IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.Doi Number

Phishing Website Detection Using Deep Learning Models

Ume Zara [1], Kashif Ayub[1], Hikmat Ullah Khan[2], Ali Daud[3], Tariq Alsahfi[4], Saima Gulzar[1]
[1]Department of Computer Science, COMSATS University Islamabad - Wah Campus, Islamabad, Pakistan.
[2]Department of Information Technology, University of Sargodha, 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, Saudi Arabia

Corresponding Authors: Ali Daud (alimsdb@gmail.com); Hikmat Ullah Khan (dr.hikmat.niazi@gmail.com)

ABSTRACT Detecting website phishing is crucial for protecting sensitive information, including personal data and financial details. It helps maintain trust and reputation for both businesses and users while preventing malware infections and cyber-attacks. This research addresses the 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. To evaluate the efficacy of the proposed method, the top features are ranked using information gain, gain ratio, and Principal Component Analysis (PCA) which help to classify a website as phishing or non-phishing. The statistical exploratory data analysis using data visualization techniques helps to provide deep insights into the dataset. The proposed system is trained using a dataset that covers 11,055 websites. The ensemble learning model applied achieved an impressive accuracy of as high as 99% in predicting phishing websites, surpassing previous models, and setting a new benchmark in the field. The findings highlight the effectiveness of the proposed model, and such advanced approaches may help to maintain the stability and security of online platforms.

INDEX TERMS Data Visualization, Deep Learning, Feature Selection, Machine Learning, Website Phishing Detection.

I. INTRODUCTION

Communication technology has been the major contender in this development, continually altering to meet consumers' ever-changing needs, giving real-time interactions, information access, and a worldwide feeling of connectedness. However, there are certain opponents who are adapting to exploit in this interconnected society and develop illegal ways for their ill-intensions 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. Website phishing detection also ensures compliance with legal requirements, reduces the spread of phishing, and contributes to educating users about online security threats. Safeguarding the broader digital ecosystem is another significant benefit, as it helps maintain the stability and security of websites.

The process of website phishing is usually conducted in a typical manner. The phisher sends out bait that mimics the genuine website and watches for victims. When a user falls for the phisher's fraud and believes the mimicked page, the phisher wins. Figure 1 shows 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 recently focused their attention on phishing attempts [2]. Many organizations, notably NS focus and the Anti-Phishing Working Group (APWG), conducted surveys of attacks.

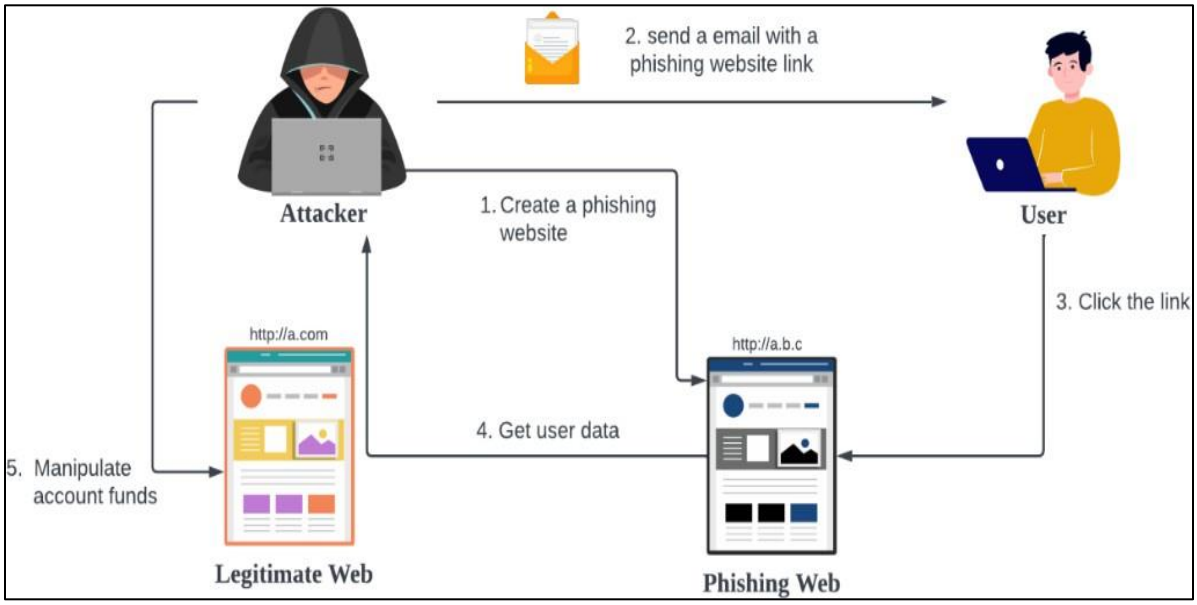


FIGURE 1. **Life cycle of phishing attack**

There are few organizations and companies which examine phishing attacks such as Bit-Defender, Symantec, McAfee, VeriSign, and other security product, service-oriented, law enforcement, trade, and international treaty organizations which are among the members of the non-profit, international APWG group. The reports of phishing trends in cyberspace are produced by APWG. As per recent APWG report (2023)[1] as shown Figure 2, states that there were 1,077,501 frauds 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 the last quarter of 2023, accounting for 42.8% of all phishing attempts. Every quarter, there is a rise in phone phishing, often known as voice phishing, also known as "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. The attacks may come via websites, emails, or malicious software.

Figure 2 shows that 2023 was the most active year for the website phishers as around half a million phishing URLs were found during the specified year.

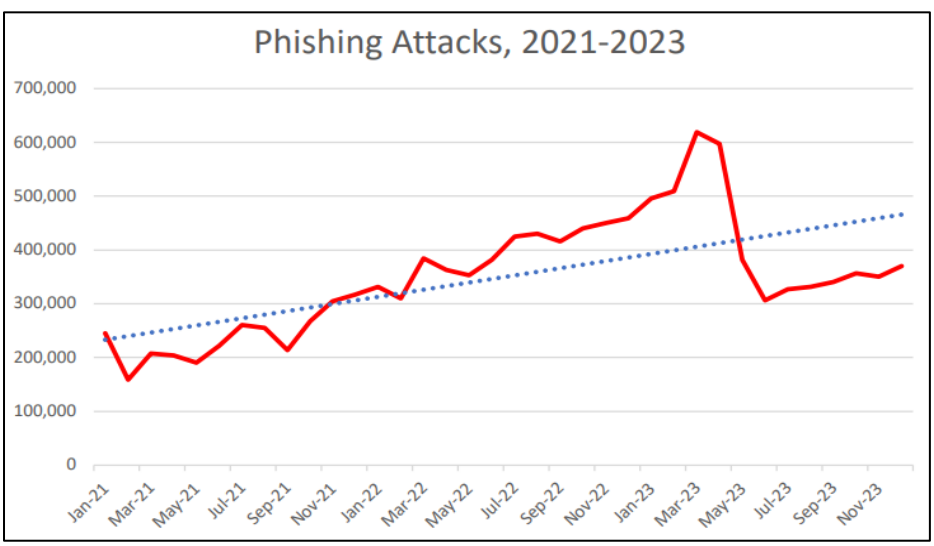


FIGURE 2. **Life Number of phishing attacks (2021 – 2023)**

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

A. BACKGROUND INFORMATION

Figure 3 shows the structure of Uniform Resource Locator. It consists of the following seven components: protocol, top-level domain, malicious domain name [3], 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 the 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[4].

