A Deep Learning for Arabic SMS Phishing Based on URLs Detection

Sadeem Alsufyani, Samah Alajmani
Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia.

Abstract—The increasing use of SMS phishing messages in Arab communities has created a major security threat, as attackers exploit these SMS services to steal users' sensitive and financial data. This threat highlights the necessity of designing models to detect SMS messages and distinguish between phishing and non-phishing messages. Given the lack of sufficient previous studies addressing Arabic SMS phishing detection, this paper proposes a model that leverages deep learning models to detect Arabic SMS messages based on the URLs they contain. The focus is on the URL aspect because it is one of the common indicators in phishing attempts. The proposed model was applied to two datasets that were in English, and one dataset was in Arabic. Two datasets were translated from English to Arabic. Three datasets included a number of Arabic SMS messages, mostly containing URLs. Three deep learning models—CNN, BiGRU, and GRU—were implemented and compared. Each model was evaluated using metrics such as precision, recall, accuracy, and F1 score. The results showed that the GRU model achieved the highest accuracy of 95.3% compared to other models, indicating its ability to capture sequential patterns in URLs extracted from Arabic SMS messages effectively. This paper contributes to designing a phishing detection model designed for Arab communities to enhance information security within Arab communities.

Keywords—Phishing; URL phishing; SMS phishing; GRU; BiGRU; CNN

I. Introduction

Cybersecurity refers to one or more of the following three things as a set of security measures and other activities aimed at first protecting computer hardware and networks, related hardware and software, as well as the information they contain and transmit, including software and data. This protection includes protection against attacks, disruptions, or threats. Second, the status or quality of protection from threats. Third, expand the scope for public discussions aimed at the process of implementing and improving those activities and quality [1]. Therefore, a cyberattacks is an attempt by attackers to infiltrate information systems at the level of an individual or organization in a deliberate way. A cyberattack aims to disrupt the resources of the target victim's system by stealing his confidential information and disrupting the main functions of his system. After that, network assaults come in several types. The attackers search for the type of ransom after carrying out network attacks on organizations. This threat is not limited only to large companies but also includes medium and small organizations. The reason lies in the fact that medium and small organizations do not have high-level security measures, and this makes attackers also focus on medium and small companies and find out their vulnerabilities [2]. Cyberattacks have become widespread in our daily lives, affecting government institutions, from the economic side to trade, as well as banks and hospitals. Malware, phishing, social engineering attacks, botnets, password attacks, man-in-the-middle attacks, and other types of cyberattacks are among the most prevalent types [3]. Therefore, preserving private data against social engineering threats such as phishing attempts is the primary objective of information security. To protect private data from these types of social engineering assaults, consumers, website developers, and experts have been particularly concerned about security vulnerabilities in every company [4]. Social engineering refers to the tactic of manipulating individuals to obtain unauthorized access to data. This method falls under the broader category of information security. People are the weakest point, the focal point of most organizations because they pose a threat to their organizations. If sensitive information about the organization is compromised, it falls into the wrong hands. Organizations usually use advanced security measures to minimize the chances of unauthorized individuals accessing that information. An organization needs to prevent its employees from succumbing to social engineering attacks. Most humans react emotionally, so they are more vulnerable than machines most often. The greatest threat that an organization poses to having sensitive information is human, not technical protection because they constitute the essence of an organization. As a result, attackers have concluded that using a human to get unauthorized access to an organization's data and communication technology infrastructure is more straightforward than attempting to breach security mechanisms [5]. Phishing attacks are prevalent in social engineering attacks. The attacker entices users to send fake messages such as winning a prize, sending a message from a fake social media account, or hacking passwords. These messages seem to be an order from a trusted entity, such as a bank disclosing information to achieve financial gain. Social engineering techniques with some fraudulent tactics are ingeniously used to entice users to acquire information. Fraudulent methods can connect to a message, phone, or fake email. Scammers send fake messages to many internet users. These attacks target people who lack sufficient knowledge about cyberattacks and their security. They are led to assume that the messages are from a legitimate organization. The core goal of phishing attacks is to search for the vulnerabilities of the intended user. The attacker always finds ways to make the targeted victims visit a phishing site. By designing fake messages in a way that makes them appear reliable, including a link that transports them to this fraudulent site, it is easy to target and deceive victims [4]. Phishing includes several types of attacks targeting users, such as voice phishing, email phishing, SMS phishing, website phishing, and social media phishing [6]. SMS phishing, also referred to as smishing, is a kind of phishing that utilizes short message service (SMS) technology. This type of phishing exploits SMS messages on smartphones. The smishing happens in two major methods. The first is when an SMS message is sent from a reliable source, such as a bank or system administrator. The second way occurs when the victim receives an SMS message containing private content, such as an account block or stolen identity. Subsequently, the victim will be sent to a deceptive website or contacted via phone number to confirm their data [7].

Therefore, with the increasing prevalence of phishing attacks received through SMS, research targeting Arabic-speaking communities to detect Arabic SMS messages containing URLs remains insufficient. This gap poses a significant security threat to Arabic-speaking users and can lead to the loss of sensitive personal and financial data. While existing phishing detection solutions have targeted SMS messages in English, Arabic SMS messages based on URLs are relatively unexplored. The significance of this paper lies in its contribution to addressing the gap in phishing detection models specifically designed for Arabic SMS messages containing URLs. By leveraging deep learning models, this paper aims to mitigate the risk of phishing threats faced by Arabic-speaking users. Thus, the key paper challenges addressed in this work are: (1) how can URLs extracted from Arabic SMS messages be analyzed to determine whether the message is phishing or non-phishing? (2) what is the appropriate deep learning model for effective classification and detection of Arabic SMS phishing based on URLs? Accordingly, the paper proposes a proposed model to bridge this gap through several contributions. First, provide a model for detecting Arabic SMS messages based on URLs and determine the type of message, whether phishing or non-phishing, depending on the analysis of the URL contained in the Arabic SMS message. Second, we created a dataset of 16,521 Arabic SMS messages, which helps provide a dataset for future research in the field of Arabic SMS detection, then extracted and analyzed URLs from Arabic SMS using deep learning models: CNN, BiGRU, and GRU. Finally, evaluate and compare the performance of models for deep learning in the classification of Arabic SMS messages based on URLs, whether phishing or non-phishing. The results of this paper can benefit financial institutions and telecommunications service providers by providing an effective tool to protect sensitive data and reduce financial losses caused by SMS-based phishing threats.

