Deep learning in phishing mitigation: a uniform resource locator-based predictive model

Hamzah Salah[1], Hiba Zuhair[2]
[1]Department of Networks Engineering, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
[2]Department of Systems Engineering, College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

Article Info

Article history:

Received Aug 3, 2022
Revised Sep 12, 2022
Accepted Oct 1, 2022

Keywords:

Convolutional neural network with bi-directional long short-term memory
Deep learning
Phish website
Phishing detection

ABSTRACT

To mitigate the evolution of phish websites, various phishing prediction8 schemes are being optimized eventually. However, the optimized methods produce gratuitous performance overhead due to the limited exploration of advanced phishing cues. Thus, a phishing uniform resource locator-based predictive model is enhanced by this work to defeat this deficiency using deep learning algorithms. This model's architecture encompasses pre-processing of the effective feature space that is made up of 60 mutual uniform resource locator (URL) phishing features, and a dual deep learning-based model of convolution neural network with bi-directional long short-term memory (CNN-BiLSTM). The proposed predictive model is trained and tested on a dataset of 14,000 phish URLs and 28,074 legitimate URLs. Experimentally, the performance outputs are remarked with a 0.01% false positive rate (FPR) and 99.27% testing accuracy.

Corresponding Author:

Hamzah Salah
Department of Networks Engineering, College of Information Engineering, Al-Nahrain University
Baghdad, Iraq
Email: ha.salim94@gmail.com

1. INTRODUCTION

Phishing is a social engineering attack aiming at stealing the sensitive information of victim users and enterprises like credit card numbers, login credentials, products' quick response codes, and even encrypted passwords. Usually, phishers evolve fake websites (phish websites) that closely resemble trustworthy websites of any enterprise, to redirect the victims through spoofing links for illegal gains, industrial espionage, cyber-crimes and so far, cyber-terrorism and cyber-warfare [1], [2]. To defeat phishing and mitigate its impacts, many detection schemes have been developed by applying black-lists, white-list, visual similarity, machine learning, and then deep learning algorithms as well as anti-phishing tools that are surplus to their requirements like Google secure browsing and PhishTank, or the Denylist which are based on users voting [3]. Although all of them have produced acceptable rates of detection accuracies with low false-positive rates, they are still fragile at predicting all uniform resource locator (URL) cues of evolutionary phish websites [1]–[3].

To thwart this evolution in the vast cyber-space, deep learning knowledge has been evoked for superior performance among other phishing detection schemes [4], [5]. However, they have suffered from categorizing the most mutual URL features that have been exploited by new phish websites as well as their shortage at predicting the forecasting change of these URLs in the future [5], [6]. Thus, the mutual phishing feature space needs to narrow with efficacious and lightweight deep learning-algorithms configuration. That, in turn, has become an ultimate solution to deep learning in phishing mitigation at the present and in the near future [5], [6]. To provide this ultima, a set of 60 mutual features with the least flaws have been filtered from

111 examined phishing features to be fed into a hybrid deep learning predictive model for better learning optimization. The hybrid deep learning predictive model was of dual architecture that encompassed the adapting exclamations of convolution neural network (CNN) and the decisive parameters of bidirectional long short-term memory (Bi-LSTM). From examining the frontiers of deep learning-based phishing anticipating approaches that have been studied in our previous research work [6], this work aims at providing efficacious phish website predictor through the following contributions: Exploring 111 adversarial URL phishing features that belong to URL property and domains (41 features), URL directory (18 features), the URL file name (18 features), URL parameters (20 features) as well as URL resolving and external services (14 features) to convolve 60 potential features. And handling deep-learning induction limits by hyper-parameterizing decisive margins of CNN and Bi-LSTM into a dual deep learning model.

People's life has become more dependent on cyberspace, data on the cloud, social media networking, web transactions, e-healthcare, e-business, e-learning and education, and e-government services over the last decade, particularly during the coronavirus disease 2019 (COVID-19) epidemic [7]. Active cyber-space users have surpassed 4.66 billion (or 59.5% of the global population) through the channels and services as reported by World Digital Population Report 2021 [8]. Eventually, a lot of sensitive data has been transmitted and stored via cloud computing, which has given hackers many opportunities to impersonate trustworthy enterprises and services to intrude on computer-based systems and mobile platforms illegally using social engineering mimics [9]. Among the most potential social engineering-based cyber-attack, are phish websites that aim to steal users' credentials like credit card numbers, logins, form submissions, file uploading, and passwords [10]. To do so, phishers often register their websites with fake layouts and domain addresses that resemble trustworthy websites of their targets (known enterprises and communities). Next, phishers have sent their mass notifications to victim users acquiring them to click spoof links to redirect them to fake websites. Then, victim users have sent the required credentials and phishers catch their digital identities [7]–[10]. Thus, the potential inflicts of phishing have reached billions of dollars to the cyber-space enterprises, data loss, and cyber-security defeat [7]–[10].

To compete with this steady escalation of phishing and to mitigate its social and economic impacts, researchers have devised various schemes of list-based, heuristics-based, visual similarity-based, and machine learning-based phishing mitigation [1], [3]. Typically, list-based phishing mitigation has utilized either whitelists or blacklists of authorized URLs or phishing URLs, respectively [3], [7], [8]. However, they have produced significant false negatives because they have needed major and frequent modifications to their employed lists. Whilst heuristic-based schemes have deployed extracted heuristics that are usually exploited by phish websites to characterize them, they have fallen short at the holistic characterization of all patterns of phish websites in high detection levels with the least false detections [3], [7], [8]. Contrarily, the visual similarity-based phishing mitigation schemes target phishers' replicas of trustworthy websites that are similar in their images, pictures, data items, and the source code's tree structure. However, they have suffered from time-consuming layout processing, similarity/dissimilarity matching, and a lot of processing and memory footprints for layout traits and storage. That in turn, has yielded high false detections [6], [8]. On the other hand, the machine learning-based phishing mitigation schemes [1], [2] have been employed with different algorithms like random forest (RF), k-nearest neighbor (KNN), naive Bayes, artificial neural network (ANN), support vector machine (SVM), and decision tree (DT) in a single or an ensemble design. They have constructed effective features-based classifiers that could classify typical patterns of phish websites among legitimate websites [1], [2], [6]. By product, such classifiers have outperformed their competitors at detecting prevalent phishing activities (i.e. those exploiting generic features) professionally but they have limited at predicting those adversarial features adaptively in the near future [2].

Oppositely, deep learning-based phishing mitigation schemes peer to outperform their formers because they could identify and prevent zero-hour phish websites that sophisticated their features [3], [4]. They have been trained on phishing URLs, phishing who is lookups, and phishing web page layouts. However, employing various features together has caused mild false detections on big cyber-data due to the variety of examined features as well as the diversity of their relevance and redundancy to phishing. To the best of our knowledge, phishing has shared many cues with malware and ransomware attacks that target computer-based systems and mobile-based systems [4], [6], [11]. To cope with this challenge, researchers need to improve the proactive mitigation of phish websites by advancing the architecture of deep learning classifiers. By applying self-learning, unsupervised, supervised, and hybrid learning; the prediction models could characterize the mutually relevant phishing features on the vast cyber-data without human tuning [9], [10]. To this point, this presented work attempts to improve a deep learning-based classifier that effectively characterizes and categorizes URL phishing features as described in the next sections.

The paper organized into six sections as follows: the literature review depicted in: section 2 which depicts phishing and its mitigation by synthesizing and categorizing the literature. Section 3 depicts the method and materials needed to evoke the hybrid architecture of deep learning model with a list of 111 features alongside the used method to select the 60 informative features. Experiments, datasets, and evaluations are described and discussed in section 4. Section 5 discuss the result of the experiments. Finally, section 6 concludes the whole work.

2. LITERATURE REVIEW

This section studies the literature on deep learning-based detection schemes in a critical context. Among the literature, was that improved by [12] for phishing detection using deep neural network (DNN) across a big dataset of Ebbu2017 to obtain up to 98% as the detection accuracy overall examined URLs. Ultimately, the authors of [13] attempted to boost that detection accuracy rate through a blended approach of DNN and features weighting algorithms like genetic algorithm (GA) to classify phish websites by their most exploiting features. While researchers of [14], applied a multi-headed and self-attentional CNN on an imbalanced dataset throughout a generative adversarial network (GAN) with a large number of URL features. However, their work fell short of fixing the length of examined URL strings among other URL features. Therefore, their work achieved 97.20% of detection accuracy with a rare focus on optimization parameters for fewer processing resources. Likely, Yerima and Alzaylaee [15] designed a single-based classifier using the CNN algorithm to detect phishing from legitimate websites with a higher rate of detection accuracy (98.2%) and low rate of false detections by automating the selection of features via their key influencing parameters. But Kumar *et al.* [16] attempted to create a neural network using swarm intelligence binary bat algorithm for website classification across a big dataset collected from Kaggle. The proposed approach achieved 94.8% detection accuracy but an unacceptable rate of false detections. That was due to the never-been tuning model in terms of the number of epochs, the learning rate, and the data batch size. Then, a faster recurrent neural network (R-CNN) was developed by [17] for website logo recognition with an improved feature pyramid network (FPN) criterion. Both FlickrLogos-32 and FlickrLogos-32plus datasets were aggregated to obtain detection accuracy of (98.9%) and (94.6%). Also, a long short-term memory (LSTM) model was developed for phishing URL detection with one-hot encoding to encode URL strings. As such, each encoded character vector could be fed into the LSTM classifier on a big dataset retrieved from PhishTank and Common Crawl with a detection accuracy of 93.5%. Similarly, the deep auto-encoder model was obtained by [18] and achieved a detection accuracy of 97.34% at carious crawling zero-day phish samples using three categories of character, string, and address code feature. Ultimately, researchers of [19] employed LSTM and CNN algorithms in an ensemble model to detect phishing activities on a dataset of up to 200K samples. They got a detection accuracy of 96% via decision voting. However, both approaches obtained unstable detection accuracy rates across the escalating dataset.