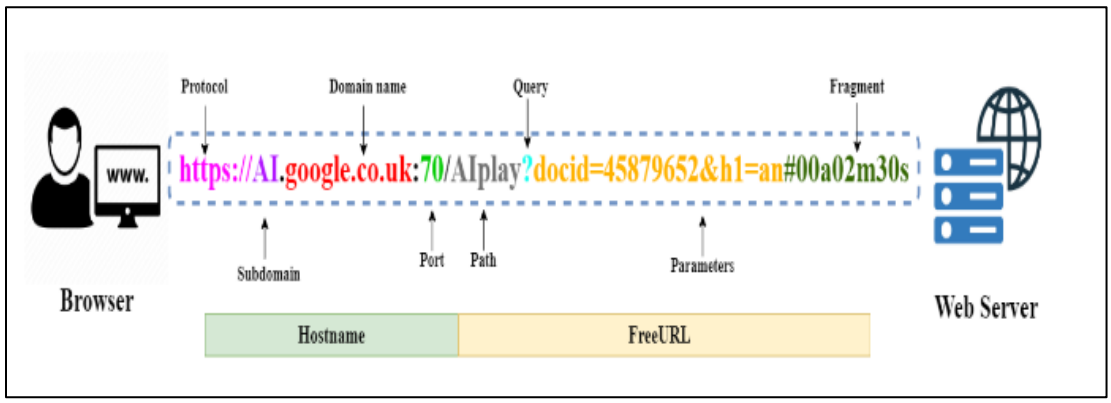


FIGURE 3. **Anatomy of URL**

According to Cybersecurity Ventures, cybercrime is expected to cost 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 greatest complained-about kind of cybercrime, and consequential in damages exceeding USD 54 million[2]. For web users it is important to be vigilant to differentiate between legitimate and fraudulent websites. 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 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 set.
- 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 effectiveness of all models.

The remaining portion of the study is structured as goes along. 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. RELTED 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.

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

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. In [5], a whitelist-based system is developed that creates a whitelist by keeping track on the IP address of each site with a login page where people may enter their details. The Windows 2008 system alerts the users about the inconsistency of registered information details when they utilize the login interface. Therefore, when consumers visit reputable websites for the first time, this system immediately mistrusts them. In another study [6], a system is developed that automatically updates and maintains the whitelist regularly, alerting the users when they come across a phishing website. The taking out of properties concealed the connection between 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 of 86.02%, with a FN score of 1.48% [7]. Based on the data of URLs referred to be phishing websites, blacklists were compiled. The 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 [8] 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, [9] 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 [10], 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 [11]. Blacklist-based systems verify the address against the list and deny access to URLs that are not 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 are more frequent in phishing URLs [12]. The purpose of research using this type of technology is to categorize data more actively by using useful qualities. These systems follow specified rules. When these rules are employed to train the system, the accuracy ratio improves. In the system named CANTINA [13] , to identify phishing assaults rules, and Term Frequency - Inverse Document Frequency (TF-IDF) were employed. Furthermore, some traits and guidelines were used to develop models in related investigations. Feng *et al* [14] created the client-side anti-phishing program Bogus Biter, 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, Bogus Biter makes it possible for the authentic website to quickly detect the stolen credentials. Mohammed *et al* [15] 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 are few visible variations between them. Finding simpler to discover variations using Image processing techniques, which are not

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

immediately seen by consumers. The degree of resemblance found determines whether a website is phishing. Within the literature, as in the study [16], Certain investigations identify distinctions based on fundamental commonalities.

Chen et al. [17] 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 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 [18]. 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 recognize phishing sites is to try to impersonator 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 ON 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 users' 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 CANTINA initiative [19] was completed by the application of machine learning. TF-IDF and heuristic methods indicated that they found 90% accuracy rates. In [20], by categorizing URL data including its length, amount of special characters, web address, directory, and file name were 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 [21], 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 [22], 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 [23], 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%. The researchers put in place a mechanism for detecting phishing [24] 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. Consequently, it was reported that real-world implementation would take far longer.

In [25], 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. In another relevant study [26], email headers, content URLs, HTML content, that 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 study [27], 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 [28], 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. Another study [29] 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, most of the research is currently focused on these low-resource languages. Existing 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 [30].Ahmed and Ali created a DNN and genetic algorithm (GA) intelligence phishing detection solution [31]. 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, GA-DNN model achieved a low 89% accuracy. The parameters like quantity data used for training have a substantial impact on DL model accuracy [32]. Aljofey *et al* published a CNN architecture for the detection of phishing constructed on URLs in 2020. [33]. 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 [34].

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 RNN, using the word embedding technique. 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 [35].

Table 1 compares significant cutting-edge solutions. Although accuracy varied 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 produced 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 exactly accurate for this task, has an advantage here. It works in real-time, and it does not 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.

TABLE 1
COMPARISON OF MAJOR STATE-OF-THE-ART SOLUTION

Ref	Years	Type	Model	Dataset	Accuracy (%)
[23]	2018	Hybrid	NLP+RF	Phishing websites	97.2
[34]	2019	Deep learning	RNN + CNN	Examined Alexa, Phish Tank; 490,408 entries, half phishing, half legitimate. Utilized word embedding for content analysis.	95.79
[36]	2020	Hybrid	LBET (LR + extra tree)	Phishing Website Dataset	97.57
[35]	2020	Deep learning	Convolutional auto encoder + DNN	Website (Clients' daily requests, PhishTank) 6116 instances of rule-based features.	89.00
[37]	2020	Single	CNN+LSTM	Phishing website dataset	93.28
[19]	2020	Single	SVM+ CANTINA	Dataset collected from real phishing cases	96.0
[38]	2021	Single	Adaptive Boosting	Phish Tank, Google Search; unspecified data quantity; 30 details per information.	98.30
[39]	2021	Hybrid	Grey wolf optimizer + SVM	Websites (PhishTank) 1353 instances: 548 legitimate URLs; 805 phishing URLs	90.38
[40]	2021	Hybrid	Bagging + LMT	Phishing Website Dataset	97.42
[6]	2021	Single	automated white list	Six different Phishing URLs dataset	95.0
[20]	2021	Single	NB+DT+RF+SVM	Experiments consist of active phishing attacks + Google Safe Browsing	97.21
[41]	2022	Hybrid	ISHO + SVM	Phishing Website Dataset	98.64
[42]	2022	Single	SVM + NB + RF	Phishing Dataset	92.0
[31]	2023	Deep learning	Genetic algorithm (GA) + DNN	Phishing Website Dataset	89.50
[33]	2023	Deep learning	LSTM+CNN	phish Tank URL features, with 20,000 records of 80 features,	97.6

[4] https://www.amazon.com/b?node=21576558011

[5] https://www.phishtank.com/index.php

III. PROPOSED RESEARCH METHODOLOGY

The proposed framework is explained in this section sharing different classifiers applied in this study, feature selection strategies, the selected data set, and the PEM implemented to conduct the tests. In the first stage, to select phishing website dataset. The top features are evaluated using feature selection techniques like IG, GR, and PCA. After determining the significance of characteristics, data is separated into test and training sets. Classifiers' performance for detecting phishing websites is assessed using evaluation techniques. Figure 4 shows the proposed framework.

