IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 December 2023, accepted 5 January 2024, date of publication 9 January 2024,
date of current version 19 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351946

RESEARCH ARTICLE

Enhancing Phishing Detection: A Novel Hybrid Deep Learning Framework for Cybercrime Forensics

FAISAL S. ALSUBAEI[1], (Member, IEEE), ABDULWAHAB ALI ALMAZROI[2], AND NASIR AYUB[3], (Student Member, IEEE)

[1]Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia
[2]Department of Information Technology, College of Computing and Information Technology at Khulais, University of Jeddah, Jeddah 21959, Saudi Arabia
[3]Department of Creative Technologies, Air University Islamabad, Islamabad 44000, Pakistan

Corresponding authors: Faisal S. Alsubaei (fsalsubaei@uj.edu.sa) and Nasir Ayub (nasir.ayubse@gmail.com)

This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-21-ICI-6). The authors, therefore, acknowledge with thanks the University of Jeddah for its technical and financial support.

ABSTRACT Protecting against interference is essential at a time when wireless communications are essential for sending large amounts of data. Our research presents a novel deep learning technique, the ResNeXt method and embedded Gated Recurrent Unit (GRU) model (RNT), rigorously developed for real-time phishing attack detection. Focused on countering the escalating threat of phishing assaults and bolstering digital forensics, our systematic approach involves SMOTE for managing data imbalance during initial data processing. The model's discriminative capability is improved, particularly in the feature extraction process, when autoencoders and ResNet (EARN) are integrated with feature engineering. The ensemble technique of feature extraction reveals crucial data patterns. At the core of our AI categorization is the RNT model, optimized using hyperparameters through the Jaya optimization method (RNT-J). Rigorously tested on real phishing attack datasets, our AI model consistently outperforms state-of-the-art algorithms by a substantial margin of 11% to 19% while maintaining exceptional computing efficiency. Furthermore, our model achieves 98% accuracy, low false positive/false negative values, and a statistical execution time with a mean of 36.99s, median of 35.99s, minimum of 34.99s, maximum of 41.99s, and a standard deviation of 1.10s. Moreover, it demonstrates superior accuracy with SMOTE (98%) and without SMOTE (83%) compared to other algorithms. This state-of-the-art AI study, which focuses on digital forensics, offers enhanced security and optimized productivity for businesses and industries, signifying a breakthrough in the continuing battle against phishing attempts. Through strengthening protection against interference in wireless communication, our AI research strives to amplify data accessibility, resilience, and trustworthiness in the face of cybersecurity threats within the organizational context.

INDEX TERMS Phishing attack detection, deep learning, ResNeXt, gated recurrent unit, digital forensics, cyber security, artificial intelligence.

I. INTRODUCTION

Phishing, a fraudulent activity utilizing both social and technological tactics to acquire financial and personal information from unsuspecting customers illicitly, remains a prevalent cybercrime [1]. Among the well-established methods is email spoofing, involving the creation of deceptive emails with a forged origin, often distributed through social media platforms, impersonating reputable entities to trick users into visiting fraudulent websites and disclosing sensitive information like usernames and passwords [2]. The utilization of devices by hackers to install malicious software further compounds the risk, facilitating unauthorized access and interception of user credentials.

Phishers employ various platforms, including email, forums, URLs, messaging apps, text messages, and phone

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

calls, to obtain user information [3]. Their deceptive content often mirrors legitimate websites, enticing users to interact and divulge personal information [4]. The primary aim of phishing is financial gain or identity theft, causing disruptions to businesses worldwide [5].

One important non-profit group that gathers, examines, and regularly updates the public on worldwide phishing activity is the Anti-Phishing Study Group. The surge in phishing attempts from 2022 to 2023, with over 300,000 recorded in July 2023, highlights the urgency of cybersecurity measures [6]. Webmail remains a preferred target, with a significant increase from 600 to 1100 phishing attempts on well-known firms monthly, emphasizing the persistent threat. California's stringent laws, such as the Anti-Phishing Act of 2016 [7], [8], aim to penalize individuals involved in phishing assaults with imprisonment for up to five years or fines.

Criminals resort to creating illicit replicas of legitimate websites and communications, especially from financial institutions, deploying misleading emails to a broad audience. This phishing tactic, involving logos and phrases from reputable businesses, catches users off guard, leading them to counterfeit websites and increasing the risk of data misuse [9]. Despite the urgency of phishing prevention, many organizations need more technology for detecting fraudulent URLs. Deep Learning (DL) algorithms, including Graph Convolutional Networks (GCN) and Bayesian Addition Regression Trees (BART), have shown promise in identifying features in observed datasets [10].

Traditional URL detection relies on blocklists, but their effectiveness is challenged by the constant rise of malicious URLs not on the list, facilitated by methods like the Domain Generation Algorithm (DGA). Recent studies in Phishing Detection (PD) focus on DL methods, with models like XGBoost for text preprocessing in email bodies and URLs [11]. XGBoost, by examining email structures, web addresses, files, sender information, and metadata, efficiently processes large databases, extracting essential features and managing noise for effective phishing categorization.

Modern procedures necessitate a departure from traditional methods, incorporating a higher degree of human intervention. Modelling approaches using robust chronological datasets, including Graph Convolutional Networks (GCN), Recurrent Neural Networks (RNN), and Artificial Neural Networks (ANN), significantly enhance the efficacy of phishing detection techniques. In the realm of digital forensics, these neural network models play a crucial role in identifying and preventing phishing attempts, contributing to a more secure cyber field [12].

This study introduces noteworthy contributions and novelty to the field of cybersecurity and phishing attack detection, emphasizing innovation, adaptability, and efficiency:

1) Lightweight Model for Reduced Processing Time: The research introduces a ResNeXt-embedded Gated Recurrent Unit (RNT) model designed for real-time phishing attack detection. This lightweight strategy aim involves significantly reducing the time for processing, which raises defence system effectiveness.
2) By introducing the Jaya optimization approach (RNT-J) for tuning hyperparameters, the research raises the possibility of responses that are flexible through dynamic tuning. Optimizing the model dynamically yields optimal values that enhance responsiveness and ensure its adaptability to a range of adaptive phishing attack scenarios.
3) The paper presents a novel design that uses the SMOTE for information preparation for recognizing phishing attacks. The framework's resistance to various kinds of phishing behaviour scenarios is ensured through this new approach to data imbalanced.
4) Another significant property of the framework is its ability to adapt to new inputs readily. Such adjustable functionality gradually boosts the algorithm's reliability by ensuring that the recognition approach remains effective in confronting the effects of changing phishing techniques and attributes.
5) Thorough Runtime Network Analysis: By utilizing EARN Ensemble (autoencoders and ResNet) to introduce sophisticated feature extraction, the study surpasses conventional metrics. The model's discriminative capability is improved by the ensemble technique, making it possible to identify important patterns linked to phishing assaults.
6) Improved Model Settings with RNT-J: The study advances the use of the Jaya optimization method (RNT-J) for model hyperparameter optimization. Through constant fine-tuning, the model is made to function at its best, which improves the accuracy of phishing detection.
7) Comprehensive Evaluation and Outperformance: Through rigorous testing on real phishing attack datasets, the proposed algorithm continuously outperforms the most advanced algorithms already in use. This significant contribution highlights the model's capacity to preserve computational efficiency while achieving better outcomes, ranging from an 11% to 19% improvement across several assessment measures.

Collectively, these contributions offer a significant improvement in the efficacy, efficiency, and responsiveness of cybersecurity measures by positioning the suggested framework as a sophisticated and flexible solution for phishing attack detection.

The following is the arrangement of the document's succeeding sections: In Section II, previous research is reviewed with a focus on dataset compilation. Section III explains the research technique. Section V presents a discussion of the outcomes, whereas Section IV looks into the results and their analysis. The study's findings are finally summarized in Section VI.

II. RELATED WORK

Many academic studies have examined the results of phishing websites. Our technique leverages important ideas from previous work, whereas our current method is influenced by a thorough examination of previous attempts to identify phishing using URL characteristics. Author in [13] outlined a method for identifying phishing attacks based on URL analysis. Their research used many algorithms to examine URLs from a range of data sets, comparing results using different deep learning (DL) techniques and hierarchical structures. To determine if this approach was effective, certain URL characteristics had to be evaluated, the website's administration and functionality had to be verified, and the website's visual representation had to be evaluated.

DL algorithms were used with skill to examine various aspects of URLs and web pages. Paradoxically, the author in [14] presented a unique technique that focuses on accurate and successful phishing website identification using URL analysis. We define our novel neural network (NN) architecture in terms of many parallel components, the first of which is the removal of surface-level URL characteristics.

However, in the work of [15], scholars were able to use simple characteristics to determine the legality of URLs while still producing accurate and trustworthy deep features of URLs.The findings of each component are combined to establish the maximum efficiency of the technology. A thorough analysis of a dataset from the internet confirms that, even after devoting a reasonable amount of time to the identification of phishing websites, our system is still comparable to other detection algorithms. A system for recognizing phishing webpages using Tag Distribution Language (HTDL), HTML, and URL characteristics was suggested by [16]. They achieved this by building concise HTDL and URL characteristics, which allowed HTDL string-embedding operations to operate independently of external infrastructure. A large database including more than 31,000 HTDL and URL characteristics was tested, and the results showed that accuracy was 97.24%, True Positive (TP) rate was 4.99%, and False Negative (FN) rate was 2.74% [17]. Author in [18] recommended an intelligent phishing detection method based on website text features to combat zero-day phishing tactics.

Researchers have developed artificial intelligence (AI)–based algorithms that employ machine learning and deep learning approaches in response to the widespread cyber dangers and vulnerabilities that internet users confront [19]. The goal was to build a strong system that could identify phishing attempts and lessen the dangers of cyberattacks. The system aims at extracting this information from URLs by using a Convolutional Neural Network (CNN) using n-gram features. In order to identify the ideal settings for improved performance, the study methodically investigates the effectiveness of many n-gram feature extraction approaches. When utilizing single characters, the most notable results are noted. Only 0.007 seconds are needed for URL categorization, while 35 seconds are needed for model training using 65 characters in an epoch. Using a dataset of high-risk URLs, the algorithm notably achieves an exceptional accuracy of about 87.70%.