On the other hand, a hybrid deep learning model was proposed in [20] to classify phish websites through their character embedding and natural language processing (NLP) characteristics with DNN and bi-directional short-term memory (BiLSTM). Although the proposed model yielded a detection accuracy of 98.79% on the Ebbu2017 dataset and a detection accuracy of 99.21% on the PhishTank dataset, it yielded a long execution time and unacceptable false alarms due to the use of various character exploitations and features that needed extra processing. While a DNN with convolutional layers was developed together for phishing detection [21]. Experimental results yielded approximately 100% of detection accuracy on collected URLs due to phishing characterization using the text of URLs only. In [22], blended the CNN with LSTM for characterizing website layouts like pictures, frames, and text. The developed classifier achieved a detection accuracy of 93.28% in maximally 25 seconds of computational time. Datasets retrieved from Phish Tank and common crawl were examined by the developed classifier which was insufficient to meet the needs of the real-time practice. Further, Somesha *et al.* [23] applied feature set reduction using information gain (IG) for the most distinctive URL obfuscation, hyperlink, and third-party features. Their approach achieved a detection accuracy of 99.52% using DNN, 99.57% using LSTM, and 99.43% using CNN. Thus, they constructed their model to be subject to third-party services exclusively.

Although the afore-mentioned works have made convincing contributions to support the urgent need for efficacious phishing detection, they still have a shortage in: i) adapting decisive classification to the advanced phishing features due to exploiting the generic, immutable phishing feature space; ii) insignificant computational cost in leveraging the escalating phishing attacks due to the use of external search engines and resources; and iii) limited prediction accuracies and unacceptable rates of falsely detecting adversaries due to their decisive parameters, layered architectures, and hyper-parameters limits.

3. MATERIAL AND METHOD

3.1. The anticipating phishing features

To define the prominent anticipating features space of phishing, various phishing URLs were aggregated from three different sources to emulate their generic and sophisticated exploitations that might be employed to defeat phishing mitigation schemes. As shown in Figure 1, a generic uniform resource locator (URL) identified webpages and file resources on the web through a protocol, a hostname, and a pathname. Typically, the hostname could refer to which server that the resource could be located via its components: subdomain and domain, whereas the pathname, specifies the location of that resource on that server. Domain camouflaged the second-level domain (SLD) and the top-level domain (TLD). SLD, in turn, stood for the brand name, trademark name, or organization name [24]. On the other hand, the pathname broke into directory, filename, and arguments. Simultaneously, phish adversaries have exploited obfuscation, spoofing, and cybersquatting interchangeably in subdomain, domains and brand names, TLDs, common spelling mistakes, and re-ordering or consecutive characters in the domain names, respectively [24]. Further, phish websites are used to exploit visual spoofing and uni-coding alteration into the homographs of the target domains, as well as inserting suspicious words instead of brand names in the domain part. So far, phishing websites concealed URL property as well as hoaxing URL directory file names, and external services [25]–[27].

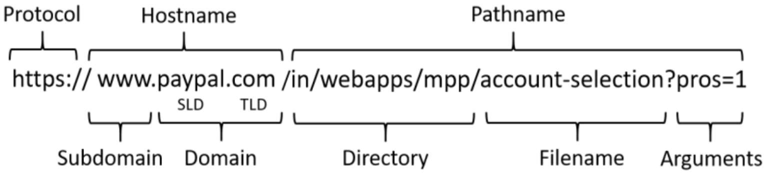


Figure 1. Generic URL components, an example of PayPal homepage

3.2. Features selection

Information gain [28] in features engineering has contributed the most essential features for further classification tasks based on discovering each informative feature's entropy that has inferred how much the feature has associated with random variable uncertainty by using information theory throughout the (1), (2), (3), and (4) [29]. As such, the information entropy of the random variable Y has been defined for the random variable X. Next, the difference in the information entropy between the random variables X and Y has been calculated as the amount of information gained by eliminating the uncertainty. Then, the value of Information Gain can be inferred by understanding the significance of the presence or absence of an individual feature to the classification task so that only the amount of information that characteristically has contributed to the classification can be used to determine its relevance. Thus, the feature has become increasingly essential as the amount of information it carried has increased. As a result, the significant features with higher information gain can be filtered to be encompassed in a subset of selective features.

$$H(X) = -\sum^{i}\ p(x_i) log\, p(x_i) \tag{1}$$

$$H(X|Y) = -\sum_j p(y_j)\ \sum_i p(x_i|y_j)\ log\, p(x_i|y_j) \tag{2}$$

$$IG\ (X,Y)\ =\ H\ (X)\ -\ H\ (X\ |\ Y) \tag{3}$$

$$IG\ (X,Y)\ =\ H\ (X)\ -\ H\ (X\ |\ Y) \tag{4}$$

3.3. Convolutional neural network

As shown in Figure 2 CNN has made up of multi-layers of convolution, subsampling, fully connected, and normalization layers, respectively. In the convolution layer, the most proficient features have been extracted and then they have been fed to the fully connected layer for their categorization. In between a sub-sampling layer has been used frequently to pixelate a picture. All layers have encompassed kernels to be grouped by two-dimensional neurons. As such, the neurons in each feature extraction layer cannot be connected to all neurons in the next layers, unlike typical neural networking algorithms [30]. As such, a feature map can be constructed by coupling only the proceeding layers into a number of spatial mappings, fixed in size, and partial in overlapping, neurons. Thus, fewer connections can be produced with less training time and overfitting risk. A filter's neurons in each layer can be linked to the same number of neurons in the prior input layer producing a feature map with the corresponding features' weights and biases. Whilst, max and/or mean pooling can be used for sub-sampling frequently such that the last layer is fully coupled with the neurons for classification, the overall weights and biases can be trained in a backpropagation context [31].

3.4. Bidirectional long short-term memory

The bidirectional long short-term memory (Bi-LSTM) model has resolved the constraints of the former LSTM by considering both the past and future context of the tracked sequence of dependencies at every time step as shown in Figure [32]–[35]. The Bi-LSTM model can learn incoming data sequentially and develop recurrent neural network models that have been relied on in the context of the previous state to save their important information [33]. Whereas, the former model of LSTM has resolved the architecture of recurrent neural network (RNN) with self-loops to preserve the gradient of the recent inputs during a long period [34]. Thus, the significant information has been accumulated to be interfaced into the next step via a memory cell structure [34]. That is behind the capability of LSTM to overcome vanishing gradients, unlike RNN by the gradient disappearance and the gradient blowing up, which has allowed the gradients to pass unmodified. However, LSTM takes a significantly long time to train, datasets among its competitors [33], [34]. In addition, LSTM only considers the forward information and does not consider the backward information. Hence, this issue has been resolved in Bidirectional LSTM [33], [34]. Experimentally, BiLSTM has been proven as a powerful method as a URL anticipating model that could improve URL categorization through its layers compared to LSTM layers. Therefore, it has been blended with CNN to train URL features in our proposed detection model [33], [34].

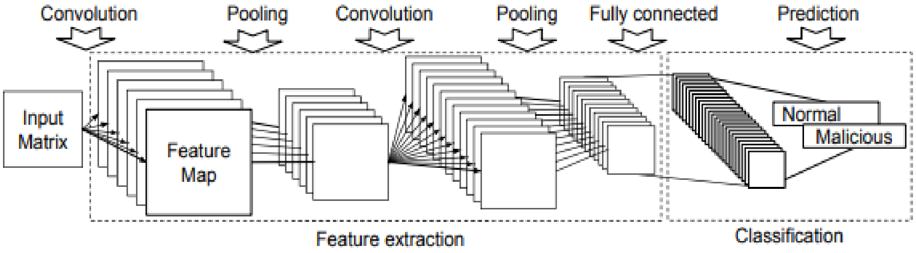


Figure 2. Architecture of CNN

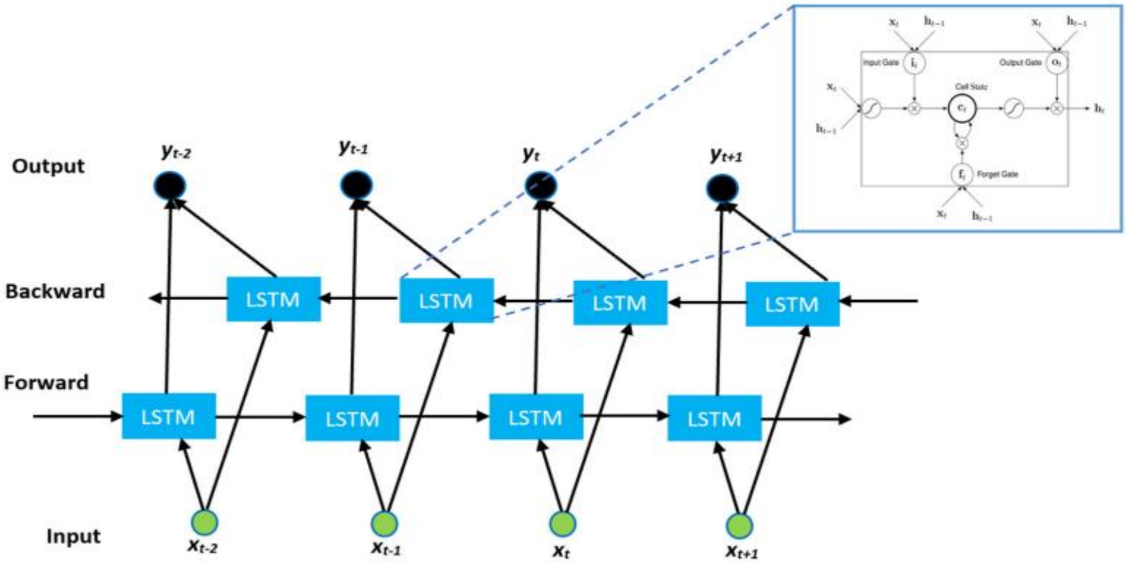


Figure 3. Architecture of bi-directional long short-term memory [36]