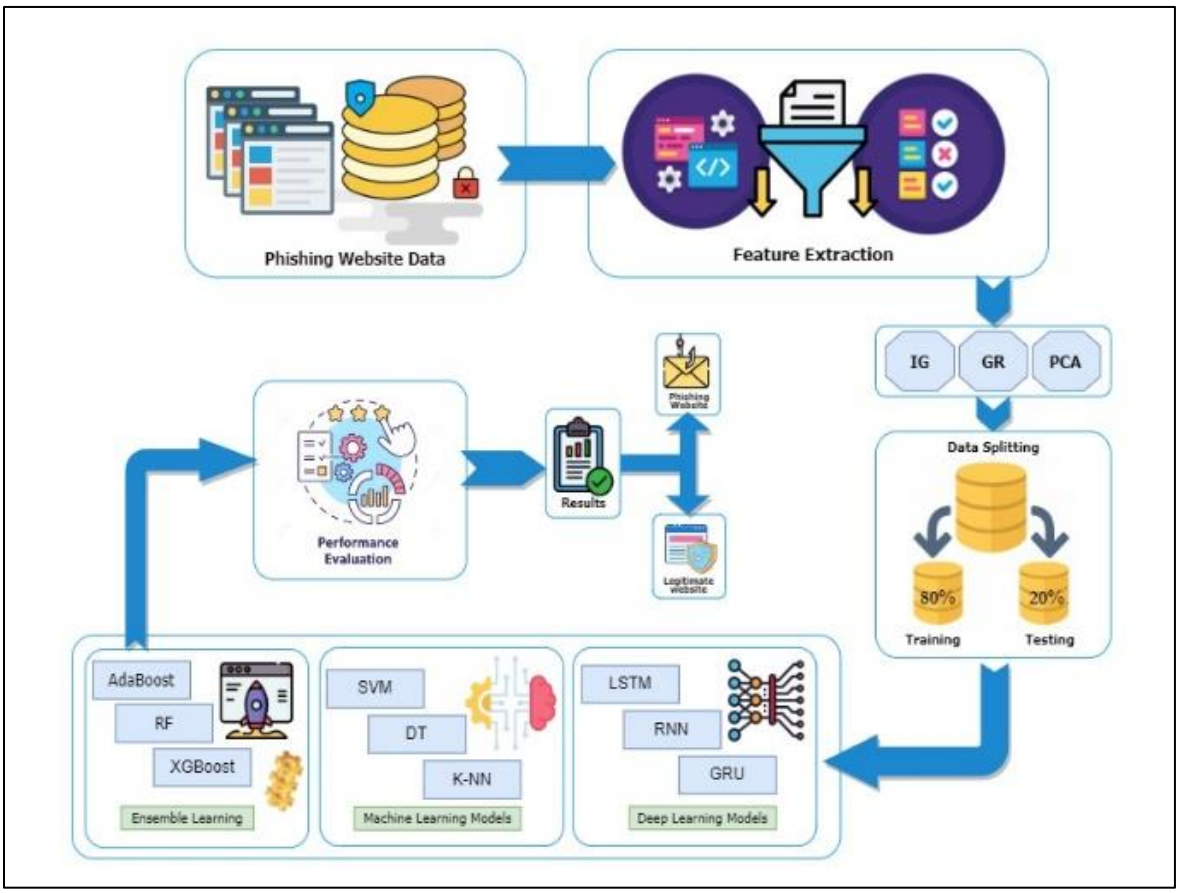


FIGURE 4. **Anatomy Steps of proposed research framework**

A. *Applied Algorithms*

Here we discuss the 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, SVM, Decision Tree, and K-Nearest Neighbors are used. The SVM algorithm is widely recognized for its ability to find the best classifier for given data, achieving excellent generalization performance. 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.

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 to enhance 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 regularized 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 are particularly good at handling time series data and sequential patterns. RNNs are intended to manage sequences of inputs by retaining hidden states that capture information from past time steps, allowing them to represent temporal relationships. However, classic RNNs have the vanishing gradient problem, which restricts their capacity to obtain long-range relationships. DL techniques, such as LSTM, RNN, and GRU, have revolutionized sequential data processing by properly capturing temporal linkages. These algorithms excel at handling time-series data and sequential patterns. RNNs are designed to manage sequences of inputs by storing hidden states that contain information from previous time steps, allowing them to reflect temporal connections. However, conventional RNNs suffer from the vanishing gradient problem, which bounds their ability to obtain long-term associations.

B. *Feature Reduction Methods*

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

1) Information Gain(IG): IG is often referred to as mutual information. It reduces the bias towards multi-valued features by considering 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.

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

In the equation 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 $\{a_1 \ldots a_v\}$ F would be in v partitions $\{F_1 \ldots F_v\}$ Eq (2) is used to calculate entropy information.

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

In equation (2), $|Fj|/|F|$ 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:

$$IG(F) = info(F) - info_A(F) \quad (3)$$

In equation (3), attributes having a top value of IG are used to categorize the document into the specified class.

2) 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. Training document is divided into v divisions based on the number of outputs for a feature.

$$\text{equalnfo}_A(E) = -\frac{\sum_{j=1}^{v}|Ej|}{|E|} X\log_2|Ej|/|E| \quad (4)$$

In equation (4), high $splitinfo$ information is sparse and consistent. Few partitions maintain peak values. GR is computed as follows:

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

3) 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:

$$Z = X \cdot W \quad (6)$$

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 co-variance matrix.

The co-variance matrix Σ is calculated as:

$$\sum = \frac{1}{n-1} \cdot (X - \bar{X})^T \cdot (X - \bar{X}) \quad (7)$$

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

4) Dataset

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

C. *Performance Evaluation Measures (PEM)*

Several PEM from the phishing site data set is 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.

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

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

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

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

$$F1 = \frac{2 \times P \times R}{P+R} \quad (11)$$

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.

IV. RESULTS AND DISCUSSION

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

A. Feature Engineering

Selecting the best features is a significant and difficult process, particularly when it comes to creating precise projections and identifications in a fresh dataset. The method entails 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 assigning is critical for accurately categorizing and comprehending the properties of websites in the dataset provided in Table 2.

TABLE 2
DATASET ATTRIBUTES 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. In 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 as follows: -1,1

B. Feature Attributes Visualization

In this part, we take a closer look at the features in a dataset by creating pictures or graphs. The visuals help us see how the different features are spread out, if they are 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 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 are frequently malicious on sites because they may be used to steal user credentials. 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.