In order to forecast whether a given URL represents a phishing link, the researchers provide a unique deep learning architecture called Texception [20]. This is an inventive method of doing so. Character-level and word-level information from the URL are incorporated into Texception, which differs from traditional techniques and minimizes the need for manually created features. By using separate parallel convolutional layers, the architecture may become either wider or deeper. Based on production data, Texception demonstrates excellent adaptation for new URLs via the Microsoft Smart Screen application dataset. With a remarkable rise of 127.6%, the genuine positive rate is still remarkably low at 0.02% in the false-positive data.

The researchers used sequencing techniques for reliable resource identification. Their suggested method showed a 95.38% TP rate in successfully identifying zero-day and phishing assaults. Past research used the text structures of websites to build frameworks for phishing detection, but phishers managed to avoid detection by including information from outside sources. Author in [21] examined the effectiveness of the long short-term memory (LSTM) classifier while investigating the field of spoofing site prediction using hyperlinks as a data source for DL models. Their study compared a new RNN-based technique with an RF classifier-based method using fourteen web address analysis criteria. In terms of average accuracy rate, an LSTM model that treated a hyperlink as a text sequence performed better than an RF classifier. The LSTM algorithm yielded 97.4% accuracy even in the lack of specialist knowledge for feature construction [22]. Their focus was just on the text-feature perspective of webpages; the model's efficacy may be increased by including additional elements like images and frame characteristics.

Author in [23] examined two URL datasets for phishing detection using CNN and CNN-LSTM logistic regression in a different assessment. Information from several sources was combined into their dataset, including lists of malware domains, ransomware domains, and phishing website domains from PhishTank and OpenPhish. More than seventy thousand URLs were used for testing, and more than sixty thousand URLs for training. Compared to other frameworks, the CNN-LSTM architecture performed better and achieved an accuracy rate of almost 97% for URL classification [24]. This design was selected because it recognized real data web addresses. However, only text-based features are used in our suggested method, which offers room for improvement by adding more features and optimizing variables for higher precision. As a result, our proposed Smart Phishing Detection System (IPDS) was built upon the limitations found in previous research. The summarized view of the existing studies is shown in Table 1.

TABLE 1. Phishing detection studies overview.

Ref	Problem Definition	Methodology	Dataset Size	Performance Metrics	Contributions
[13]	Detecting phishing attacks through URL analysis	Deep learning algorithms, hierarchical structures	Large	Accuracy, TP rate, FN rate	URL characteristic assessment, challenges in scalability
[14]	Precise identification of phishing websites via URL analysis	Innovative neural network architecture, elimination of surface-level URL characteristics	Moderate	Precision, Recall, F1-score	Effective elimination of surface-level URL traits, compromised effectiveness with highly obfuscated URLs
[15]	Determining URL legality using simplistic characteristics	Deep learning algorithms, integration of component results	Small to Moderate	Accuracy, Deep feature quality	Evaluation of URL legality, generation of accurate deep features, susceptibility to noise
[16]	Recognition of phishing webpages using HTML, HTDL, and URL traits	HTDL and URL characteristics, string-embedding operations	Large	Accuracy, TP rate, FP rate	97.24% accuracy, 4.99% TP rate, 2.74% FN rate, dependency on the quality of HTDL and URL characteristics
[18]	Intelligent phishing detection based on website text features	ResNet, DenseNet	Moderate	Accuracy, Precision, Zero-day attack mitigation	Relies on textual features, potential oversight of visual-based phishing
[19]	Construction of a robust system for identifying phishing attempts	Convolutional Neural Network (CNN), n-gram features	Large	Accuracy, Performance under dynamic phishing attacks	Exceptional accuracy around 87.70%, limitations in dynamic phishing attacks
[20]	Unique deep learning architecture (Texception) for predicting phishing URLs	Texception architecture, character-level, and word-level information	Moderate	Genuine Positive Rate, False Positive Rate	Significant rise in genuine positive rate (127.6%), 0.02% false-positive rate, resource-intensive implementation
[21]	Utilizing LSTM classifier for spoofing site prediction	LSTM classifier, RNN-based technique	Moderate	Accuracy, Feature construction efficiency	97.4% accuracy, efficient even without specialist knowledge for feature construction, constrained to a text-feature perspective
[23]	Examination of URL datasets for phishing detection using CNN and CNN-LSTM logistic regression	CNN, CNN-LSTM logistic regression	Large	Accuracy, Comparative performance metrics	Almost 97% accuracy in URL classification, neglects consideration for non-textual elements like images in URLs

Multiple studies have advanced our knowledge and use of deep learning techniques in the field of phishing detection. Notably, the authors of the study by [25] provide a thorough taxonomy, discuss present difficulties, and suggest potential paths for using deep learning to phishing detection. Propose a deeper phishing detection tool that applies deep learning to URLs in [26], highlighting the effectiveness of their methodology. A phishing detection system using a combination of LSTM-CNN is proposed by the author in [27], demonstrating the integration of many deep learning architectures. Using deep learning in contemporary security, researchers in [28] describe a novel method for phishing detection with an emphasis on Uniform Resource Locators (URLs). By providing a method for identifying and mitigating phishing attempts, the author in [29] advances the area and provides guidance on how to bolster security protocols. To improve the accuracy and efficacy of phishing detection systems, researchers in [30] investigate intelligent phishing website detection using deep learning. These research works highlight the many ways that deep learning may be used to counteract phishing attacks and offer insightful information for the advancement and enhancement of phishing detection systems.

Using machine learning classifiers to improve detection accuracy, the authors of [31] perform a comparative analysis that focuses on semantic characteristics for phishing detection. A machine learning method utilizing hyperlink information for phishing detection is proposed by researchers in [32], which helps to explore various attributes to enhance detection skills. An earlier study by [33] focused on client-side phishing website detection and offered a machine learning-based method to improve the detection of such dangerous websites. All of this research looks at various aspects and techniques to improve the overall efficacy of detection systems, which advances the area of phishing detection.

A. PROBLEM STATEMENT

The increasing complexity of phishing attacks presents a significant threat to cybersecurity. Phishing is an increasingly common danger to both individuals and businesses. It is a method that combines social engineering and technology. Phishing primarily aims to undercover acquire sensitive information, such as personal or financial details, endangering both one's privacy and financial security [34], [35]. Email spoofing is a popular tactic employed by hackers,

in which phony emails with fictitious sender addresses are sent. These emails frequently pose as coming from reliable sources in an attempt to fool recipients into responding to phony links or divulging private information. Phishing efforts have a greater impact and reach when they are disseminated through social media platforms. This can result in issues beyond only money losses, such as identity theft [36], [37]. The frequency of attacks using phishing is increasing in spite of countermeasures. The changing techniques of attackers prove to be too much for traditional solutions, particularly those that rely on static blocklists. Novel ways to anti-phishing efforts are required since sophisticated techniques like Domain Generation Algorithms (DGA) make detection even more difficult [38]. Present approaches often concentrate on certain facets of phishing, such as text analysis in emails or URLs. Comprehensive models that take into account the different components of phishing schemes–such as attachments, URLs, sender information, pictures, and textual content–are critically lacking [39].

Our research proposes an improved hybrid strategy to address these issues and get beyond the limits of current anti-phishing systems. Our architecture utilizes the RNT-J, which is capable of deep learning, to offer more resilient and adaptable protection. By including the Jaya Algorithm, the model is further improved and becomes more adaptive to the changing risks posed by phishing. This study is in line with our overarching objective of creating a robust RNT-J Deep Learning Framework for Cybercrime Forensics, and it makes a substantial contribution to the current endeavor to fortify cybersecurity against advanced phishing attempts.

III. PROPOSED SYSTEM MODEL

The conceptual framework developed for phishing webpage detection and flagging is presented in this section. Our model uses a combination of ensemble approaches to address the difficulties involved in detecting fraudulent web content, realizing the importance of accurate and reliable phishing detection tools. Feature engineering, reduction of dimensionality, and a new classification technique are all incorporated into this comprehensive strategy to maintain operational effectiveness while effectively identifying misleading activity. By addressing the interpretability criteria essential for successful cybersecurity in multiple domains, the proposed system model seeks to improve phishing detection accuracy and accommodate the dynamic nature of deceptive actions. The suggested model and the related steps are shown in Figure 1.

First, our model receives the features from the dataset as input. The preprocessing stage is started and entails addressing missing data, normalization, and other things. Resolving this imbalance is essential for robust training of models because of the dataset's class imbalance, which shows that occurrences of phishing websites only account for 3.27% of all observations. We begin data preparation using the SMOTE technique in order to address class imbalance. SMOTE is used to create artificial samples for the minority class and improve overall dataset balance by balancing the distribution of classes within the dataset. We employ the Ensemble AutoEncoder with ResNet (EARN) model for obtaining features after finishing this crucial preprocessing step. ''par'' The EARN model is a crucial tool for combining low-dimensional and multidimensional variables and provides a thorough approach to handling data variances. During the feature extraction phase, we generate various ResNet and autoencoder algorithms with different layer configurations and architectural details. Encoders that using the EARN method are trained unsupervised observation in an effort to reduce the input although preserving its fundamental features. For every autoencoder which has been trained individually loss functions like mean error must be used in order to help in the extraction of attributes within the encoder component. Regardless of class labels, this phase operates solely to extract hidden attributes required for additional categorization.

By integrating data generated by automated encoding processes alongside the ResNet method, possible to generate a combination feature model during the Features Ensembler Classifiers step. Upon base of this combined features vectors, an algorithm with an RNT-J categorization level made up of bypassed links and pertinent layer is subsequently created. By connecting the collected attributes to labels that indicate classes, the algorithm enhances its categorization skills and gains the capacity to classify instances. Additionally, we employ an innovative technique called ensemble learning (EARN) to boost the model's resilience and effectiveness. To create an EARN model, we vary the starting seeds, autoencoder architectures, and ResNet configurations.

A. DATASET DESCRIPTION

In this study, our focus revolves around a meticulously curated dataset comprising 10,000 instances, each characterized by various attributes associated with URLs and web pages. The dataset, obtained from Kaggle [40], is specifically designed for the identification of phishing websites, aiming to discern between reliable websites and potential threats.