3.5. Dual predictive model

In the dual predictive model (CNN-BiLSTM), each input sequence in the input vector is convolved using one of the 20 convolution filters. A one-dimensional vector is set as the filter size. Downsampling and spatial dimensionality functions are performed by the maximum pooling layer. To obtain the highest value feasible, input features in each filter kernel’s pool are added together. Bidirectional LSTM networks have an output that is shared by two hidden layers that are connected in different directions. Reproductive deep learning is used in the BiLSTM network’s production layer to obtain knowledge sequences from both previous and future states. LSTM layer memory cells could distribute the results of earlier data features into the output layer. A reduced performance for the machine learning system is also the result of the features being learned just in the forward direction, neglecting the backward relation. This flaw is addressed by the bidirectional recurrent network approach, which processes data both forward and backward. Respectively, each LSTM cell performs four discrete computations based on four gates: *input* (), *forget* (), *candidate* (), and *output* (). The following is an introduction and definition of the equations for these gates:

$$f_t = \sigma(W_f\, x_t + w_f\, h_{t-1} + w_{cf}\, c_{t-1} + b_f) \tag{5}$$

$$i_t = \sigma(W_i\, x_t + w_i\, h_{t-1} + b_i) \tag{6}$$

$$o_t = \sigma(W_o\, x_t + w_o\, h_{t-1} + b_o) \tag{7}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes tanh(W_c\, x_t + W_f\, h_{t-1} + b_c) \tag{8}$$

$$h_t = o_t \otimes tanh(c_t) \tag{9}$$

$$H_t = (h_t^{\rightarrow} : h_t^{\leftarrow}) \tag{10}$$

Altogether, the forget, input, and output gate vectors are represented by f, i and o, respectively. *b*, *W*, *w*, and $\otimes$ stand for bias, input weights, recurrent output weights, and element-wise multiplication, respectively. h_t is the LSTM cell's output, while $(h_t^{\rightarrow} : h_t^{\leftarrow})$ represents the concatenation of the outputs from the forward and backward layers, as shown in Figure 4.

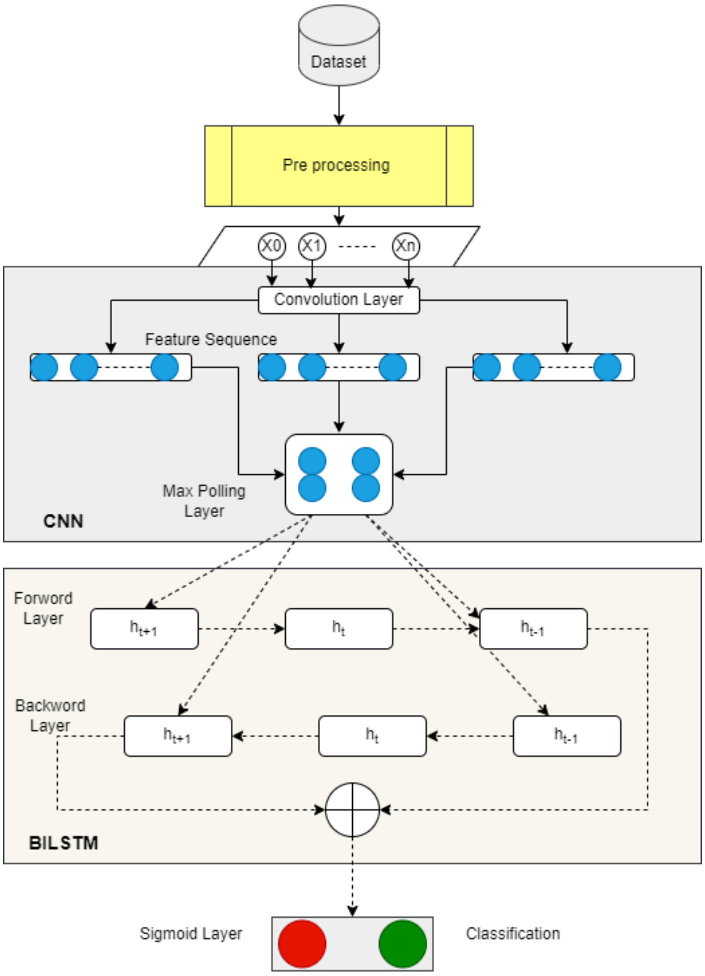


Figure 4. The dual predictive model (CNN-BiLSTM)

3.6. Feature space pre-processing

The feature space was constructed through three steps of cleaning, shuffling, and selection as shown in Figure 5. First, URLs have been chopped as a bag of words and given a unique integer value for each word in the bag via Python's string package and printable class. As such, all penetrated URLs would have a similar size. All null values of the penetrated URLs have been eliminated in the cleaning step before feeding them to the shuffling step to avoid any obstruction during the URL training. Next, the shuffling step omitted overfitting values to reduce the random abundance of URL classes (phish and legitimate). Then, the selection step has filtered the mass-informative features from the miss-informative features to the classification accuracy via information gain (IG) criterion. Correspondingly, only the topmost potential and mutually relevant features to characterize the adversarial phish URLs have been selected to generate the feature space, as shown in Figures 6 to 11. Table 1**Error! Reference source not found.** shows the optimum features, with the value of information Gain after applying IG algorithm to the large feature set.

Then, the feature space was split into feature vectors and target vectors. The target vectors have been made up of URLs' classes as either "0" for legitimate or "1" for phish. Whereas feature vectors have consisted of the values of all 60 mutual features. As such, the dual prediction model (CNN-BiLSTM) would optimize its decisive parameters of convolution neural network and bi-directional long short-term memory by obtaining the representative feature space of 60 features for training without the undue time and memory consumption versus the sophisticated phish URLs.

Training features space has been fed to the training pipeline particularly the features extraction module to extract feature vector. Feature vectors have been passed into the input layer. Then, URL features have been chained by CNN max-pooling with an activation function to be fed into the BiLSTM layer. That, in turn, employed a hyperbolic tangent function to activate the output with a size of 32 through a dense layer of two neurons alongside the sigmoid activation function for actual classification. Altogether output classes have been passed to the output layer for categorizing them into phish and/or legitimate labels. Both binary cross-entropy Adam optimizer and dropouts have been exploited throughout all hidden layers. On the other hand, the testing pipeline gets the advantage of the dual predictive model (CNN-BiLSMT) in the training pipeline to classify the extracted features in the testing features space as either phish or legitimate URLs iteratively. Then, produced dual predictive model (CNN-BiLSTM) has been run throughout the framework in Figure 12 that has evoked a web request while the user browsing the websites and checked up whether that requesting URL might be safe to visit by notifying the user.