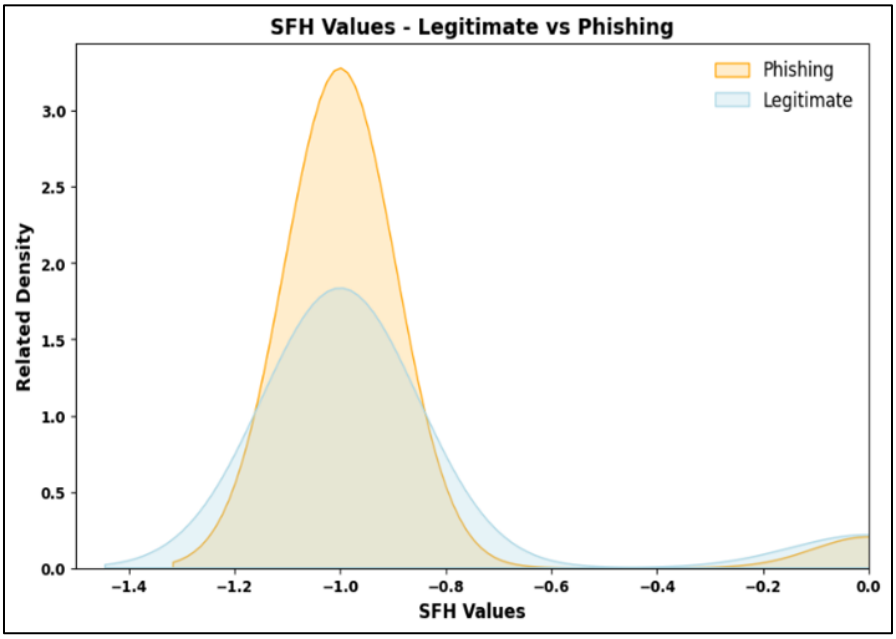


FIGURE 5. **SFH and result visualization**

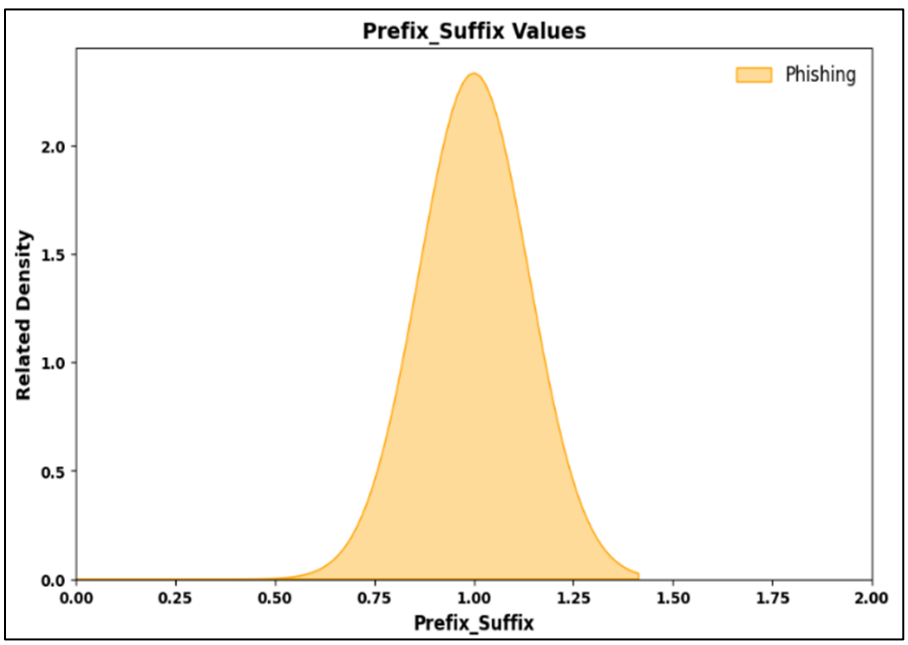


FIGURE 6. **Prefix Suffix and results**

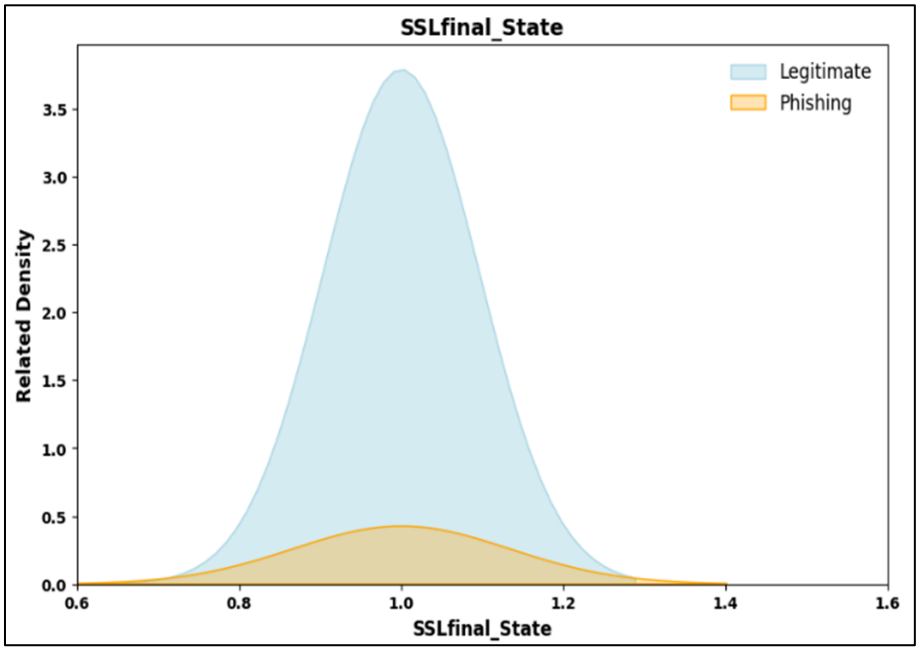


FIGURE 7. **SSLFinal_State and result visualization**

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 website.

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 represents the Secure Sockets Layer's final state, shows a delicate balance of IG and GR, indicating that it is important in the dataset. Furthermore, its correlation with the Favicon in PCA makes a significant contribution to the total variance, indicating probable linkages between SSLfinal_State and Favicon-related variables as shown in table 3.

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

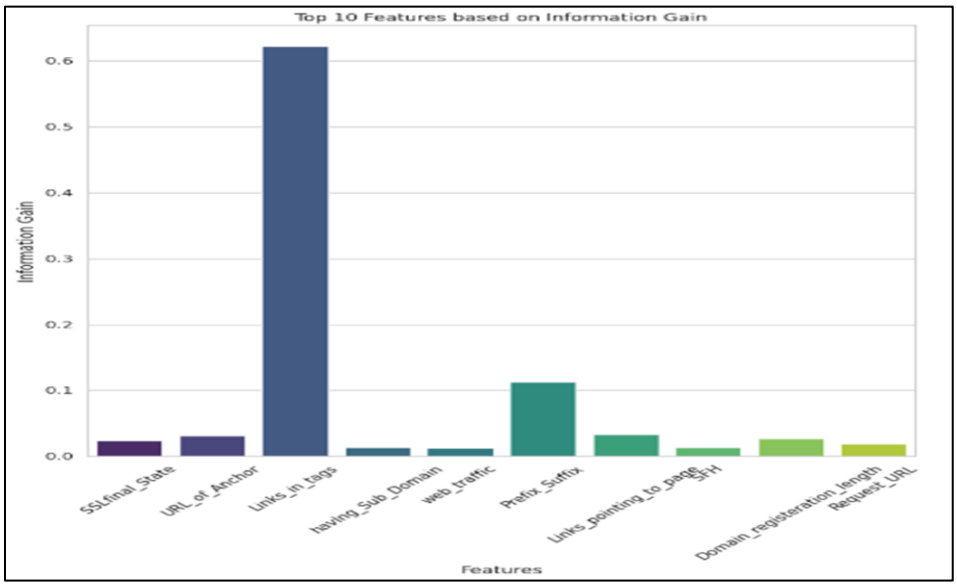
Moving on to URL_of_Anchor, the feature has a separate profile with a slightly higher IG but a lower GR. The correlation with popUpWidnow in PCA highlights its significant contribution to the dataset's variability, highlighting the possible influence of URL_of_Anchor on pop-up window performance. This refinement evaluation assists the researchers in determining not just the individual significance of characteristics, but also their interactions in various analytical situations.

Links_in_tags emerge as a noteworthy feature, with a remarkably high IG, suggesting its significant importance. In PCA, the alignment with the port indicates a relationship between the number of linkages in tags and particular port-related properties. This knowledge can help researchers grasp the underlying trends in the dataset, helping them improve their hypotheses and model-building procedures.

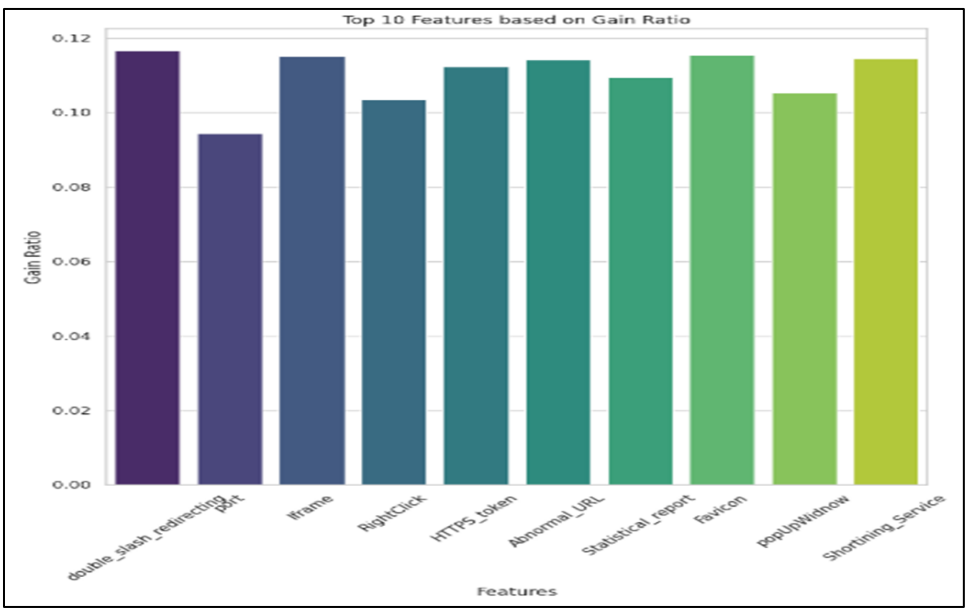
On the other hand, having_Sub_Domain has a lesser IG but compensates with a moderate GR. Its relationship with

on_mouseover in PCA enhances the analysis by identifying a link between subdomain existence and mouseover interactions. This dual-metric method enables a more complex assessment of feature relevance, accounting for both the breadth and depth of their influence. web_traffic, with its moderate IG and GR, is like Submitting_to_email in PCA. This connection throws light on correlations between online traffic patterns and email submission behaviors, offering useful insights into the dynamics of user interactions.