The rest of this paper is organized as follows: Section II provides a review of related works. Section III presents the methodology. Section IV presents the results and discussions. Finally, Section V presents the conclusion and future work of the paper.

II. Related works

In this section, we review several related works on SMS phishing detection with deep learning and machine learning methods.

Mishra & Soni, 2023 [8] presented a two-stage SMS phishing detection model. The first stage was URL validation domain checking. The second stage was SMS classification. The URL domain was checked, and SMS classification categorized the messages' text content and extracted some useful features. Finally, the system used a backpropagation algorithm to classify the messages and was evaluated using the SMS dataset. The results showed an accuracy rate of 97.93%.

Mishra & Soni, 2020 [9] suggested the Smishing Detector model, which detected SMS phishing messages with minimal false positives. The model analyzed content through a Naïve Bayes classification. Four modules provided this model: APK Download Detector, SMS Content Analyzer, URL Filter, and Source Code Analyzer. The results demonstrated an overall accuracy rate of 96.29%.

Agrawal et al., 2023 [10] developed a model to detect fraudulent SMS. The model's design involved two phases: the first phase used a hybrid model for SMS message classification, whereas the examination of URLs was the second phase. Random Forest, Naive Bayes, and Extra Tree classifiers were used in their hybrid model. The results showed that the Random Forest, Multinomial Bayes classifier, and Extra Tree classifier achieved 96.25% accuracy and 99.38% precision.

Prasanna Bharathi et al., 2021 [11] applied two well-known algorithms to categorize spam SMS: a Support Vector Machine and Naive Bayes. 96.19% accuracy percent was achieved by the Naive Bayes algorithm. 98.77% accuracy was achieved with the Support Vector Machine algorithm approach.

Wu et al., 2018 [12] introduced a novel approach to detecting SMS phishing utilizing oversampling technology to enhance feature selection and improve accuracy. They utilized three types of features, namely symbol features, subject features, language query features, and word calculation (LIWC). They applied one of the oversampling methods called the Adasyn adaptive synthetic sampling approach. The BPSO binary particle swarm was used to analyze the three feature types and then select the optimal combination of all the features. The experiment was performed on the Almeida et al. dataset, which contained 5574 messages in English. They used the Random Forest classification algorithm to obtain detection findings. The findings showed that the two methods offered by ADASYN and BPSO achieved the highest accuracy rate of 99.01%.

Oswald et al., 2022 [13] proposed an intent-based approach that efficiently handled the filtering of SMS spam, textual and semantic features of SMS messages were created using 13 pre-defined intent labels. Multiple pre-trained NLP models were applied to generate textual contextual embeddings. For the pre-defined labels, intent scores were computed. Several supervised learning classifiers were used to filter spam or ham. The results showed that the DistilBERT+SVM (Poly) model performed well with an accuracy (98.07%), precision, and recall (∼0.97).

Tuan et al., 2023 [14] evaluated five algorithms on three various Vietnamese datasets: Support Vector Machine, Random Forests, Naïve Bayes, Convolutional Neural, and Long Short-Term Memory to evaluate the efficiency of spam detection in Vietnamese SMS. The results showed that the CNN and LSTM, supported by the transformer PhoBert model, were more effective than the conventional models for machine learning. The LSTM model obtained the greatest accuracy of 97.77%, on the Vietnamese full-dialect dataset, while the CNN and PhoBert models showed a high accuracy of 95.56% on the non-diacritic Vietnamese dataset.

The University of Baghdad et al., 2021 [15] suggested a new approach for detecting SMS spam that focused on improving the binary particle swarm based on fuzzy rule selection. Initially, the significant features of the SMS spam dataset were extracted. Then, a fuzzy collection of rules was produced using the features that were extracted. The most reliable fuzzy rules, which lowered complexity and enhanced model performance, were finally chosen using a binary particle swarm. The findings demonstrated that the suggested model achieved an F-measure of 94.6%, recall of 98.8%, accuracy of 98.5%, and precision of 90.8%.

Amir Sjarif et al., 2020 [16] proposed a method for classifying spam SMS messages through a variety of techniques for data mining. Algorithms such as Multinominal Naïve Bayes, Support Vector Machine, Naïve Bayes, and K Nearest Neighbor with different values of K = 1, 3, and 5 were trained and assessed using the dataset from the UCI machine learning repository. Each algorithm's performance was compared to determine which best-fitting classifier performed better in terms of accuracy, error, processing time, kappa statistics, and the lowest number of false positives. The SVM algorithm outperformed the other classifiers in terms of accuracy, with an average accuracy of 98.9% for detecting and labeling spam text messages. In terms of the error coefficient, the KNN algorithm had the highest error with K = 3 and K = 5, while SVM had the lowest error, followed by the Multinominal Naïve Bayes algorithm.

Uddin et al., 2024 [17] addressed spam detection using a transformer-based Large Language Models (LLMs) approach that was refined and optimized. The benchmark SMS spam dataset was used to detect spam messages. The imbalance problem in the data was mitigated by implementing methods for data augmentation, such as back translation. In addition, calculated the scores of positive and negative coefficients that detected and explained the transparency of the fine-tuned model in detecting spam messages using explainable artificial intelligence (XAI) techniques. Traditional models for machine learning and transformer-based models' performances were compared. The experiments showed that the refined and optimized BERT model with the variant model RoBERTa obtained the highest accuracy of 99.84%.

