1.
age_of_domain	The age of the web address corresponds to the lifetime of the website that you are maintaining. A longer domain name indicates a more trustworthy website. The bogus website has a short-lived domain.
having_IP	In most cases, DNS is used instead of an IP address to identify a genuine website. So, if an IP address is displayed in the place of a domain name in the web address, it is a fraudulent site. Its value characteristics are as follows: -1,1
Favicon	Indicates whether the website has a favicon (small icon displayed in the browser tab). Lack of a favicon may be a red flag, but it is not always indicative of phishing. Its value characteristics are as follows: -1,1
Iframe	Checks if the site uses iframes. Fraudulent sites may use iframes to load malicious content or mask the true site. Its value characteristics are as follows: -1,1
Google_Index	Indicates if the site is indexed by Google. Sites not indexed may be new or suspicious. Its value characteristics are as follows: -1,1.
DNSRecord	Shows the DNS records associated with the domain. Valid DNS records indicate a legitimate site; anomalies can be suspicious. Its value characteristics are as follows: -1,1.
Prefix_Suffix	Refers to the existence of prefixes or suffixes in the URL. Frequently used to make a URL shows legitimate when it is not. Its value characteristics are as follows: -1,1.
having_At_Symbol	Checks for the existence of the "@" symbol in the URL. If appear, it could imply a suspicious or phishing URL. Its key features are as follows: -1,0,1.

assigning is critical for accurately categorizing and comprehending the properties of websites in the dataset provided in Table 2.

B. FEATURE ATTRIBUTES VISUALIZATION

In this part, we're taking a closer look at the features in a dataset by creating pictures or graphs. These visuals help us see how the different features are spread out, if they're related to each other, and how they might affect how well models work. Imagine it as drawing a map of data to understand its landscape better, seeing where things are, how they're connected, and figuring out how it all influences models.

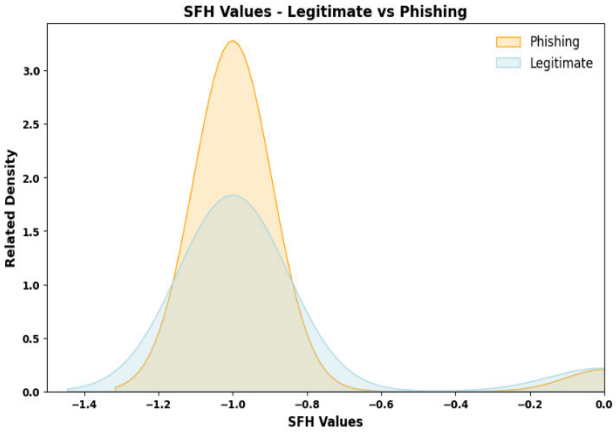


FIGURE 5. SFH and the result visualaization.

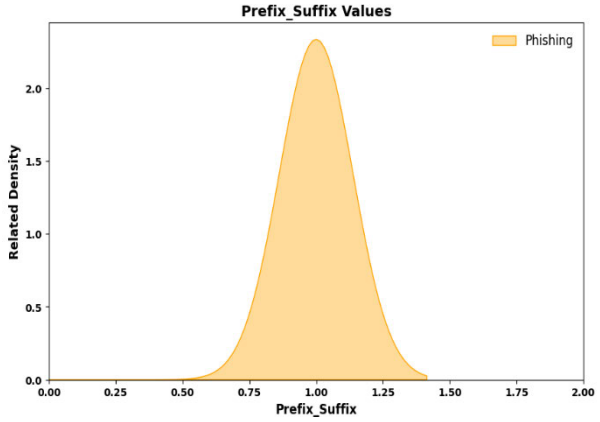


FIGURE 6. Prefix_Suffix and the result.

FIGURE 5 Shows a comparison between the result attribute and SFH. There are more phishing values in the SFH attribute. Where values are valid, the graph gets broader and narrower; where values are becoming more phishing, the graph gets thinner and higher. Out of all the qualities.

Prefix_Suffix has the second-highest ranking. The Prefix_Suffix property includes every one of the phishing values displayed in FIGURE 6. The visualization demonstrates that Prefix_Suffix is frequently malicious on sites because it may be used to steal user credentials.

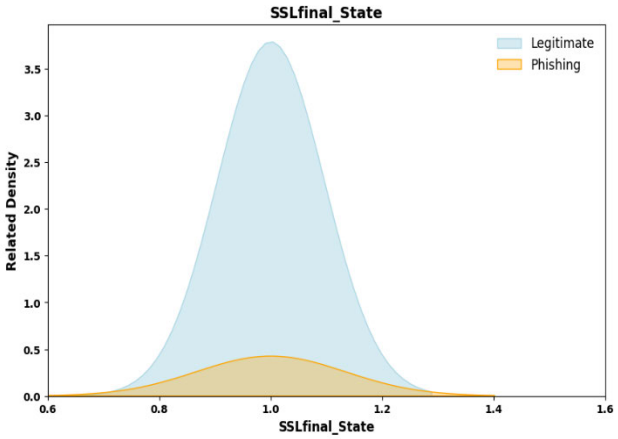


FIGURE 7. SSLfinal_State and the result visualization.

FIGURE 7. Phishing values are surpassed by valid values in SSLfinal_State. In comparison to the graph with phishing values, the one with more valid values is higher and thinner.

C. TOP RANKED FEATURES

In this study on detecting phishing websites, utilized three techniques PCA, IG, and GR to streamline the dataset, which initially included 32 features extracted from each website. As a result of applying these techniques, identified the top ten most significant features. These values, along with additional insights, are presented in the table below for a comprehensive understanding of the selected features and their importance in enhancing the detection of phishing websites.