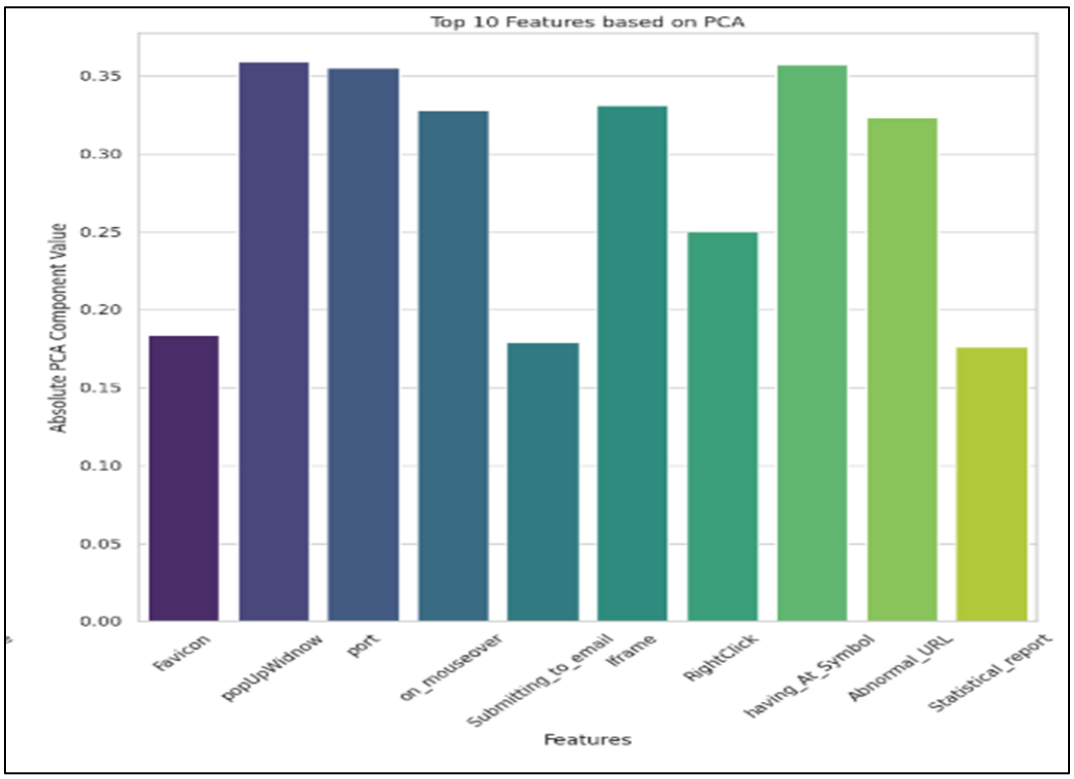
Figure 8 Graphical depiction of the top ten ranking features based on IG GR and PCA Finally, this complete feature analysis, which includes numerous metrics and PCA, ranks features based on their relevance while also revealing deep correlations between them. Researchers may use this knowledge to make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



(a)



(b)



(c)

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

D. *Classification Results*

In this section, 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.

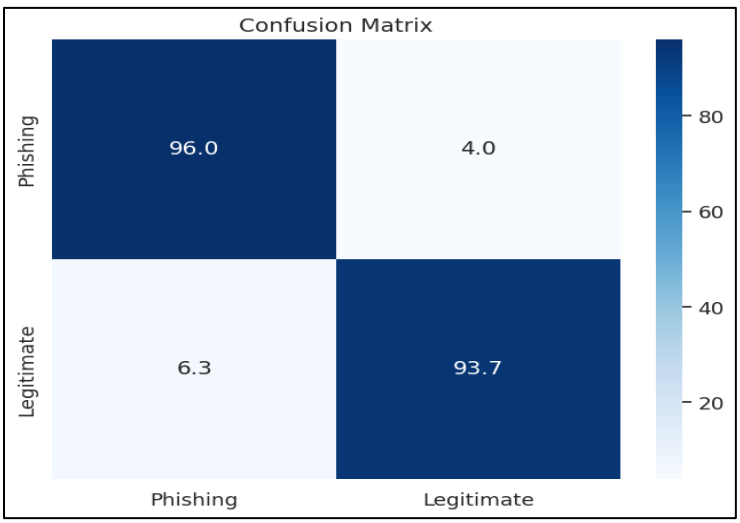
Table 4 shows the machine learning domain, where three distinct models were used to detect phishing sites. SVM achieved a high accuracy of 96.0%, suggesting its ability to properly detect genuine phishing incidents. It also has a recall of 93.7%, indicating that it can detect a large proportion of real 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.

TABLE 4
THE RESULTS OF ML ALGORITHMS ON FEATURES SET (%)

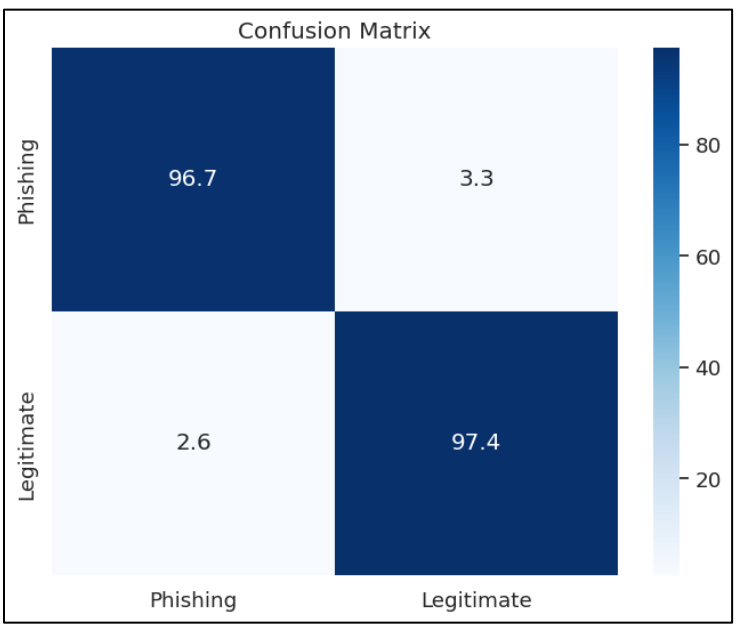
ML Algo	Precision	Recall	Accuracy	F-Measure
SVM	96.0	93.7	94.4	94.8
DT	96.7	97.4	96.7	97.7
KNN	94.0	93.4	95.3	93.7

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 positive cases. The total accuracy shows the model's correctness, whereas the F-measure