Ali et al., 2023 [18] proposed a new model for detecting SMS spam using (MLP) Multiple Linear Regression to extract seven features from each message. The message detection process was entrusted to the feature weight and Extreme Learning Machine (ELM). MLR was used to weigh the seven extracted features. The SMS was classified as spam or ham by ELM. The suggested model was evaluated for recall, F-measure, precision, and accuracy, and showed scores of 98.7%, 95.9%, 93.3%, and 98.2%, respectively.

Sonowal, 2020 [19] identified the greatest collection of features for the detection of SMS phishing by employing four ranking algorithms: Spearman's rank correlation, Pearson rank correlation, Kendall rank correlation, and Point biserial rank correlation, along with machine learning algorithms. According to the findings, the AdaBoost classifier provided the highest accuracy. When compared to other correlation algorithms, the Kendall rank correlation algorithm provided the best accuracy. Therefore, this finding proved that the ranking algorithm could provide 98.40% accuracy and 61.53% reduction in feature dimensions.

Giri et al., 2023 [20] suggested various deep neural networks for spam SMS classification. The Tiago dataset was used, and some steps were taken to start with preprocessing and then feeding these preprocessed messages into two different models of deep learning (Long Short-Term Memory Network with Convolution Neural Network) with simple architectures. Word embedding techniques (BUNOW and GloVe) were combined to enhance the two basic architectures' accuracy with the deep learning models. The results after using the two-word embedding techniques in text categorization, demonstrated an accuracy of 98.44% with the CNN LSTM BUNOW model.

Table I summarizes the previous studies that contributed to SMS phishing message detection solutions.

TABLE I. COMPARISON OF PREVIOUS STUDIES ON SMS PHISHING USING MACHINE LEARNING AND DEEP LEARNING DETECTION

Ref.	Model architecture used	Dataset language	Result %
[8]	Backpropagation algorithm	English	97.93%
[9]	Naive Bayes	English	96.29%.
[10]	Random Forest, Naive Bayes, and Extra tree classifiers	English	96.25%
[11]	Support vector machine, and Naive Bayes.	English	98.77%
[12]	ADASYN, BPSO	English	99.01%
[13]	DistilBERT+SVM (Poly)	English	98.07%
[14]	Support Vector Machine, Random Forests, Naive Bayes, LSTM, and CNN	Vietnamese	97.77%, on the Vietnamese full-dialect and 95.56%on the non-diacritic Vietnamese
[15]	Binary particle swarm based on fuzzy rule selection.	English	98.5%
[16]	Support Vector Machine, Multinominal Naïve Bayes, Naïve Bayes, and K Nearest Neighbor with different values	English	98.9%
[17]	explainable artificial intelligence (XAI), transformer-based Large Language Models, refined and optimized BERT model with the variant model RoBERTa	English	99.84%
[18]	Multiple linear regression, extreme learning machine ELM.	English	98.2%
[19]	Spearman's rank correlation, Pearson rank correlation, Kendall rank correlation, and Point biserial rank correlation	English	98.40%
[20]	CNN LSTM BUNOW	English	98.44%

III. METHODOLOGY

The proposed methodology is based on the process of detecting Arabic SMS phishing messages based on URLs, which uses models for deep learning such as GRU, CNN, and BiGRU to examine and categorize these Arabic SMS messages. The process begins with the step of identifying SMS messages that contain URLs, which are often indicative of phishing tries.

Once identified, the Arabic SMS messages are classified based on the presence of URLs for further analysis. This analysis step is fundamental for understanding the type of content of Arabic SMS messages, with a focus on the URLs embedded within the Arabic SMS messages. The methodology evaluates patterns and characteristics that are usually related to phishing, such as suspicious domains or malicious URLs. The cleaned dataset experiences an inspecting process, where each Arabic SMS message is evaluated for the presence of a URL. After that, the URLs are extracted from the Arabic SMS messages and passed to the URL-based classification. The classification step utilizes models for deep learning, such as GRU, CNN, and BiGRU, to process and analyze the extracted URL dataset. These models were selected for their capability to capture sequential patterns, spatial features, and contextual dependencies, which are important in the process of detecting phishing tries.

Fig. 1 illustrates the overall methodology proposed in this paper, illustrating the steps in classifying Arabic SMS messages based on URLs based on the proposed models for deep learning.

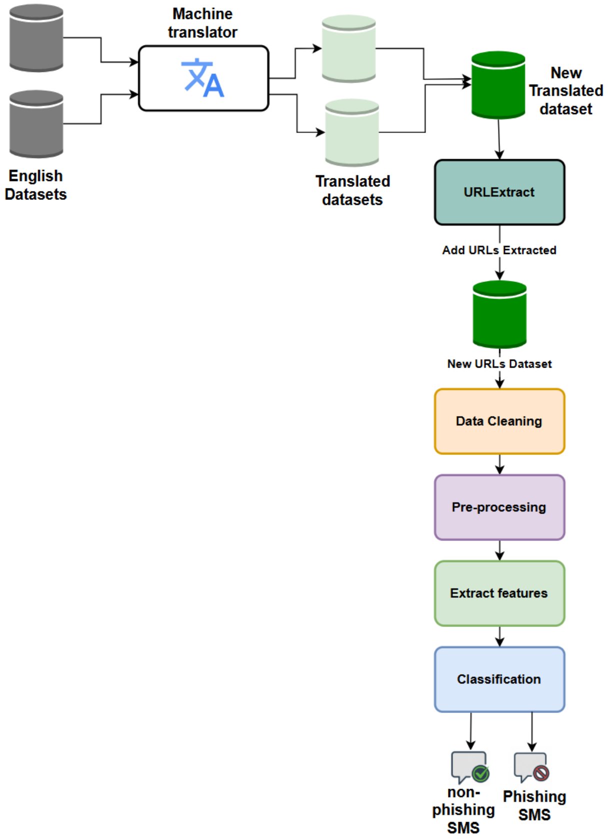


Fig. 1. The process of proposed Arabic SMS messages based on URLs detection.