The comprehensive examination of features is conducted with a multifaceted approach, incorporating IG, GR, and PCA Variance Ratio to gain a comprehensive understanding of their significance. The SSLfinal_State feature, which reflects the Secure Sockets Layer's final state, has a delicate balance of IG and GR, showing that it is quite significant in the dataset. Furthermore, its connection with the Favicon in PCA contributes significantly to the total variance, revealing possible links between SSLfinal_State and Favicon-related variables.

Moving on to URL_of_Anchor, the feature has its own rating with slightly greater IG but lower GR. The association with popUpWidnow in PCA reveals its major contribution to the dataset's variability, indicating that URL_of_Anchor may impact pop-up window performance. This refining evaluation helps researchers determine not only the individual relevance of traits, but also how they combine in different analytical scenarios.

Links_in_tags appear as a notable characteristic, with a surprisingly high IG, indicating its relevance. In PCA, alignment with the port implies a probable association between the number of links in tags and port-specific features. This insight can assist researchers comprehend deeper patterns in the dataset, allowing them to better their hypotheses and model-building techniques.

TABLE 3. Top feature ranking by IG, GR, and PCA.

	Information Gain		Gain Ratio		PCA	
Sr. No	Ranked Feature	Values	Ranked Features	Values	PC	Variance Ratio
1	SSLfinal_State	0.0253	double_slash redirecting	0.1151	Favicon	0.1836
2	URL_of_Anchor	0.0302	port	0.0860	popUpWidnow	0.3591
3	Links_in_tags	0.6222	Iframe	0.1002	Port	0.3550
4	having_Sub_Domain	0.0131	Abnormal_URL	0.1186	on_mouseover	0.3280
5	web_traffic	0.0146	RightClick	0.1099	Submitting_to email	0.1790
6	Prefix_Suffix	0.1125	on_mouseover	0.1147	Iframe	0.3312
7	Links_pointing_to_page	0.0333	Statistical_report	0.1129	RightClick	0.2501
8	Request_URL	0.0139	Shortining_Service	0.1215	having_At_Symbol	0.3573
9	SFH	0.0267	HTTPS_token	0.1071	Abnormal_URL	0.3237
10	Domain_registeration length	0.0178	popUpWidnow	0.1150	Statistical_report	0.1761

On the other hand, having_Sub_Domain has a lower IG but a significant GR. Its association with on_mouseover in PCA improves the analysis by revealing a correlation between subdomain presence and mouseover interactions. This dual-metric technique allows for a more nuanced assessment of feature significance, taking into consideration both the size and scope of their effect. web_traffic, with its mild IG and GR, is analogous to Submitting_to_email in PCA. This relationship sheds light on potential connections between internet traffic patterns and email submission behaviours, providing important insights into the dynamics of user engagement.

FIGURE 8 Graphical depiction of the top ten ranking features based on IG GR and PCA Finally, this complete feature analysis, which includes numerous metrics and PCA, ranks features based on their relevance while also revealing deep correlations between them. Researchers may use this knowledge to make better judgments regarding feature selection, model interpretation, and hypothesis development, resulting in a more robust and informative study conclusion.

D. CLASSIFICATION RESULTS

In this part, the results of a detailed evaluation of several classifiers applied to the provided feature set are presented. Traditional classifiers' performance is assessed using carefully picked features that capture critical parts of the underlying data. Comparative findings are examined via exact experimentation and rigorous assessment, giving insight into each classifier's unique strengths and limits in interpreting the underlying information hidden within the chosen features. This study aims to unravel the refining interaction between classifiers and features, providing a thorough examination of their combined efficacy in the framework of predictive modeling.

E. RESULTS USING MACHINE LEARNING MODELS

In this section, we begin with a deep examination of ML models, notably SVM, DT, and KNN, as applied to the chosen feature set. Our goal is to identify the unique performance features of each model within the framework of our investigation. We want to find predictive skills and identify subtleties in these models' strengths and flaws by carefully examining and analyzing them. The next subsections conduct a detailed assessment of the experimental outcomes, providing insights into significant metrics such as precision, recall, accuracy, and F-measure. Through this inquiry, we want to contribute to a thorough knowledge of SVM, DT, and KNN models in dealing with the complexities of our selected feature set.

TABLE 4. The results of ML algorithms on features set (%).

ML Algo	Precision	Recall	Accuracy	F-Measure
SVM	96.0	93.7	94.4	94.8
DT	96.7	97.4	96.7	97.7
KNN	94.0	93.4	95.3	93.7

Table 4. Shows the machine learning domain, where three distinct models were used to detect phishing sites. SVM achieved a high accuracy of 96.0%, suggesting its ability to properly detect genuine phishing incidents. It also has a recall of 93.7%, indicating that it can detect a large proportion of real phishing websites. DT performed admirably with an accuracy of 96.7% and a recall of 97.4%, demonstrating its capacity to reduce false positives and negatives. KNN obtained a commendable 95.3% accuracy, with 94.0% precision and 93.4% recall. These measurements show the success of ML algorithms in spotting fraudulent websites.