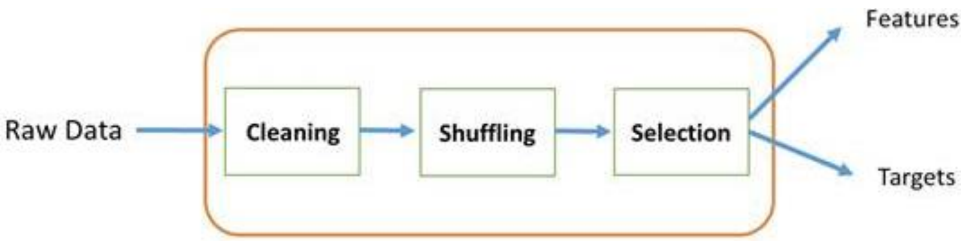


Figure 5. Feature space construction

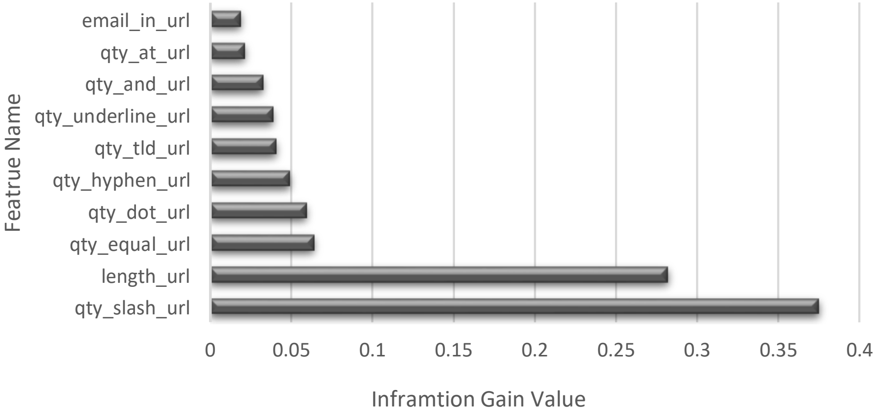


Figure 6. Most effective feature of whole URL properties

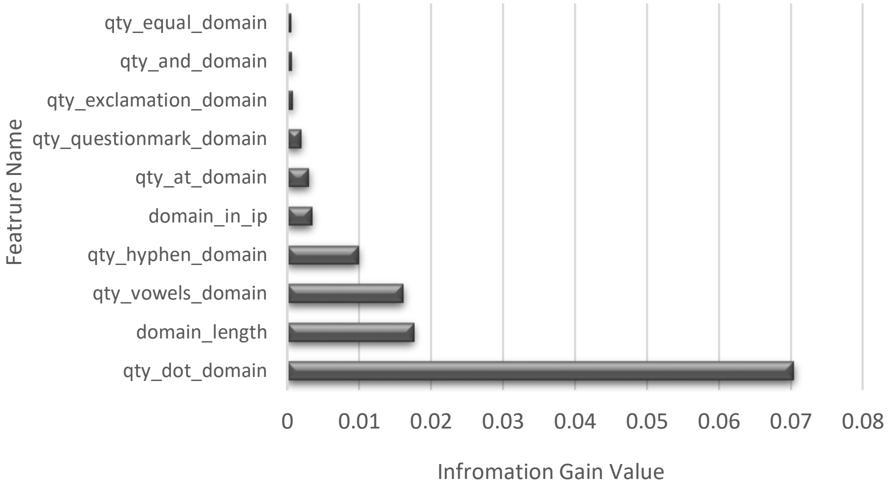


Figure 7. Most effective feature of domain properties

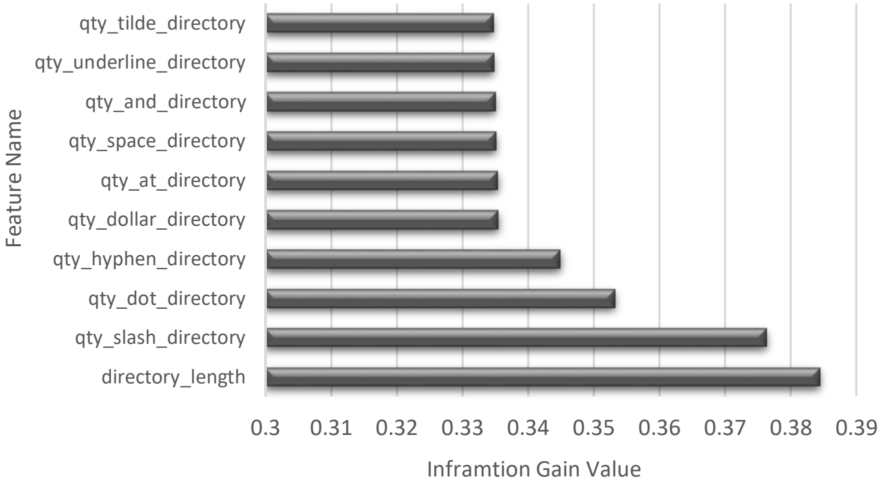


Figure 8. Most effective feature of directory properties

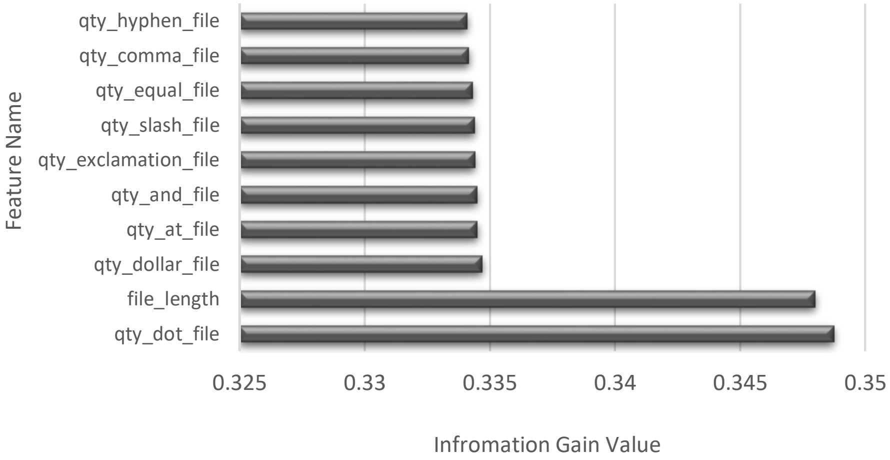


Figure 9. Most effective feature of file properties

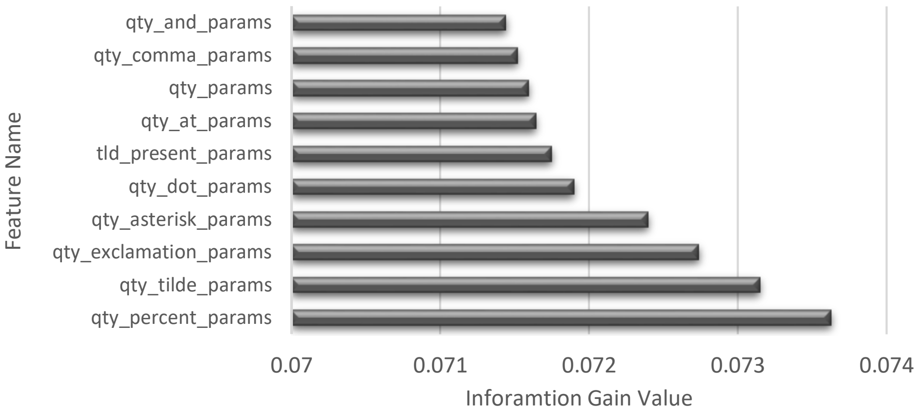


Figure 10. Most effective feature of parameter properties

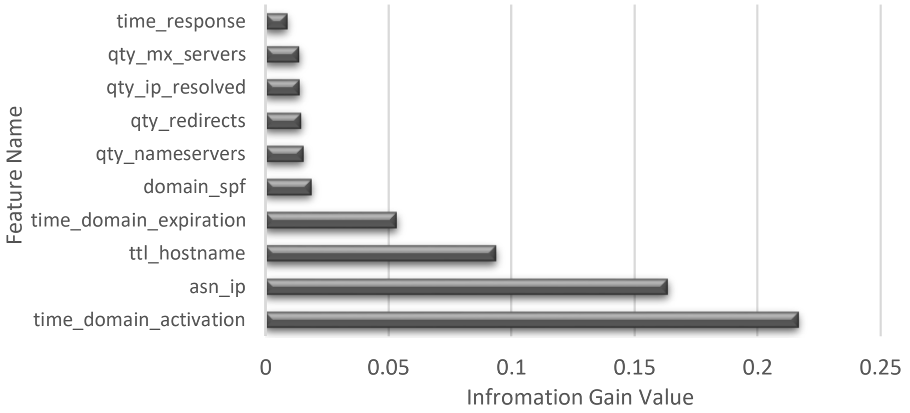


Figure 11. Most effective feature of resolve and third-party properties

Table 1. Optimum feature set based on IG algorithm

ID	Feature Name	IG	ID	Feature Name	IG
F1	*qty_slash_url*	0.374893	F31	*qty_dot_file*	0.348736
F2	*length_url*	0.281893	F32	*file_length*	0.347988
F3	*qty_equal_url*	0.063778	F33	*qty_dollar_file*	0.334679
F4	*qty_dot_url*	0.059265	F34	*qty_at_file*	0.33449
F5	*qty_hyphen_url*	0.048873	F35	*qty_and_file*	0.334483
F6	*qty_tld_url*	0.040417	F36	*qty_exclamation_file*	0.334395
F7	*qty_underline_url*	0.038548	F37	*qty_slash_file*	0.334388
F8	*qty_and_url*	0.032214	F38	*qty_equal_file*	0.334304
F9	*qty_at_url*	0.021126	F39	*qty_comma_file*	0.334149
F10	*email_in_url*	0.018615	F40	*qty_hyphen_file*	0.334089
F11	*qty_dot_domain*	0.070389	F41	*qty_percent_params*	0.073628
F12	*domain_length*	0.017554	F42	*qty_tilde_params*	0.073148
F13	*qty_vowels_domain*	0.016099	F43	*qty_exclamation_params*	0.072738
F14	*qty_hyphen_domain*	0.009891	F44	*qty_asterisk_params*	0.072396
F15	*domain_in_ip*	0.003394	F45	*qty_dot_params*	0.071897
F16	*qty_at_domain*	0.002991	F46	*tld_present_params*	0.071747
F17	*qty_questionmark_domain*	0.001893	F47	*qty_at_params*	0.071641
F18	*qty_exclamation_domain*	0.000702	F48	*qty_params*	0.071593
F19	*qty_and_domain*	0.000617	F49	*qty_comma_params*	0.071518
F20	*qty_equal_domain*	0.000493	F50	*qty_and_params*	0.07144
F21	*directory_length*	0.384384	F51	*time_domain_activation*	0.216502
F22	*qty_slash_directory*	0.37632	F52	*asn_ip*	0.163469
F23	*qty_dot_directory*	0.353212	F53	*ttl_hostname*	0.093621
F24	*qty_hyphen_directory*	0.344885	F54	*time_domain_expiration*	0.053202
F25	*qty_dollar_directory*	0.33543	F55	*domain_spf*	0.018492
F26	*qty_at_directory*	0.335368	F56	*qty_nameservers*	0.015278
F27	*qty_space_directory*	0.335083	F57	*qty_redirects*	0.014389
F28	*qty_and_directory*	0.334989	F58	*qty_ip_resolved*	0.013776
F29	*qty_underline_directory*	0.334768	F59	*qty_mx_servers*	0.013495
F30	*qty_tilde_directory*	0.334662	F60	*time_response*	0.008607

