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.2017.DOI

Enhancing Smishing Detection: A Deep Learning Approach for Improved Accuracy and Reduced False Positives

Muhammad Khalid Mehmood [1], Humaira Arshad [1] [0000−0003−3615−6202] and Moatsum Alawida [2] [0000−0001−8146−5843] and Abid Mehmood [2] [0000−0002−1468−7259]

[1]Department of Computer Science, The Islamia University of Bahawalpur, Baghdad-ul-Jadeed Campus, Bahawalpur 63100, Pakistan
[2]Department of Computer Sciences, Abu Dhabi University, Abu Dhabi 59911,United Arab Emirates

Corresponding authors: Moatsum Alawida (e-mail: moatsum.alawida@adu.ac.ae)

ABSTRACT The widespread use of smartphones and their constant connection to the Internet makes them vulnerable to phishing attacks. Phishing is the act of sending malicious content such as emails to unsuspecting individuals. Smishing, a hybrid of Short Message Service (SMS) and phishing is a well-known cybersecurity problem in which attackers send malicious SMS messages to their targets. This practice is deceptive and aims to mislead individuals into exposing personal information or completing certain activities through text messages. Although researchers have presented various techniques to detect smishing, there is still a lack of methods to significantly reduce false-positive predictions, which are incorrect classifications of legitimate messages as malicious. The proposed method leverages the effectiveness of deep learning to automatically extract significant features from text messages to determine whether it is smish or legitimate. Comparative analysis is performed with traditional machine learning models to highlight the superiority of deep learning models in smishing attack detection. This work aggregates Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). Results reveal that the proposed CNN-LSTM architecture shows robust performance by achieving a 0.9974 accuracy score and a high precision score, indicating a low number of false positives in detecting smishing attacks. The model also demonstrated high recall and F1-score, indicating robust performance. The proposed method has a lot of real-world implications, such as helping to design proactive defence mechanisms against smishing attacks and improving cybersecurity in the mobile communication sector.

INDEX TERMS SMS Classification, Smishing attacks, Cybersecurity, Deep learning, CNN, LSTM

I. INTRODUCTION

The advancement and evolution of information technology have resulted in Internet connectivity and increased smartphone usage. Smartphones are becoming increasingly luxurious, and users are connecting to the internet worldwide [1]. Facebook and Google are always working to enhance the usability of the internet. As technology becomes more affordable, the number of smartphone users continues to increase [2]. Because of the enormous number of Internet users, there has been a significant increase in cybercrime occurrences such as phishing [3]. As consumers store sensitive information, such as banking or credit card data, on their smartphones, the concern for user privacy is increasingly becoming a critical issue in the realm of smart devices. Attackers hope to obtain sensitive information, such as user names, passwords, and financial information by impersonating a respectable company [4]. Mobile phone users are especially vulnerable to phishing attempts for various reasons: they are unable to verify their authenticity, adopt unsafe behaviors, and enter their credentials whenever prompted [5].

Short Message Service (SMS) is a communication technology used to exchange text messages between mobile devices. It is a frequently used messaging application that allows users to send brief text-based messages to one another. Text messaging is extensively employed in advertising [6]. The more common use of text messages is advantageous to attackers. Every hour, approximately 9132 million text messages are transmitted worldwide [7]. In less than three seconds, more than 90% of SMSs are opened [8]. This demonstrates that consumers have an urgency-related feeling, which is crucial for focusing on cyberattacks.

An SMS phishing attack involves sending a message to a user containing links directing them to malicious websites, applications, or user interfaces where they are prompted to input their credentials. SMS is preferred by attackers over email for phishing [9] for the following reasons: individuals respond more quickly to text messages, attackers can send SMS messages easily, and mobile devices have small displays that can obscure signs of phishing. Smishing is a combination of the SMS and Phishing methods. The term "SMISHING" was first used by McAfee's David Rayhawk in 2006 [4]. Smishing, or SMS-based phishing, involves sending infected SMS messages to a target with the intent of prompting them to disclose personal information to an attacker, who is referred to as the 'smisher.' A sample smishing message is shown in Figure 1.

According to the Anti-Phishing Working Group (APWG) [10], the number of phishing websites is increasing annually. The number of phishing and smishing attacks increased significantly during the COVID-19 pandemic. During the COVID-19 crisis, fraudulent text messages were claimed from health organizations, urging recipients to click on links for information or testing, potentially leading to a malicious website aimed at collecting personal information.

During the pandemic, the United Nations (UN) reported a 350% spike in phishing assault [11]. Among the many communication media, such as telephones, SMS, and email, attackers favor text messaging for phishing assaults. Owing to the high response rate of consumers, attackers favor text messages. According to CallHub, the response rate to SMS messages is notably higher than that observed for email messages [12].

Several techniques for identifying smishing have been reported in the literature. The majority of solutions include specific instructions, such as being mindful of unfamiliar links in the SMS. However, the challenge of identifying smishing signals has been briefly discussed in the literature. When analyzing existing techniques, it is observed that most employ blacklist, heuristic, and whitelist approaches [13] for smishing attack detection. However, the blacklist approach is ineffective because the URL domains often change and employ URL shortcut services. Consequently, attackers routinely change the names of domains for phishing attempts, rendering blacklisting-based strategies useless. Several researchers have employed heuristic techniques in their detection methods. While the presence of specific terms and a URL in a text message could raise suspicion of smishing, it is important to consider that it might also be legitimate communication from a service provider or a website. These approaches tend to generate too many false positives, which are rendered useless without an additional layer of scrutiny. Unless the URL points to a malicious website to obtain user credentials or to start the automated download of a dangerous item, the message should not be categorically labelled as smishing; it could be spam. Hence, before classifying communication as smishing, it is critical to verify the URL included inside it [14]. Existing approaches for detecting smishing messages have several limitations that make it impossible to realize a smishing-identifying mechanism in real-world applications. Therefore, it is critical to develop a novel and effective strategy for detecting smishing communication while preventing false-positive results.

Presently, Deep learning models are at the forefront of artificial intelligence, enabling machines to learn and make complex decisions by simulating the human neural network in the brain [15]. The goal is to provide the output along with the given input sets so that the neural network can compare it with the expected output. The authors employed a flow-based technique to perform classification using several aspects of SMS and URLs [7]. The aforementioned methodologies classify the data using machine learning.

This study handles the urgent problem of smishing detection, which is becoming increasingly critical due to the widespread use of smartphones and the rising threat of cybercrime. Current smishing detection methods have limitations, with high false-positive rates and difficulty adapting to new attack strategies. This highlights the need for new, more accurate techniques capable of distinguishing smishing messages from legitimate ones. We aim to contribute valuable insights into the development of more reliable and efficient smishing detection systems. Through the identification of various feature extraction techniques and the evaluation of model performance, our research seeks to address the limitations of existing methods and pave the way for more effective strategies in combating smishing attacks. The goal of this research is to create an intelligent and accurate smish detection technique that utilizes a deep-learning model. The contributions of this study are as follows.

- A unified framework of deep learning models is proposed to achieve high detection accuracy and reduce the number of false positives for detecting smishing attacks. To this end, we introduce a multi-layered deep learning architecture called CNN-LSTM, which leverages the benefits of both CNN and LSTM structures. The models are combined sequentially, with CNN handling feature extraction and LSTM capturing the context and long-range dependencies within the SMS message to detect phishing attacks.
- We aim to evaluate and compare the effectiveness of various deep learning architectures, including Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), bidirectional LSTM (BiLSTM), and Gated recurrent unit (GRU models), in accurately detecting smishing attacks. Additionally, we compare the performance of our proposed model with traditional rule-based approaches and other machine learning techniques to reliably distinguish smishing SMS deposits from legitimate SMS deposits. The contribution of our approach lies in its ability to achieve significantly lower false positives compared to other models. By effectively reducing the number of false positives, our model enhances the reliability and accuracy of smishing attack

FIGURE 1: Smishing SMS Sample

detection, thereby improving overall security measures in communication systems.

- To raise awareness about the prevalence and severity of smishing attacks among both researchers and practitioners in the cybersecurity community. By highlighting the effectiveness of advanced detection methodologies, this study underscores the importance of prioritizing smishing detection efforts in cybersecurity issues worldwide.

The following sections of this paper are organized as follows. Section II delves into previous research on Smishing SMS deposits. In Section III, we highlight the research gap in existing state-of-the-art papers. Section IV provides an overview of the materials and methods employed in the experiments to enhance efficacy. The results are presented and discussed in Section V. Finally, Section VI concludes the paper and outlines potential avenues for future work.

II. RELATED WORK

Researchers have focused more on smishing in recent years because of its prominence in mobile smishing attacks. Some studies have designed methods to identify smishing communications [16]–[20]. Some approaches for spam detection that employ similar techniques have also been addressed in [21], [22]. Many scholars have studied smishing tactics and detection methodologies to raise awareness among users and researchers [23]–[26]. This section presents numerous studies by other researchers in the field of smishing detection. We have covered several phishing detection methods and comparable strategies utilized for phishing detection [27]–[30].

The authors proposed an S-detector method for recognizing smishing communications [18]. The SMS monitor, database, SMS analyzer, and determinant are the four components of an S-Detector. This model examines the URL to determine whether the Android Application Package (APK) file was downloaded. An SMS is classified as a smishing message when it downloads an executable file. They analyzed the content of the communication using the Naive Bayes classification technique. Another study predesigned a rule-based strategy for detecting smishing communication [19]. They identified nine principles to distinguish smishing messages from real mail messages. These rules were also taught using several classification methods, such as PRISM. The model evaluation demonstrated a true negative rate greater than 98%. The authors in [17] presented a model called SmiDCA (SMIshing Detection based on Correlation Algorithm) to identify smishing messages using a supervised model. Using the correlation approach, the suggested model collected thirty-nine attributes from smishing SMS. Supervised machine learning techniques were utilized, and tests were conducted on a variety of datasets. Using the Random Forest classifier, the experimental assessment by SmiDCA yielded an accuracy of 96.40. In another study [21], a hybrid CNN-LSTM model was proposed to detect SMS spam messages in both Arabic and English, achieving an accuracy of 0.9837. Similarly, the authors of [24] used SVM to detect smishing attacks, and their experimental results showed an accuracy of 0.9839. Another study [31] proposed an ensemble of NB, RF, ETC for the detection of spam SMS, achieving an accuracy score of 0.969.