The steps are explained as follows:

A. Step 1: SMS Messages Dataset and Translation

The datasets used in this process were obtained from three different sources: the first dataset from [21], the second dataset from the UCI Repository [22] and the third dataset from Kaggle [23]. The two sources contain [22] and [23] English datasets, which necessitate the use of machine translation, such as Google Translate, to convert SMS messages from English to Arabic. This translation is necessary for the process of checking whether the SMS messages contain URLs or not, to classify Arabic SMS messages based on the type of the URL, whether it is phishing or non-phishing. SMS messages can be automatically translated using machine translation, a technology built on artificial intelligence systems.

Google Translate was utilized in our proposed model to translate two English datasets into Arabic. The file upload feature in Google Translate allowed us to translate the content of both datasets completely and comprehensively and convert them into Arabic to support the proposed model. After the translation process was successful, we downloaded the translated dataset file. This step is essential to support our proposed model to ensure the availability of an Arabic SMS dataset. We translated the two datasets [22] and [23] that were originally in English and then translated into Arabic using Google Translate.

B. Step 2: Merge the Dataset

This step creates a dataset containing a large number of SMS messages translated into Arabic, some of which include URLs in their content. The merging process was according to some steps:

1) First step: Column data type is identical. This rule indicates that the first column represents the label of each Arabic SMS message, whether it is non-phishing or phishing. The second column contains the text of the Arabic SMS message, which is of type text.

2) Second step: The number of columns is identical. This rule indicates that all the datasets have the same number of columns. Our dataset contains only two columns: The first column represents the label of the Arabic SMS message, categorizing it as either non-phishing or phishing. The second column represents the text of the Arabic SMS message.

After the merging process, we reached three datasets comprising a total of 16,521 Arabic SMS messages. Most of these messages include URLs, which will achieve our goal of detecting Arabic SMS phishing messages based on the URL they contain.

C. Step 3: URLs Dataset

We collected a dataset of URLs to support expanding the URL dataset to train and accurately classify deep learning models. The auxiliary dataset was collected from [24], which includes 20,000 URLs, categorized as either non-phishing or phishing. The dataset consists of two columns: the first column represents lists of the URLs, and the second column represents the type of URLs, whether non-phishing or phishing.

D. Step 4: URL Extraction and Merging Dataset

The dataset is processed by the URL classification component, starting with the extraction of URLs from the SMS messages using the URLExtract library, a Python library that extracts URLs from Arabic SMS messages. All URLs extracted from Arabic SMS messages are saved and merged with a new dataset [24] containing a large number of URLs. Merging these URLs with the new URL dataset enhances the expansion of the

URL dataset, which improves the accuracy of the training and classification using deep learning models.

E. Step 5: Data Cleaning

After completing the previous step of preparing the URL dataset, the next step is cleaning the dataset. Any unnecessary elements are removed. The cleaning process includes the following:

1) Removing duplicate URLs: This refers to eliminating duplicate entries where the same URL or the same rows are repeated.

2) Removing null values: This refers to removing cells that contain null or missing values, i.e., the URL is not provided or the label is missing.

F. Step 6: Pre-Processing Process

Preprocessing is an important step in improving data quality. This directly contributes to enhancing the accuracy and capability of the model. By thoroughly preparing the dataset, we ensure that it enhances feature extraction, helping the model provide more reliable and precise classification results.

One of the fundamental preprocessing tasks includes dealing with the URLs within the dataset. This includes various tasks that contribute to normalizing and standardizing URLs to remove inconsistencies and duplication. Key steps include:

1) Converting characters to lowercase: All characters in URLs are changed to lowercase. This helps to avoid handling the same URL with different letter cases as individual entities.

For example: ForExample.com

After the conversion process, it forexample.com

2) Removal of numbers: Any numeric values within URLs that are not appropriate to the classification task are removed to streamline the dataset and reduce noise.

3) Removal of extra spaces: Unnecessary spaces within URLs are removed to ensure consistency in data formatting and prevent errors during processing and classification.

4) Removal of symbols: Many of the elements that do not significantly contribute to the classification process are removed to avoid unnecessary complexity.

By performing these preprocessing steps, the dataset becomes more accurate, allowing the model to focus on the important aspects of the data. This approach lays the foundation for a more effective feature extraction process, leading to enhanced performance in data classification.

G. Step 7: Extraction Features

Features are extracted using lexical features. Lexical features are derived from the textual and structural components of URLs. The motivation for using lexical features is to rely on the appearance of a URL to determine the type of phishing or non-phishing. These features are commonly used in phishing detection systems and machine models to classify URLs as phishing or non-phishing [25].

1) Features based on length: These features depend on the length of many URL components:

a) URL length: Refers to the overall number of characters in the URL, including the protocols, hostname, path, queries, and any additional parameters.

b) Path length: Refers to the length of the URL path, which indicates a specific page or resource on the site.

c) Hostname length: Refers to the length of the part of the URL that identifies the server or site.

d) Top-level domain length: Refers to the top-level domain's length, indicating the type or geographic region of the site.

e) First directory length: Refers to the length of the first directory, which is the first part after '/' in the path.