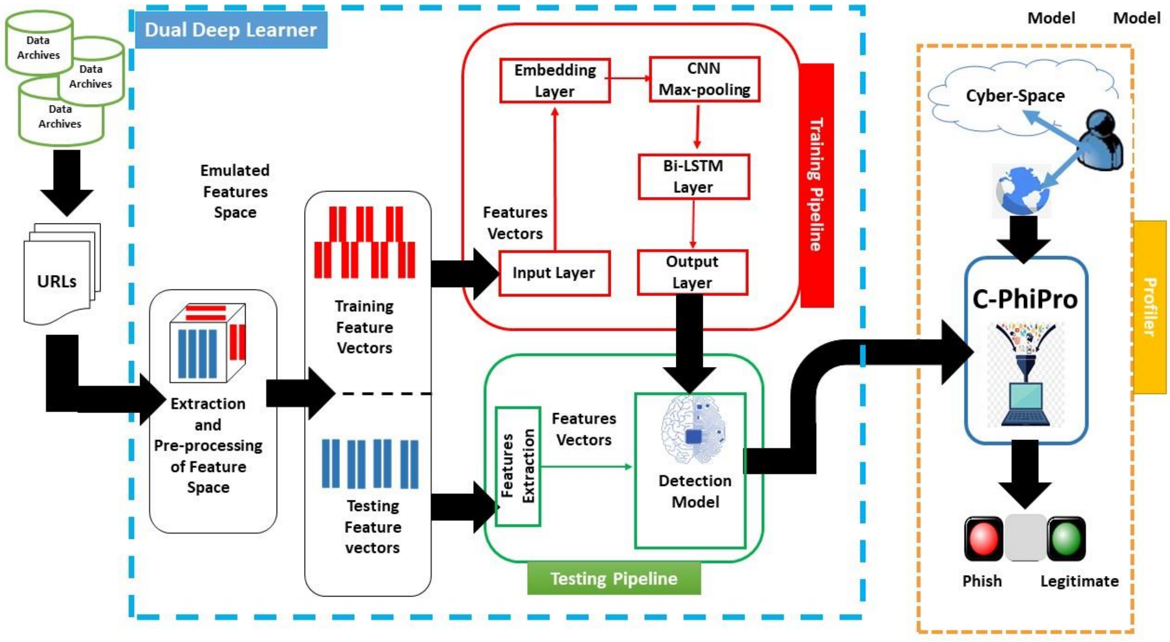


Figure 12. Framework of dual deep learning predictive model

4. EXPERIMENTS AND DISCUSSION

Experiments, datasets, and evaluations are described and discussed systematically in this section. The dataset has been built, then select the most potential features based on the IG value. Finally, proposed a hybrid deep learning-based classification model that hybridizes convolution neural network and bi-directional long short-term memory (CNN-BiLSTM).

4.1. The dataset collection

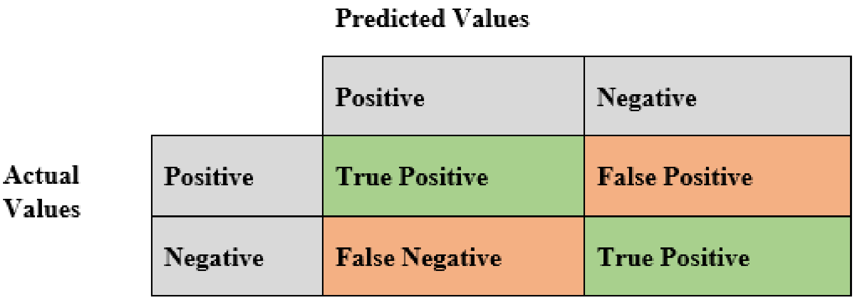
For an effective phishing predictive model, training and testing data must own item quality, source reliability, and class abundance greatly affect detection performance [37], [38]. Thus, we emulated a dataset of phish and legitimate URLs that were collected from various archives of phishing and legitimate URLs. Thus, the emulated dataset consisted of 14,000 valid sophisticated and generic phish URLs as well as 28,074 valid legitimate URLs. The goal of the imbalanced dataset is to simulate a real-world situation in which there are more legitimate websites. The dataset contains 111 features are separated into six categories: characteristics that are dependent on the whole URL property (20 features), characteristics based on domain URL (21 features), characteristics based on the directory of the URL (18 features), characteristics based on the URL file name (18 features), characteristics based on URL parameters (20 features) and characteristics based on resolving URL and external services (14 features). The dataset was collected from PhishTank and OpenPhish data sources for phishing sites, and domkop data sources for legitimated websites during the end of November 2021. The description of emulating dataset is present in Table 2.

Table 2. Description of emulating dataset [39]–[41]

Source	Number of Phishes	Number of Legitimate	Type
Phish-Tank	13,000	Zero	Phish Archive
Open-Fish	1,000	Zero	Phish Archive
domcop	Zero	28074	Legitimate Archive

4.2. Performance evaluation measurements

To estimate the effectiveness of the proposed dual deep learning predictive model, the standard confusion matrix was employed to infer the values true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) listed (FN) by counting its rows, columns, and diagonals as shown in Figure 13. Overall, TP and TN can reveal the correctly classified negative and positive instances. FP and FN can infer the incorrectly classified positive and negative instances, respectively. Accordingly, detection accuracy, precision, recall, and f-measure described in Table 3, can be computed as the proportion of the correctly identified URLs to the total number of examined URLs; the rate of valid positive URLs to the anticipated positive URLs; the rate of valid positive URLs regarding all positive URLs; and the harmonically computed rate of both accuracy and recall as presented in Table 3.



		Predicted Values	
		Positive	Negative
Actual Values	Positive	True Positive	False Positive
	Negative	False Negative	True Positive

Figure 13. Confusion matrix

Table 3. Description of performance evaluation measurements

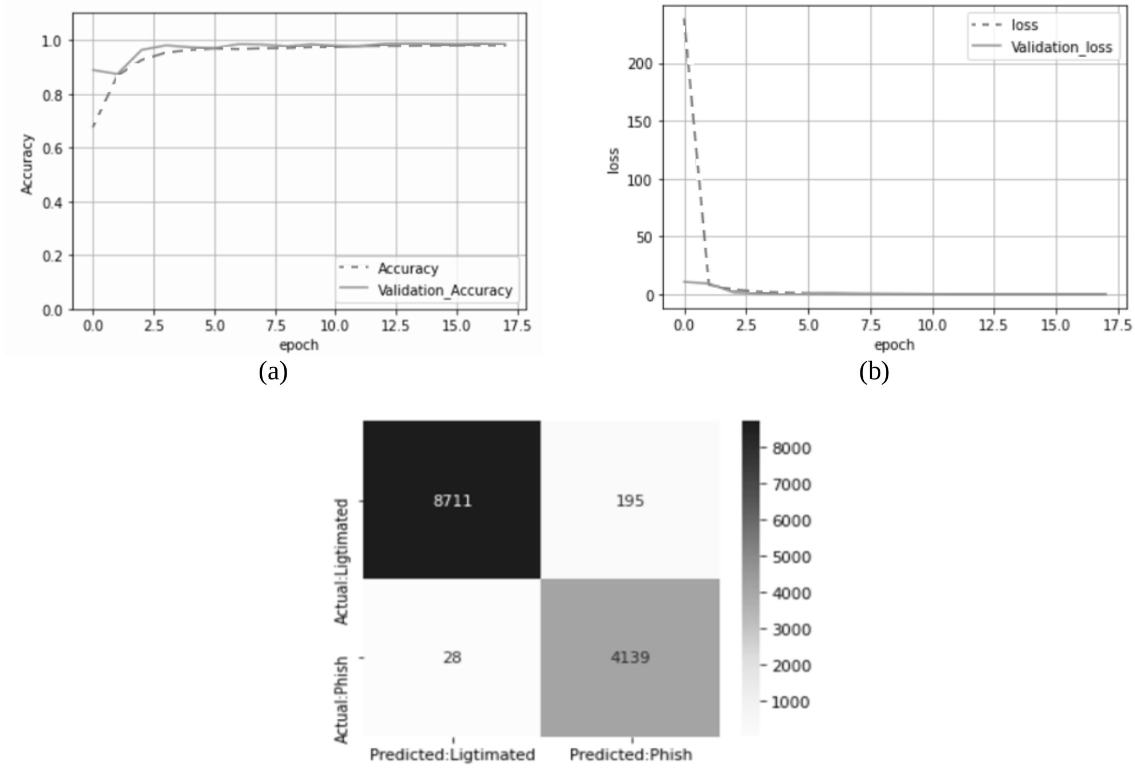
Measurement	Formula
Accuracy	$Accuracy\ =\ (TP\ +\ TN)/(TP\ +\ TN + \ FN + \ FP\ )$
Precision	$P\ = TP/(TP\ +\ FP\ )$
Recall	$R\ = TP/(TP\ +\ FN\ )$
F-measure	$F =\ (2\ *\ P\ * R)/(P + R)$

5. RESULTS AND DISCUSSION

Altogether, CNN, LSTM, BiLSTM, and the proposed dual predictive model of (CNN-BiLSTM) have been carried out via Scikit Learn Library in Python. All examined models have been trained across (75%) of the emulating dataset and tested on (25%) of the emulating datasets respectively. The performance of the CNN 1D model is shown in Figure 14, the training accuracy of the CNN 1D model shown in Figure 14(a) and training loss of the CNN 1D model is shown in Figure 14(b), and Figure 14(c) shows the confusion matrix of the CNN model. The convolution layers with 20, and the kernel size is 2. The optimizer function was decided to be adaptive moment estimation (Adam) that is one of the most used optimization techniques for estimating large-scale neural network parameters, it works best with extremely large amounts of data when the gradients are maintained “tighter” during numerous learning iterations. Rectified linear unit (ReLU) was employed as the hidden layers’ activation function because it can 55 disrupt linearity between layers and prevent gradients from becoming unsaturated, while the sigmoid function was used for the output layer. The model has trained with 18 epochs.