The authors suggested a smishing detection technique called smishing classifier for recognizing smishing communications [16]. Using the Nave Bayesian Classifier, this system examines the content and keywords of SMS. The proposed approach validates the presence of URLs in the SMS and examines the sender's cell number. This model also evaluates the appearance of the homepage and file download. The authors addressed numerous forms of smishing and phishing attacks [32]. They devised a URL test to validate the URL supplied in SMS text. They also explored the smishing box technique as a precaution against downloading files during smishing attacks.

Researchers have employed content-based techniques to smish SMS deposits [20]. The most commonly used terms for smishing SMS deposits were identified using machine learning techniques. In addition to assessing the appearance of the login page and a.apk file download, this model evaluates URL maliciousness. Researchers have developed a feature-based model that leverages machine learning

techniques to identify smishing SMS deposits [33]. The proposed model retrieved various attributes from smishing messages and analyzed the findings on the same dataset using five machine learning methods. Using similar techniques, other researchers have presented spam SMS detection models [23]. The authors proposed a spam detection model called 'SMSAssassin,' which is based on the Naive Bayes method and a Support Vector Machine. The authors utilized Crowdsourcing to maintain the database. Users with the SMS Assassin app installed on their mobile devices can contribute by updating the list of spam keywords with any spam messages they encounter, thereby enhancing the system's performance. This paper also describes the development of a content-based SMS filtering system for spam text detection. They created and built a mobile application that managed various types of messages, making the SMS administration interface easy to use.

Other researchers have raised user and researcher awareness by addressing smishing detection techniques. The authors discussed different taxonomies for detecting phishing on mobile phones [34]. They mentioned and commented on numerous phishing strategies such as voice phishing, Bluetooth, and mobile app phishing. The authors illustrated several forms of phishing attacks and related technical approaches [35]. They also covered several phishing mitigation tactics and best practices for preventing phishing assaults on mobile devices. Their efforts were aimed at raising security awareness among mobile users. Researchers have examined phishing attempts thoroughly [36]. They addressed numerous mobile phishing attempts by attackers and their defences. They also covered the taxonomy of phishing solutions, author strategies, numerous obstacles associated with the identification of phishing assaults, and the development of various phishing datasets. The authors of [20] described several smishing techniques used by attackers. The author also recommended certain measures that users could employ to combat smishing attacks. This paper also discusses several approaches and methodologies for countering smishing attacks.

The authors created a model named 'PhiDMA' with five levels to identify phish ing websites [37]. They created a prototype of this model that included an interface for visually challenged people. The experimental findings showed 92.72% accuracy. Researchers have created a phishing detection approach called " PhisherCop, " which identifies phishing attempts on mobile devices [38]. The proposed model is based on the SVC and SGD for mobile devices. The proposed approach evaluates the source code, the URL, and the IP address of a website to determine its maliciousness. The authors presented a method that inspects URLs using phishing URL properties such as lexical and hosted features [39]. They introduced a model to identify phishing websites using selected features. They demonstrated the high accuracy of their proposed approach by testing it on numerous datasets. Catherine [40] proposed a technique for classifying phishing websites, analyzed the efficient aspects of detecting phishing sites, and created a rule-based approach. The author used the frequency of attributes in their datasets to determine the common characteristics for detecting phishing websites.

A phishing website detection approach using Term Frequency-Inverse Document frequency (TF-IDF) analysis content was proposed in [41]. Each word in a document has its TF-IDF score computed and the top five words are selected. A lexical signature is obtained by providing a search engine with a set of chosen words. The authors say that a website is legitimate if its domain name matches one of the top N search results; otherwise, it is labelled as a phishing website. A method for identifying phishing websites has been proposed by the authors [42]. They devised a heuristic approach called the twin support vector machine classifier (TWSVM), which relies on the features of phishing URLs and online sites. They performed phishing detection by analyzing home and login pages, and according to the testing results, the combination of HTML- and URL-based elements was quite efficient in identifying authentic websites. A literature review revealed that using machine learning algorithms only on URL links and hyperlinks cannot identify diverse smishing tendencies. Table 1 summarizes the related work on Smishing SMS detection.

III. RESEARCH GAP

Although studies have been conducted on smishing detection, significant gaps still exist that provide opportunities for additional exploration and improvement. The research gaps are as follows.

- There is a research gap in analyzing and comparing the efficacy of various deep learning architectures and methods designed particularly for smishing detection with established rule-based approaches and other machine learning methods. Limited attention has been paid to deep learning techniques; despite the potential of deep learning being observed in several fields, smishing detection can be one of the few applications.
- Some studies have considered the linguistic features for smishing detection. However, further research is needed to delve more deeply into these aspects. There is a need to identify and analyze the patterns present in smishing messages that can aid in accurate detection. Incorporating advanced feature extraction methods and natural language processing tools can enhance the effectiveness of smishing detection models.
- Access to big, diversified, and precisely labelled datasets is crucial to developing reliable smishing detection algorithms. However, the availability of such datasets for training and assessing smishing detection algorithms is limited.

IV. MATERIAL AND METHODS

This section presents the dataset, feature extraction methods, machine learning models [43], and deep learning models [44] used in the experimental design. The experimental

TABLE 1: Summary of Studies on Smishing Detection

Ref.	Year	Dataset	Method/Approach	Findings
[16]	2018	SMS text messages	Content and keywords analysis using the Naive Bayesian Classifier	Detection of smishing SMS with URL validation and sender cell number check.
[17]	2018	5578 English and 1893 non-English datasets	SmiDCA - Supervised a model with a correlation approach	Achieved 96.40% accuracy on English Dataset.
[19]	2018	5169 Text Messages	Rule-based strategy for smishing detection	Demonstrated true negative rate over 98%.
[20]	2019	5572 text Messages	Machine learning, URL and content Analysis	Analyzed malicious keywords and malicious behavior of the URL.
[33]	2019	5169 Text Messages	Machine learning	Tested on 5 Machine Learning methods for accuracy assessment.
[39]	2020	URLs collected from Phish-tank.com	Phishing URL Detection	Achieved 87% accuracy with their proposed approach.
[21]	2020	8304 text messages	A hybrid of CNN-LSTM	Achieved 98.37% accuracy.
[13]	2021	5574 English text messages, 2730 Arabic messages	A hybrid security model called "SM-Detector"	Identified malicious URLs and performed expression analysis.
[38]	2022	5574 SMS	Model-based on SVC and SGD for mobiles. PhisherCop - Phishing detection on mobile devices	Achieved 96% accuracy.
[32]	2022	5858 text messages	'Smishing Detector' based on Neural Network	Found that the URL feature is most significant and achieved 94% accuracy.
[24]	2022	5572 text messages	Proposed SVM for the detection of spam SMS.	The machine learning model achieved an accuracy score of 98.39%.
[31]	2023	5169 text messages	Devised a hybrid of NB, RF, ETC.	The proposed model yielded an accuracy of 96.9%.

approach is illustrated in Figure 2. While Figure 3 presents the workflow adopted in the proposed approach.

A. DATASET

In the current study, we conducted experiments to detect smishing SMS deposits using two datasets. One of the datasets, named 'SMS PHISH- ING DATASET, "was sourced from Mendeley, while the other dataset, known as 'SMS Smishing Collection Data Set," was obtained from Kaggle. Both datasets were merged into a single consolidated dataset to ensure a sufficiently large dataset for experimentation. Despite their different sources, both datasets consist of two classes: 'Smish' and 'Ham' SMS. By merging datasets from different sources, we wanted to provide the proposed model with a rich variety of smishing examples. This diversity helped the model learn intricate patterns and features associated with smishing behavior, improving its effectiveness. Table 2 shows the distribution of records across each dataset.

TABLE 2: The dataset's record count

Dataset	Source Smish	Ham	Total SMS
DATASET 1	Mendeley 1127	4844	5971
DATASET 2	Kaggle 747	4827	5574
Total	1874	9671	11545

B. PREPROCESSING STEPS

These datasets contain unstructured or semi-structured data, which may not necessarily contain meaningful information. The training process for the model is slowed by such extraneous inputs, which might also affect the model's performance. Preprocessing is essential for ML models to operate more effectively while using fewer computational resources. Enhancing a model's capacity for dependable outcome prediction involves text preparation, which encompasses processes such as tokenization, case conversion, stop-word elimination, and number removal. This study employed several preprocessing techniques, including number removal, stop word and punctuation removal, lowercase conversion, stemming, and lemmatization. The elimination of raw data will assist in lowering feature complexity because text sometimes contains extraneous text, which can affect the performance of the models. Python's natural language toolkit (NLTK) package was used for preprocessing. Removing punctuation and numbers: As punctuation and numerals are not relevant characteristics for model training, eliminating them from texts helps to simplify the features.