2) Features based on count: Refers to the dependence of features on the number of times certain patterns appear within URLs. They are useful for analyzing URLs and discovering patterns that indicate phishing or non-phishing.

a) Number of dashes: Refers to the number of '-' symbols repeated within the URL. Its significance lies in identifying suspicious URLs that use many dashes in the process of dividing long parts of the URL.

b) Number of @ in the URLs: Refers to the total number of '@' symbols that appear. Its importance lies in the fact that some URLs contain the '@' symbol, which indicates attempts to redirect users within the URL.

c) Number of question marks: Refers to the count of '?' symbols in the URL. These are often used for creating queries, which attackers may use to collect user data.

d) Number of percentage signs: Refers to counting the number of '%' symbols in the URL, often used in encoding. Attackers may exploit this to hide parts of the URL or include special characters.

e) Number of HTTP instances: Count how many times HTTP appears in the URL. Some phishing URLs misuse HTTP to redirect the user.

f) Number of HTTPS instances: Count how many times HTTPS appears in the URL.

g) Number of WWW instances: Count the repetitions of WWW in the URL.

h) Number of dots in the URLs: Refers to the count of '.' dots. Phishing URLs may use excessive dots in domain names to appear similar to legitimate sites.

i) Number of equal signs in the URLs: Refers to the number of '=' symbols repeated in the URL, often used in query transactions. A high frequency may indicate data-collection attempts.

3) Features based on binary: Malicious URLs often use techniques to obscure their true identity, complicating detection by users and security systems. One popular tactic is replacing domain names with IP addresses (IPv4 or IPv6).

For example, instead of using a domain like: http://forexample.com/phishing

An attacker may use an IPv4 address and convert it to: http://196.168.1.1/phishing

This is for the case where the domain name changed from the identifiable to IPv4.

As for if the attacker uses an IPv6 instead of using the domain name.

For example: http://forexample.com/phishing

and converted it to: http://2001:db 8:ff 00:42:8329/phishing

Attackers exploit users' limited familiarity with IP addresses compared to domain names, making them more likely to click on these URLs without suspicion. However, using IP addresses instead of domain names can bypass detection systems that rely on domain-based pattern analysis, enhancing phishing or malware distribution capabilities.

H. Step 8: Classification Process

After completing the previous steps, which include dataset splitting, model building, and classification, the process is as follows:

1) *Data splitting:* The dataset is split into two groups:

a) *Training Group:* Refers to the data used to train the models.

b) *Testing Group:* Refers to the data used to evaluate the performance of the model.

c) The dataset is split with 70% for training and 30% for testing.

2) *Model building:* In our proposed model, the datasets are passed to different deep learning models, namely GRU, CNN, and BiGRU.

a) *CNN model:* This model is known as Convolutional Neural Network. In terms of data, it was developed as a method to handle it in various types. The structure and operation of the brain's visual cortex served as the model's inspiration [26].

b) *BiGRU model:* This model stands for Bidirectional GRU and contains a two-layer reinforcement neural network. This design allows the two layers of the output layer to fully integrate the contextual data of the input data sequence at every moment. The basic concept behind this model is that the input sequence is processed by both the forward and backward neural networks [27].

c) *GRU model:* It is a type of RNN, GRU short for Gated Recurrent Units. It contains GRU units, which are used for deep learning, particularly effective in processing sequential data for applications [28].

3) *Classification:* URLs are classified using deep learning models such as GRU, CNN, and BiGRU.

The results then indicate that this URL-based SMS is either phishing or a non-phishing message.

I. Step 9: Model Evaluation

The model evaluation process plays a crucial role in the evaluation performance of three models for deep learning in our proposed model. The evaluation process is based on four major criteria: precision, accuracy, recall, and F1 score. These criteria are essential to providing a comprehensive understanding of the model's ability to classify Arabic SMS messages containing URLs as non-phishing or phishing. By analyzing these criteria, we can determine the model that performs best in the particular task of detecting phishing in Arabic SMS. Following is an explanation of each of the four evaluation criteria:

First, the parameters used to evaluate the performance of models for deep learning are explained:

- True Positive (TP): Represents the number of URLs that were correctly classified as positive, indicating that phishing URLs were correctly classified as phishing.
- True Negative (TN): Represents the number of URLs that were correctly classified as negative, indicating that non-phishing URLs were correctly classified as non-phishing.
- False Positive (FP): Represents the number of URLs that were incorrectly classified as positive category, i.e., non-phishing URLs that were incorrectly classified as phishing.
- False Negative (FN): Represents the number of URLs incorrectly classified as belonging to the negative category, i.e., phishing URLs that were incorrectly classified as non-phishing.

Next, we will explain the four evaluation criteria:

- Accuracy: This metric is used to evaluate the quality of classification. It considers the rate of correct classification across all categories, rather than the distribution of the dataset. It reflects the number of correct predictions made by the model, whether the classifications are positive, i.e., identifying URLs as phishing, or negative, i.e., identifying URLs as non-phishing. A higher accuracy value indicates that the model is effectively classifying Arabic SMS messages based on URLs. It is represented by the following Eq. (1):

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN+FP+FN+TP}} \tag{1}$$

- Recall: It represents the percentage of actual phishing URLs correctly identified by the model. It indicates the model's ability to detect all phishing instances. A higher recall rate means that the model is less likely to ignore phishing URLs. It is represented by the following Eq. (2):

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

- F1 Score: It refers to the average between precision and recall, providing an integrated view of model performance, commonly used to assess the performance of the model in unbalanced classification problems. It is represented by the following Eq. (3):

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{3}$$

- Precision: It refers to correct positive predictions, i.e., URLs that were correctly identified as phishing. An increase in precision indicates the model is less likely to mistakenly classify non-phishing URLs as phishing. It is represented by the following Eq. (4):

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (4)$$

IV. Results and Discussions

A. Results

In this section, we present the performance of the proposed models for detecting Arabic SMS messages based on URLs. Three deep learning models were utilized: CNN, BiGRU, and GRU. Fig. 2 illustrates a comparison of the deep learning models' accuracy.