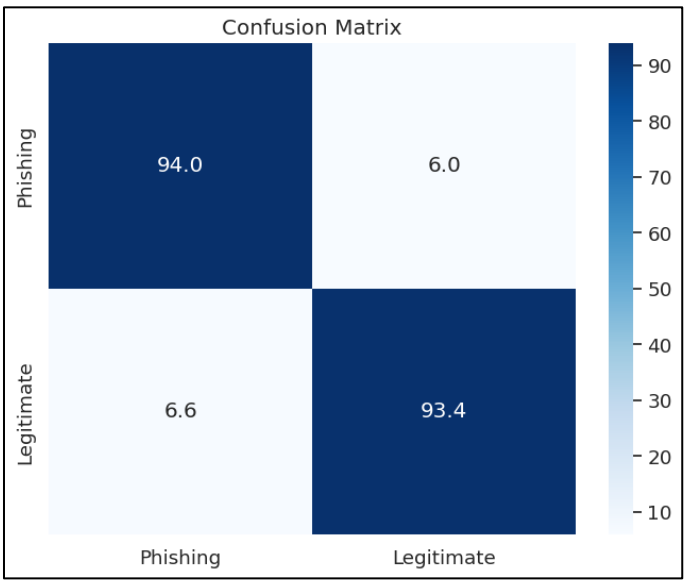
indicates the balance of precision and recall. Similarly, DT exhibits good precision and recall, resulting in an accurate and balanced F-measure. KNN, which focuses on precision and recall, aids in accurate categorization through its unique methodology. Understanding the intricacies in the confusion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



(a)



(b)



(c)

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

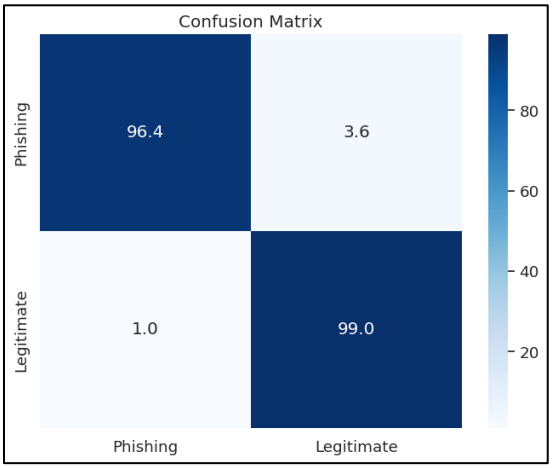
F. *Results using Deep Learning Models*

In this section, we discuss deep investigation of results 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, 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 in detecting the complicated patterns associated with phishing sites.

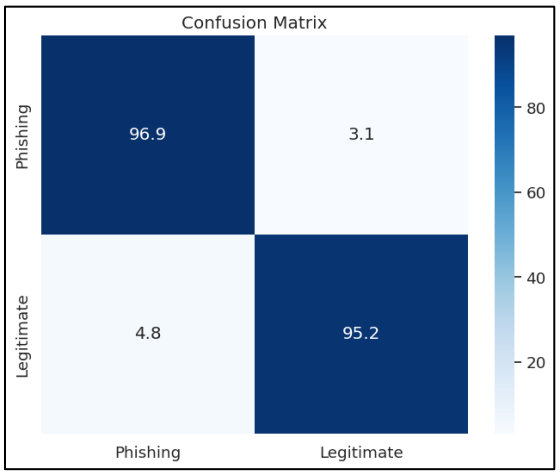
Figure 10. The confusion matrices for the deep learning models GRU, LSTM, and RNN show noteworthy patterns. In the case of GRU, the precision score reflects its accuracy in positive predictions, whereas recall demonstrates its efficacy in catching positive events. The total accuracy reflects the soundness 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.

TABLE 5
THE RESULTS OF DL ALGORITHMS ON FEATURES SET (%)

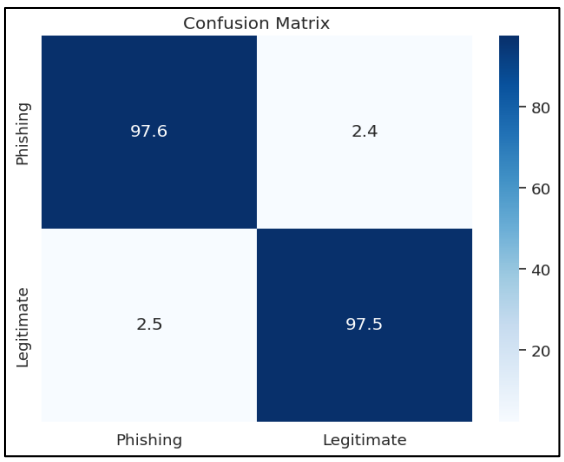
DL Algo	Precision	Recall	Accuracy	F-Measure
LSTM	96.4	99.0	97.4	97.7
RNN	96.9	95.2	95.7	96.1
GRU	97.6	97.5	97.3	97.6



(a)



(b)



(c)

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

G. *Results using Ensemble Learning Models*

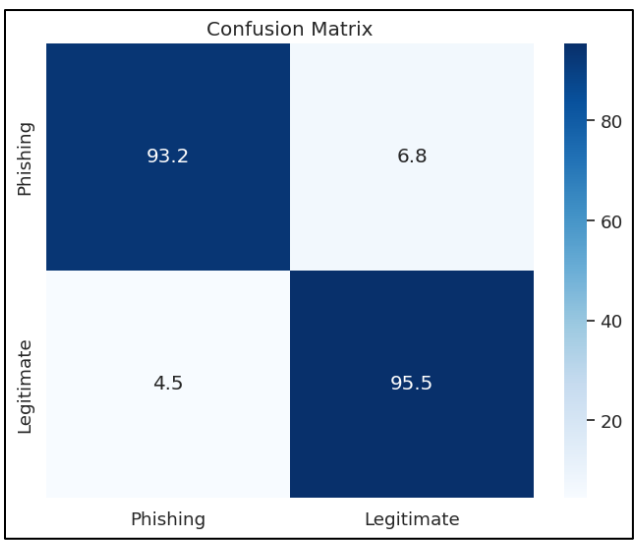
In this section, we discuss ensemble learning approaches in detail, including how AdaBoost, RF, and XGBoost are used to our carefully picked feature set. The key goal is to reveal the distinct performance features inherent in each ensemble model within the scope of our research. Through thorough inspection 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 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 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.

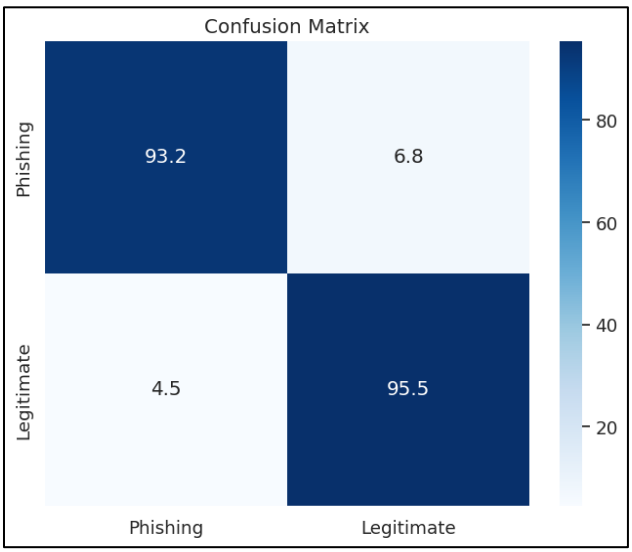
TABLE 6
THE RESULTS OF EL ALGORITHMS ON FEATURES SET (%).

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

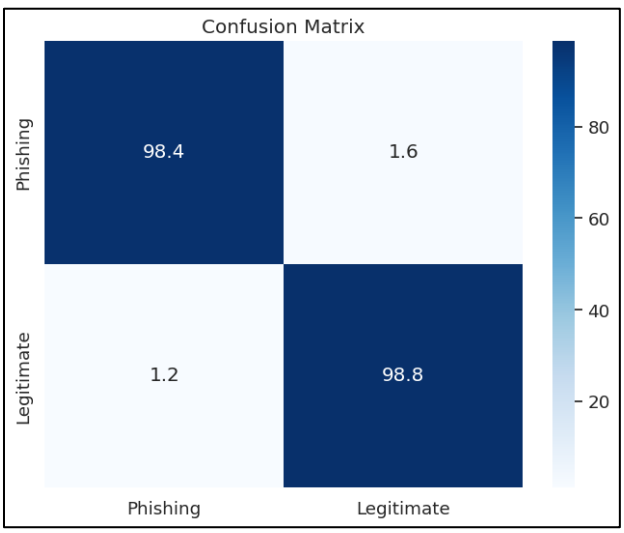
Figure 11. Analyzing the confusion matrices for the ensemble learning models Adaboost, RF, and XGBoost reveals significant trends. Adaboost precision score indicates its accuracy in positive predictions, while recall demonstrates its effectiveness in catching positive events. The overall accuracy measures Adaboost 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.