1) DATA COLLECTION PROCESS AND CHARACTERISTICS

Compiling the dataset required a thorough procedure of gathering instances that display various characteristics related to phishing attempts. These phishing attack datasets were compiled from carefully selected sources to provide a complete and representative sample. These features were selected in order to collect a wide range of data, including the length of the URL, the number of dots in the URL, the presence of subdomains, and other pertinent data indicative of phishing activity.

The collection includes a wide range of characteristics, each of which provides information about a distinct facet of the properties and structure of a URL. A thorough review of these features is given in Table 2, which gives a thorough rundown of the features of the dataset. Several essential

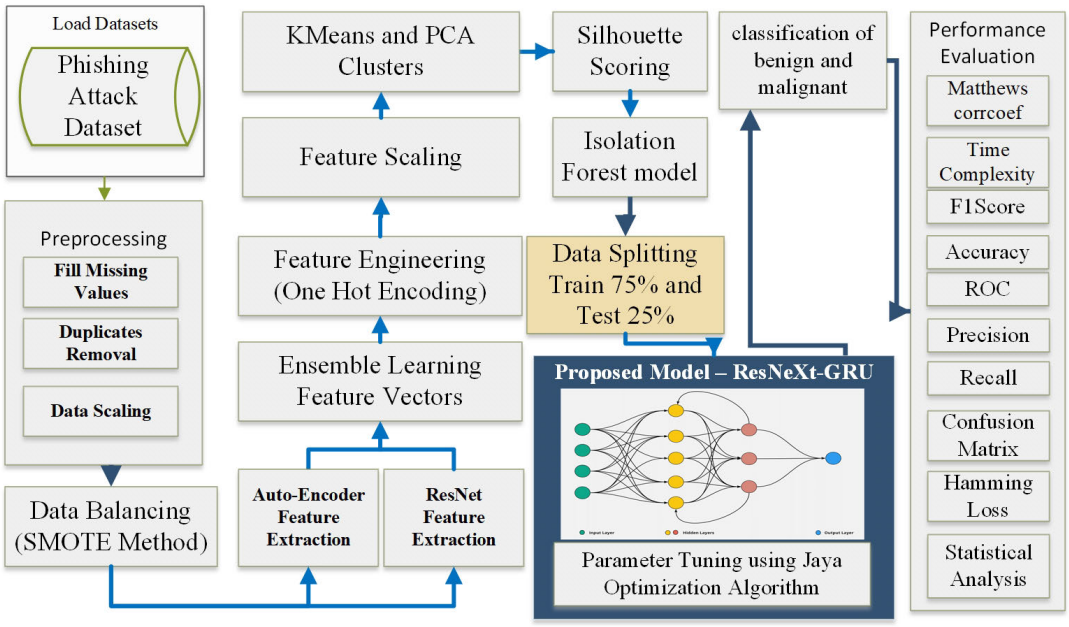

FIGURE 1. Proposed framework for phishing attack identification.

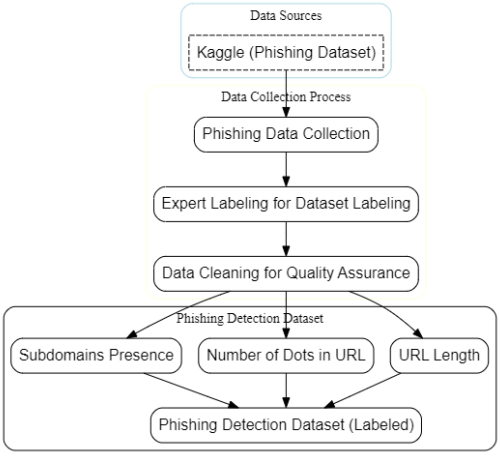

FIGURE 2. Data collection process.

characteristics for phishing identification include the quantity of query elements, the existence of double slashes in the path, unusual form actions, the proportion of external resource URLs, and other signs.

2) INTEGRATION INTO PHISHING DETECTION SYSTEM

This dataset plays a pivotal role in the training and evaluation of our proposed Phishing Detection System (IPDS). The system integrates a diverse array of machine learning and deep learning methodologies to effectively detect and flag potential phishing websites. The rich information provided by the dataset's features empowers the model to make informed decisions during the detection phase. The flow of data collection is shown in Figure 2.

B. PREPROCESSING OF DATA

The first and most important stage in preparing datasets for phishing website detection analysis and modeling is to perform basic preparation operations. Handling uncompleted information These methods involve data sizing standardization and the elimination of unnecessary data. Through the completion of these tasks, enhance the consistency and improvement of the information, laying the foundation for precise evaluation and effective modelling in seeking for likely phishing attack scenarios.

Managing Absent Data: Addressing handle of the discrepancies in the information we have that result from missing or insufficient data is the fundamental task of managing values that are missing. One method that is frequently used for this is mean imputation. In mean imputation, the missing entries are replaced by using the mean value, which is determined by averaging the known data points within a certain characteristic. A concise representation of the process is provided by Equation 1 [41]:

$$F_{\text{filled}} = \frac{1}{C} \sum_{j=1}^{C} F_j \tag{1}$$

F_{filled} signifies the values that are anticipated to fill in the gaps, F_j represents the values that were initially observed, and C indicates the total number of non-missing values.

Removing of Duplicate Records: Removal of duplicate records helps to minimize bias brought on by duplications, keep our data accurate and dependable, and make sure that our dataset is made up of unique data points. All records need to be analyzed and contrasted with one another in order to retain just the unique ones.

$$S_{\text{distinct}} = \{s_i \in S : \text{No identical entry in } S \text{ matches } s_i\} \tag{2}$$

TABLE 2. Summary of features in the phishing detection dataset.

Attribute	Description	Attribute	Description
NumQueryComponents	Number of query components	NumHash	Number of hash symbols
DoubleSlashInPath	Presence of double slashes in the path	NumAmpersand	Number of ampersands
AbnormalFormAction	Abnormal form action	RandomString	Presence of random strings
PctExtResourceUrlsRT	Percentage of external resource URLs with respect to the root	ExtMetaScriptLinkRT	Presence of external meta script links with respect to the root
HostnameLength	Length of the hostname	RightClickDisabled	Right-click disabled
PctNullSelfRedirectHyperlinksRT	Percentage of external null self-redirect hyperlinks with respect to the root	ExtFormAction	Presence of external form action
PctExtResourceUrls	Percentage of external resource URLs	AbnormalExtFormActionR	Abnormal external form action with respect to the root
NoHttps	Absence of 'https' in the URL	SubdomainLevelRT	Subdomain level with respect to the root
NumPercent	Number of percent symbols	ExtFavicon	Presence of external favicon
EmbeddedBrandName	Presence of embedded brand names	SubdomainLevel	Level of subdomains
ExtFormAction	Presence of external form action	PathLength	Length of the path
NumDots	Number of dots in the URL	NumNumericChars	Number of numeric characters
NumSensitiveWords	Number of sensitive words	SubmitInfoToEmail	Submission of information to email
NumDashInHostname	Number of dashes in the hostname	PathLevel	Level of the path in the URL
QueryLength	Length of the query	NumUnderscore	Number of underscores
FrequentDomainNameMismatch	Frequent domain name mismatch	ImagesOnlyInForm	Presence of images only in forms
DomainInPaths	Presence of the domain in paths	PctExtNullSelfRedirectHyperlinksRT	Percentage of external null self-redirect hyperlinks with respect to the root
IpAddress	Presence of an IP address	NumDash	Number of dashes
MissingTitle	Missing webpage title	HttpsInHostname	Presence of 'https' in the hostname
NumNumericChars	Number of numeric characters	DomainInSubdomains	Presence of domain in subdomains
PopUpWindow	Presence of pop-up windows	IframeOrFrame	Presence of iframes or frames
AbnormalExtFormActionR	Abnormal external form action with respect to the root	TildeSymbol	Presence of ' ' symbol
PctExtHyperlinks	Percentage of external hyperlinks	UrlLengthRT	URL length with respect to the root
RelativeFormAction	Presence of relative form action	AtSymbol	Presence of '@' symbol
InsecureForms	Presence of insecure forms	SubdomainLevel	Level of subdomains
RightClickDisabled	Right-click disabled	ExtFormAction	Presence of external form action
FakeLinkInStatusBar	Presence of fake links in the status bar	NoHttps	Absence of 'https' in the URL
HttpsInHostname	Presence of 'https' in the hostname	DoubleSlashInPath	Presence of double slashes in the path
TildeSymbol	Presence of ' ' symbol	HttpsInHostname	Presence of 'https' in the hostname
NumUnderscore	Number of underscores	PctExtResourceUrlsRT	Percentage of external resource URLs with respect to the root
NumDashInHostname	Number of dashes in the hostname	MissingTitle	Missing webpage title
PctExtResourceUrlsRT	Percentage of external resource URLs with respect to the root	SubdomainLevelRT	Subdomain level with respect to the root

S_{distinct} denotes the dataset with all duplicate entries eliminated. Within this dataset, si stands for a single, unique entry, and S for the original collection, which could have included duplicates.

Data scaling is an essential stage in the data preparation process that guarantees consistency and standardization among all numerical parameters. The two most popular techniques for doing this are min-max scaling and standardization. The process of standardization involves transforming a specific characteristic, X such that its standard deviation equals one and its mean, or average, value, is equal to zero. We can compare feature X with other features in a dataset efficiently thanks to this transformation [42], as shown in Equation 3;

$$Y_{\text{normalized}} = \frac{Y - \mu}{\sigma} \tag{3}$$

In the above equation, $Y_{\text{normalized}}$ denotes the normalized feature, Y represents the original feature, μ is the mean, and σ is the standard deviation.However, a feature Y is adjusted using min-max scaling to fit inside a certain range [0, 1] [42].

$$Y_{\text{normalized}} = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}} \tag{4}$$

The variable $Y_{\text{normalized}}$ now stands for a feature that has undergone normalization or rescaling based on its original value (Y), its minimum value ($Y_{\min}$), and its maximum value ($Y_{\max}$).

C. SMOTE METHOD TO HANDLE DATA BALANCING

One major challenge in identifying phishing websites is handling the imbalance between reputable websites and phishing websites. This mismatch makes traditional algorithms to detect phishing occurrences effectively. We analyze the SMOTE [43], an advanced approach, to overcome this for the phishing class.