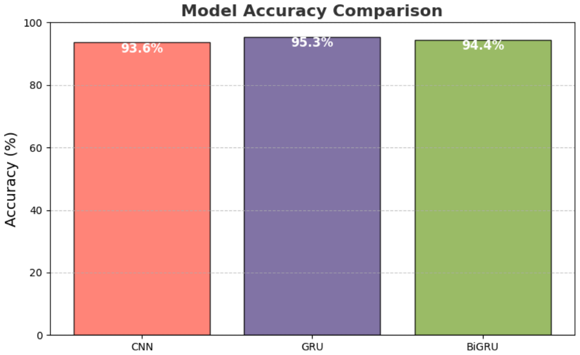
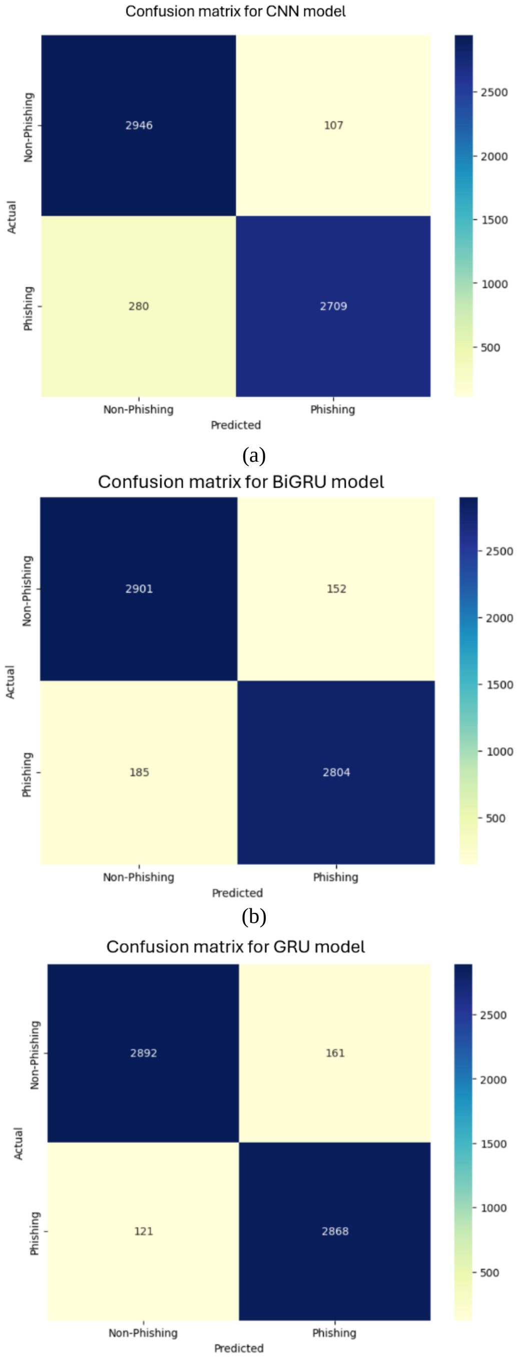


Fig. 2. Accuracy performance of three models for deep learning: CNN, GRU, and BiGRU.

Fig. 2 provides a detailed comparison of the performance of the three models proposed for deep learning in our proposed model for detecting Arabic SMS messages based on URLs. The classification focuses on determining whether an Arabic SMS message is phishing or non-phishing based on the type of URL extracted. The comparison was based on the accuracy achieved by each model. The GRU model demonstrates the best performance among the three models, achieving a superior accuracy rate of 95.33%. This high accuracy focuses on the GRU model's ability to effectively learn temporal dependencies in sequential data, making it particularly suitable for analyzing datasets. While the BiGRU model ranked second with an accuracy rate of 94.42%, slightly lower than the GRU model, the BiGRU's bidirectional architecture enables it to capture context in both the forward and backward directions. The CNN model achieved an accuracy rate of 93.59%, which, although lower than the GRU and BiGRU models.

From the graph in Fig. 2, it is clear that the GRU model outperforms the others in terms of accuracy. This superior performance demonstrates that the GRU is the most effective for the task of classifying Arabic SMS messages based on URLs in this paper.

Fig. 3, presents a confusion matrix for the three deep learning models, CNN, BiGRU, and GRU, used to classify Arabic SMS messages based on URL type as phishing or non-phishing. Confusion matrix (a) shows the performance of the CNN model. It correctly classified 2709 phishing URLs as phishing and correctly classified 2946 non-phishing URLs as non-phishing. However, it incorrectly classified 107 non-phishing URLs as phishing and incorrectly classified 280 phishing URLs as non-phishing. Confusion matrix (b) displays the performance of the BiGRU model, which correctly classified 2901 non-phishing URLs as non-phishing and correctly classified 2804 phishing URLs as phishing. Nevertheless, it incorrectly classified 152 non-phishing URLs as phishing and incorrectly classified 185 phishing URLs as non-phishing. Confusion matrix (c) illustrates the results of the GRU model, which correctly classified 2892 non-phishing URLs as non-phishing and correctly classified 2868 phishing URLs as phishing. However, incorrectly classifying 161 non-phishing URLs as phishing and incorrectly classifying 121 phishing URLs as non-phishing.



(a)

(b)

(c)

Fig. 3. Confusion matrix, (a) CNN, (b) BiGRU, (c) GRU.