(a)



(b)



(c)

FIGURE 11. **Confusion Matrices using (a) Adaboost (b) RF (c) XGB**

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 12. **Performance Metrics for Different Models**

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*

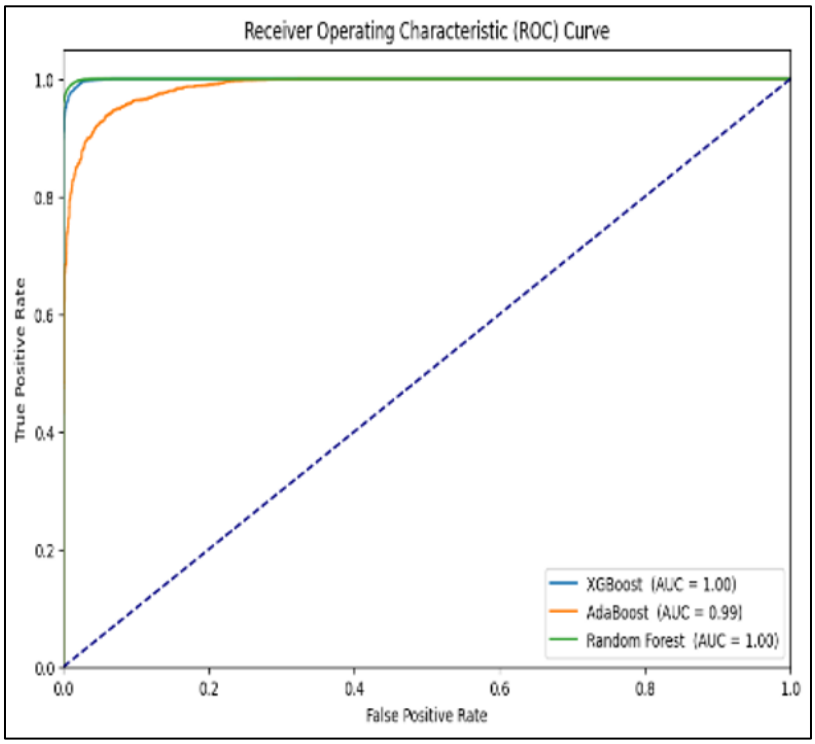
Table 7 provides a comparative analysis of several algorithms and the accuracies that correlate with them, as described in numerous references. Reference [43] explores ensemble techniques such as bagging, boosting, and stacking, achieving accuracy of 95.4%. In contrast, Reference [44] introduces the Adab-Forest PA-PWDM algorithm, which demonstrates a higher accuracy of 96.5%. Reference [45] 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, highlighting the feasibility of the required model in achieving superior accuracy compared to the referenced methodologies. The core findings of the whole experiment justify the credibility of the EL models.

TABLE 7
PROPOSED METHOD WITH EXISTING METHODS.

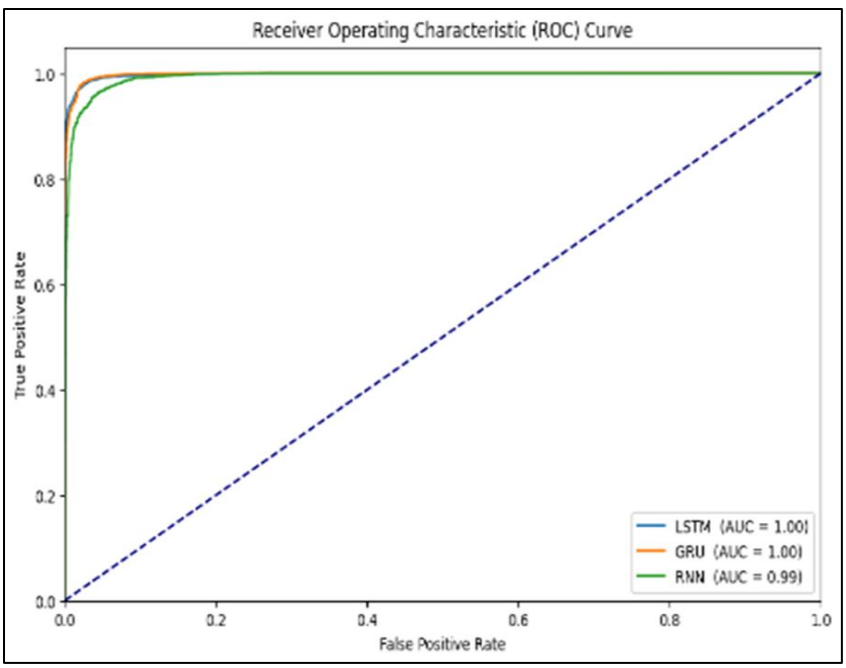
Ref.	Algorithms	Accuracy
[43]	Ensemble bagging, boosting, stacking	95.4%
[44]	Adab-Forest PA-PWDM	96.5%
[45]	DNN +Adam	96.00%
This Study	**RF**	**99.0 %**

As RF model is considered the best deep learning model when applied to the text dataset. The main concept of the proposed study is to extract important features and then transfer it 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.

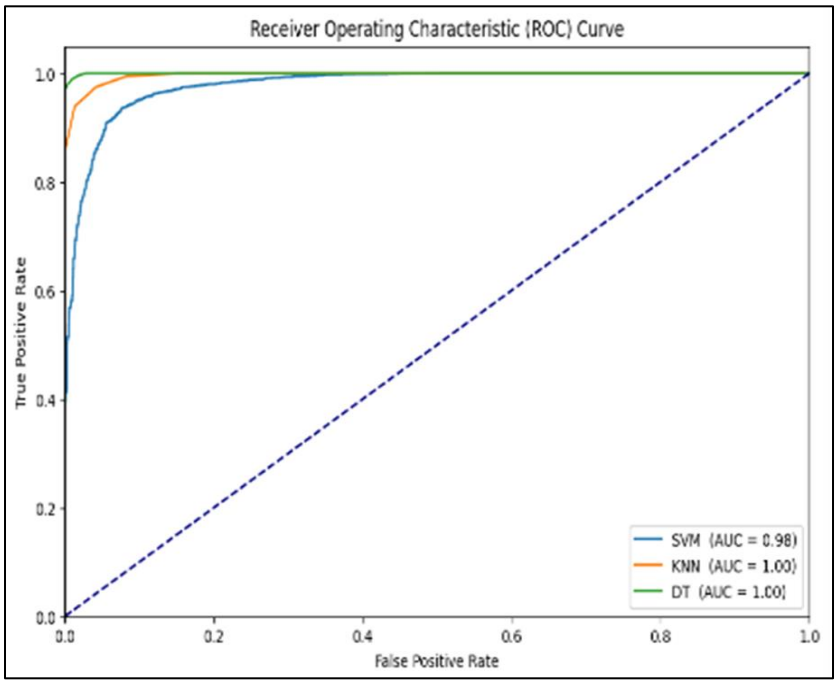
This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3486462



(a)



(b)



(c)

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

V. CONCLUSION AND FUTURE DIRECTIONS

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 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 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 towards 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 *Proceedings of the 16th International Conference on Availability, Reliability and Security*, in ARES '21. New York, NY, USA: Association for Computing Machinery, 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. Front.*, vol. 22, no. 1, pp. 45–57, Feb. 2020, doi: 10.1007/s10796-020-09984-5.

[3] 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. IJSWIS*, vol. 17, no. 3, pp. 1–17, Jul. 2021, doi: 10.4018/IJSWIS.2021070101.