Synthetic instances are created between a fake website (m1) and its nearest neighbors in the region of features (x01 and x02) as part of the procedure. A random number between 0 and 1 is added (random(0, 1)) to increase the degree of randomization. Then, the variations in the features of the initial phishing site and its nearest neighbors are supplemented with this random value. New data points that aid in closing the gap between the overlooked phishing class and its surrounding data are the outcome. Equation 5 [44] illustrates how this improves the visibility and depiction of phishing incidents in the entire dataset.

$$(M1; M2) = (m1; m2) + \text{random}(0; 1) \cdot (x01 - x1; x02 - x2) \tag{5}$$

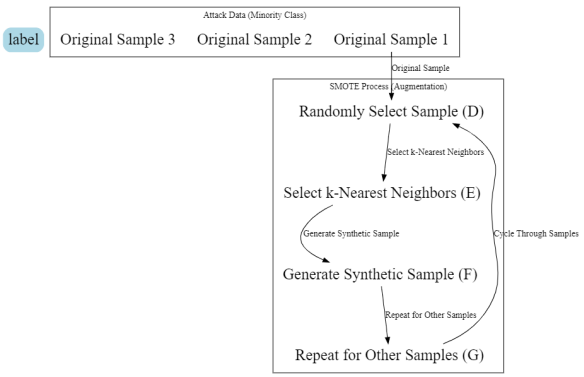

FIGURE 3. Data augmentation process (SMOTE).

A random number between 0 and 1 is produced by using Random(0, 1). We calculate the difference, represented as (x01 - x1; x02 - x2), between the feature values of the given instance and those of its nearest neighbors. To create several fake examples for the underrepresented class, this process is repeated several times. The SMOTE algorithm helps alleviate the problem of class imbalance and improves the model's capacity to identify phishing websites when it is used in our phishing detection system. The process of data augemntation through SMOTE is shown visually in Figure 3.

D. FEATURE ENSEMBLE LEARNING

This section provides insight into our approach to ensemble learning, combining the advantages of feature extraction using both autoencoders and ResNet models within the framework of a well-balanced dataset. The aim is to strengthen our ensemble model, augmenting its robustness and discriminative powers for enhanced performance on a variety of tasks.

1) EXTRACTING FEATURES FROM AUTOENCODERS

When working with balanced datasets, autoencoders are very good at obtaining insightful representations from the input data. To obtain attributes from the balanced dataset, we use autoencoders in our ensemble technique. E(input) is the encoder function that converts the input values x into an efficient format called `ensmb_auto`:

$$\text{ensmble\_autoenc} = Ex(\text{inputData}) \tag{6}$$

2) EXTRACTION OF FEATURE WITH ResNet

Our feature extraction method is substantially strengthened by the use of ResNet architectures. Using previously trained ResNet algorithms that have been fine-tuned on our balanced dataset, we capture complex hierarchical features. The feature vector that ResNet extracts is `f_resnet`, and it originates from the network's topmost fully connected layers:

$$f_{\text{resnet}} = Func(inputVal) \tag{7}$$

3) ENSEMBLE INTEGRATION

Our method combines feature vectors from ResNet models and autoencoders into a single representation that efficiently captures a wide variety of characteristics extracted from a well-balanced dataset. The ensemble feature vector, represented as `ensemble_feature`, is created through concatenation or weighted averaging.

4) CONCATENATION

The two feature vectors are merged directly, forming a single ensemble feature vector:

$$\text{feature\_ensemble} = [\text{ensmb\_autoenc}, \text{resnet\_func}] \tag{8}$$

5) AVERAGING BY WEIGHT

The feature vectors from autoencoders and ResNet are given particular weights (w_{auto} and w_{resnet}). Next, these feature vectors are combined using the allocated weights to generate the ensemble feature vector:

$$\text{ens\_feat} = w_{\text{auto}} \cdot \text{ensmb\_autoenc} + w_{\text{resnet}} \cdot \text{func\_resnet} \tag{9}$$

This attribute vector collectively captures a broad spectrum of data by combining high-level qualities identified by ResNet with specific information from autoencoders. The algorithm performs exceedingly well on problems requiring properly-balanced datasets as a result of this integrative method, which improves the model's ability of sequence detection.

E. ENHANCING FEATURES AND POST-PROCESSING

Improving the accuracy of vectors of attributes generated through pair learning is essential when trying to set up these for subsequent machine learning operations. The following are a few crucial tasks at the next process phase:

1) UNIFIED CODING SYSTEM

At this point, the classification attributes within feature vectors are handled, and the consequent classes are converted into vectors of binary values. This ensures that the algorithm is compatible with machine learning techniques to enhance its functionality.

2) COMPONENT ANALYSIS CONVERSION

reducing data while retaining substantial connections and trends is the objective of PCA conversion. The formula which follows is used for achieving this, dividing an element of data D into its separate elements:

$$\text{PCA} = D \cdot V \tag{10}$$

3) EVALUATION OF CLUSTER UTILIZING THE SILHOUETTE APPROACH SCORE

A key statistic to measure the level of accuracy of groups formed throughout a clustering study is the Silhouette Rating [44]. subsequently assesses how tightly a component

within a single cluster corresponds to the others (cohesion) rather than with items in different clusters (separation). Essential data regarding the efficacy of clustering can be gathered by the Silhouette Rating, which has a number between −1 and 1.

- Effective cluster can be determined by an aspect of evidence which fits perfectly into its cluster as well as distinguishes away from endpoints within other clusters, as well as by a significant affirmative silhouette rating with a value near to 1.
- A data point that is close to the border between two nearby clusters but does not have a significant correlation with either is indicated by a silhouette score of about 0.
- Since a data point with a silhouette score of about −1 is more comparable to points from a different cluster, it may have been incorrectly assigned to a cluster.

In assessing clustering algorithms, this scoring system is essential for figuring out how many clusters are best for efficiently separating and grouping related data points.

4) ANOMALY DETECTION WITH ISOLATION FOREST MODELING (IFM)

As prior research has shown [45], IFM is particularly good at quickly locating and separating outliers from a dataset. In contrast to conventional techniques that concentrate on simulating typical data points, the isolation forest builds a collection of decision trees in order to identify anomalies. The identification process only requires a few many sections in a structure resembling a tree to separate anomalies from most data points. Using this method to detect anomalous or fraudulent activity is particularly helpful when dealing with datasets that have uneven class distributions. During the testing phase, every observation is given a score for anomalies by the IFM based on its training. The effectiveness of the model in recognizing odd data sets is demonstrated by higher scores for anomalies, which also imply a larger chance of outliers or anomalies in areas like cybersecurity fraud detection and defect control.

F. CLASSIFICATION USING RNT-J

Its specific purpose is to identify scams. A reliable and flexible architecture for classifying information from phishing websites is the ResNeXt-GRU model. The proposed approach combines the continuous training and contextually comprehension built into GRUs with the powerful feature mining abilities of the ResNeXt architecture. Figure 4 shows how the internal structure of the ResNeXt GRU Model is organized.

Feature Extraction using ResNeXt: The initial raw data input from phishing websites is when the feature extraction procedure begins. This data goes through a series of changes inside the ResNeXt component. In the ResNeXt block, a thorough description of this feature extraction stage is given for every route i:

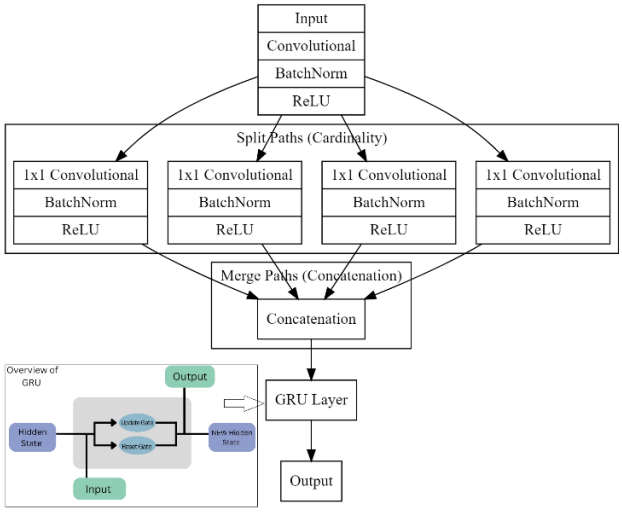

FIGURE 4. ReNeXt-GRU model.

- Convolutional Layer: $Conv_i$ = Convolutional Layer$_i$ ($input$)
- Batch Normalization: BN_i = Batch Normalization$_i$ ($Conv_i$)
- ReLU Activation: $ReLU_i = \text{ReLU}(BN_i)$

Here, $input$ represents the raw data that was obtained via phishing websites. The variables $Conv_i$, BN_i, and $ReLU_i$, respectively, represent the batch-normalized output of the route i following batch normalization, the output of the convolutional layer, and the outcome of applying the ReLU activation function.

Path Cardinality: Cardinality-based segmentation is strategically used to divide the data into many routes, which increases the model's ability to capture a variety of characteristics. This process is known as path cardinality and concatenation. Through the ResNeXt architecture, each route processes the data independently. These route outputs are concatenated to provide a thorough feature representation after being enhanced with additional information:

$$\text{Concatenation} = [RLUFunc1, RLUFunc2, RLUFunc3, RLUFunc4] \quad (11)$$

Concatenation creates a single representation by combining the outputs (RLU_i) from all paths–four in this case–into one.

1) GRU-BASED SEQUENTIAL MODELING

In a GRU layer, the combined attributes are processed. This stage adds a time-based analysis layer to the model to assist it recognize the sequential linkages and changing patterns observed in phishing websites. The following is a summary of the mathematical model of GRU operations [46], [47]:

$$\alpha = \sigma(W_\alpha \cdot [\text{Concatenation}, \beta_{t-1}] + b_\alpha) \quad (12)$$

$$\beta = \sigma(W_\beta \cdot [\text{Concatenation}, \beta_{t-1}] + b_\beta) \quad (13)$$

$$\gamma_t = \tanh(W_\gamma \cdot [\text{Concatenation}, \alpha \cdot \beta_{t-1}] + b_\gamma) \quad (14)$$

$$\beta_t = (1 - \beta) \cdot \beta_{t-1} + \beta \cdot \gamma_t \quad (15)$$

TABLE 3. Optimized hyperparameter values.