FIGURE 9. Examining the confusion matrices for the ML models SVM, DT, and KNN reveals significant trends. SVM's accuracy score represents its ability to properly create positive predictions, whilst recall demonstrates its success in catching real positive cases. The total accuracy shows the model's correctness, whereas the F-measure indicates the balance of precision and recall. Similarly, DT displays good precision and recall, resulting in an accurate and balanced

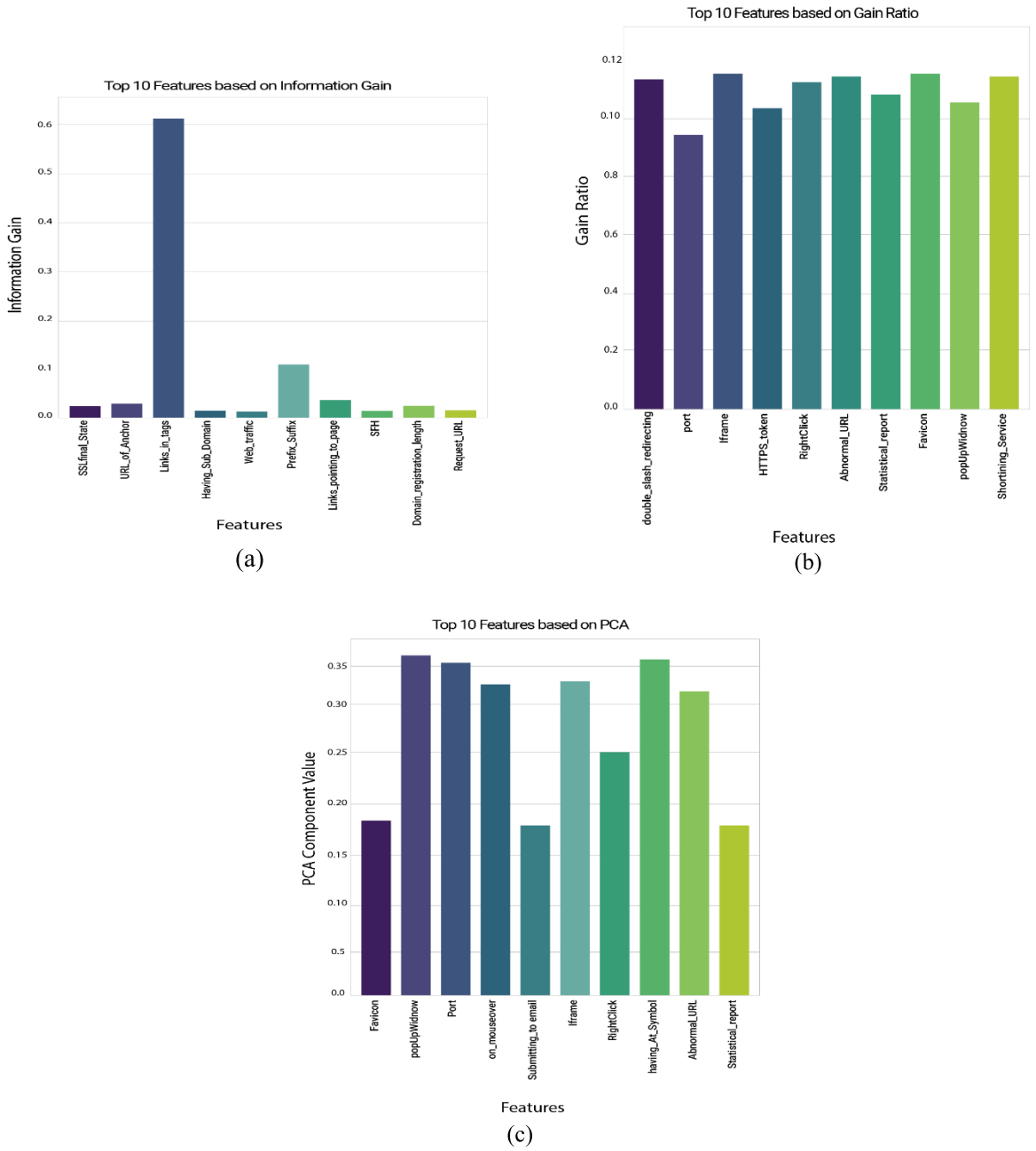


FIGURE 8. (a) IG (b) GR (c) PCA Graphical representation on top ranked feature according.

F-measure. KNN, which focuses on precision and recall, aids in accurate categorization through its unique methodology. Understanding the intricacies in the confusion matrices improves our understanding of the model's performance characteristics and helps us make educated judgments regarding their applicability for various tasks.

F. RESULTS USING DEEP LEARNING MODELS

In this section, we begin a deep investigation of Deep Learning models, specifically LSTM, RNN, and GRU, as applied to the chosen feature set. Our goal is to highlight each model's unique performance features within the framework of our investigation. We want to reveal their prediction capacities and distinguish variations in their strengths and shortcomings by carefully assessing and analyzing these models. The next subsections give an in-depth review of the experimental data, revealing light on critical metrics like precision, recall, accuracy, and the F-measure. This work contributes to a deeper knowledge of the LSTM, RNN, and GRU models while dealing with the complexities inherent in our selected feature set.

Table 5. Several deep learning algorithms, including LSTM, RNN, and GRU, demonstrated amazing accuracy in detecting phishing websites. LSTM obtained an accuracy of 96.4% and an outstanding recall of 99.0%, demonstrating its capacity to reliably detect phishing attempts while minimizing false negatives. RNN displayed balanced performance,