Then, the same anticipating feature space was used to train BiLSTM deep learning algorithm with 8 epochs, Adam optimizer has used with ReLU activation function. The model consists of three layers, the first of which has seven cells and the second of which has three cells, and last layer has 1 cell. Finally, the sigmoid activation function has used. Figure 15 shows the Performance outcomes of the BiLSTM model, the training accuracy ratio in Figure 15(a), the loss and confusion matrix in Figures 15(b) and 15(c) respectively.

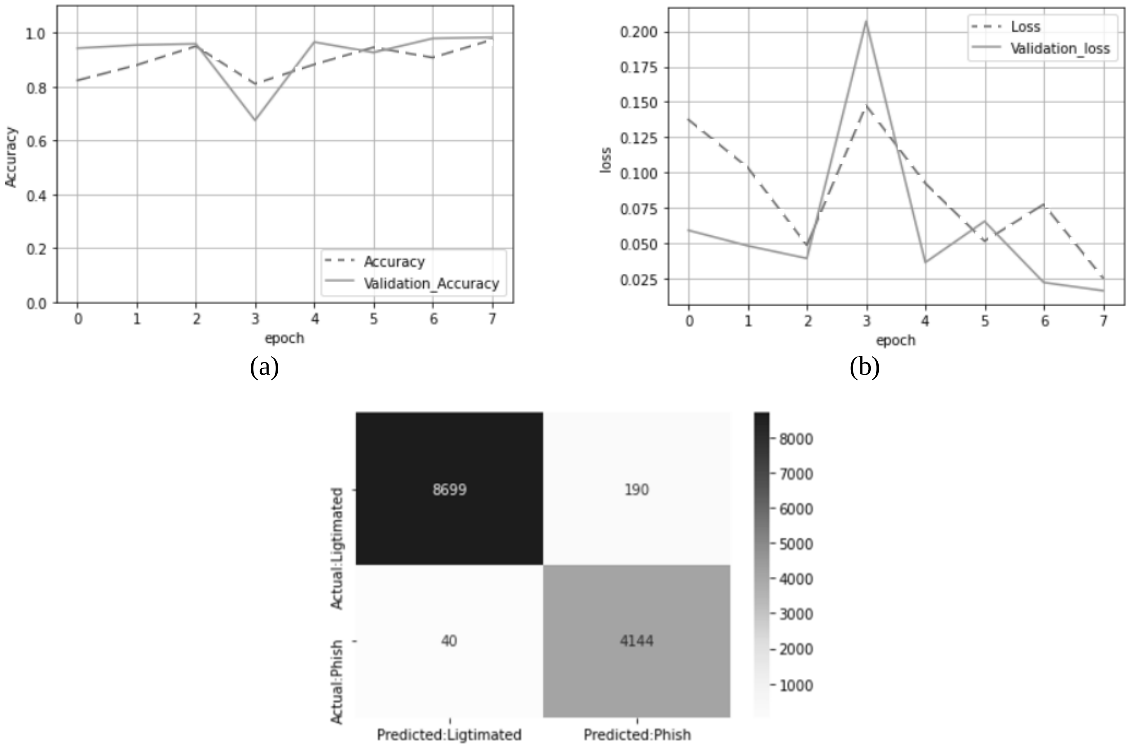
Our developed dual deep learning algorithm of CNN+BiLSTM outperformed CNN, and BiLSTM because the neural network model was influenced by the size of the data set as well as other factors such as the number of hidden layers, activations employed, and drop out to avoid overfitting. As such, the Input URL is reshaped to be input to the CNN model, in this layer generates a tensor of outputs by convolving the layer input with the convolution kernel over a single spatial (or temporal) dimension. Since the number of features is 60, the input shape of the first convolution layer is (60,1). The convolution layers with 20, and the kernel size is 2. ReLU is the activation function used. The output of the CNN layer passed through the max pooling layer with a size of (2,1). Then the BiLSTM layer had 4 neurons followed by a flattened layer. A fully connected layer was used to be the last layer with a sigmoid activation function. The Adam optimizer, and Cross entropy loss function were used, as it proves to be the best for binary classification problems. **Error! Reference source not found.** shows the performance results of the proposed model CNN-BiLSTM with the training accuracy, and loss in Figure 16(a), and Figure 16(b) respectively, and Figure 16(c) shows the confusion matrix of the proposed model.



(a) (b)

(c)

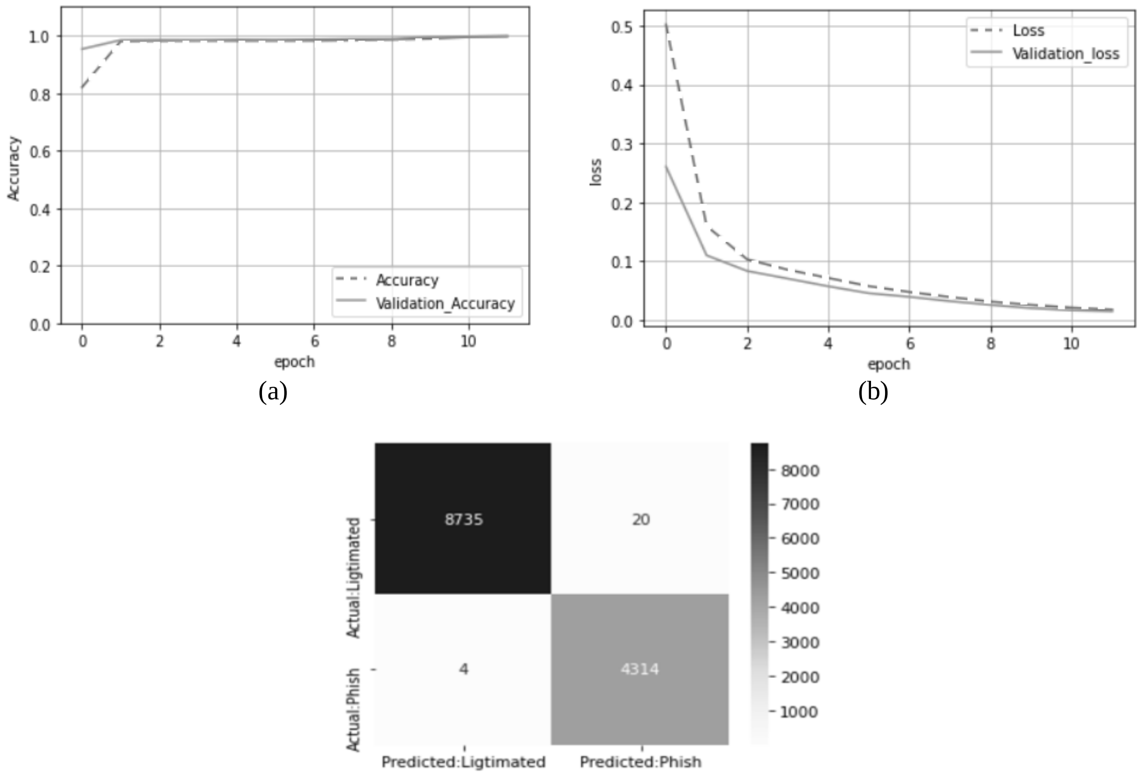
Figure 14. Performance outcomes of CNN 1D model (a) training accuracy of CNN, (b) training loss of CNN, and (c) confusion matrix of CNN



(a) (b)

(c)

Figure 15. Performance outcomes of BiLSTM model, (a) training accuracy of BiLSTM, (b) training loss of BiLSTM, and (c) confusion matrix of BiLSTM



(a) (b)

(c)