[4] 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.

[5] A. K. Jain and B. B. Gupta, "PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning," M. U. Bokhari, N. Agrawal, and D. Saini, Eds., in Advances in Intelligent Systems and Computing, vol. 729. Singapore: Springer Singapore, 2018, pp. 467–474. doi: 10.1007/978-981-10-8536-9_44.

[6] 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, p. 102328, Sep. 2021, doi: 10.1016/j.cose.2021.102328.

[7] M. K. Hayat *et al.*, "Towards Deep Learning Prospects: Insights for Social Media Analytics," *IEEE Access*, vol. 7, pp. 36958–36979, 2019, doi: 10.1109/ACCESS.2019.2905101.

[8] A. K. Murthy and Suresha, "XML URL Classification Based on their Semantic Structure Orientation for Web Mining Applications," *Procedia Comput. Sci.*, vol. 46, pp. 143–150, Jan. 2015, doi: 10.1016/j.procs.2015.02.005.

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

[10] 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, p. 012077, Feb. 2021, doi: 10.1088/1757-899X/1055/1/012077.

[11] 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 *2021 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, Dec. 2021, pp. 1–4. doi: 10.1109/ICMLANT53170.2021.9690540.

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

[13] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*, in WWW '07. New York, NY, USA: Association for Computing Machinery, May 2007, pp. 639–648. doi: 10.1145/1242572.1242659.

[14] 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. Humaniz. Comput.*, vol. 15, no. 3, pp. 1865–1879, Mar. 2024, doi: 10.1007/s12652-018-0786-3.

[15] 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.

[16] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, in SecureComm '08. New York, NY, USA: Association for Computing Machinery, Sep. 2008, pp. 1–6. doi: 10.1145/1460877.1460905.

[17] J.-L. Chen, Y.-W. Ma, and K.-L. Huang, "Intelligent Visual Similarity-Based Phishing Websites Detection," *Symmetry*, vol. 12, no. 10, Art. no. 10, Oct. 2020, doi: 10.3390/sym12101681.

[18] 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 *2019 25th Asia-Pacific Conference on Communications (APCC)*, Nov. 2019, pp. 280–285. doi: 10.1109/APCC47188.2019.9026498.

[19] 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.

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

[21] 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.

[22] E. Buber, B. Dırı, and O. K. Sahingoz, "Detecting phishing attacks from URL by using NLP techniques," in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct. 2017, pp. 337–342. doi: 10.1109/UBMK.2017.8093406.

[23] E. Buber, B. Diri, and O. K. Sahingoz, "NLP Based Phishing Attack Detection from URLs," in *Intelligent Systems Design and Applications*, vol. 736, A. Abraham, P. Kr. Muhuri, A. K. Muda, and N. Gandhi, Eds., in Advances in Intelligent Systems and Computing, vol. 736. , Cham: Springer International Publishing, 2018, pp. 608–618. doi: 10.1007/978-3-319-76348-4_59.

[24] R. Mohammad, "Predicting Phishing Websites based on Self-Structuring Neural Network," *Neural Comput. Appl.*, Dec. 2013.

[25] 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, 2020, doi: 10.4218/etrij.2018-0553.

[26] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," in *Special interest tracks and posters of the 14th international conference on World Wide Web*, in WWW '05. New York, NY, USA: Association for Computing Machinery, May 2005, pp. 1060–1061. doi: 10.1145/1062745.1062868.

[27] 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.

[28] T. Peng, I. Harris, and Y. Sawa, "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, Jan. 2018, pp. 300–301. doi: 10.1109/ICSC.2018.00056.

[29] S. Kazi, S. Khoja, and A. Daud, "A survey of deep learning techniques for machine reading comprehension," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 2509–2569, Nov. 2023, doi: 10.1007/s10462-023-10583-4.

[30] 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," *Nat. Lang. Process. J.*, vol. 4, p. 100026, Sep. 2023, doi: 10.1016/j.nlp.2023.100026.

[31] 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.

[32] 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.

[33] A. Aljofey *et al.*, "An effective detection approach for phishing websites using URL and HTML features," *Sci. Rep.*, vol. 12, p. 8842, May 2022, doi: 10.1038/s41598-022-10841-5.

[34] W. Wang, F. Zhang, X. Luo, and S. Zhang, "PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks," *Secur. Commun. Netw.*, vol. 2019, p. e2595794, Oct. 2019, doi: 10.1155/2019/2595794.

[35] 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.

[36] 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.

[37] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *J. Enterp. Inf. Manag.*, vol. 36, no. 3, pp. 747–766, Jan. 2020, doi: 10.1108/JEIM-01-2020-0036.

[38] 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.

[39] 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.

[40] 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*, M. Anbar, N. Abdullah, and S. Manickam, Eds., in Communications in Computer and Information Science. Singapore: Springer, 2021, pp. 627–641. doi: 10.1007/978-981-33-6835-4_41.

[41] M. Sabahno and F. Safara, "ISHO: improved spotted hyena optimization algorithm for phishing website detection," *Multimed. Tools Appl.*, vol. 81, no. 24, pp. 34677–34696, Oct. 2022, doi: 10.1007/s11042-021-10678-6.

[42] A. Mandadi, S. Boppana, V. Ravella, and R. Kavitha, "Phishing Website Detection Using Machine Learning," in *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, Apr. 2022, pp. 1–4. doi: 10.1109/I2CT54291.2022.9824801.

[43] 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, 2019, doi: 10.14569/IJACSA.2019.0100133.

[44] Y. A. Alsariera, A. V. Elijah, and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," *Arab. J. Sci. Eng.*, vol. 45, no. 12, pp. 10459–10470, Dec. 2020, doi: 10.1007/s13369-020-04802-1.

[45] 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," *Wirel. Pers. Commun.*, vol. 118, no. 4, pp. 3549–3564, Jun. 2021, doi: 10.1007/s11277-021-08196-7.

Hikmat Ullah Khan received 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/ Chairman, 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.

Ume Zara received a master's degree in computer science from COMSATS University Islamabad–Wah, Wah Cantonment, Pakistan. Her research interests include data mining, machine learning, information retrieval, and sentiment analysis.

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

Tariq Alsahfi received the B.S. degree in Computer Science from King Abdul Azizi University, Saudi Arabia, in 2011, the M.S and PhD degrees in Computer Science from The University of Texas at Arlington, USA in 2020. He became an Assistant Professor for the Department of Information Systems and Technology at University of Jeddah, Saudi Arabia. His current research interest is in the field of data science, deep learning, machine learning, geographical information systems, trajectory data, and enhance road traffic safety in Intelligent Transportation System

Ali Daud have obtained his PhD degree in Computer Science from Tsinghua University, Beijing, China in July 2010. Currently, he is working as a Full Professor at Faculty of Resilience, Rabdan Academy, Abu Dhabi, United Arab Emirates. He has 13 years' post-PhD experience of teaching, supervision and research at BS, MS, and PhD level. He has published more than a hundred research papers in reputed international Impact factor journals and conferences. He has taken part in many research projects as well and have written and acquired many research funding's. He has proven and experience in Data Mining, Artificial Intelligence (Machine Learning / Deep Learning) applications to Social Networks, Data Science, Natural Language Processing, Internet of Things, etc.

Saima Gulzar Ahmad

Dr. Saima Gulzar Ahmad received her master's degree from COMSATS University Islamabad, Wah Campus, Pakistan in 2012. She completed her PhD in Computer Science from University Malaya, Malaysia in 2017. Currently, she is working as Assistant Professor in Department of Computer Science at COMSATS University Islamabad, Wah Campus, Pakistan. Her professional affiliations are with Pakistan Engineering Council (PEC) and IEEE. She is an HEC approved supervisor, and her research interests are parallel and distributed computing, heterogeneous computer networks (cloud/grid), machine learning and artificial intelligence. She is the author of quality research publications.