Hyperparameter	Value
Population Size	10
Config of Block in ResNeXt	4 width, 4 blocks, 32 depth
Size of processing Batch	32
Hidden Neuron Units in GRU	128
Convergence Threshold	1e-5
Epochs	50
Reduce the data frequency	0.2
Weight Decay	0.0001
Exploration Range	[0.2, 0.8]
Learning Rate	0.0001

The characteristics of the updated GRU layer are expressed by these formulae. In this case, "βt" represents the state that is hidden at the time point 't,' and the two separate barriers or 'α and 'β', oversee the information flow and related control mechanisms.

The intricate design of the suggested Gated Recurrent Unit (GRU) ensemble with ResNeXt model for phishing detection is shown in Figure 5. In order to create embedding vectors, the method starts with the input layer and channels URL characteristics via the embedding layer. The residual features and intermediate representations are extracted in the multiple-layer ResNeXt Block that follows. Following that, these characteristics go via the Gated Recurrent Unit (GRU) Block, collecting and honing sequential information and temporal relationships. The Attention Mechanism, when positioned strategically, improves the model's concentration on important segments of the input sequence. Sequential learning and robust feature extraction are enhanced by the synergistic combination of ResNeXt, GRU, and Attention Mechanism. The Output Layer presents a complete model for real-time phishing attack identification by generating phishing detection probabilities based on the learnt attributes.

2) TUNNING WITH JA

During the Jaya Algorithm (JA) fine-tuning phase, we concentrate on improving hyperparameters that are essential to our ResNeXt-GRU model's efficiency, integration, and generalization. The table below provides specific hyperparameters along with their corresponding values:

The repeated analysis and update of hyperparameter values is guided by the Jaya Algorithm, which draws inspiration from population collaboration and improvement. The stages outlined in Algorithm are followed in the optimization process, which evaluates the effectiveness of the model on validation data using an objective function. 1.

The Jaya Algorithm optimization process for hyperparameter tuning in the context of phishing website detection is shown in Algorithm 1. Several symbols, each with a distinct meaning, are used in the algorithm. The population of hyperparameter settings is P. f(selected_solution) is the objective function that is used to assess the performance of a chosen configuration. $[L, U]$ defines each hyperparameter's exploration range, which makes sure configurations stay

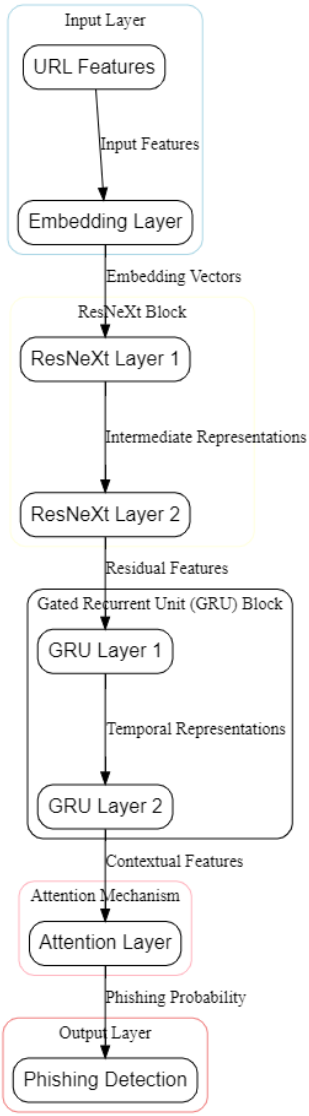

FIGURE 5. ReNeXt-GRU architecture.

inside reasonable bounds. ϵ is the symbol for the convergence threshold. In the optimization process, the optimal configuration chosen at each iteration is represented by C_{best}. In addition, C_i indicates a unique configuration inside population P. In order to converge on the ideal collection of hyperparameters, the method updates these configurations repeatedly.

IV. DISCUSSION

A. ENHANCED MODEL GENERALIZATION AND ADAPTABILITY IN PHISHING DETECTION

The model performs well in a variety of scenarios seen in the testing data, displaying a respectable level of generalization. This implies that it can extend the patterns it learnt in training to brand-new, untested data points. The extensive ensemble approach that combines ResNet models

Algorithm 1 Enhancing Hyperparameter Optimization for Identifying Phishing Websites With JA

Data as Input:
Ensemble of base-parameters configurations PO
Objective/Goal calling function f(solution_chosen) for Phishing Websites
Range for exploration $[G, H$ with every tuning-parameter
Converging point ϵ
Output Data: Optimal hyperparameter ensemble
procedure RefineHyperparametersForPhishing
    Populate Po with randomly generated tuning_param sets.
    setup optimal configuration PTLoptimal using a randomly chosen configuration from PO.
    while No Achievement towards convergence **do**
        for Every configuration C_i within PO **do**
            Create an arbitrary integer. rn fluently distributed within [0, 1].
            Update the configuration:
            $PTL_i = PTL_i + r \cdot (PTL_{\text{optimal}} - PTL_i)$
            Ensure configurations stay within the exploration range:
            $PTL_i = \min(G, \max(H, PTL_i))$
        end for
        choose configuration along superior objective procedure output-value as PTL_{optimal}.
    end while
end procedure

with autoencoders helps the model adapt to a variety of phishing attack types. Combining ensemble learning with the various seedings, autoencoder architectures, and ResNet configurations in the EARN model improves its ability to recognize and adjust to the many subtleties seen in dynamic phishing strategies. In real-world circumstances, where the proportion of phishing incidents may be substantially lower, the model demonstrates resilience in tackling the problem of class imbalance by utilizing the SMOTE. The algorithm's flexibility in identifying small evidence predictive of phishing attempts is enhanced by the combination of depiction of features, which includes both a high degree qualities identified by ResNet and precise information from encoders. The attributes are additionally enhanced by the use of binary subsequent processing steps, PCA transform clusters assessing, and recognition of anomaly, that increase the algorithm's adaptability in recognizing irregularities and unique phishing patterns. when the RNT framework is applied to classification, an ongoing learning component is added. thereby it possible for the framework to identify temporal patterns and adapt to the changing characteristics of phishing attempts. In the long run, a cohesive strategy that considers the many challenges posed by the unpredictable circumstances of phishing scams, continuous improvement, and dynamic fusion of diverse approaches has led to an extensive adaptation of the model's parameters.

B. SCALABILITY ANALYSIS AND REAL-WORLD EFFICIENCY

Furthermore, our study on scalable emphasizes the adaptability of our strategy in other cybersecurity contexts. After undergoing extensive training on a variety of datasets, our phishing detection technology has an impressive ability to adjust and forecast accurately in a wide range of settings. This flexibility, which enables the model to dynamically modify itself based on particular characteristics of various circumstances, is an essential feature. Through training, our model proves its effectiveness in a variety of scenarios, which improves its predictive power and guarantees consistent performance in scenarios outside of the training set. In real-world applications, where phishing attack landscapes are always changing, this adaptable characteristic is essential. Our model's adaptability, acquired through a variety of training situations, places it in a position to offer a reliable and scalable solution for complex problems faced by extensive, real-world cybersecurity deployments.

C. INTERPRETABILITY MEASURES

Our technique prioritizes interpretability to increase the predictability of the model and satisfy the critical need to understand AI choices in the cybersecurity domain. Our model's interpretability feature was purposefully included in the design to prevent the decision-making process from being viewed as a black box. We provide a better understanding of our model's reasoning by including interpretability measurements, which is essential for fostering understanding and confidence in the cybersecurity community. The incorporation of ensemble learning techniques enables our framework to adopt a diverse approach to interpretability. The collection of ResNet models and autoencoders offers an organized and thorough perspective of the attributes impacting the model's forecasts. The PCA transformation, cluster assessment, and binary encoding processes also help to improve the interpretability of the model's conclusions. The capacity to examine and understand AI-driven decisions is crucial for efficient threat analysis and response, which is why this interpretability capability is so useful in cybersecurity applications.

D. DATA DIVERSITY

The cornerstone of any strong phishing detection model is its capacity to recognize and adjust to a wide range of attack routes. Our dataset in this study, which is described in great length in the feature description that is provided (Table 2), is evidence of the purposeful selection of features intended to capture the complex nature of phishing attempts. Our feature set includes structural aspects like the path's double slashes, the hostname's length, and the URL's constituent parts, as well as more subtle semantic markers like email submissions of information and the appearance of random

sequences. With this all-encompassing approach, we can be sure that our model is sensitive to the intricacies present in complex phishing efforts, in addition to being able to identify patterns that are obviously malevolent. The breadth of our feature set suggests a deliberate attempt to cover different aspects of phishing activities, even if the details of the test datasets are described in detail. We specifically list the variety of phishing attack vectors that our test datasets include in order to increase the transparency of our study. By doing this, we are able to present a more comprehensive picture of the variety that our model has experienced and assimilated during the training and assessment stages.

We understand the dynamic nature of cyber threats and the need of regularly testing our model against various phishing attack variants. This proactive approach not only demonstrates our dedication to strengthening and improving our model, but it also supports an industry-wide necessity that encourages cooperation in order to remain ahead of the constantly changing strategies used by cyber attackers.

V. SIMULATION RESULTS

During this stage, TensorFlow was used within IDE enviroment of Google-Colab setting by using the powerful processing unit resources by improving our system's ability to identify phishing websites. The implementation was carried out in the Python programming language within the Spyder IDE from the Anaconda distribution. Our proposed model underwent evaluation using three datasets related to phishing incidents. The outcomes of these evaluations are discussed below.

The correlation matrices in Figures 6 to 10 provide a comprehensive overview of the relationships between attributes in the dataset, organized in groups of 10 attributes per figure. Each cell in the matrix represents the correlation coefficient, ranging from −1 to 1, indicating the strength and direction of the relationship. Observing these matrices helps identify patterns of strong positive or negative correlations, highlighting potential linear relationships between attributes. Additionally, noting low or zero correlations between certain attributes suggests independence. Analyzing these correlation matrices aids in feature selection, reveals potential multicollinearity, and provides insights into the intricate relationships within the dataset, which is crucial for enhancing the effectiveness of phishing website detection models.