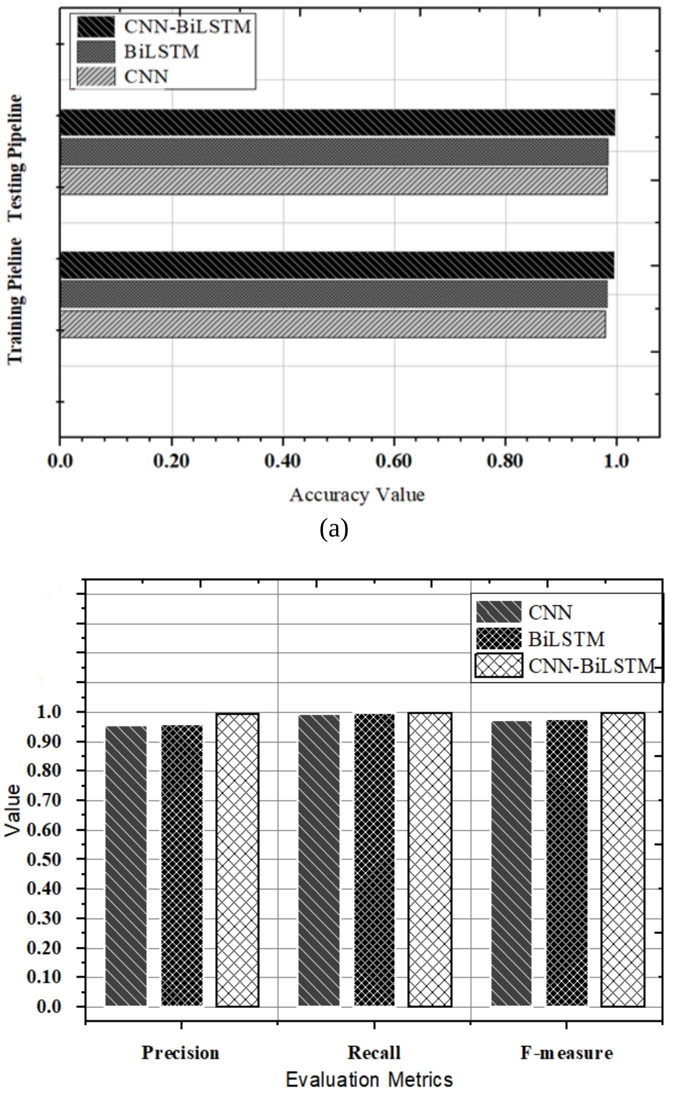
Figure 16. Performance outcomes of CNN-BiLSTM model, (a) training accuracy of CNN-BiLSTM, (b) training loss of CNN-BiLSTM, and (c) confusion matrix of CNN-BiLSTM

Due to harmonizing the mean of models' precision and recall together, the proposed dual predictive model of (CNN-BiLSTM) achieved the greatest F-measure of 98.88%, accuracy of 99.27%, Recall of 98.85% rather than CNN, and BiLSTM individually, as shown in **Error! Reference source not found.** and in Table 4. Figure 17(a) shows the comparison training, and testing accuracy among the CCN, BILSTM, and CNN-BILSTM.

Figure 17(b) compares recall, precision, and F-measure performance. Thus, the experimental results revealed that the three different deep learning-based predictive models varied in their performance and performance overhead. These experimental outputs raised the following question: "What factors should be attained to provide the highest detection accuracy?"

It can be observed from **Error! Reference source not found.** that the numbers of filters in CNN-BiLSTM compared to CNN and BiLSTM models, improved the performance outcomes of phishing prediction. Indeed, the increasing number of filters produced a higher training accuracy with a lower training loss. Correspondingly, the obtained F-measure was (98.88), and testing accuracy was 99.27 which inferred the highest rates. However, increasing the number of features rather they were significantly informative ones, would require leveraging more filters that in turn might maximize the complexity of the detection model and the number of hyper-parameters to train. Adding more filters (i.e., more layers) was unlikely to improve the training accuracy and minimize training loss. Furthermore, the training loss might not match up the increasing number of epochs closely, which would reflect how the detection model was over-fitted across imbalanced instances in the training dataset. That would maximize the training time and hence no outperformance can be obtained. On the other hand, the variance in filters' length across the examined deep learning predictive models causes varied performance from low, moderate, and mild. Amongst, was the highest F-measure (98.88) of the proposed predictive model (CNN-BiLSTM) in comparison to a filter length of 4, 6, and 8, respectively.

That inferred the outperformance of the deep learning predictive model depending on how to customary cross-combine more than deep learning algorithms at high detection accuracy versus the cyber-data stream [7]. Moreover, adapting the prediction of the new URL features of phishing in a real-time environment [6]. Contrarily, a solo deep learning predictive model (not a cross-combination or blended) might output overfitting versus new phishing features by comparing the rates of false detections obtained via the training pipeline and testing pipeline along with the number of epochs. Thus, the best improvement to margin the prediction setting between under-fitting and overfitting can be obtained by the early stop of running epochs (no more than ten via CNN-BiLSTM) [16], [27], [42]. In addition, fine-tuning neurons in the running layers could significantly affect deep learning prediction accuracy and minimize the elapsed time (time consumption by training and testing pipelines). Hence, the dual deep learning predictive model like that of CNN-BiLSTM architecture could emulate the training imbalanced cyber-data stream into an informative feature space without compromising its effectiveness and efficiency in a sandboxing context [42], [43]. According to the results shown in Table 5, our technique offers comparable accuracy, recall, and precision to the current state-of-the-art models. To identify phishing websites, some related works utilized various sets of generic phishing features, however, their suggestions did not make a significant contribution to innovative phishing prediction.



(a)

(b)

Figure 17. Comparative performance evaluation (a) training and testing accuracies of comparable classifiers and (b) precision, recall, and F-measure of comparable classifiers

Table 4. Comparative among the proposed models

Model	Training accuracy	Testing accuracy	Precision	f1-score	Recall
CNN 1D	0.9789	0.9813	0.9490	0.9710	0.9941

BI-LSTM	0.9831	0.9855	0.9584	0.9777	0.9978
CNN-BI-LSTM	0.9913	0.9927	0.9792	0.98881	0.9985

Table 5. Comparative review of the related works

Related Work	Deep Learning Algorithm	Method	Data Source	Performance evaluation
Kumar *et al.* 2021 [16]	CNN	It is improved with Swarm Intelligence Binary Bat Algorithm to learn URL features	KAGGLE	Accuracy: 94.8% 0.2 % False detections

Xiao *et al.* 2021 [14]	CNN	It is a multi-headed and self-attentional algorithm assisted by the Generative Adversarial Network (GAN) and URL features	Five thousand legitimate websites [36], Phish Tank	Accuracy: 97.20%
				Detection Time: 174 as
				Recall= 95.60%
				Precision = 98.76%
				F1= 97.15%

Jiang *et al.* 2018 [44]	CNN	Classifies URL at character-level by the features of URL length, URL separators, number of dots, and other categorical and lexical features	Google, DMOZ Phish Tank, Virus Total	Miss Rates: 4/1000 URLs
Ozcan *et al.*, 2021 [20]	DNN+ LSTM, DNN + Bi-LSTM	It is important for NLP features and character embedding features classification	Ebbu2017	DL model Accuracy DNN+ LSTM 98.62% DNN+ BILSTM 98.79%
			Phish Tank	DNN+ LSTM 98.98% DNN+ BILSTM 99.21% DNN+ LSTM 98.62%
Somesha *et al.* 2020 [23]	DNN+CNN+LSTM	It uses information gain (IG) for features ranking and optimizing a set of URL obfuscation features, hyperlink-based features, and third party-based features.	Phish Tank, Alexa	Accuracy varied from: 99.57%, to 99.43%
Zhang and Li 2017 [45]	DBN	Borderline Smote-based classification across a set of URL features, Page contents, and Images	Phish Tank	Recall: 90.7% Precision: 96.5%
Huang *et al.* 2019 [46]	CNN+ RNN	It segments specific Viterbi and URL features for phishing classification	Phish Tank, Alexa, Open-Fish	Accuracy: 97.905% FPR= 0.020% Precision: 98.958%
Our Model	CNN-BILSTM	It uses information gain (IG) for features ranking and optimizing a set of URL obfuscation features, and third party-based features.	Phish Tank, Open-Fish DomCop	Accuracy: 99.27% F-measure: 98.88% Recall: 99.85%

Since we employed a tuned dataset by applying the IG algorithm with a sizable amount of data, our approach performed more accurately than the previous studies.

6. CONCLUSIONS AND FUTURE WORK

This work investigates how the mutual phishing feature space, and a dual deep learning-based predictive model can be satisfactorily blended to improve deep learning in phishing mitigation. To do so, 111 phishing features were assessed and filtered for the 60 most mutual features by information gain. Then, the 60 mutual phishing features were employed to construct the required feature space. Based on this assessment, a dual architecture of deep learning predictive model was proposed to extract these features from fetched URLs. For a more elastic and manageable evaluation, the proposed predictive model was evaluated with an emulated dataset consisting of 14,000 phishing URLs and 28074 legitimate URLs. The performance outcomes indicate an agreeable Elapsed time of less than 90 ms, accuracy is 99.27%, and F-measure of (98.88%). As such, our work presents a superior predictive model when compared with those of the literature.

In the future, the predictive model will apply to a desktop anti-phishing tool that can examine and detect any URL inserted into the tool, then notify the user whether the fetched website is phishing or legitimate. Also, we will consider implementing the proposed predictive model for proper phishing profiler and investigating the appropriateness of that phishing profiler versus the emerging IoT phishing attacks. manageable evaluation, the proposed predictive model was evaluated with an emulated dataset consisting of 14,000 phishing URLs and 28074 legitimate URLs. The performance outcomes indicate an agreeable Elapsed time of less than 90 ms and a F-measure of (98.88%). As such, our work presents a superior predictive model when compared with those of the literature.

REFERENCES

[1] H. Zuhair and A. Selamat, "Phishing classification models: issues and perspectives," *International Journal of Digital Enterprise Technology*, vol. 1, no. 3, 2019, doi: 10.1504/IJDET.2019.10019065.