TABLE II. A COMPREHENSIVE COMPARISON OF THE THREE PROPOSED MODELS, NAMELY CNN, BIGRU, AND GRU

Models	Evaluation Metrics %				Time (seconds)	
	Accuracy	Precision	Recall	F1 score	Train	Test
CNN	93.59	96.20	90.63	93.33	8.19	0.70
GRU	95.33	94.68	95.95	95.31	32.16	1.33
BiGRU	94.42	94.86	93.81	94.33	190.69	3.27

Based on Table II, a comprehensive comparison of the three proposed models, namely CNN, BiGRU, and GRU, is presented. This comparison is based on several performance metrics, namely precision, F1, recall, and accuracy, as well as two additional elements: the time required for training and testing. This analysis helps to understand the strengths and weaknesses of each model. The GRU model achieved the highest accuracy rate compared to the other two models at 95.33%, which indicates its strength in classification. While the precision was 94.68%, slightly lower than that of the CNN model, the recall was 95.95%, the highest among the models, indicating the model's ability to detect Arabic SMS phishing messages more effectively based on the URLs. The F1 score also had the highest result at 95.31%, reflecting a strong and balanced performance. As for the training and testing time performance, the training time was 32.16 seconds, and the testing time was 1.33 seconds, which is slower than CNN but faster than BiGRU. The BiGRU model followed, achieving a lower accuracy than GRU, at 94.42%, which is the average between the two models, CNN and GRU. It achieved a recall percentage that was lower than GRU but higher than CNN, which was 93.81%. As for the precision, it was also average among the other models, at 94.86%, slightly lower than GRU. The F1 score was 94.33%. In terms of training and testing time, it achieved a training time of 190.69 seconds and a testing time of 3.27 seconds, making it the slowest model in both training and testing, due to the processing of texts in both directions, i.e., reading from beginning to end and from end to beginning. The last model was the CNN model, which achieved an accuracy rate of 93.59%, which is a reasonable performance, but it is lower compared to the other models. In terms of recall, it was the lowest, indicating that it may miss some phishing messages, which was 90.63%. However, in terms of precision, it was the best among the other models, indicating that the model avoids false positives to a large extent, with a precision of 96.20%. The F1 score was 93.33%, reflecting a balanced result between precision and recall. In terms of training and testing time, the training time was 8.19 seconds, and the testing time was 0.70 seconds. This indicates that it is the fastest model in both training and testing among all the models, making it an excellent choice for practical applications in time-critical situations.

B. Discussions

The results indicate that GRU is the most effective model for classifying Arabic SMS messages based on URLs, due to its high accuracy, recall, and balanced F1 score. This makes it able to learn temporal dependencies perfectly in analyzing sequential data, such as Arabic SMS messages containing URLs. Although the BiGRU model was able to capture context from both directions, forward and backward, it took longer training and testing time, which can limit its practical application in real-time scenarios. While the CNN model was the fastest, it performed poorly in accuracy and recall, making it an excellent choice when speed is considered. Therefore, the results emphasized that the GRU model outperforms both BiGRU and CNN in terms of accuracy and recall, which indicates its strength in processing sequential data and provides the best balance between speed and classification, making it the superior choice for detecting Arabic SMS phishing messages based on URLs. However, the CNN model provided the fastest training and testing time, but it provided the lowest recall, indicating a higher probability of missing Arabic SMS phishing messages based on URLs, which reduces its reliability compared to GRU and BiGRU. The BiGRU model is an alternative solution when the demand for contextual understanding is high. Therefore, based on the paper's goal of choosing a deep learning model that provides better accuracy in detecting Arabic SMS phishing messages based on URLs, the GRU model is the most suitable choice that achieves this goal based on the previous results.

V. CONCLUSION AND FUTURE WORK

The rapid development and widespread use of smartphones have led to an increase in cyber-attacks targeting smartphones, including SMS phishing attacks. This paper proposed a model for detecting Arabic SMS phishing messages based on URLs using models for deep learning, namely GRU, CNN, and BiGRU. We assessed the performance of these deep learning models and compared their accuracy and effectiveness in the detection process. The GRU model illustrates superior performance with an accuracy of 95.3%, demonstrating its capability to effectively process data sequences and capture contextual relations within the dataset. This high level of accuracy makes the GRU model an excellent candidate for applications where accuracy is critical. Although the CNN model achieved a slightly lower accuracy of 93.6%, it was capable of better in faster training time compared to the GRU model. This makes CNN a strong option for real-time scenarios requiring faster processing as a priority. The BiGRU model achieved an accuracy of 94.4%, which is lower than GRU but higher than CNN, although it did not outperform GRU in terms of performance. Its bidirectional structure allowed it to capture contextual data in both forward and backward directions, making it the suitable option in certain applications. These results emphasized the significance of selecting the appropriate model based on certain requirements, such as accuracy or speed.

This paper has achieved valuable objectives, but it has some limitations. First, the dataset used was relatively small and translated from English to Arabic due to the lack of a supporting Arabic dataset in this field, which may affect the results. Second, the models were assessed based on URLs as a phishing indicator, excluding other indicators that may be used as phishing processes, such as email and phone numbers. In future work, we aim to expand the Arabic dataset, compare the proposed models with other deep learning techniques to mitigate phishing detection in Arabic SMS messages and extend the proposed model to include other indicators such as email and phone numbers. These proposals aim to create a more comprehensive solution to mitigate SMS phishing messages in Arabic-speaking communities.

REFERENCES

[1] E. A. Fischer, “Cybersecurity Issues and Challenges: In Brief,” 2014.

[2] K. M. Sudar, P. Deepalakshmi, P. Nagaraj, and V. Muneeswaran, “Analysis of Cyberattacks and its Detection Mechanisms,” in 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Bangalore, India: IEEE, Nov. 2020, pp. 12–16. doi: 10.1109/ICRCICN50933.2020.9296178.

[3] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” Electronics, vol. 12, no. 6, p. 1333, Mar. 2023, doi: 10.3390/electronics12061333.

[4] S. Gupta, A. Singhal, and A. Kapoor, “A literature survey on social engineering attacks: Phishing attack,” in 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India: IEEE, Apr. 2016, pp. 537–540. doi: 10.1109/CCAA.2016.7813778.

[5] F. Mouton, L. Leenen, M. M. Malan, and H. S. Venter, “Towards an Ontological Model Defining the Social Engineering Domain,” in ICT and Society, K. Kimppa, D. Whitehouse, T. Kuusela, and J. Phahlamohlaka, Eds., Berlin, Heidelberg: Springer, 2014, pp. 266–279. doi: 10.1007/978-3-662-44208-1_22.