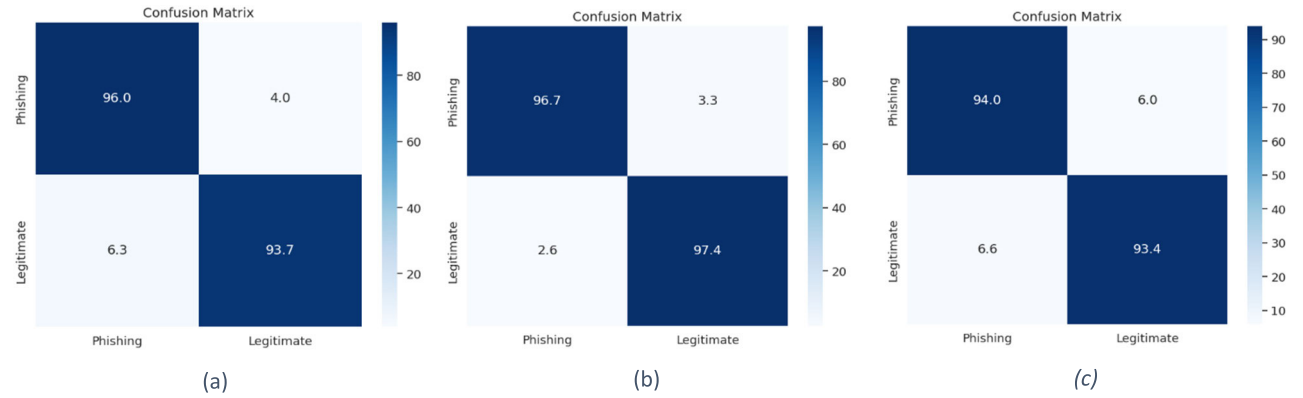


FIGURE 9. Confusion matrices using (a) SVM (b) DT (c) KNN.

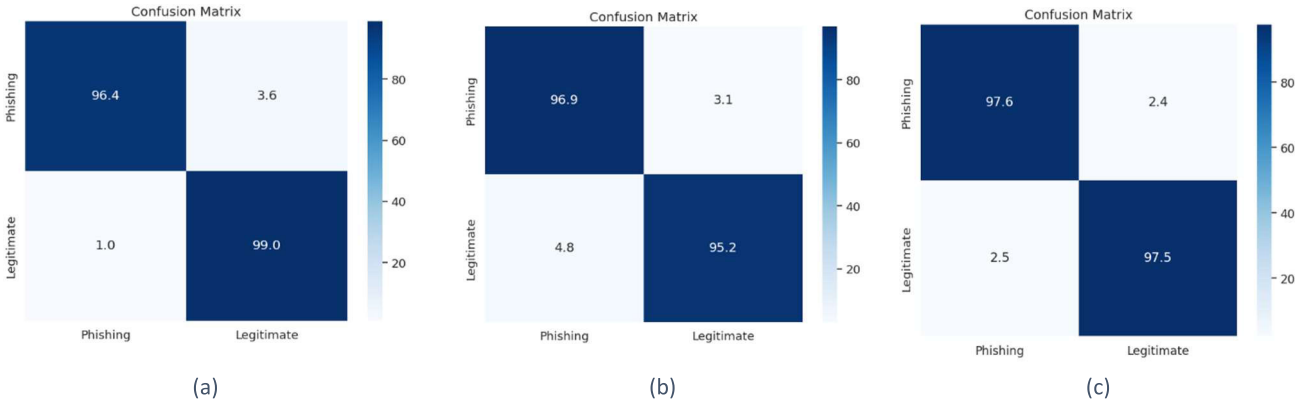


FIGURE 10. Confusion matrices using (a) LSTM (b) RNN (c) GRU.

TABLE 5. The results of DL algorithms on features set (%).

DL Algo	Precision	Recall	Accuracy	F-Measure
LSTM	96.4	99.0	97.4	97.7
RNN	96.9	95.2	95.7	96.1
GRU	97.6	97.5	97.3	97.6

with an accuracy of 96.9% and a recall of 95.2%, demonstrating its ability to retain precision without sacrificing recall. GRU demonstrated good precision (97.6%) and recall (97.5%), for an accuracy of 97.3%. These deep learning models, with their sequential learning capabilities, proved to be extremely good at detecting the complicated patterns associated with phishing sites.

FIGURE 10. The confusion matrices for the deep learning models GRU, LSTM, and RNN show noteworthy patterns. In the case of GRU, the precision score reflects its accuracy in positive predictions, whereas recall demonstrates its efficacy in catching real positive events. The total accuracy reflects the soundness of the GRU model, while the F-measure provides a more nuanced view of the balance between precision and recall. Similarly, LSTM has great precision and recall, yielding an accurate and balanced F-measure. RNN, with its emphasis on precision and recall, contributes to accurate categorization because of its unique methodology. comprehension of the intricacies in the confusion matrices improves our comprehension of the performance characteristics of GRU, LSTM, and RNN models, allowing us to make more educated decisions about their suitability for certain tasks.

G. RESULTS USING ENSEMBLE LEARNING MODELS

In this section, we'll go over Ensemble Learning approaches in detail, including how AdaBoost, RF, and XGBoost are used to our carefully picked feature set. The key goal is to reveal the distinct performance features inherent in each ensemble model within the scope of our research. Through thorough inspection and detailed analysis, we want to demonstrate the predictive capability of these models while identifying minor differences in their strengths and flaws. The next sections give a thorough analysis of experimental results, shedding light on critical metrics including precision, recall, accuracy, and the F-measure. This analytical method greatly adds to a thorough grasp of how AdaBoost, Random Forest, and XGBoost negotiate the complexity included in our chosen feature set.

Table 6. provides performance measurements for three major ensemble learning algorithms: Adaboost, RF, and XGBoost. Precision, or the accuracy of positive predictions, is 93.2% for Adaboost, 98.8% for RF, and 98.4% with XGBoost. The recall rate, which indicates the capacity to