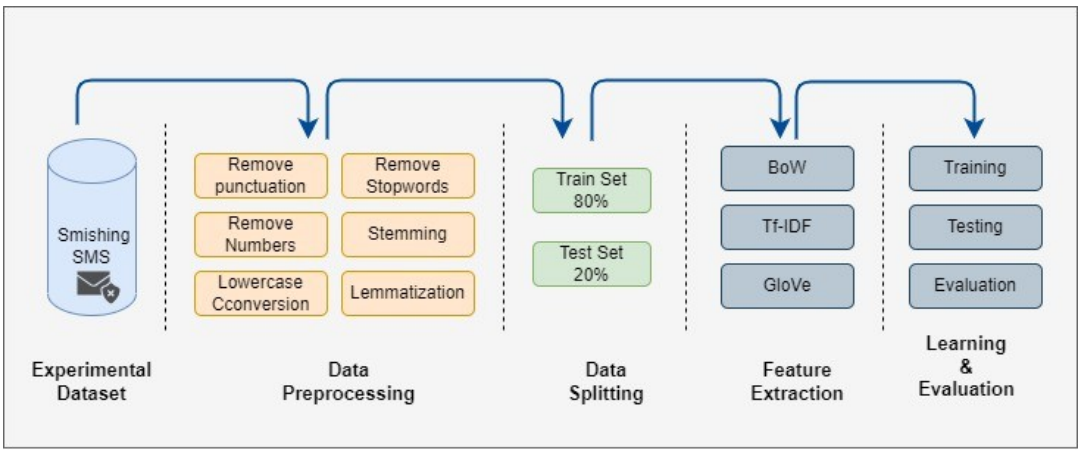


FIGURE 2: Architecture of methodologies adopted for Machine Learning Models

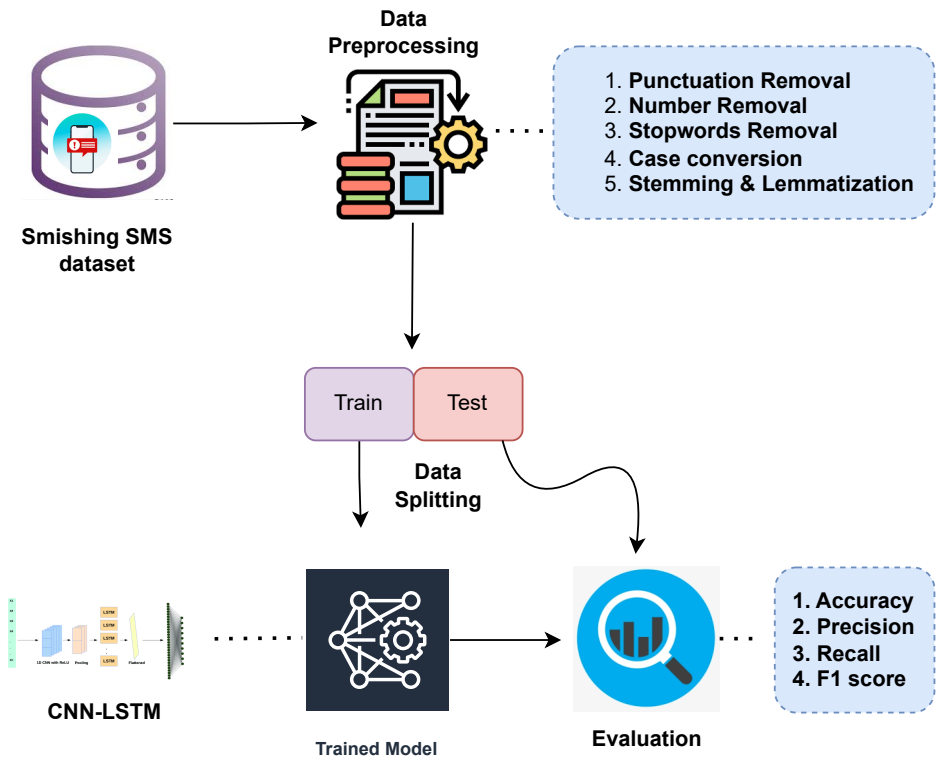


FIGURE 3: Architecture of methodologies adopted for the proposed Model

Regular expressions were used at this stage. Lowercase conversion: Converting all text to lowercase reduces feature space complexity and helps machine learning models perform better. Removal of stop words: The stop words were removed to improve the performance of the classifiers. Stemming eliminates and 'es after the word.' Lemmatization and Stemming: Lemmatization is more suitable because it considers the context of a word and converts it into the correct base form. The experiments in this study used Word Net Lemmatizer and NLTK Porter Stemmer libraries.

C. FEATURE EXTRACTION

Classifiers require textual content to be represented as vectors for training. For this purpose, textual data must be transformed into numbers without loss of information. This translation is possible by utilizing strategies that assign a number to each word, such as bag-of-words (BOW) and TF-IDF.

1) BoW

BoW is a basic and extensively utilized approach in natural language processing (NLP) to represent text data. It is a

straightforward technique that focuses on the occurrence and frequency of terms in a document while ignoring syntax, word order, and context. BoW performs remarkably well for issues such as text categorization and language modelling. Bow uses a count vectorizer to extract key characteristics. The count vectorizer operates like the term frequency. Each feature in the BoW has a value assigned to it, which denotes the frequency of that characteristic.

2) TF-IDF

The TF-IDF approach helps identify important and distinct words that may be used to separate papers within a corpus. It can be used for text categorization, information retrieval, keyword extraction, and other NLP applications. The utilization of TF-IDF values improves the representation of text data by considering both local word frequency and global term rarity.

$$W_{i,j} = [1 + \log(tf_{i,j})] \times \left[\log(\frac{N}{df})\right] \quad (1)$$

where N represents the total documents, $tf_{i,j}$ denotes the term frequency within a specific document, and df,t indicates a document containing text t.

D. MACHINE LEARNING MODELS

This study used various machine learning models to validate the proposed methodology, including SVC, AC, RF, ETC, GBM, and NB. Table 3 lists these models. The hyperparameters most appropriate for the dataset were used in these models. When selecting the best hyperparameters, the value ranges were adjusted for maximum performance.

E. DEEP LEARNING MODELS

This section provides an overview of the deep learning models employed in the experiments, including CNN, LSTM, BiLSTM, GRU, and CNN-LSTM. Figure 4 illustrates the hierarchical architecture of these deep-learning models.

1) CNN

A Convolutional Neural Network (CNN) is a deep learning architecture consisting of convolution and pooling layers designed to capture complex features [51]. CNNs are typically applied to tasks, such as image classification [52], hydrodynamic simulator [53] and climate forecasting [54]. The CNN model, which is end-to-end trained, is known for its reliability. The authors in [55] discussed uncertainty quantification of neural networks. In this feed-forward network, filters are applied to the output of each layer to extract features. Additionally, the CNN model incorporates fully connected, dropout, and pooling layers. The fully connected layer, which receives input from the previous layers, contributes to the final decision-making process. Pooling layers, whether employing maximum or average pooling, aid in the feature selection. Equation 2 utilizes the ReLU as the activation function.

$$y = max(0, i) \quad (2)$$

where y is the activation output, and i signifies its input. Convolutional layers utilize weights to obtain valuable information during the training process. Cross-entropy is a loss function that can be computed as follows.

$$crossEntropy = -(i.log(p) + (1-i).log(1-p)) \quad (3)$$

where p is the predicted probability and i is the class label. Because it is an improved version of the backpropagation model, the sigmoid error function is utilized to forecast CNN output. The CNN model produces two target classes.

2) LSTM

Long Short-Term Memory (LSTM) [56] is a cutting-edge deep learning algorithm that is commonly used to solve text categorization problems. An LSTM comprises three gates: the input gate (ik), the forget gate (fk), and the output gate (ok). As the data passes through these gates, the gates preserve important information while discarding unimportant data based on the dropout value. Memory-chunk Ck contains critical information. There are several LSTM versions; the one used in this study is shown in Equations 4, 5, 6, and 7.

$$i_k = \sigma(W_i s_k + V_i h_{k-1} + b_i) \quad (4)$$
$$f_k = \sigma(W_f s_k + V_f h_{k-1} + b_f) \quad (5)$$
$$o_k = \sigma(W_o s_k + V_o h_{k-1} + b_o) \quad (6)$$
$$c_k = \tanh(W_c x_k + V_c h_{k-1} + b_c) \quad (7)$$

W and V denote the appropriate weights for the matrix members. where sk represents the input at a certain moment, b represents the bias, and h represents the hidden state up to k1 time steps. For each k1 time step, the memory block cell c is updated. Every neuron in the dense layer communicates with every other neuron in the LSTM short-term memory output layer. LSTM has been applied in different fields [57].

3) GRU

Gated Recurrent Unit (GRU) [58] represents an alternative type of recurrent neural network architecture that simplifies the LSTM gating mechanisms while achieving comparable outcomes. It consists of two gates, a reset gate and an update gate. The reset gate assists the network in determining which previous information is discarded, whereas the update gate assists the network in determining how much of the current input is integrated. GRUs have gained popularity because of their simplified structure and competitive performance. for various sequential tasks. Given their capacity to regulate information flow and handle long-term dependencies, gated neural networks have been successful in modelling sequences and time-dependent patterns. They are extensively utilized in various applications, including speech synthesis, sentiment analysis, video analysis, and machine translation. The networks in these designs can learn when to recall, forget, and update information owing to the gating

TABLE 3: Machine Learning Models used in the study

Reference	Model	Strengths	Weaknesses
[45]	RF	RF is a versatile ensemble machine learning algorithm utilized for classification and regression. It aggregates multiple trees after training by random subsets of data to improve accuracy and reduce overfitting. With bagging and feature randomization techniques, it enhances robustness and generalization. RF can handle noisy, huge datasets, and it provides feature importance rankings. It's widely used in various domains due to its simplicity and excellent performance.	May prone to overfitting without proper tuning.
[46]	AC	AC is an ensemble machine learning algorithm that aggregates multiple weak learners, often decision trees, to create a robust and accurate predictive model. It focuses on previously misclassified data points by assigning them higher weights in each iteration, allowing the model to improve its performance over time. AdaBoost employs weighted voting to make predictions and is effective in boosting classification accuracy, especially in handling complex datasets. It also provides insights into feature importance.	Susceptible to overfitting with noisy data
[47]	GBM	is an ensemble machine learning algorithm known for its exceptional predictive accuracy and versatility. To build a strong predictive model, it integrates the outputs of several weak learners, often decision trees. GBM employs a gradient descent optimization technique, sequentially training decision trees to fix errors from prior trees. It is effective for both regression and classification tasks and offers robustness against overfitting. GBM can provide insights into feature importance and is widely applicable across various domains. Efficient implementations are XGBoost and LightGBM.	Sequential training may lead to longer training times. Makes strong independence assumptions between features
[48]	NB	A probabilistic machine learning technique called NB is largely utilized for classification tasks, particularly in text analysis and natural language processing. It leverages Bayes' theorem to calculate the probability of data points belonging to specific classes. Despite making a strong independence assumption between features (hence the "naive" label), Naive Bayes can perform well in various real-world applications. It comes in different variants like Gaussian, Multinomial, and Bernoulli, catering to different types of data. Naive Bayes is efficient, particularly in high-dimensional data scenarios like text analysis, and provides insight into feature importance. It's widely used in spam detection and sentiment analysis.	Makes strong independent assumptions between features.
[49]	SVC	The SVC is a machine learning algorithm primarily used for binary classification tasks but also applicable to regression and outlier detection. The goal of SVMs is to identify an ideal decision boundary, or hyperplane, that maximises the margin between data points belonging to various classes. This margin helps improve generalization and robustness. SVMs can handle non-linear data using a technique called the kernel trick and are effective in high-dimensional spaces. They offer a trade-off parameter (C) to balance margin maximization and error minimization. SVMs are robust to overfitting, provide interpretability through support vectors, and find applications in various domains, including text classification, image recognition, and medical diagnosis.	Requires careful selection of kernel and hyperparameters.
[50]	ETC	For classification tasks, ETC is an ensemble learning technique. It's similar to Random Forest but adds more randomness by selecting features and thresholds for splits randomly. This randomness reduces overfitting and makes it robust against noisy data. It uses bagging, parallelization, and regularization techniques, making it efficient for high-dimensional datasets. Tuning its hyperparameters is important for optimal performance. Due to its adaptability and capacity for handling complicated data, Extra Trees is frequently employed across several disciplines.	More randomness may lead to increased computational complexity.