[6] R. Alabdan, “Phishing Attacks Survey: Types, Vectors, and Technical Approaches,” Future Internet, vol. 12, no. 10, Art. no. 10, Oct. 2020, doi: 10.3390/fi12100168.

[7] College of Computers and Information Technology, Taif University, Saudi Arabia et al., “Four Most Famous Cyber Attacks for Financial Gains,” IJEAT, vol. 9, no. 2, pp. 2131–2139, Dec. 2019, doi: 10.35940/ijeat.B3601.129219.

[8] S. Mishra and D. Soni, “DSmishSMS-A System to Detect Smishing SMS,” Neural Comput & Applic, vol. 35, no. 7, pp. 4975–4992, Mar. 2023, doi: 10.1007/s00521-021-06305-y.

[9] 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, Jul. 2020, doi: 10.1016/j.future.2020.03.021.

[10] N. Agrawal, A. Bajpai, K. Dubey, and B. D. Patro, An Effective Approach to Classify Fraud SMS Using Hybrid Machine Learning Models. 2023, p. 6. doi: 10.1109/I2CT57861.2023.10126300.

[11] P. Prasanna Bharathi, G. Pavani, K. Krishna Varshitha, and V. Radhesyam, “Spam SMS Filtering Using Support Vector Machines,” in Intelligent Data Communication Technologies and Internet of Things, vol. 57, J. Hemanth, R. Bestak, and J. I.-Z. Chen, Eds., in Lecture Notes on Data Engineering and Communications Technologies, vol. 57. , Singapore: Springer Singapore, 2021, pp. 653–661. doi: 10.1007/978-981-15-9509-7_53.

[12] T. Wu, K. Zheng, C. Wu, and X. Wang, “SMS Phishing Detection Using Oversampling and Feature Optimization Method,” dtcse, no. iece, Dec. 2018, doi: 10.12783/dtcse/iece2018/26634.

[13] C. Oswald, S. E. Simon, and A. Bhattacharya, “SpotSpam: Intention Analysis–driven SMS Spam Detection Using BERT Embeddings,” ACM Trans. Web, vol. 16, no. 3, pp. 1–27, Aug. 2022, doi: 10.1145/3538491.

[14] V. M. Tuan, N. X. Thang, and T. Q. Anh, “Evaluating the Efficiency of Vietnamese SMS Spam Detection Techniques,” ISJ, vol. 1, no. 18, Jun. 2023, doi: 10.54654/isj.v1i18.932.

[15] University of Baghdad, S. Hameed, Z. Ali, and Mustansiriyah University, “SMS Spam Detection Based on Fuzzy Rules and Binary Particle Swarm Optimization,” IJIES, vol. 14, no. 2, pp. 314–322, Apr. 2021, doi: 10.22266/ijies2021.0430.28.

[16] N. N. Amir Sjarif, Y. Yahya, S. Chuprat, and N. H. F. Mohd Azmi, “Support Vector Machine Algorithm for SMS Spam Classification in The Telecommunication Industry,” International Journal on Advanced Science, Engineering and Information Technology, vol. 10, no. 2, p. 635, Apr. 2020, doi: 10.18517/ijaseit.10.2.10175.

[17] M. A. Uddin, M. N. Islam, L. Maglaras, H. Janicke, and I. H. Sarker, “ExplainableDetector: Exploring Transformer-based Language Modeling Approach for SMS Spam Detection with Explainability Analysis,” May 12, 2024, arXiv: arXiv:2405.08026. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/2405.08026

[18] Z. H. Ali, H. M. Salman, and A. H. Harif, “SMS Spam Detection Using Multiple Linear Regression and Extreme Learning Machines,” Iraqi Journal of Science, pp. 6342–6351, Oct. 2023, doi: 10.24996/ijs.2023.64.10.45.

[19] G. Sonowal, “Detecting Phishing SMS Based on Multiple Correlation Algorithms,” SN COMPUT. SCI., vol. 1, no. 6, p. 361, Nov. 2020, doi: 10.1007/s42979-020-00377-8.

[20] S. Giri, S. Das, S. B. Das, and S. Banerjee, “SMS Spam Classification–Simple Deep Learning Models with Higher Accuracy using BUNOW and GloVe Word Embedding”, doi: http://dx.doi.org/10.6180/jase.202310_26(10).0015.

[21] A. Ibrahim, S. Alyousef, H. Alajmi, R. Aldossari, and F. Masmoudi, “Phishing Detection in Arabic SMS Messages using Natural Language Processing,” in 2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU), Riyadh, Saudi Arabia: IEEE, Mar. 2024, pp. 141–146. doi: 10.1109/WiDS-PSU61003.2024.00040.

[22] J. H. Tiago Almeida, “SMS Spam Collection.” UCI Machine Learning Repository, 2011. doi: 10.24432/C5CC84.

[23] “Spam / Ham SMS DataSet.” Accessed: Oct. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/vivekchutke/spam-ham-sms-dataset

[24] E. S. Aung and H. Yamana, “Segmentation-based Phishing URL Detection,” in IEEE/WIC/ACM International Conference on Web Intelligence, ESSENDON VIC Australia: ACM, Dec. 2021, pp. 550–556. doi: 10.1145/3486622.3493983.

[25] D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,” Aug. 21, 2019, arXiv: arXiv:1701.07179. doi: 10.48550/arXiv.1701.07179.

[26] S. Min, B. Lee, and S. Yoon, “Deep Learning in Bioinformatics,” vol. 47(6), pp. 366–382, Dec. 2023, doi: https://doi.org/10.55730/1300-0152.2671.

[27] P. Li et al., “Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation,” IJGI, vol. 9, no. 11, p. 635, Oct. 2020, doi: 10.3390/ijgi9110635.

[28] “Gated Recurrent Unit Definition | DeepAI.” Accessed: Oct. 03, 2024. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/gated-recurrent-unit.