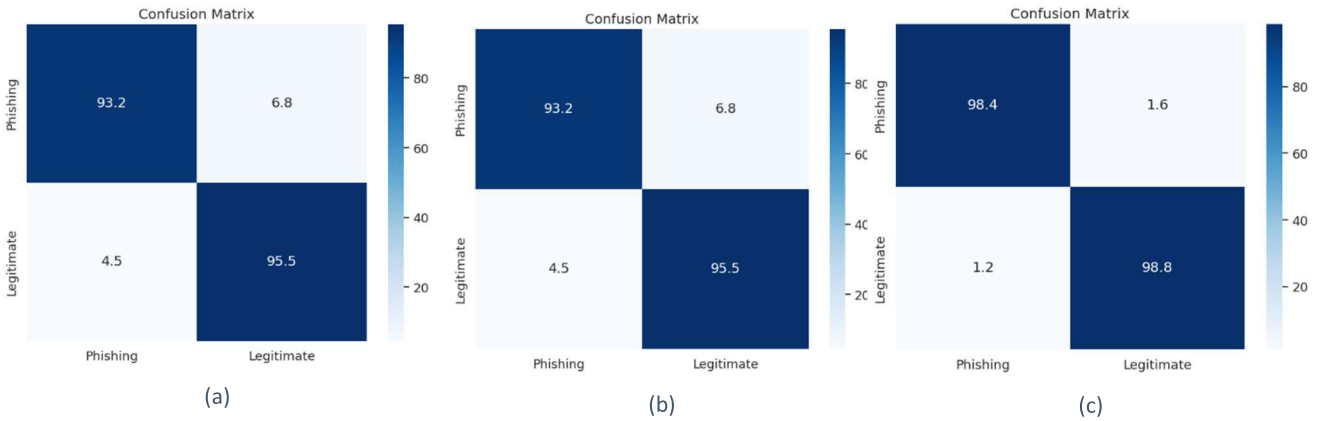


FIGURE 11. Confusion matrices using (a) Adaboost (b) RF (c) XGboost.

TABLE 6. The results of EL algorithms on features set (%).

EL Algo	Precision	Recall	Accuracy	F-Measure
Ada-Boost	93.2	95.5	93.7	94.4
RF	98.8	99.3	98.9	99.0
XG-Boost	98.4	98.8	98.4	98.6

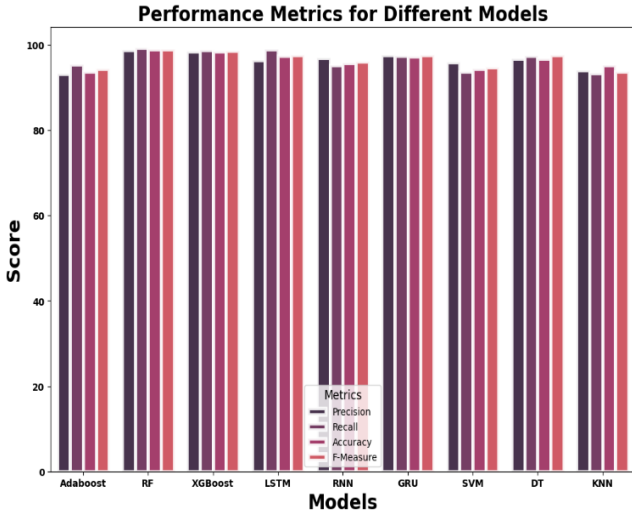


FIGURE 12. Performance metrics for different models.

catch genuine positive cases, is 95.5% for Adaboost, 99.3% for RF, and 98.8% for XGBoost. The accuracy, or total correctness of predictions, is 93.7% for Adaboost, 98.9% for RF, and 98.4% for XGBoost. F-Measure, a balanced metric of precision and recall, achieves 94.4%, 99.0%, and 98.6% for Adaboost, RF, and XGBoost, demonstrating the ensemble approaches' robustness and good performance in classification tasks. The findings show that RF has the best overall performance, outperforming in both precision and recall, while XGBoost also shows good classification ability.

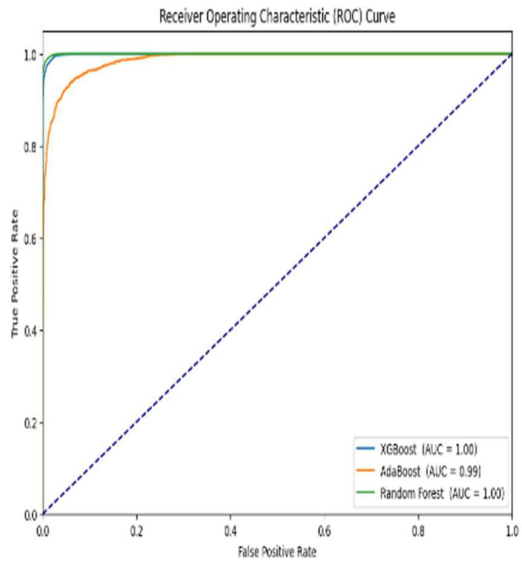
FIGURE 11. Analyzing the confusion matrices for the ensemble learning models Adaboost, RF, and XGBoost reveals significant trends. Adaboost's precision score indicates its accuracy in positive predictions, while recall demonstrates its effectiveness in catching real positive events. The overall accuracy measures Adaboost's correctness, but the F-measure offers more subtle information on the balance of precision and recall. Similarly, RF has great precision and recall, yielding an accurate and well-balanced F-measure. XGBoost, which focuses on accuracy and recall, supports accurate categorization with its unique ensemble technique. understanding the complexities of the confusion matrices improves our understanding of the performance features of Adaboost, RF, and XGBoost models, allowing us to make better decisions regarding their suitability for certain jobs.

FIGURE 12. Show us a graphical representation of all model performance matrices. The performance study demonstrates the efficiency of several ways for identifying phishing websites. The first set of data demonstrates that machine learning algorithms are capable of effectively recognizing phishing cases. The second set of data highlights the capabilities of deep learning models, focusing on their capacity to detect complicated patterns associated with phishing websites. Finally, the final set of data demonstrates the effectiveness of ensemble learning approaches in improving detection accuracy by capitalizing on the strengths of individual models. Overall, this detailed study sheds light on the various strengths and applicability of these different approaches to the difficult issue of detecting phishing websites.