mechanisms, which also enables them to record intricate connections within sequences. A GRU is a distinct variant

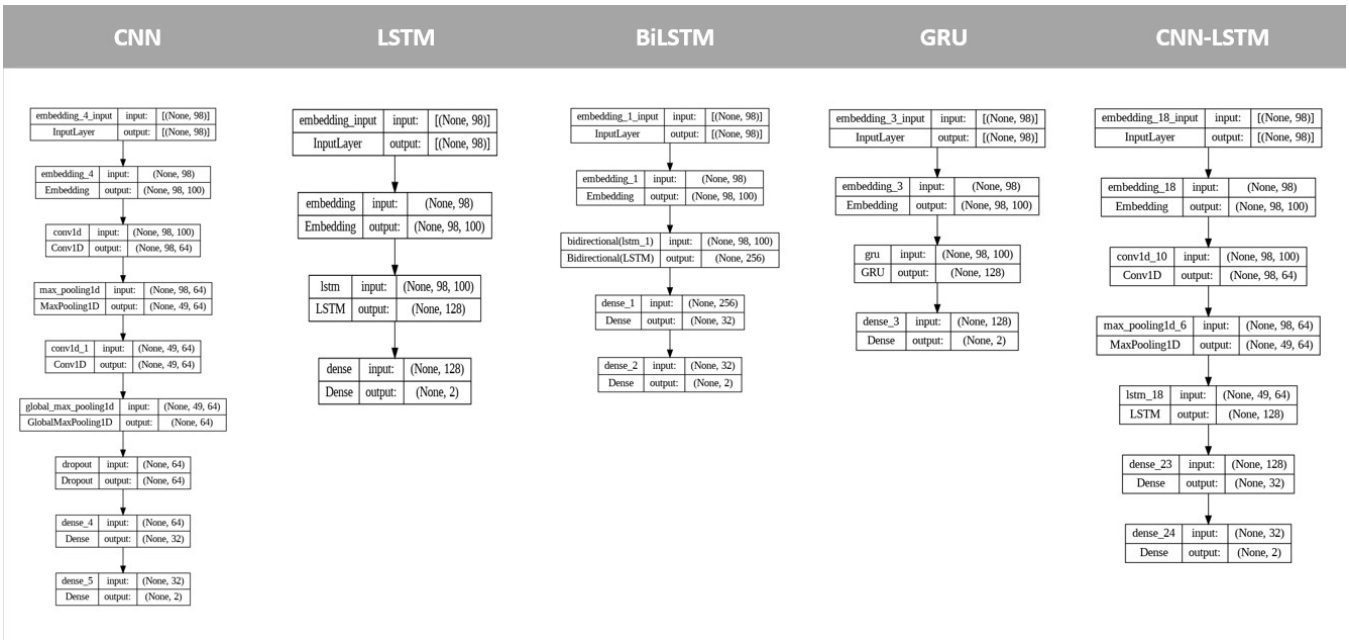


FIGURE 4: Architectural Layout of Deep Learning Models

of recurrent neural networks (RNNs) equipped with gating mechanisms to control the flow of input. Below, you can find the equations for the reset gate, update gate, and hidden state:

$$r_t = \sigma(W_r.x_t + U_r.h_{t-1} + b_r) \tag{8}$$

$$z_t = \sigma(W_z.x_t + U_z.h_{t-1} + b_z) \tag{9}$$

$$\tilde{h}_t = \tanh(W.x_t + U.(r_t \odot h_{t-1} + b)) \tag{10}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{11}$$

where xt is the input, ht is the hidden state, rt is the reset gate, and zt is the output gate at t time step. U and W matrices show weights, and b represents the bias. σ represents the sigmoid function and tanh represents a hyperbolic tangent function.

4) BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) [59] is a form of RNN that can process sequences in both directions. This design improves context comprehension by considering both past and future data. The input sequence is handled in the same manner as a normal LSTM in the forward pass. The input from each time step was merged with the hidden state and memory cell from the previous time step to create a new hidden state and memory cell. This information was then sent to the sequence. The input sequence was reversed, and each time step's input was coupled with the hidden state and memory cell of the time step after it during the backward pass. Through backward propagation, dependencies that might not be observed from the forward pass alone can be detected by the network. The final hidden state of each time step is created by concatenating the forward and backwards hidden states. With this integrated form, the sequence's present and potential future are jointly encoded. BiLSTMs are beneficial for tasks requiring bidirectional interdependence, such as language processing., they are, more computationally demanding than ordinary LSTMs. While BiLSTMs are useful, other models, such as transformers, are also gaining popularity owing to their parallel processing and long-range dependency managing capabilities.

F. PROPOSED CNN-LSTM ARCHITECTURE

The approach for classifying Smishing SMS is presented in this section along with an overview of the proposed framework. Figure 5 shows a visual depiction of the framework's architecture. Deep learning models such as CNNs can automatically detect important features from textual inputs. They excel in capturing hierarchical patterns, local associations, and long-term dependencies within the text, thereby facilitating the extraction of meaningful representations. This study introduces an ensemble deep learning model that combines CNN and LSTM layers to enhance Smishing SMS detection.

To improve the classification accuracy for smishing and ham SMS messages, we propose a framework that uses a labelled dataset collected from a publicly available repository. The dataset was first cleaned using a series of pre-processing steps to improve the quality of the text and make it easier to process. The dataset was then divided into an 80:20 ratio for training and testing. The CNN-LSTM architecture is used to combine two neural network types sequentially, as they are both well suited for processing textual data. This ultimately leads to improved classification accuracy.

In the CNN-LSTM model architecture, the CNN layer

extracts features from the input data and the LSTM layer uses these features to make predictions. This creates a streamlined flow of information as the output of the CNN layer is directly passed to the LSTM layer. The architecture, as depicted in Figure 5, was initiated with an embedding layer featuring a vocabulary size of 5000 and an output dimension of 200. This embedding layer is followed by the CNN layer, which processes the input data and generates a meaningful sequence of features for the LSTM layer. To mitigate complexity, a dropout layer with a dropout rate of 0.5, is incorporated immediately after both the CNN and LSTM layers.

The LSTM layer, equipped with 64 units, further processed the information passed from the CNN layer. Subsequently, a dense layer housing 16 neurons was employed to manage the sparse output produced by the LSTM layer. Finally, an output layer is added, accommodating a varying number of neurons based on the target classes for Smishing SMS detection. The CNN-LSTM model was compiled using the categorical cross-entropy function, as this problem involves multiple classes, and training was conducted using the 'Adam optimizer' over 100 epochs.

G. EVALUATION METRICS

Model evaluation involves the assessment of their effectiveness in classifying deep fake text using commonly employed metrics. These metrics include the accuracy, recall, precision, and F1 score. Accuracy provides a general overview of a model's performance by measuring the ratio of correctly classified examples to total occurrences, reflecting its overall correctness.

$$\text{Accuracy} = \frac{\text{Number of correctly classified predictions}}{\text{Total predictions}} \tag{12}$$

while in the case of binary classification, accuracy is measured as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{13}$$

While the terms can be defined as follows:

- True positive (TP) is the number of positive predictions.
- False positive (FP) refers to the number of negative predictions that are incorrectly predicted as positive.
- True negative (TN) is the number of negative predictions.
- False negative (FN) refers to the number of positive and negative predictions.

Precision measures the accuracy of the model in classifying positive cases by evaluating the ratio of correctly classified positive instances to the total anticipated positive instances. This emphasizes the capability of the model to minimize false positives [60]. This emphasizes the capability of the model to minimize false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

Recall, also known as the sensitivity or true positive rate (TPR), measures the ability of a model to correctly identify positive samples. It was calculated as the number of true positives divided by the total number of positive samples. Its significance becomes particularly pronounced in scenarios in which the correct identification of positive cases is of utmost importance, as seen in situations such as disease diagnosis.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

The F1 score is a popular metric for binary classification problems. It uses the harmonic mean to integrate the accuracy and recall into a single number, making it especially beneficial for unbalanced datasets. This aids in balancing the precision-to-recall trade-off, with higher scores suggesting better model performance. The F1 score, which ranges from 0 to 1, is useful for measuring model correctness and completeness in a variety of disciplines, including information retrieval and healthcare.

$$F_1\text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

V. RESULTS AND DISCUSSION

This section details and examines the experiments conducted in this investigation. To detect smishing SMSs, this study employed machine-learning and deep-learning models. The following machine learning models were used to verify the suggested approach: SVC, AC, RF, ETC, GBM, and NB. Table 3 lists these models. The models were trained using the optimal hyperparameters for the dataset. When determining the ideal hyperparameters, the value ranges were fine-tuned to obtain the best performance.

A. PERFORMANCE OF MACHINE LEARNING MODELS FOR SMISHING SMS DETECTION

This section provides an in-depth exploration of the experiments conducted during this research, along with an analysis of the results obtained. We investigated the outcomes of the experiments by employing diverse feature engineering techniques to detect Smishing SMS messages. The performance of different supervised machine learning models was assessed using frequency-based approaches, such as BoW and TF-IDF. These models including the SVC, AC, RF, ETC, GBM, and NB were assessed in terms of accuracy, precision, recall, and F1 scores. The effectiveness of each model varied depending on the chosen feature extraction method.