In our analysis, When comparing the number of occurrences between phishing and authentic websites, we first noticed an imbalance in the dataset. To address this issue, we employed the SMOTE algorithm, a method designed to balance imbalanced datasets. Figure 11 illustrates the initial state of the dataset, highlighting the disproportionate distribution of phishing and legitimate instances. The clear discrepancy in the number of data points for each class is evident in this visualization. Following the application of the SMOTE algorithm, as depicted in Figure 12, we achieved a balanced dataset. The figure demonstrates how SMOTE

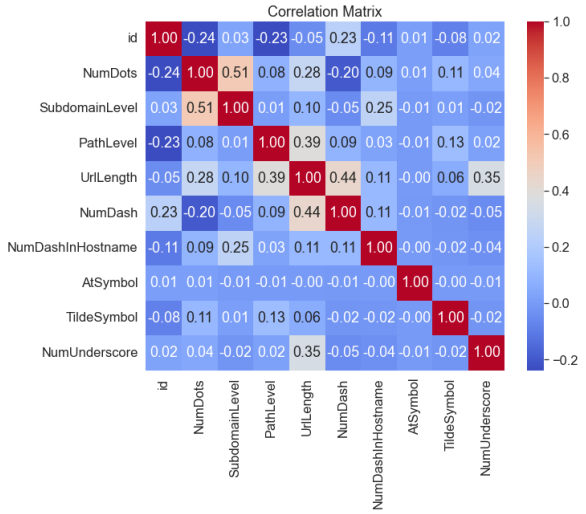

FIGURE 6. Confusion matrix of attribute 1 to 10.

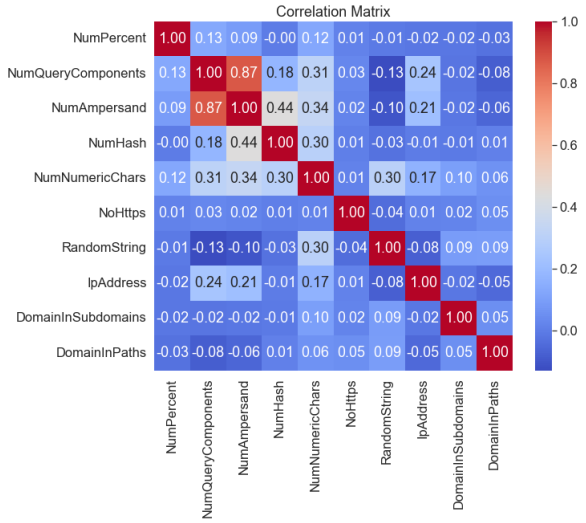

FIGURE 7. Confusion matrix of attribute 11 to 20.

effectively generated synthetic instances for the minority class, resulting in an equalized representation of both phishing and legitimate instances. In order to prevent the machine learning model from becoming skewed towards a particular class, a balanced dataset is essential for training the model.

Figure 13 visually represents the results of applying Principal Component Analysis (PCA) for dimensionality reduction after the feature extraction and selection process using autoencoders and a shallow neural network. In this visualization, the selected features from the dataset have been transformed into a two-dimensional space using PCA. Each point on the scatter plot corresponds to a data instance, and the color of the points is determined by the target variable (phishing or legitimate). The scatter plot demonstrates the distribution and separation of instances based on the reduced features. The goal of this visualization is to showcase how

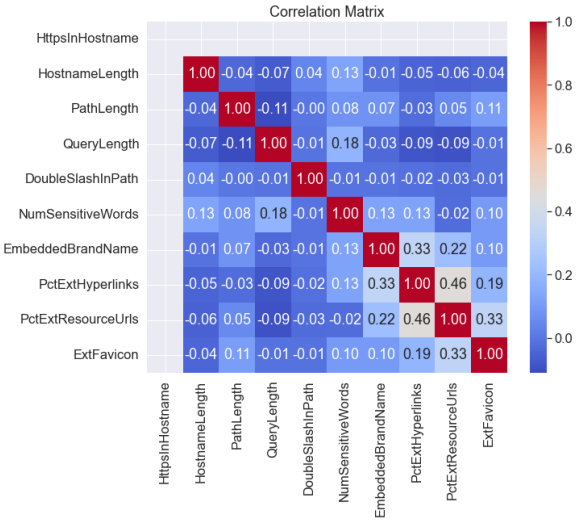

FIGURE 8. Confusion matrix of attribute 21 to 30.

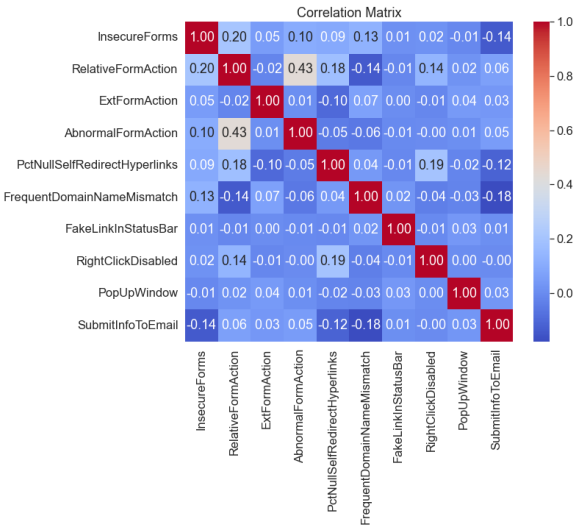

FIGURE 9. Confusion matrix of attribute 31 to 40.

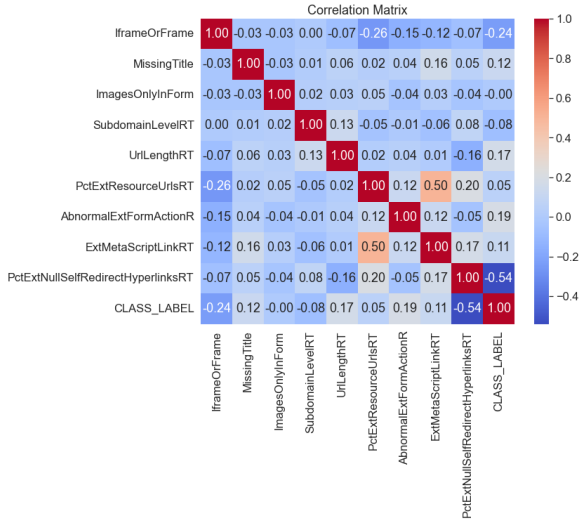

FIGURE 10. Confusion matrix of attribute 41 to 50.

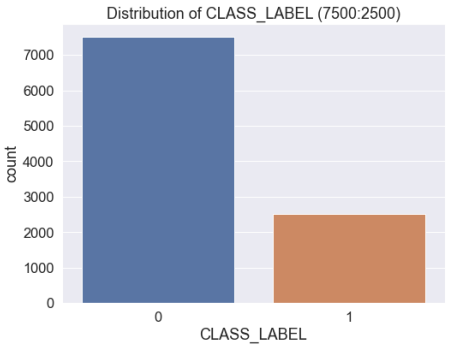

FIGURE 11. Imbalance data.

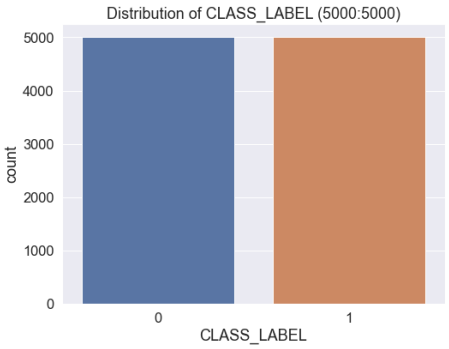

FIGURE 12. Balanced using SMOTE.

well the selected features, obtained through the combined autoencoder and shallow neural network approach, contribute to distinguishing between phishing and legitimate websites in a lower-dimensional space. The plot provides insights into the clustering and separation of data points, offering a visual assessment of the effectiveness of the feature extraction and selection technique for phishing website detection.

We used the optimization technique JA to send the hyperparameters through RNT-J and evaluate the model's performance. Furthermore, we assessed our model by calculating its True Positive as well as True Negative values, which are displayed for the suggested and current approaches in Figure 14.

The confusion matrices corresponding to the classification techniques used in this work are shown in Figures 15 to 19. It offers a graphic depiction of the model's performance in terms of TP, TN, FP, and FN, specifically. Among the numerous approaches that are currently in use, the suggested RNT-J method is noteworthy for having the lowest percentages of false positives and false negatives. This discovery highlights the effectiveness of RNT-J in reducing misclassifications, demonstrating its superiority over other approaches in obtaining a more accurate and dependable phishing detection.

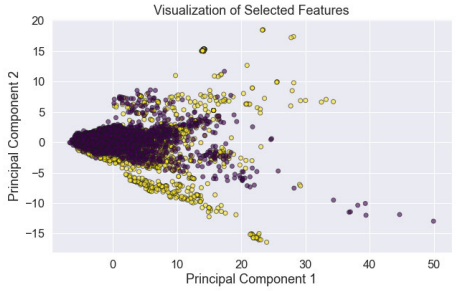

FIGURE 13. Component analysis of the features.

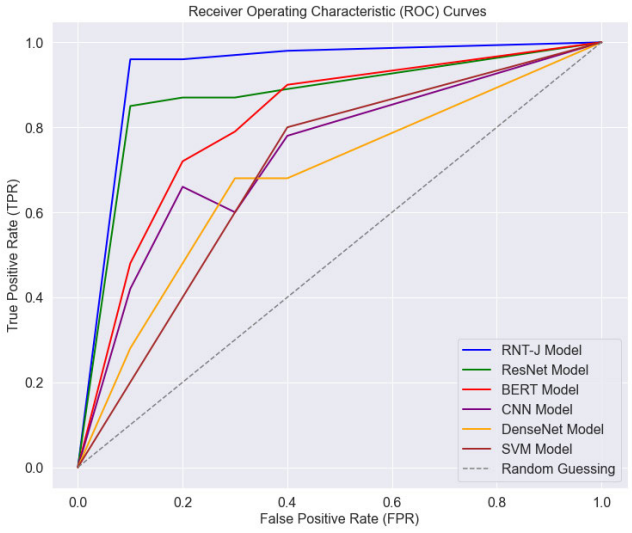

FIGURE 14. Suggested method's and the current approaches' ROC curves.