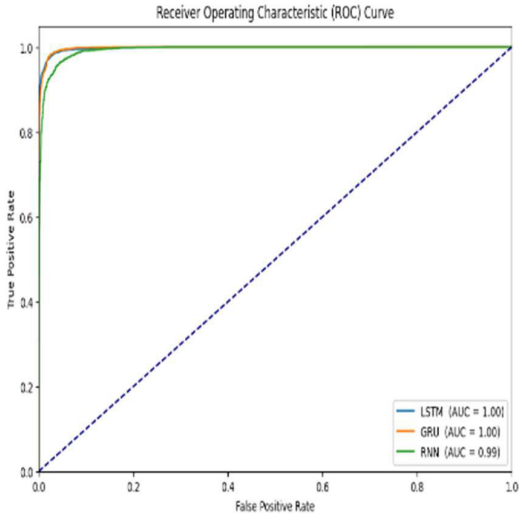
FIGURE 13. demonstrates the trade-off between sensitivity (actual positive rate) and specificity (true negative rate). A model with a bigger area under the curve (AUC) has stronger overall discriminatory power, since it captures the balance of true positive and false positive rates across various threshold settings.

H. COMPARING RESULTS WITH EXISTING METHODS

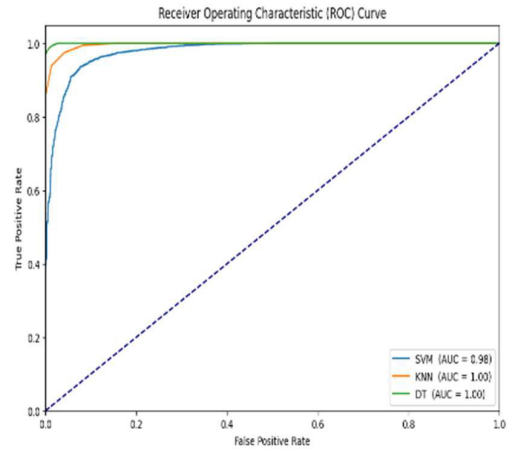
Table 4. Provides a comparative analysis of several algorithms and the accuracies that correlate with them, as



(a)



(b)



(c)

FIGURE 13. (a) EL, (b) DL, (c) ML ROC curve.

described in numerous references. Reference [44] explores ensemble techniques such as bagging, boosting, and stacking, achieving an accuracy of 95.4%. In contrast, [45] introduces the Adab-Forest PAPWDM algorithm, which demonstrates a higher accuracy of 96.5%. Reference [46] utilizes a Deep Neural Network (DNN) with the Adam optimizer, resulting in an accuracy of 96.00%. The proposed model in which DL, ML, and EL models are used after executing all the models EL models yielding the highest reported accuracy of 99%. The table provides a concise summary of these algorithms and their associated performance metrics, showcasing the feasibility of the required model in achieving superior accuracy compared to the referenced methodologies.

TABLE 7. Proposed method with existing methods.

Ref.	Algorithms	Accuracy
[44]	Ensemble bagging, boosting, stacking	95.4%
[45]	Adab-Forest PA-PWDM	96.5%
[46]	DNN +Adam	96.00%
This Study	**RF**	**99.0 %**

The core findings of the whole experiment justify the credibility of the EL models. As the RF model is considered the best Deep learning model when applied to the test dataset. The main concept of the proposed study is to extract important features and then transfer them into different DL models as input. The results clearly show that in all the comparisons being made, RF outperformed other models and existing studies when it comes to dealing with text data. Relative to other ML models, RF performs more quickly and produces better outcomes.

V. CONCLUSION AND FUTURE WORK

The increasing popularity of phishing websites stands as a significant and evolving threat within the digital domain. These platforms are particularly designed to mislead users, give in sensitive information, and propose substantial risks to cybersecurity infrastructure. Considering the scope of these dangers, it is essential to take the detection of phishing websites very seriously. This study has investigated phishing detection in great detail, evaluating the value and efficacy of a wide range of DL and ML models. By comparing the performance of several models, such as SVM, DT, RF, KNN, GRU, LSTM, RNN, and ensemble learning models like XGBoost, AdaBoost, and RF, the study established a distinction between authentic and phishing domains. Among these many models, the ensemble learning strategy that is, RF in particular has proven quite effective, with 99% accuracy. This outperforms other models stated in the existing literature.

Looking forward, future endeavors in this domain may explore additional algorithmic approaches and maintain a vigilant stance toward emerging threats. This ongoing

commitment to refinement and adaptation ensures the continual enhancement of cybersecurity measures to safeguard against the dynamic challenges posed by phishing activities.

REFERENCES

[1] C. Rupa, G. Srivastava, S. Bhattacharya, P. Reddy, and T. R. Gadekallu, "A machine learning driven threat intelligence system for malicious URL detection," in *Proc. 16th Int. Conf. Availability, Rel. Secur.*, New York, NY, USA, Aug. 2021, pp. 1–7, doi: 10.1145/3465481.3470029.

[2] J. K. Lee, Y. Chang, H. Y. Kwon, and B. Kim, "Reconciliation of privacy with preventive cybersecurity: The bright internet approach," *Inf. Syst. Frontiers*, vol. 22, no. 1, pp. 45–57, Feb. 2020, doi: 10.1007/s10796-020-09984-5.

[3] *APWG | Phishing Activity Trends Reports*. Accessed: Jun. 11, 2023. [Online]. Available: https://apwg.org/trendsreports/

[4] A. A. Alshdadi, A. S. Alghamdi, A. Daud, and S. Hussain, "Blog backlinks malicious domain name detection via supervised learning," *Int. J. Semantic Web Inf. Syst.*, vol. 17, no. 3, pp. 1–17, Jul. 2021, doi: 10.4018/ijswis.2021070101.