1) Evaluation of Classifiers using BoW Features

Table 4 compares the performance of various machine learning models. The findings indicate that when utilizing the BoW technique, SVC outperforms the others, achieving an accuracy of 0.9826 and a value of 0.98 for precision, recall, and F1 Score. Similarly, AC and GBM demonstrated comparable results, achieving 0.96 scores of precision, recall, and F1 for Smishing SMS classification. Moreover,

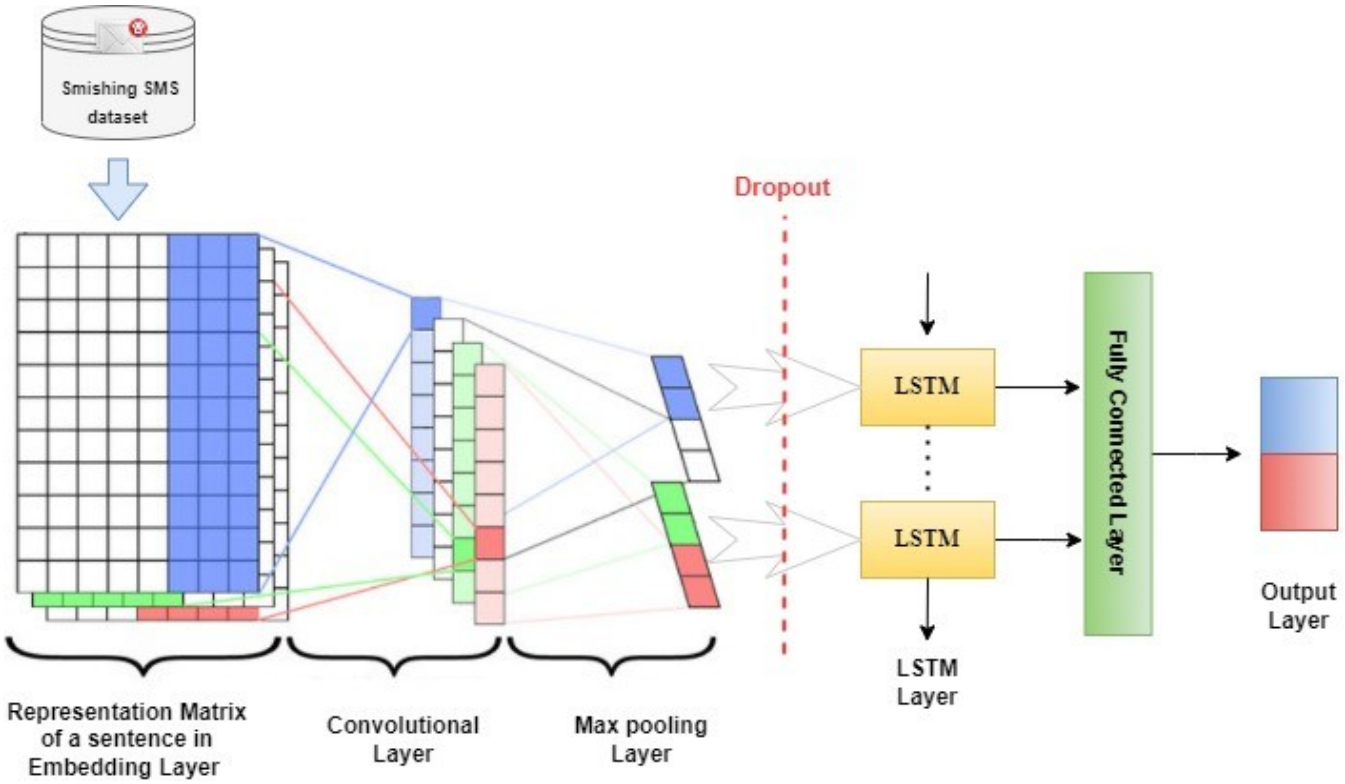


FIGURE 5: Architecture of the proposed CNN-LSTM

RF exhibits remarkable performance with an accuracy of 0.9745, precision of 0.98, recall of 0.97, and F1 score of 0.97.

TABLE 4: Classification results using BoW

Model	Accuracy	Precision	Recall	F1-Score
SVC	0.9826	0.98	0.98	0.98
AC	0.9605	0.96	0.96	0.96
RF	0.9745	0.98	0.97	0.97
ETC	0.9779	0.98	0.98	0.98
GBM	0.9632	0.96	0.96	0.96
NB	0.9739	0.97	0.97	0.97

2) Evaluation of Classifiers using TF-IDF Features

Table 5 shows the results of applying TF-IDF for classifying Smishing SMS using various machine learning models. Interestingly, employing TF-IDF did not appear to have led to any noticeable enhancement in the classifier performance. Among the models tested, SVC utilizing TF-IDF achieved the highest results, boasting an accuracy score of 0.9749, as well as precision, recall, and F1 scores, all at a value of 0.97. RF, ETC, and NB yielded similar outcomes, each attaining a precision, recall, and F1 score of 0.97.

B. RESULTS OF DEEP LEARNING MODELS

In this research, a set of advanced deep learning models, including CNN, LSTM, BiLSTM, GRU, and CNN-LSTM,

TABLE 5: Classification results using TF-IDF

Model	Accuracy	Precision	Recall	F1-Score
SVC	0.9749	0.97	0.97	0.97
AC	0.9565	0.96	0.96	0.96
RF	0.9674	0.97	0.97	0.97
ETC	0.9657	0.97	0.97	0.97
GBM	0.9531	0.95	0.95	0.95
NB	0.9715	0.97	0.97	0.97

were employed. These models have gained significant popularity in text classification applications within the research community. Detailed information regarding the architectural specifications of these deep learning models is provided in Figure 4. Among the various models, including both machine and deep learning approaches, the ensemble recurrent model CNN-LSTM distinguishes itself by delivering exceptional performance. Its outstanding accuracy, which represents the peak achievement, was emphasized. CNN-LSTM demonstrated superior performance across a spectrum of evaluation metrics, including accuracy, precision, recall, and F1 score.

The experimental results revealed that BiLSTM and GRU surpassed the performance of traditional machine learning models, achieving accuracies of 0.9833 and 0.9883, respectively. Both models demonstrated comparable outcomes, with a precision of 0.98, a recall of 0.97, and an F1 score of 0.98. As depicted in Table 6, LSTM displayed superior performance in Smishing SMS detection compared to CNN,

achieving an accuracy score of 0.9858, while CNN yielded an accuracy score of 0.9824. Notably, the combination of CNN and LSTM into an ensemble in Table 6 showcased superior performance with an accuracy score of 0.9974. These findings emphasize the effectiveness of the CNN-LSTM model proposed in this study, leveraging the strengths of both CNN and LSTM through its ensemble structure. This versatile approach positions CNN-LSTM as a significant model, highlighting its superiority over using either model individually.

TABLE 6: Classification results using Deep Learning models

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.9824	0.98	0.97	0.98
LSTM	0.9858	0.96	0.95	0.96
BiLSTM	0.9833	0.98	0.97	0.98
GRU	0.9883	0.98	0.97	0.98
CNN-LSTM	0.9974	0.99	0.99	0.99

C. DISCUSSION

A performance comparison between machine learning models and deep learning models was performed on a large dataset, which was developed by combining two benchmark datasets for Smishing SMS detection. The impact of two feature engineering techniques, BoW and TF-IDF, was also evaluated in combination with machine learning models. Deep learning models require large datasets to provide a good fit, which is possible by combining the two datasets. Figure 6 presents a comparative analysis of the deep learning models based on accuracy, precision, recall, and F1 scores. We base the reduction of false positives by the models on the principle that precision is inversely proportional to the number of false positives. The results demonstrate that the deep learning models outperform their machine learning counterparts individually. However, CNN-LSTM outperforms the other models with a 0.99 score of accuracy, precision, recall, and F1 score. The highest precision score of CNN-LSTM indicates that the proposed model has a reduced number of false positives. In cybersecurity, a false positive occurs when benign messages are incorrectly flagged as malicious. False positives are particularly important in the context of Smishing detection due to the potential negative impacts on user trust, operational efficiency, communication reliability, and overall security posture. For organizations, handling false positives can be resource-intensive, requiring manual review and additional verification steps, which diverts valuable time and resources away from addressing actual threats. In the context of SMS, where messages are often critical communications such as banking alerts, verification codes, and emergency notifications, false positives can disrupt essential services and cause considerable inconvenience or even harm to users. Moreover, a high rate of false positives can lead to complacency among users and administrators, increasing the risk that genuine threats might be overlooked as they become inundated with false alerts. Our approach addresses these concerns by leveraging deep learning techniques to enhance detection accuracy while significantly reducing false positives, thereby providing a more robust and user-friendly solution for cybersecurity.

Figure 8 presents the accuracy curve of the proposed CNN-LSTM model. A smooth curve for both the training and validation accuracy shows that the model learns effectively and improves its performance as it passes through each training epoch. It also shows that the model gradually converges towards a better fit to the data without significant overfitting or underfitting. Figure 9 shows the moss curves of the proposed CNN-LSTM model. A smooth loss curve during training reflects the ability of the model to make accurate predictions.

The confusion matrix for the CNN-LSTM model and the model's performance metrics is presented in Figure 7. This matrix illustrates the following: 9653 messages were correctly identified as Ham (True Negatives, TN), 18 messages were incorrectly identified as Smish (False Positives, FP), 19 messages were incorrectly identified as Ham (False Negatives, FN), and 1855 messages were correctly identified as Smish (True Positives, TP). This detailed confusion matrix supports our claim by providing a clear insight into the model's performance, particularly its effectiveness in reducing false positives, which is crucial for enhancing user trust and operational efficiency in cybersecurity.