[2] H. Zuhair, A. Selamat, and M. Salleh, "Feature selection for phishing detection: a review of research," *International Journal of Intelligent Systems Technologies and Applications*, vol. 15, no. 2, pp. 147–162, 2016, doi: 10.1504/IJISTA.2016.076495.

[3] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro, and E. Alegre, "Impact of current phishing strategies in machine learning models for phishing detection," in *Advances in Intelligent Systems and Computing*, vol. 1267, 2021, pp. 87–96.

[4] R. Mu and X. Zeng, "A review of deep learning research," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 4, pp. 1738–1764, Apr. 2019, doi: 10.3837/tiis.2019.04.001.

[5] N. Q. Do, A. Selamat, O. Krejcar, T. Yokoi, and H. Fujita, "Phishing webpage classification via deep learning-based algorithms: an empirical study," *Applied Sciences*, vol. 11, no. 19, Oct. 2021, doi: 10.3390/app11199210.

[6] H. Salah and H. Zuhair, "Catching a Phish: frontiers of deep learning-based anticipating detection engines BT-advances on intelligent informatics and computing," in *International Conference of Reliable Information and Communication Technology*, 2022, pp. 483–497.

[7] M. Vijayalakshmi, S. Mercy Shalinie, M. H. Yang, and R. M. U., "Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions," *IET Networks*, vol. 9, no. 5, pp. 235–246, Sep. 2020, doi: 10.1049/iet-net.2020.0078.

[8] A. Hannousse and S. Yahiouche, "Towards benchmark datasets for machine learning based website phishing detection: An experimental study," *Engineering Applications of Artificial Intelligence*, vol. 104, 2021, doi: 10.1016/j.engappai.2021.104347.

[9] G. Petrič and K. Roer, "The impact of formal and informal organizational norms on susceptibility to phishing: Combining survey and field experiment data," *Telematics and Informatics*, vol. 67, Feb. 2022, doi: 10.1016/j.tele.2021.101766.

[10] 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, doi: 10.1109/ACCESS.2022.3151903.

[11] L. Tang and Q. H. Mahmoud, "A deep learning-based framework for phishing website detection," *IEEE Access*, vol. 10, pp. 1509–1521, 2022, doi: 10.1109/ACCESS.2021.3137636.

[12] S. Singh, M. P. Singh, and R. Pandey, "Phishing detection from URLs using deep learning approach," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, Oct. 2020, pp. 1–4, doi: 10.1109/ICCCS49678.2020.9277459.

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

[14] X. Xiao *et al.*, "Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets," *Computers and Security*, vol. 108, Sep. 2021, doi: 10.1016/j.cose.2021.102372.

[15] S. Y. Yerima and M. K. Alzaylaee, "High accuracy phishing detection based on convolutional neural networks," in *2020 3rd International Conference on Computer Applications and Information Security (ICCAIS)*, Mar. 2020, pp. 1–6, doi: 10.1109/ICCAIS48893.2020.9096869.

[16] P. P. Kumar, T. Jaya, and V. Rajendran, "SI-BBA-A novel phishing website detection based on Swarm intelligence with deep learning," *Materials Today: Proceedings*, Jul. 2021, doi: 10.1016/j.matpr.2021.07.178.

[17] A. Al-Alyan and S. Al-Ahmadi, "Robust URL phishing detection based on deep learning," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 7, pp. 2752–2768, Jul. 2020, doi: 10.3837/tiis.2020.07.001.

[18] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, no. Ii, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.

[19] Y. V. M, B. Janet, and S. Reddy, "Anti-phishing system using LSTM and CNN," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, Nov. 2020, pp. 1–5, doi: 10.1109/INOCON50539.2020.9298298.

[20] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN-LSTM model for detecting phishing URLs," *Neural Computing and Applications*, vol. 0123456789, Aug. 2021, doi: 10.1007/s00521-021-06401-z.

[21] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Computer Networks*, vol. 178, Sep. 2020, doi: 10.1016/j.comnet.2020.107275.

[22] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *Journal of Enterprise Information Management*, 2020, doi: 10.1108/JEIM-01-2020-0036.

[23] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, "Efficient deep learning techniques for the detection of phishing websites," *Sādhanā*, vol. 45, no. 1, Dec. 2020, doi: 10.1007/s12046-020-01392-4.

[24] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, Feb. 2022, doi: 10.1016/j.jksuci.2019.12.005.

[25] A. Aljofey *et al.*, "An effective detection approach for phishing websites using URL and HTML features," *Scientific Reports*, vol. 12, no. 1, May 2022, doi: 10.1038/s41598-022-10841-5.

[26] T. Farral, "Nation-state attacks: practical defences against advanced adversaries," *Network Security*, no. 9, pp. 5–7, Sep. 2017, doi: 10.1016/S1353-4858(17)30111-3.

[27] S. Maurya and A. Jain, "Deep learning to combat phishing," *Journal of Statistics and Management Systems*, vol. 23, no. 6, pp. 945–957, Aug. 2020, doi: 10.1080/09720510.2020.1799496.

[28] E. O. Omuya, G. O. Okeyo, and M. W. Kimwele, "Feature selection for classification using principal component analysis and information gain," *Expert Systems with Applications*, vol. 174, 2021.

[29] A. Zamir *et al.*, "Phishing web site detection using diverse machine learning algorithms," *The Electronic Library*, vol. 38, no. 1, pp. 65–80, Mar. 2020, doi: 10.1108/EL-05-2019-0118.

[30] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, Jan. 2019, doi: 10.1016/j.neucom.2018.09.038.

[31] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[32] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[33] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[34] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi: 10.1140/epjst/e2019-900046-x.

[35] M. Kowsher *et al.*, "LSTM-ANN and BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy," *Procedia Computer Science*, vol. 193, pp. 131–140, 2021, doi: 10.1016/j.procs.2021.10.013.

[36] M. M. Rahman, Y. Watanobe, and K. Nakamura, "A bidirectional LSTM language model for code evaluation and repair," *Symmetry*, vol. 13, no. 2, Feb. 2021, doi: 10.3390/sym13020247.

[37] V. Zeng, S. Baki, A. El Aassal, R. Verma, L. F. T. De Moraes, and A. Das, "Diverse datasets and a customizable benchmarking framework for phishing," in *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, Mar. 2020, pp. 35–41, doi: 10.1145/3375708.3380313.

[38] R. M. Verma, V. Zeng, and H. Faridi, "Data quality for security challenges: case studies of phishing, malware and intrusion detection datasets," *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2605–2607, 2019.

[39] "Top 10 million websites," *DomCop*. https://www.domcop.com/top-10-million-websites (accessed Nov. 06, 2021).

[40] "Join the fight against phishing," *PhishTank*. http://data.phishtank.com/data/online-valid.csv (accessed Nov. 20, 2021).

[41] "Timely. Accurate. Relevant Phishing Intelligence," *OpenPhish*. https://openphish.com/feed.txt (accessed Nov. 06, 2021).

[42] P. Anki and A. Bustamam, "Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 23, no. 1, pp. 197–205, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp197-205.

[43] S. Liu, P. Feng, S. Wang, K. Sun, and J. Cao, "Enhancing malware analysis sandboxes with emulated user behavior," *Computers and Security*, vol. 115, Apr. 2022, doi: 10.1016/j.cose.2022.102613.

[44] J. Jiang *et al.*, "A deep learning based online malicious URL and DNS detection scheme," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 238, Springer International Publishing, 2018, pp. 438–448.

[45] J. Zhang and X. Li, "Phishing detection method based on borderline-smote deep belief network," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS 2016)*, 2017, pp. 45–53.

[46] Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL detection via CNN and attention-based hierarchical RNN," in *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, Aug. 2019, pp. 112–119, doi: 10.1109/TrustCom/BigDataSE.2019.00024.

BIOGRAPHIES OF AUTORS

Hamzah Salah is a postgraduate student in the department of networks engineering, college of information engineering, at Al-Nahrain University, Baghdad, Iraq. He is currently studying M.Sc. under the supervision of Asst. Prof. Dr. Hiba Zuhair. Prior to this, he received the BSc. in computing and informatics from Al-Khawarzmi College of Engineering, University of Baghdad in Baghdad, Iraq in 2016. He can be contacted at email: ha.salim94@gmail.com.

Hiba Zuhair received the B.S. degree in computer science from Al-Nahrain University, Baghdad, Iraq, in 2000. Then she was awarded M.Sc. degree in information security mechanical engineering from Al-Nahrain University, Baghdad, Iraq, in 2002. Then, she was awarded the Ph.D. degree in cyber-security with the high distinction of the best postgraduate student from school of computing, faculty of engineering, Universiti Teknologi Malaysia, Johor Bahru, and Johor, Malaysia in 2017. From 2005 to 2013, she was a Research Assistant and senior lecturer at college of information engineering, Al-Nahrain University in Baghdad, Iraq. Then, she was honored Assistant Professor in 2017. From 2013 till now she has worked in the development of cyber-security for critical cyber-physical infrastructures. Her research of interest and publication are in the fields of intrusion detection systems, phishing, and ransomware attacks in android and IoT systems as well as deep learning and machine learning techniques. She is also currently serving on the Editorial Board of Open Computer Science (Gemb/DeGruyter/Poland/Clarivate). Furthermore, she was honored to be the reviewer in Journal of Information security and its Applications JISAS as well as Computers and Security (Elsevier, The Netherlands), that are the best of Elsevier-Scopus indexed journals in information security, computing, and networking fields. She can be contacted at email: hiba.zuhair.pcs2013@nahrainunive.edu.iq.