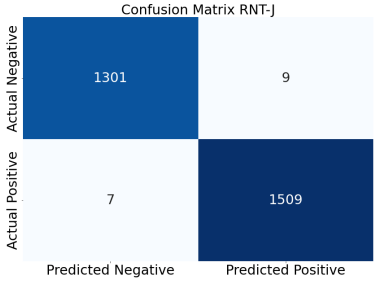

FIGURE 15. Confusion matrix of proposed method.

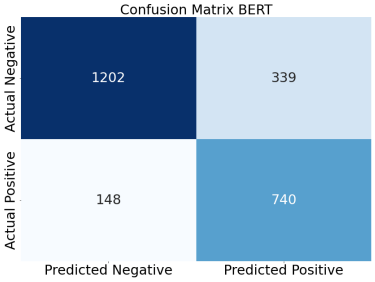

FIGURE 16. Confusion matrix of BERT.

The accuracy values of several strategies used to identify financial fraud using the dataset are shown in Figure 20.

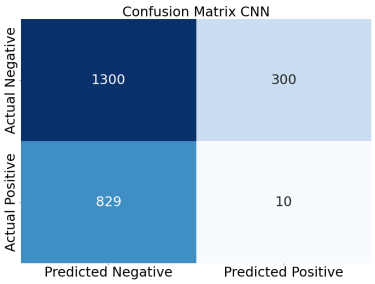

FIGURE 17. Confusion matrix of CNN.

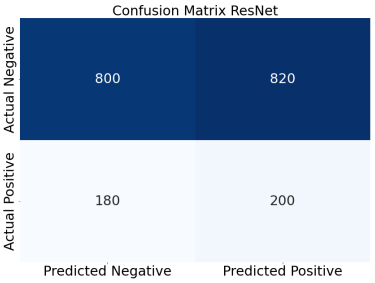

FIGURE 18. Confusion matrix of ResNet.

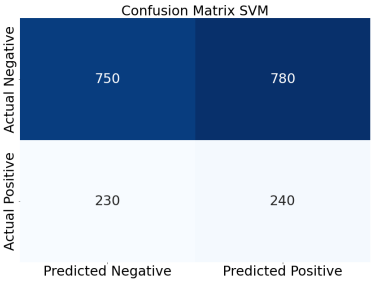

FIGURE 19. Confusion matrix of SVM.

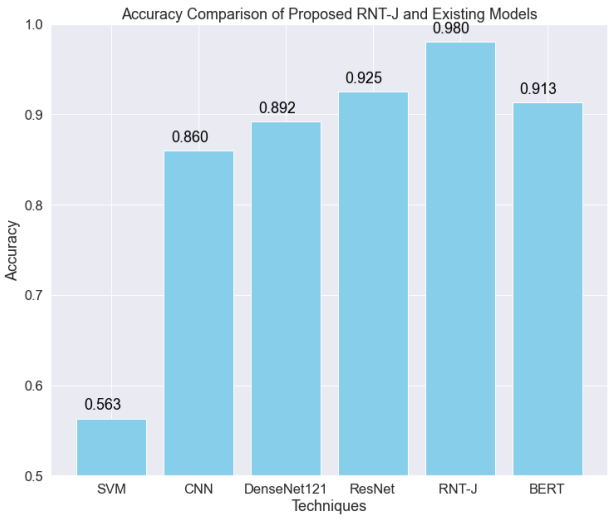

FIGURE 20. Precision of both suggested and current (Dataset).

Higher accuracy levels indicate greater performance. Accuracy quantifies how effectively each approach properly identifies fraud events.

TABLE 4. Comparing the proposed RNT-J model's performance with current models.

Techniques	ROC	AUC	Log Loss	Recall	MCC	Precision	F1-Score	Accuracy
SVM [16]	0.710	0.786	0.982	0.92	0.313	0.510	0.664	0.563
CNN [19]	0.908	0.944	0.215	0.86	0.795	0.818	0.868	0.860
DenseNet121 [18] [12]	0.897	0.963	0.257	0.895	0.826	0.85	0.899	0.892
ResNet [18]	0.942	0.981	0.124	0.92	0.940	0.913	0.943	0.925
BERT (Transformer) [13]	0.919	0.973	0.200	0.91	0.847	0.871	0.917	0.913
CNN-LSTM [23]	0.880	0.920	0.180	0.89	0.820	0.830	0.860	0.870
RNN [21]	0.850	0.900	0.210	0.87	0.800	0.810	0.840	0.860
RNT-J (Proposed)	0.992	0.998	0.047	0.99	0.985	0.978	0.988	0.980

TABLE 5. Statistical analysis of the execution time.

Model	Mean	Median	Min	Max	Range	Standard Deviation
DenseNet [18]	84.99s	83.99s	74.99s	96.99s	20.99s	4.94s
CNN [22]	85.99s	88.99s	72.99s	95.99s	21.99s	5.38s
ResNet [35]	86.99s	84.99s	77.99s	94.99s	15.99s	4.67s
SVM [31]	85.99s	86.99s	74.99s	96.99s	20.99s	5.20s
BERT [35]	84.99s	84.99s	73.99s	95.99s	20.99s	5.35s
CNN-LSTM	78.99s	80.99s	72.99s	86.99s	14.99s	3.67s
RNN	75.99s	74.99s	68.99s	80.99s	12.99s	2.89s
RNT-J	36.99s	35.99s	34.99s	41.99s	5.99s	1.10s

Using a variety of Established Models with the phishing dataset, Table 4 offers an impressive analysis of the Proposed RNT-J model. With an astounding accuracy record of 98%, RNT-J has demonstrated remarkable performance. With this great precision, RNT-J is clearly a very successful approach for identifying phishing inside the dataset, demonstrating its outstanding capacity to recognize phishing attacks. RNT-J's improved performance compared to the evaluated existing models signifies a noteworthy advancement in security, providing a dependable and strong instrument for strengthening fraud detection systems and improving the overall accuracy and reliability of phishing attempt tracking.

Table 5 presents a statistical study of the processing time for many models intended to forecast the identification of phishing websites. The table provides insightful information by displaying how long each model takes to execute in order to provide predictions. It gives data on the degree of variance in these durations as well as the average time, median value, and lowest and maximum execution times.

SMOTE's Effect on Data Balance and Feature Selection: We evaluated several feature selection strategies in Table 6 by doing hypothesis testing on both equal (SMOTE) and data that is unbalanced. The outcomes show a notable difference between the two circumstances, underscoring the important role that data balance plays. Compared to the post-SMOTE condition before SMOTE was used, The whole information balance and the number of pertinent characteristics chosen varied significantly. This emphasizes how important SMOTE is for managing data imbalance and affecting how well feature selection techniques work.

TABLE 6. Performance assessment using PCA with various feature choices. (SFeat: Sample of feature).

Selection of Features with PCA Approach		SFeat A	SFeat B	SFeat C	SFeat D	SFeat E
	No. of Features >	18	20	34	38	50
Accuracy with SMOTE	Proposed	97.99	97.09	94.213	93.99	92.99
	ResNet [18]	91.19	87.19	82.19	82.20	82.09
	DenseNet121 [18]	87.89	83.89	78.89	75.19	74.99
	BERT [13]	90.59	88.49	88.49	86.39	86.49
	SVM [16]	80.49	76.49	71.49	71.21	70.45
	CNN [19]	54.99	50.99	53.99	52.99	51.99
	CNN-LSTM [23]	65.99	61.99	64.99	63.99	62.99
	RNN [21]	78.99	75.99	72.99	72.21	71.45
Accuracy without SMOTE	Proposed	82.99	81.99	79.213	72.99	67.99
	ResNet [18]	76.19	72.19	67.19	67.20	67.09
	DenseNet121 [18]	72.89	68.89	63.89	60.19	59.99
	BERT [13]	75.59	73.49	73.49	71.39	71.49
	SVM [16]M	65.49	61.49	56.49	56.21	55.45
	CNN [19]	39.99	35.99	38.99	37.99	36.99
	CNN-LSTM [23] [23]	48.99	45.99	47.99	46.99	45.99
	RNN [21]	68.99	65.99	62.99	62.21	61.45

SFeat means Feature Sample

VI. CONCLUSION

With significant implications for digital forensics and online security, our work represents a breakthrough in the ongoing problem of phishing website detection. Despite technical advancements, creative solutions are still necessary to counter growing phishing threats. Our model RNT-J, specifically designed for real-time phishing website analysis within the digital forensics framework, handles modern complexity and big datasets related to phishing threats with exceptional performance. RNT-J is better able to identify and assess phishing behaviour in addition to digital forensics. In a comparative analysis against ResNet, DenseNet, BERT, and ELMo, RNT-J outperforms these methods in terms of accuracy, swift pattern identification, and efficient integration with digital forensics capabilities. Two key elements of RNT-J's superiority are the considerable improvement in phishing website recognition accuracy and the quick discovery of hitherto unknown patterns. This concept provides a comprehensive strategy that effectively addresses inefficiencies in the present methods employed in the field of digital forensics. Our model aptly reflects the need for more comprehensive comparative studies, demonstrating the potential of RNT-J for performance evaluations based on real-time information obtained from phishing websites. This methodology, when seen within the broader context of digital forensics, represents a major advancement in the fight against phishing attempts, leading to improved security and operational efficacy.

Our study contributes significantly to defence online activities in the field of cyber security, where sophisticated algorithms are necessary to fortify defences. It also closely studies digital crimes associated with phishing attempts. We want to use more sophisticated hybrid algorithms in the future, which will improve the accuracy and efficiency with which we identify phishing attacks.

ABBREVIATIONS

The following abbreviations are used in this study.

TABLE 7. Abbreviations.

Abbreviation	Full Form
PCA	Principal Component Analysis
SVM	Support Vector Machine
RNT-J	ResNeXt-GRU
BERT	Bidirectional Encoder Representations from Transformers
IFM	Isolation Forest Modeling
GRU	Gated Recurrent Unit
JA	Jaya Algorithm
CNN	Convolutional Neural Network
EARN	Ensemble of Autoencoders and ResNet Models
URL	Uniform Resource Locator
FN	False Negative
IPDS	Phishing Detection System
RNT	ResNeXt-embedded Gated Recurrent Unit model
DL	Deep Learning
HTDL	Tag Distribution Language
LSTM	Long Short-Term Memory
TP	True Positive
RF	Random Forest
DGA	Domain Generation Algorithm
SMOTE	Synthetic Minority Over-sampling Technique
MSE	Mean Squared Error

ACKNOWLEDGMENT

The authors would like to thank the University of Jeddah for its technical support.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] A. K. Dutta, "Detecting phishing websites using machine learning technique," *PLoS One*, vol. 16, no. 10, Oct. 2021, Art. no. e0258361.