The superior performance of the CNN-LSTM ensemble model in identifying smishing SMS stems from its adept utilization of the substantial dataset created by merging two datasets. Its robust performance across diverse evaluation metrics and its capability to discern nuanced patterns establish it as a valuable tool for text categorization and anomaly identification. This study underscores the significance of employing deep learning algorithms and leveraging diverse data resources to effectively address the challenging task of smishing SMS detection. The effectiveness of our model in accurately identifying smishing attacks lays the foundation for its integration into existing cybersecurity systems, mobile security applications, and spam filtering mechanisms. By leveraging advanced deep learning techniques, such as the CNN-LSTM architecture, our model offers a promising solution for combating the growing threat of spam SMS messages. Additionally, the deployment of our model has the potential to enhance user awareness and protection against phishing attempts via SMS, thereby contributing to a safer online environment for mobile users. Future research and development efforts can further explore the practical implementation and scalability of our model in real-world settings, with a focus on usability, interoperability, and user privacy considerations.

This study contributes novel insights to the field of smishing SMS detection through a thorough comparative analysis of machine learning and deep learning models. We highlight the superior performance of the CNN-LSTM model, emphasizing its efficacy in accurately identifying

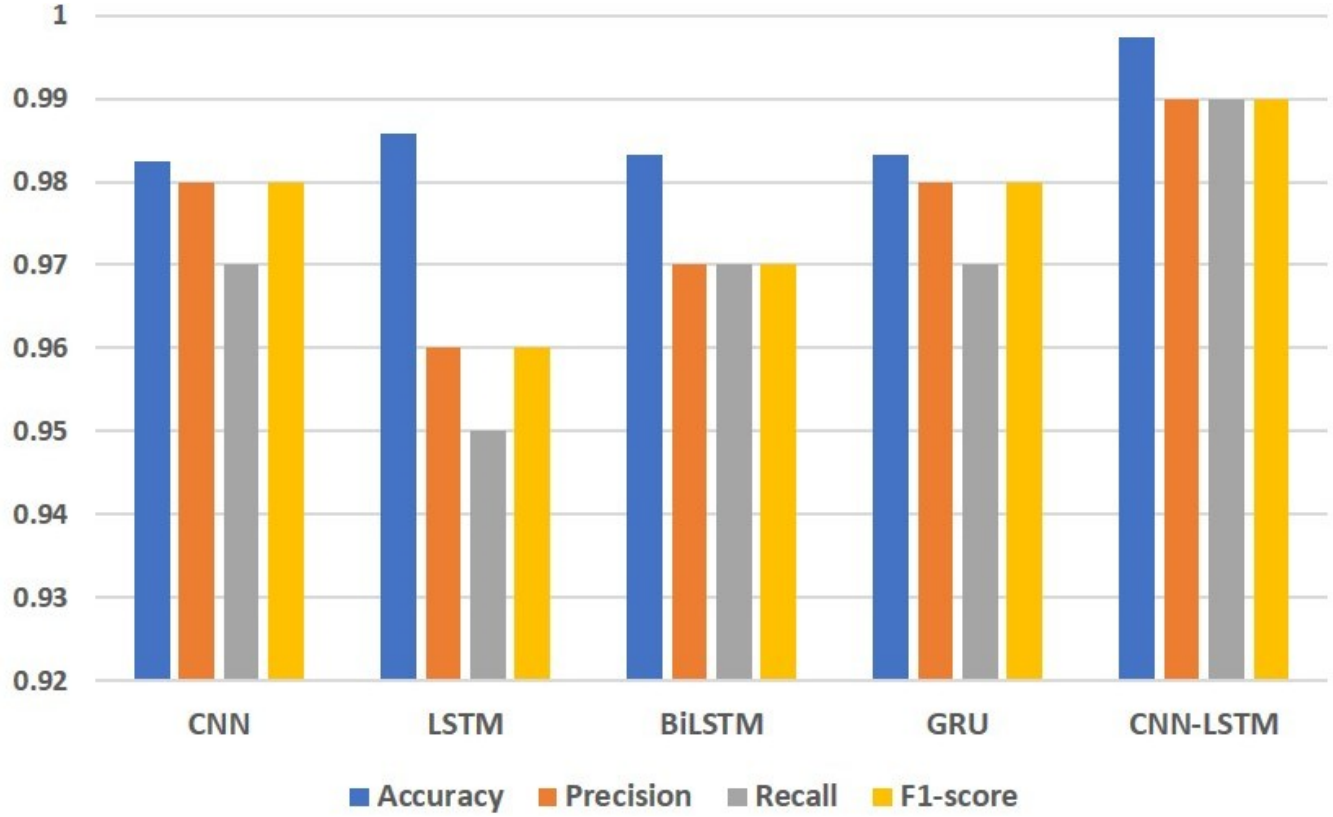


FIGURE 6: Performance Comparison of Deep Learning Models

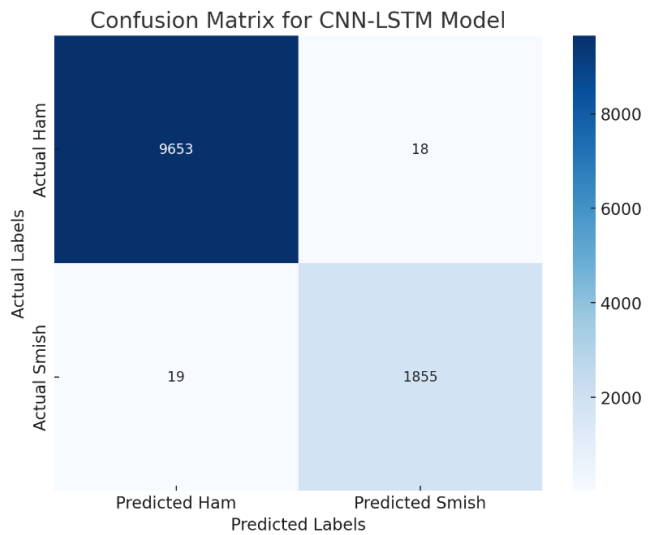


FIGURE 7: Confusion Matrix of the proposed CNN-LSTM

smishing attacks. Our findings underscore the importance of leveraging deep learning algorithms and extensive data resources for effective cybersecurity measures. However, it's imperative to acknowledge the limitations of our study, particularly regarding data availability and methodological constraints. Moving forward, future research should explore avenues for improving dataset quality and enhancing model robustness. Moreover, the implementation of our results and methods holds promise for addressing real-world industry problems related to cybersecurity, offering potential applications in developing advanced threat detection systems and bolstering security measures against smishing attacks.

D. STATISTICAL SIGNIFICANCE OF THE PROPOSED APPROACH

We have conducted a rigorous statistical analysis, specifically a t-test, comparing the performance of our proposed CNN-LSTM model with all other deep learning models presented in the study. We aimed to determine whether the proposed model exhibits statistically significant superiority in smishing detection. The t-test results demonstrate that the proposed CNN-LSTM model outperforms all other models with statistical significance. We reject the null hypothesis and conclude that our CNN-LSTM model is indeed statistically superior to the other models evaluated. In Table 7, we provide a summary showcasing the model names along with their corresponding p-values obtained from the t-test.

E. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS

To evaluate the importance of the CNN-LSTM model presented in this research, a comprehensive comparative performance analysis was performed against other cutting-edge approaches in the realm of Smishing SMS detection.

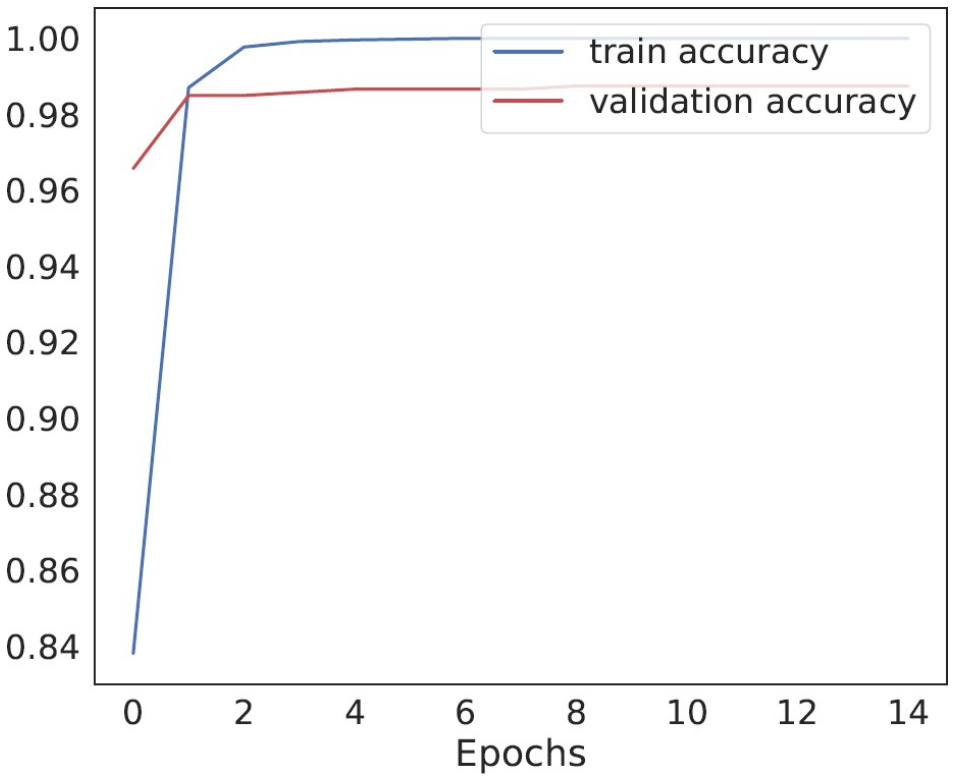


FIGURE 8: Accuracy Curve of the proposed CNN-LSTM Model

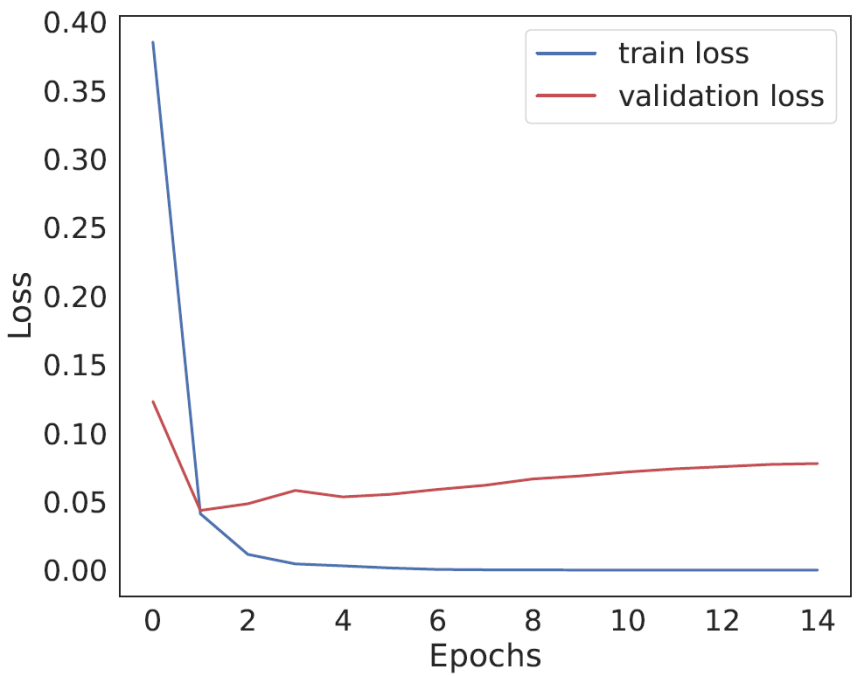


FIGURE 9: Loss Curve of the proposed CNN-LSTM Model

TABLE 7: Statistical result.