[5] S. Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, "A survey of intelligent detection designs of HTML URL phishing attacks," *IEEE Access*, vol. 11, pp. 6421–6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[6] A. K. Jain and B. B. Gupta, "PHISH-SAFE: URL features-based phishing detection system using machine learning," in *Cyber Security* (Advances in Intelligent Systems and Computing), vol. 729, M. U. Bokhari, N. Agrawal, and D. Saini, Eds., Singapore: Springer, 2018, pp. 467–474, doi: 10.1007/978-981-10-8536-9_44.

[7] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting phishing attacks," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102328, doi: 10.1016/j.cose.2021.102328.

[8] M. K. Hayat, A. Daud, A. A. Alshdadi, A. Banjar, R. A. Abbasi, Y. Bao, and H. Dawood, "Towards deep learning prospects: Insights for social media analytics," *IEEE Access*, vol. 7, pp. 36958–36979, 2019, doi: 10.1109/ACCESS.2019.2905101.

[9] A. K. Murthy and Suresha, "XML URL classification based on their semantic structure orientation for web mining applications," *Proc. Comput. Sci.*, vol. 46, pp. 143–150, Jan. 2015, doi: 10.1016/j.procs.2015.02.005.

[10] P. George and P. Vinod, "Composite email features for spam identification," in *Cyber Security* (Advances in Intelligent Systems and Computing), M. U. Bokhari, N. Agrawal, and D. Saini, Eds., Singapore: Springer, 2018, pp. 281–289, doi: 10.1007/978-981-10-8536-9_28.

[11] V. Rajasekar, J. Premalatha, K. Sathya, S. D. Raakul, and M. Saracevic, "An enhanced anti-phishing scheme to detect phishing website," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1055, no. 1, Feb. 2021, Art. no. 012077, doi: 10.1088/1757-899x/1055/1/012077.

[12] T. N. Bac, P. T. Duy, and V.-H. Pham, "PWDGAN: Generating adversarial malicious URL examples for deceiving black-box phishing website detector using GANs," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. Netw. Technol. (ICMLANT)*, Dec. 2021, pp. 1–4, doi: 10.1109/ICMLANT53170.2021.9690540.

[13] M. Kihal and L. Hamza, "Robust multimedia spam filtering based on visual, textual, and audio deep features and random forest," *Multimedia Tools Appl.*, vol. 82, no. 26, pp. 40819–40837, Nov. 2023, doi: 10.1007/s11042-023-15170-x.

[14] Y. Zhang, J. I. Hong, and L. F. Cranor, "CANTINA: A content-based approach to detecting phishing web sites," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 639–648, doi: 10.1145/1242572.1242659.

[15] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han, and J. Wang, "The application of a novel neural network in the detection of phishing websites," *J. Ambient Intell. Humanized Comput.*, vol. 15, no. 3, pp. 1865–1879, Mar. 2024, doi: 10.1007/s12652-018-0786-3.

[16] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, Mar. 2018, doi: 10.1016/j.dss.2018.01.001.

[17] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*. New York, NY, USA: Association for Computing Machinery, Sep. 2008, pp. 1–6, doi: 10.1145/1460877.1460905.

[18] J.-L. Chen, Y.-W. Ma, and K.-L. Huang, "Intelligent visual similarity-based phishing websites detection," *Symmetry*, vol. 12, no. 10, p. 1681, Oct. 2020, doi: 10.3390/sym12101681.

[19] S. Haruta, F. Yamazaki, H. Asahina, and I. Sasase, "A novel visual similarity-based phishing detection scheme using hue information with auto updating database," in *Proc. 25th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2019, pp. 280–285, doi: 10.1109/APCC47188.2019.9026498.

[20] E. R. S. and R. Ravi, "A performance analysis of software defined network based prevention on phishing attack in cyberspace using a deep machine learning with CANTINA approach (DMLCA)," *Comput. Commun.*, vol. 153, pp. 375–381, Mar. 2020, doi: 10.1016/j.comcom.2019.11.047.

[21] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight URL-based phishing detection," *Future Internet*, vol. 13, no. 6, p. 154, Jun. 2021, doi: 10.3390/fi13060154.

[22] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Comput.*, vol. 23, no. 12, pp. 4315–4327, Jun. 2019, doi: 10.1007/s00500-018-3084-2.

[23] E. Buber, B. Diri, and O. K. Sahingoz, "Detecting phishing attacks from URL by using NLP techniques," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 337–342, doi: 10.1109/UBMK.2017.8093406.

[24] E. Buber, B. Diri, and O. K. Sahingoz, "NLP based phishing attack detection from URLs," in *Intelligent Systems Design and Applications* (Advances in Intelligent Systems and Computing), vol. 736, A. Abraham, P. Kr. Muhuri, A. K. Muda, and N. Gandhi, Eds., Cham, Switzerland: Springer, 2018, pp. 608–618, doi: 10.1007/978-3-319-76348-4_59.

[25] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Comput. Appl.*, vol. 25, no. 2, pp. 443–458, Aug. 2014.

[26] W. Khan, A. Daud, F. Alotaibi, N. Aljohani, and S. Arafat, "Deep recurrent neural networks with word embeddings for Urdu named entity recognition," *ETRI J.*, vol. 42, no. 1, pp. 90–100, Feb. 2020, doi: 10.4218/etrij.2018-0553.

[27] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web*, May 2005, p. 1060, doi: 10.1145/1062745.1062868.

[28] O. K. Sahingoz, E. BUBEr, and E. Kugu, "DEPHIDES: Deep learning based phishing detection system," *IEEE Access*, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[29] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in *Proc. IEEE 12th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2018, pp. 300–301, doi: 10.1109/ICSC.2018.00056.

[30] S. Kazi, S. Khoja, and A. Daud, "A survey of deep learning techniques for machine reading comprehension," *Artif. Intell. Rev.*, vol. 56, no. S2, pp. 2509–2569, Nov. 2023, doi: 10.1007/s10462-023-10583-4.