[2] M. Mijwil, I. E. Salem, and M. M. Ismaeel, "The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review," *Iraqi J. Comput. Sci. Math.*, vol. 4, no. 1, pp. 87–101, Jan. 2023.

[3] M. S. Pour, C. Nader, K. Friday, and E. Bou-Harb, "A comprehensive survey of recent Internet measurement techniques for cyber security," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103123.

[4] C. C. L. Tan, K. L. Chiew, K. S. C. Yong, Y. Sebastian, J. C. M. Than, and W. K. Tiong, "Hybrid phishing detection using joint visual and textual identity," *Expert Syst. Appl.*, vol. 220, Jun. 2023, Art. no. 119723.

[5] M. F. Alghenaim, M. A. A. Bakar, and F. A. Rahim, "Awareness of phishing attacks in the public sector: Review types and technical approaches," in *Proc. 2nd Int. Conf. Emerg. Technol. Intell. Syst. (ICETIS)*, Al Buraimi, Oman, 2022, pp. 616–629.

[6] S. Patil and S. Dhage, "A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, Mar. 2019, pp. 588–593.

[7] Y. Su, "Research on website phishing detection based on LSTM RNN," in *Proc. IEEE 4th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Chongqing, China, Jun. 2020, pp. 284–288.

[8] K. Althobaiti, N. Meng, and K. Vaniea, "I don't need an expert! Making URL phishing features human comprehensible," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Yokohama, Japan, May 2021, p. 117.

[9] A. Ali, B. A. S. Al-Rimy, A. A. Almazroi, F. S. Alsubaei, A. A. Almazroi, and F. Saeed, "Securing secrets in cyber-physical systems: A cutting-edge privacy approach with consortium blockchain," *Sensors*, vol. 23, no. 16, p. 7162, Aug. 2023.

[10] C. Opara, B. Wei, and Y. Chen, "HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, p. 18.

[11] H. M. Abualrejal, R. ShahiraRohmat, T. K. Al-Ormuza, M. Mohamad, A. A. Almazroi, and M. A. Kafi, "Factors affecting online banking adoption among students during COVID-19: Case of Universiti Utara Malaysia-UUM," in *Proc. Int. Conf. Intell. Technol., Syst. Service Internet Everything (ITSS-IoE)*, Dec. 2022, pp. 1–4.

[12] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing webpage classification via deep learning-based algorithms: An empirical study," *Appl. Sci.*, vol. 11, no. 19, p. 9210, Oct. 2021.

[13] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics*, vol. 9, no. 9, p. 1514, Sep. 2020.

[14] W. Abu-Ulbeh, M. Altalhi, L. Abualigah, A. A. Almazroi, P. Sumari, and A. H. Gandomi, "Cyberstalking victimization model using criminological theory: A systematic literature review, taxonomies, applications, tools, and validations," *Electronics*, vol. 10, no. 14, p. 1670, Jul. 2021.

[15] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Riyadh, Saudi Arabia, Mar. 2020, pp. 1–6.

[16] I. Ahmad, M. A. Alqarni, A. Ali Almazroi, and A. Tariq, "Experimental evaluation of clickbait detection using machine learning models," *Intell. Autom. Soft Comput.*, vol. 26, no. 4, pp. 1335–1344, 2020.

[17] M. Gopinath and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," *Comput. Sci. Rev.*, vol. 47, Feb. 2023, Art. no. 100529.

[18] Y. Zhu, M. Wang, X. Yin, J. Zhang, E. Meijering, and J. Hu, "Deep learning in diverse intelligent sensor based systems," *Sensors*, vol. 23, no. 1, p. 62, Dec. 2022.

[19] 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.

[20] R. Shajahan and P. L. Lekshmy, "Hybrid learning approach for e-mail spam detection and classification," in *Proc. Intell. Cyber Phys. Syst. Internet Things (ICoICI)*, Coimbatore, India, 2022, pp. 781–794.

[21] E. Benavides-Astudillo, W. Fuertes, S. Sanchez-Gordon, G. Rodriguez-Galan, V. Martinez-Cepeda, and D. Nuez-Agurto, "Comparative study of deep learning algorithms in the detection of phishing attacks based on HTML and text obtained from web pages," in *Proc. 4th Int. Conf. Appl. Technol. (ICAT)*, Quito, Ecuador, 2022, pp. 386–398.

[22] 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.

[23] P. Ponni and D. Prabha, "Randomized active learning to identify phishing URL," in *Proc. 1st Int. Conf. Adv. Commun. Intell. Syst. (ICACIS)*, 2022, pp. 533–539.

[24] J. Zhou, H. Cui, X. Li, W. Yang, and X. Wu, "A novel phishing website detection model based on LightGBM and domain name features," *Symmetry*, vol. 15, no. 1, p. 180, Jan. 2023.

[25] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022.

[26] A. Dawabsheh, M. Jazzar, A. Eleyan, T. Bejaoui, and S. Popoola, "An enhanced phishing detection tool using deep learning from URL," in *Proc. Int. Conf. Smart Appl., Commun. Netw. (SmartNets)*, Palapye, Botswana, Nov. 2022, pp. 1–6.

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

[28] E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad, and A. I. A. Alzahrani, "A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators," *Sensors*, vol. 23, no. 9, p. 4403, Apr. 2023.

[29] H. T. M. Fetooh, M. M. El-Gayar, and A. Aboelfetouh, "Detection technique and mitigation against a phishing attack," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 177–188, 2021.

[30] A. Assefa and R. Katarya, "Intelligent phishing website detection using deep learning," in *Proc. 8th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, Mar. 2022, pp. 1741–1745.

[31] A. Almomani, M. Alauthman, M. T. Shatnawi, M. Alweshah, A. Alrosan, W. Alomoush, B. B. Gupta, B. B. Gupta, and B. B. Gupta, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *Int. J. Semantic Web Inf. Syst.*, vol. 18, no. 1, pp. 1–24, Feb. 2022.
[32] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 2015–2028, May 2019.
[33] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommun. Syst.*, vol. 68, pp. 687–700, Dec. 2017.
[34] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," *Frontiers Comput. Sci.*, vol. 3, Mar. 2021, Art. no. 563060.
[35] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 3043–3070, Jun. 2023.
[36] Y. Li, A. Yazdanmehr, J. Wang, and H. R. Rao, "Responding to identity theft: A victimization perspective," *Decis. Support Syst.*, vol. 121, pp. 13–24, Jun. 2019.
[37] M. A. Ali, M. A. Azad, M. P. Centeno, F. Hao, and A. van Moorsel, "Consumer-facing technology fraud: Economics, attack methods and potential solutions," *Future Gener. Comput. Syst.*, vol. 100, pp. 408–427, Nov. 2019.
[38] L. Elluri, V. Mandalapu, P. Vyas, and N. Roy, "Recent advancements in machine learning for cybercrime prediction," *J. Comput. Inf. Syst.*, vol. 1, pp. 1–15, Oct. 2023.
[39] T. Stojnic, D. Vatsalan, and N. A. G. Arachchilage, "Phishing email strategies: Understanding cybercriminals' strategies of crafting phishing emails," *Secur. Privacy*, vol. 4, no. 5, p. e165, Sep. 2021.
[40] Kaggle. *Phishing Dataset*. Accessed: Dec. 2, 2023. [Online]. Available: https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning
[41] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: Methods and prospects," *Big Data Analytics*, vol. 1, no. 1, pp. 1–22, Dec. 2016.
[42] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, Jul. 2021.
[43] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[44] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance," *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019.
[45] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2020, pp. 747–748.
[46] Y. Chen and W. Wu, "Isolation forest as an alternative data-driven mineral prospectivity mapping method with a higher data-processing efficiency," *Natural Resour. Res.*, vol. 28, no. 1, pp. 31–46, Jan. 2019.
[47] S. Viswanathan, M. A. Kumar, and K. P. Soman, "A sequence-based machine comprehension modeling using LSTM and GRU," in *Proc. Int. Conf. Emerg. Res. Electron., Comput. Sci. Technol. (ICERECT)*, 2019, pp. 47–55.

FAISAL S. ALSUBAEI (Member, IEEE) received the B.S.Eds. degree in computer science from King Abdulaziz University, Saudi Arabia, the M.Sc. degree in computer science, concentrating in security in computing from RMIT University, Australia, and the Ph.D. degree in computer science from The University of Memphis, USA. He is currently the Vice-Dean of the Deanship of Scientific Research and an Assistant Professor with the Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Saudi Arabia. He was a Software Engineer with Shaker and Associates Pty Ltd., Australia. He is a Microsoft Certified Technology Specialist, a Microsoft Certified Professional, and a Cisco Certified Entry Networking Technician. He is also an active member of Australian Computer Society and Linux Users of Victoria. His research interests include security and privacy in the IoMT and cloud computing.

ABDULWAHAB ALI ALMAZROI received the M.Sc. degree in computer science from the University of Science, Malaysia, and the Ph.D. degree in computer science from Flinders University, Australia. He is currently an Associate Professor with the Department of Information Technology, College of Computing and Information Technology at Khulais, University of Jeddah, Saudi Arabia. His research interests include parallel computing, cloud computing, wireless communication, and data mining.

NASIR AYUB (Student Member, IEEE) received the M.S. degree in the domain of smart grid and data science from COMSATS University Islamabad, Pakistan. He is currently pursuing the Ph.D. degree in computer science with the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad. Previously, he was a Lecturer and a Senior Lecturer with FUUAST Islamabad and CUST University Islamabad, respectively. He is also a Faculty Member with the Department of Creative Technologies, Air University Islamabad, Pakistan. His research interests include various areas, including smart grid, machine learning, deep learning, natural language processing, and blockchain. He actively contributes to the academic community as a Reviewer for IEEE Access, Elsevier, MDPI, and Peer journals. Furthermore, he is also serving as an Academic Editor for *PLOS One* and other journals.