Model	P-Value
CNN	0.0001
LSTM	<0.0001
BiLSTM	<0.0001
GRU	<0.0001

In this context, two recent studies have been selected that employed deep learning models for the identification of small SMS.

In this study [21], the authors proposed a hybrid deep learning model for detecting SMS spam messages. The model combines two deep learning methods, CNN and LSTM, and the dataset comprises mixed text messages in both Arabic and English. The achieved accuracy is approximately 0.9837. One study [24] conducted smishing attack detection using an SVM on a dataset comprising 5572 SMS messages and achieved an accuracy score of 0.9839. They leveraged the TF-IDF method for feature extraction. Another study [31] employed a hybrid machine learning model that combined NB, RF, ETC on a dataset containing 5169 SMSs, yielding an accuracy score of 0.969.

Table 8 presents a detailed comparative analysis of the current work with the outcomes of the proposed CNN-LSTM model. The comparison shows that the CNN-LSTM model outperformed the other approaches, achieving a remarkable accuracy score of 99.74%. This suggests that the proposed model exhibits a superior performance, particularly when handling a substantial number of SMS messages. The results highlight the effectiveness of deep learning models, such as CNN-LSTM, in handling intricate and dynamic security concerns. The adaptability and accuracy of models such as CNN-LSTM are essential for staying ahead of new threats as the threat landscape continues to evolve. This study highlights the value of utilizing cutting-edge methods and data sources to protect against smishing attacks in the digital age effectively. Table 9 describes all abbreviations used in the paper.

TABLE 8: Comparative analysis using cutting-edge models from the literature

Reference	Year	Dataset Size	Model	Accuracy
Ref. [21]	2020	8304 SMS	CNN-LSTM Proposed	0.9837 0.9921
Ref. [24]	2022	5572 SMS	SVM Proposed	0.9839 0.9941
Ref. [31]	2023	5169 SMS	Hybrid ML model Proposed	0.969 0.9899
Proposed	**2023**	**11545 SMS**	**CNN-LSTM**	**0.9974**

To further evaluate the effectiveness of our proposed model, we assessed its performance on datasets used in previous research Ref. [21], [24], [31]. Our proposed approach outperforms in detecting phishing attacks from the datasets acquired from the literature.

VI. CONCLUSION AND FUTURE WORK

In conclusion, the escalating threat of phishing, especially through text messages, poses a significant risk to consumers. This paper specifically delves into the emergent danger of smishing attacks on smartphones, where SMS and phishing tactics intertwine to trick users into divulging sensitive information or engaging in compromising activities. Despite prior attempts to detect smishing, the persistent challenge lies in minimizing false positives. To tackle this issue, our study employed deep learning, specifically the CNN-LSTM combination. The proposed method holds substantial real-world implications, contributing to establishing proactive defence mechanisms against smishing attacks and advancements in cybersecurity within the mobile communication sector. Through a comprehensive comparison with conventional machine learning models, this research underscores the benefits of deep learning in smishing detection. The results showcase the CNN-LSTM architecture's remarkable accuracy score of 0.9974 in identifying smishing attacks, underscoring the efficacy of deep learning in bolstering cybersecurity within the mobile communication industry. Furthermore, the model yielded a precision score of 0.99, indicating a reduced number of false positives produced by the proposed model. In terms of recall and F1-score, the model also exhibited improved performance, achieving a score of 0.99 for both metrics. Strengthening defence mechanisms against smishing empowers organizations and individuals to shield themselves against this insidious threat better. Future research avenues may explore the feasibility of real-time detection capabilities and the integration of the system into mobile security applications.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Bajpai and D. Radha, "Smart phone as a controlling device for smart home using speech recognition," in 2019 International Conference on Communication and Signal Processing (ICCSP), pp. 0701–0705, IEEE, 2019.

[2] R. E. Madrid, F. Ashur Ramallo, D. E. Barraza, and R. E. Chaile, "Smartphone-based biosensor devices for healthcare: Technologies, trends, and adoption by end-users," Bioengineering, vol. 9, no. 3, p. 101, 2022.

[3] K. K. Ibrahim and A. J. Obaid, "Fraud usage detection in internet users based on log data," International Journal of Nonlinear Analysis and Applications, vol. 12, no. 2, pp. 2179–2188, 2021.

[4] G. Sonowal and G. Sonowal, "Introduction to phishing," Phishing and Communication Channels: A Guide to Identifying and Mitigating Phishing Attacks, pp. 1–24, 2022.

[5] G. Desolda, L. S. Ferro, A. Marrella, T. Catarci, and M. F. Costabile, "Human factors in phishing attacks: a systematic literature review," ACM Computing Surveys (CSUR), vol. 54, no. 8, pp. 1–35, 2021.

[6] G. Brown and P. M. Greenfield, "Staying connected during stay-at-home: Communication with family and friends and its association with well-being," Human Behavior and Emerging Technologies, vol. 3, no. 1, pp. 147–156, 2021.

[7] S. Mishra and D. Soni, "Smishing detector: A security model to detect smishing through sms content analysis and url behavior analysis," Future Generation Computer Systems, vol. 108, pp. 803–815, 2020.

[8] H. A. Wahsheh and M. S. Al-Zahrani, "Lightweight cryptographic and artificial intelligence models for anti-smishing," in Proceedings of International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2021 Volume 2, pp. 483–496, Springer, 2022.

[9] G. Canova, M. Volkamer, C. Bergmann, R. Borza, B. Reinheimer, S. Stockhardt, and R. Tenberg, "Learn to spot phishing urls with the android nophish app," in Information Security Education Across the Curriculum: 9th IFIP WG 11.8 World Conference, WISE9, Hamburg, Germany, May 26–28, 2015, Proceedings 9, pp. 87–100, Springer, 2015.

[10] A. U. Surwade, "Phishing e-mail is an increasing menace," International Journal of Information Technology, vol. 12, no. 2, pp. 611–617, 2020.

[11] K. Gradoń, "Crime in the time of the plague: Fake news pandemic and the challenges to law-enforcement and intelligence community," Society Register, vol. 4, no. 2, pp. 133–148, 2020.

[12] S. J. Delany, M. Buckley, and D. Greene, "Sms spam filtering: Methods and data," Expert Systems with Applications, vol. 39, no. 10, pp. 9899–9908, 2012.

[13] A. Ghourabi, "Sm-detector: A security model based on bert to detect smishing messages in mobile environments," Concurrency and Computation: Practice and Experience, vol. 33, no. 24, p. e6452, 2021.

[14] A. K. Jain, D. Goel, S. Agarwal, Y. Singh, and G. Bajaj, "Predicting spam messages using back propagation neural network," Wireless Personal Communications, vol. 110, pp. 403–422, 2020.

[15] O. Sova, O. Turinskyi, A. Shyshatskyi, V. Dudnyk, R. Zhyvotovskyi, Y. Prokopenko, T. Hurskyi, V. Hordiichuk, A. Nikitenko, and A. Remez, "Development of an algorithm to train artificial neural networks for intelligent decision support systems," Eastern-European Journal of Enterprise Technologies, vol. 1, no. 9, p. 103, 2020.

[16] D. Goel and A. K. Jain, "Smishing-classifier: a novel framework for detection of smishing attack in mobile environment," in Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3, pp. 502–512, Springer, 2018.

[17] G. Sonowal and K. Kuppusamy, "Smidca: an anti-smishing model with machine learning approach," The Computer Journal, vol. 61, no. 8, pp. 1143–1157, 2018.

[18] J. W. Joo, S. Y. Moon, S. Singh, and J. H. Park, "S-detector: an enhanced security model for detecting smishing attack for mobile computing," Telecommunication Systems, vol. 66, pp. 29–38, 2017.

[19] A. K. Jain and B. Gupta, "Rule-based framework for detection of smishing messages in mobile environment," Procedia Computer Science, vol. 125, pp. 617–623, 2018.

[20] S. Mishra and D. Soni, "A content-based approach for detecting smishing in mobile environment," in Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India, 2019.

[21] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid cnn-lstm model for sms spam detection in arabic and english messages," Future Internet, vol. 12, no. 9, p. 156, 2020.

[22] K. M. Kotsifakou, D. Kotsifakos, and C. Douligeris, "Neural network for spam recognition in short message services as an instructional application for students of vocational education and training," in 2022 IEEE Global Engineering Education Conference (EDUCON), pp. 1405–1412, IEEE, 2022.

[23] J. Kamau, D. Kaburu, et al., "A review of smishing attaks mitigation strategies," International Journal of Computer and Information Technology (2279-0764), vol. 11, no. 1, 2022.

[24] R. E. Ulfath, I. H. Sarker, M. J. M. Chowdhury, and M. Hammoudeh, "Detecting smishing attacks using feature extraction and classification techniques," in Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021, pp. 677–689, Springer, 2022.