[31] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, "Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends," *Natural Lang. Process. J.*, vol. 4, Sep. 2023, Art. no. 100026, doi: 10.1016/j.nlp.2023.100026.

[32] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Inf. Secur.*, vol. 13, no. 6, pp. 659–669, Nov. 2019, doi: 10.1049/iet-ifs.2019.0006.

[33] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, Jan. 2021, doi: 10.1007/s11235-020-00733-2.

[34] A. Aljofey, Q. Jiang, A. Rasool, H. Chen, W. Liu, Q. Qu, and Y. Wang, "An effective detection approach for phishing websites using URL and HTML features," *Sci. Rep.*, vol. 12, no. 1, p. 8842, May 2022, doi: 10.1038/s41598-022-10841-5.

[35] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Oct. 2019, doi: 10.1155/2019/2595794.

[36] 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, doi: 10.1109/ACCESS.2024.3351946.

[37] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI meta-learners and extra-trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 8, pp. 142532–142542, 2020, doi: 10.1109/ACCESS.2020.3013699.
[38] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *J. Enterprise Inf. Manage.*, vol. 36, no. 3, pp. 747–766, Apr. 2023, doi: 10.1108/jeim-01-2020-0036.
[39] A. Odeh and I. Keshta, "PhiBoost—A novel phishing detection model using adaptive boosting approach," *Jordanian J. Comput. Inf. Technol.*, vol. 7, no. 1, p. 64, 2021, doi: 10.5455/jjcit.71-1600061738.
[40] S. Anupam and A. K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms," *Telecommun. Syst.*, vol. 76, no. 1, pp. 17–32, Jan. 2021, doi: 10.1007/s11235-020-00739-w.
[41] V. E. Adeyemo, A. O. Balogun, H. A. Mojeed, N. O. Akande, and K. S. Adewole, "Ensemble-based logistic model trees for website phishing detection," in *Advances in Cyber Security* (Communications in Computer and Information Science), M. Anbar, N. Abdullah, and S. Manickam, Eds., Singapore: Springer, 2021, pp. 627–641, doi: 10.1007/978-981-33-6835-4_41.
[42] M. Sabahno and F. Safara, "ISHO: Improved spotted hyena optimization algorithm for phishing website detection," *Multimedia Tools Appl.*, vol. 81, no. 24, pp. 34677–34696, Oct. 2022, doi: 10.1007/s11042-021-10678-6.
[43] A. Mandadi, S. Boppana, V. Ravella, and R. Kavitha, "Phishing website detection using machine learning," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT)*, Apr. 2022, pp. 1–4, doi: 10.1109/I2CT54291.2022.9824801.
[44] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Phishing website detection: An improved accuracy through feature selection and ensemble learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 252–257, 2019, doi: 10.14569/ijacsa.2019.0100133.
[45] Y. A. Alsariera, A. V. Elijah, and A. O. Balogun, "Phishing website detection: Forest by penalizing attributes algorithm and its enhanced variations," *Arabian J. Sci. Eng.*, vol. 45, no. 12, pp. 10459–10470, Dec. 2020, doi: 10.1007/s13369-020-04802-1.
[46] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, "Smart phishing detection in web pages using supervised deep learning classification and optimization technique Adam," *Wireless Pers. Commun.*, vol. 118, no. 4, pp. 3549–3564, Jun. 2021, doi: 10.1007/s11277-021-08196-7.

UME ZARA received the master's degree in computer science from COMSATS University Islamabad, Wah Campus, Wah, Pakistan. Her research interests include data mining, machine learning, information retrieval, and sentiment analysis.

KASHIF AYYUB received the master's degree in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 2002. He is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah, Pakistan. His research interests include algorithms, machine learning, and sentiment analysis

HIKMAT ULLAH KHAN received the master's and Ph.D. degrees in computer science from International Islamic University, Islamabad. He has been an Active Researcher for the last ten years. He is currently a Professor/Chairperson of the Department of Information Technology, University of Sargodha, Pakistan. He has authored more than 50 papers in top peer-reviewed journals and international conferences. His research interests include social web mining, semantic web, data science, information retrieval, and scientometrics. He is an editorial board member of several prestigious impact factor journals.

ALI DAUD received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in July 2010. He is currently a Full Professor with the Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates. He has 13 years' post-Ph.D. experience of teaching, supervision, and research at B.S., M.S., and Ph.D. level. He has published more than hundred research papers in reputed international impact factor journals and conferences. He has taken part in many research projects and have written and acquired many research funding's. He has proven and extensive experience in data mining, artificial intelligence (machine learning/deep learning) applications to social networks, data science, and natural language processing, and the Internet of Things.

TARIQ ALSAHFI received the B.S. degree in computer science from King AbdulAziz University, Saudi Arabia, in 2011, and the M.S. and Ph.D. degrees in computer science from The University of Texas at Arlington, USA, in 2020. He became an Assistant Professor with the Department of Information Systems and Technology, University of Jeddah, Saudi Arabia. His current research interests include data science, deep learning, machine learning, geographical information systems, trajectory data, and enhance road traffic safety in intelligent transportation systems.

SAIMA GULZAR AHMAD received the master's degree from COMSATS University Islamabad, Wah Campus, Pakistan, in 2012, and the Ph.D. degree in computer science (distributed computing) from Universiti Malaya, Malaysia, in 2017. She is currently an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Wah Campus. Her professional affiliations are with Pakistan Engineering Council (PEC) and IEEE. She is HEC approved supervisor. She is the author of quality research publications. Her research interests include parallel and distributed computing, heterogeneous computer networks (cloud/grid), machine learning, and artificial intelligence.