[25] R. Saeki, L. Kitayama, J. Koga, M. Shimizu, and K. Oida, "Smishing strategy dynamics and evolving botnet activities in japan," IEEE Access, vol. 10, pp. 114869–114884, 2022.

[26] A. Sharaff, V. Pathak, and S. S. Paul, "Deep learning-based smishing message identification using regular expression feature generation," Expert Systems, vol. 40, no. 4, p. e13153, 2023.

[27] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: a systematic literature review," Knowledge and Information Systems, vol. 64, no. 6, pp. 1457–1500, 2022.

[28] M. M. Alani and H. Tawfik, "Phishnot: a cloud-based machine-learning approach to phishing url detection," Computer Networks, vol. 218, p. 109407, 2022.

[29] S. Raminedi, T. N. Pandey, V. A. Woonna, S. C. Mascarenhas, and A. Bharani, "Classification of phishing websites using machine learning models," in 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP), pp. 1–5, IEEE, 2023.

[30] Y. Huang, J. Qin, and W. Wen, "Phishing url detection via capsule-based neural network," in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 22–26, IEEE, 2019.

[31] N. Agrawal, A. Bajpai, K. Dubey, and B. Patro, "An effective approach to classify fraud sms using hybrid machine learning models," in 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), pp. 1–6, IEEE, 2023.

[32] S. Mishra and D. Soni, "Implementation of 'smishing detector': an efficient model for smishing detection using neural network," SN Computer Science, vol. 3, no. 3, p. 189, 2022.

[33] A. K. Jain and B. B. Gupta, "Feature based approach for detection of smishing messages in the mobile environment," Journal of Information Technology Research (JITR), vol. 12, no. 2, pp. 17–35, 2019.

[34] 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.

[35] R. Alabdan, "Phishing attacks survey: Types, vectors, and technical approaches," Future internet, vol. 12, no. 10, p. 168, 2020.

[36] A. K. Jain and B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," Enterprise Information Systems, vol. 16, no. 4, pp. 527–565, 2022.

[37] G. Sonowal and K. Kuppusamy, "Phidma–a phishing detection model with multi-filter approach," Journal of King Saud University-Computer and Information Sciences, vol. 32, no. 1, pp. 99–112, 2020.

[38] N. Noah, A. Tayachew, S. Ryan, and S. Das, "Phishercop: Developing an nlp-based automated tool for phishing detection," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 66, pp. 2093–2097, SAGE Publications Sage CA: Los Angeles, CA, 2022.

[39] F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, "An automated framework for real-time phishing url detection," in 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0335–0341, IEEE, 2020.

[40] O. D. Catherine, "An intelligent rule based phishing website detection model," Iiardpub. Org, vol. 5, no. 2, pp. 10–19, 2019.

[41] A. Alswailem, B. Alabdullah, N. Alrumayh, and A. Alsedrani, "Detecting phishing websites using machine learning," in 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–6, IEEE, 2019.

[42] R. S. Rao, A. R. Pais, and P. Anand, "A heuristic technique to detect phishing websites using twsvm classifier," Neural Computing and Applications, vol. 33, pp. 5733–5752, 2021.

[43] M. Habib, J. O'Sullivan, S. Abolfathi, and M. Salauddin, "Enhanced wave overtopping simulation at vertical breakwaters using machine learning algorithms," Plos one, vol. 18, no. 8, p. e0289318, 2023.

[44] K. Khosravi, F. Rezaie, J. R. Cooper, Z. Kalantari, S. Abolfathi, and J. Hatamiafkoueieh, "Soil water erosion susceptibility assessment using deep learning algorithms," Journal of Hydrology, vol. 618, p. 129229, 2023.

[45] L. Breiman, "Random forests," Machine learning, vol. 45, pp. 5–32, 2001.

[46] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, and X. Liu, "Ensemble adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 10, p. e4088, 2022.

[47] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Frontiers in neurorobotics, vol. 7, p. 21, 2013.

[48] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve bayes.," Encyclopedia of machine learning, vol. 15, pp. 713–714, 2010.

[49] W. S. Noble, "What is a support vector machine?," Nature biotechnology, vol. 24, no. 12, pp. 1565–1567, 2006.

[50] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine learning, vol. 63, pp. 3–42, 2006.

[51] Y. Liu, H. Pu, and D.-W. Sun, "Efficient extraction of deep image features using convolutional neural network (cnn) for applications in detecting and analysing complex food matrices," Trends in Food Science & Technology, vol. 113, pp. 193–204, 2021.

[52] R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, and K. Shankar, "Optimal feature selection-based medical image classification using deep learning model in internet of medical things," IEEE Access, vol. 8, pp. 58006–58017, 2020.

[53] J. Donnelly, A. Daneshkhah, and S. Abolfathi, "Physics-informed neural networks as surrogate models of hydrodynamic simulators," Science of the Total Environment, vol. 912, p. 168814, 2024.

[54] J. Donnelly, A. Daneshkhah, and S. Abolfathi, "Forecasting global climate drivers using gaussian processes and convolutional autoencoders," Engineering Applications of Artificial Intelligence, vol. 128, p. 107536, 2024.

[55] B. Ghiasi, R. Noori, H. Sheikhian, A. Zeynolabedin, Y. Sun, C. Jun, M. Hamouda, S. M. Bateni, and S. Abolfathi, "Uncertainty quantification of granular computing-neural network model for prediction of pollutant longitudinal dispersion coefficient in aquatic streams," Scientific reports, vol. 12, no. 1, p. 4610, 2022.

[56] G. Murthy, S. R. Allu, B. Andhavarapu, M. Bagadi, and M. Belusonti, "Text based sentiment analysis using lstm," Int. J. Eng. Res. Tech. Res, vol. 9, no. 05, 2020.

[57] M. Mahdian, R. Noori, M. M. Salamattalab, E. Heggy, S. M. Bateni, A. Nohegar, M. Hosseinzadeh, S. M. Siadatmousavi, M. R. Fadaei, and S. Abolfathi, "Anzali wetland crisis: Unraveling the decline of iran's ecological gem," Journal of Geophysical Research: Atmospheres, vol. 129, no. 4, p. e2023JD039538, 2024.

[58] S. Yu, D. Liu, W. Zhu, Y. Zhang, and S. Zhao, "Attention-based lstm, gru and cnn for short text classification," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 1, pp. 333–340, 2020.

[59] J. Deng, L. Cheng, and Z. Wang, "Attention-based bilstm fused cnn with gating mechanism model for chinese long text classification," Computer Speech & Language, vol. 68, p. 101182, 2021.

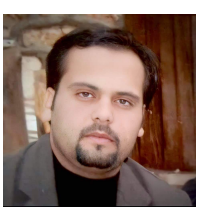
[60] V. Lenarduzzi, F. Pecorelli, N. Saarimaki, S. Lujan, and F. Palomba, "A critical comparison on six static analysis tools: Detection, agreement, and precision," Journal of Systems and Software, vol. 198, p. 111575, 2023.

TABLE 9: Abbreviations and Definitions

Abbreviation	Description
AC	AdaBoost Classifier
APK	Android Application Package
APWG	Anti-Phishing Working Group
BiLSTM	Bidirectional Long Short-Term Memory
BoW	Bag-of-words
CNN	Convolutional Neural Network
COVID-19	coronavirus disease of 2019
ETC	Extra Tree Classifier
FN	False negative
FP	False positive
GBM	Gradient boosting machine
GRU	Gated Recurrent Unit
HMTL	HyperText Markup Language
IP	Internet Protocol
LSTM	Long Short-Term Memory
ML	Machine Learning
NB	Naive Bayes
NLP	Natural language processing
NLTK	Natural language toolkit
PRISM	a rule-based algorithm
RF	Random Forest
RNN	Recurrent neural networks
SGD	Stochastic gradient descent
SMS	Short Message Service
SmiDCA	SMIshing Detection based on Correlation Algorithm
SVC	Support vector machine classifier
TF_IDF	Term Frequency-Inverse Document frequency
TN	True negative
TP	True positive
TPR	True positive rate
TWSVM	Twin support vector machine
UN	United Nations
URL	Uniform resource locator

Muhammad Khalid Mehmood is a research associate in the Department of Computer Science at The Islamia University of Bahawalpur, Pakistan. He holds a Master's and is currently pursuing a Ph.D. in Computer Science and Security. With expertise in computer security, he has actively contributed to various projects, specializing in requirements analysis, design, and solution development. His experience extends to Future Internet and Enterprise Systems, as well as Network Security Technologies, showcasing proficiency in data science.

Dr. Humaira Arshad is an experienced Associate Professor with a strong background in research. Her expertise lies in Computer Science, Digital Forensics, Social Network Forensics, Semantic Web Data Analysis, Ontology, Formal Knowledge Models, and Forensic Process Models. Holding a Doctor of Philosophy (Ph.D.) in Computer Science from the University Sains Malaysia, she currently serves as the Head of the Department of Computer Science at The Islamia University of Bahawalpur, Pakistan. Dr. Humaira Arshad has a notable publication record, with over 25 papers published in scientific journals and international conference proceedings.



MOATSUM ALAWIDA (MEMBER, IEEE) is an Assistant Professor of Cybersecurity at Abu Dhabi University. He earned his Ph.D. in Computer Science with a specialization in cybersecurity (cryptography) from the School of Computer Sciences at Universiti Sains Malaysia in 2020. With a prolific academic career, he has authored and published over 50 articles in high-impact factor journals. Notably, in the years 2021, 2022, and 2023, he achieved recognition among the top 2% of scientists worldwide.

His diverse research interests encompass chaotic systems, applications of chaos, multimedia security, drone security blockchain, cybersecurity, quantum-based cryptography, and traditional cryptography methods. Beyond his research contributions, Moatsum Alawida has actively served as a referee for esteemed journals, including IEEE, Elsevier, Springer, MDPI, and various conferences.

ABID MEHMOOD (MEMBER, IEEE) received a Ph.D. degree in computer science from Deakin University, Australia. He is currently an Assistant Professor at Abu Dhabi University. He has multiple research publications in well-reputed journals. His research interests include machine learning, privacy, information security, data mining, and cloud computing.