electronics

MDPI

Article

Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction

Muna Elsadig [1], Ashraf Osman Ibrahim [2,*], Shakila Basheer [1], Manal Abdullah Alohali [1], Sara Alshunaifi [1], Haya Alqahtani [1], Nihal Alharbi [1] and Wamda Nagmeldin [3]

[1] Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2] Faculty of Computing and Informatics, University of Malaysia Sabah, Kota Kinabalu 88400, Malaysia
[3] Department of information systems, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
* Correspondence: ashrafosman@ums.edu.my

Abstract: Recently, phishing attacks have been a crucial threat to cyberspace security. Phishing is a form of fraud that attracts people and businesses to access malicious uniform resource locators (URLs) and submit their sensitive information such as passwords, credit card ids, and personal information. Enormous intelligent attacks are launched dynamically with the aim of tricking users into thinking they are accessing a reliable website or online application to acquire account information. Researchers in cyberspace are motivated to create intelligent models and offer secure services on the web as phishing grows more intelligent and malicious every day. In this paper, a novel URL phishing detection technique based on BERT feature extraction and a deep learning method is introduced. BERT was used to extract the URLs' text from the Phishing Site Predict dataset. Then, the natural language processing (NLP) algorithm was applied to the unique data column and extracted a huge number of useful data features in terms of meaningful text information. Next, a deep convolutional neural network method was utilised to detect phishing URLs. It was used to constitute words or n-grams in order to extract higher-level features. Then, the data were classified into legitimate and phishing URLs. To evaluate the proposed method, a famous public phishing website URLs dataset was used, with a total of 549,346 entries. However, three scenarios were developed to compare the outcomes of the proposed method by using similar datasets. The feature extraction process depends on natural language processing techniques. The experiments showed that the proposed method had achieved 96.66% accuracy in the results, and then the obtained results were compared to other literature review works. The results showed that the proposed method was efficient and valid in detecting phishing websites' URLs.

Keywords: phishing detection; deep neural network; nature language processing; website URL classification

Citation: Elsadig, M.; Ibrahim, A.O.; Basheer, S.; Alohali, M.A.; Alshunaifi, S.; Alqahtani, H.; Alharbi, N.; Nagmeldin, W. Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction. *Electronics* **2022**, *11*, 3647. https://doi.org/10.3390/electronics11223647

Academic Editor: Krzysztof Szczypiorski

Received: 4 September 2022
Accepted: 31 October 2022
Published: 8 November 2022

1. Introduction

Recently, people's and governmental organizations' daily usage of technology, especially the internet, has made life easier and greatly facilitated commercial services and transactions. Banking and other electronic services rely heavily on the internet in providing their commercial services. The number of web applications visited by users daily is huge [1]. During the COVID-19 pandemic, people depended more on the internet to purchase their daily household needs, e.g., food, drinks, and clothes, and this is known as an online purchase. This created a large market for merchants, restaurants, banks, delivery services, health care providers, government agencies, and others [2,3]. However, this whetted the appetite of many fraudsters who were looking for important and valuable user data, whereby many electronic fraud methods fall under electronic crimes [4]. Figure 1 shows a report of the financial phishing attacks allocated globally during 2021, which shows the affected business sectors [5]. One of the types of cyber-attacks is phishing, which is the proposed

concern of this paper. A phishing incident activity is a form of cybersecurity threat that focuses on users specifically through text, e-mail, or direct messages. During one of these fraud operations, the attacker would present himself/herself as a confidential contact to steal the user's sensitive information, such as login information (password and username), credit card information, account numbers, and all transaction information [6,7]. Phishing is a process carried out by professionals in the computer science field, who are very familiar with the nature of the human psyche, which is largely controlled by greed, naivety, and the love of exploration. The field of science that allows using of social skills to lure people to expose their sensitive and valuable information is called social engineering [8].

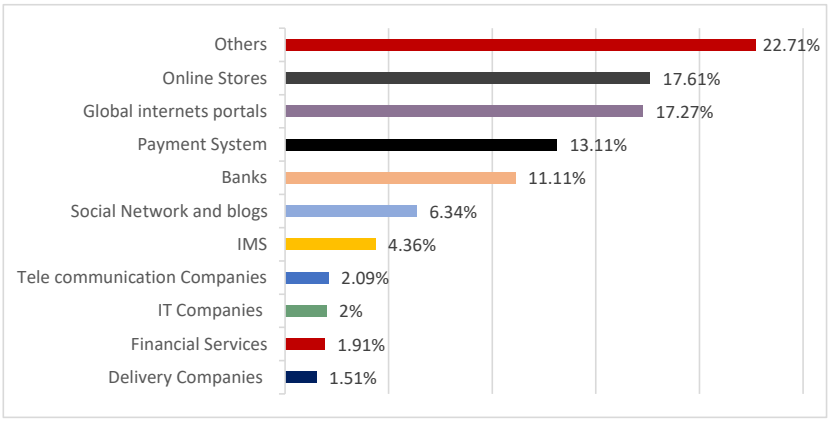


Figure 1. Percentage of phishing attack organisations.

The attacker would study the victim closely, including his general information, such as name, job, etc., and then more private information, such as hobbies, places of leisure, bank balance, e-mail address, mobile phone number, or webpage on social networking sites such as Facebook and Twitter [9]. Phishing cannot be absolutely and easily eliminated in a day, and due to its ubiquity, it will not disappear in the near future. Therefore, to resolve website phishing, comprehensive research is needed.

In light of the importance of the phishing website problem, this motivates the researcher to investigate and use complex machine learning algorithms to detect phishing websites. The proposed solution is an application that requests a website's URL from the user, extracts the website's features, and then classifies the website by using deep machine learning [1,10].

Detecting phishing websites is a kind of problem that requires an intelligence technique whereby these phishing websites' features need to be extracted and then classified as to whether a website is legitimate or otherwise. Since this deception technique follows social engineering and contains detailed complexities, traditional artificial intelligence algorithms fail to process and detect it accurately and efficiently. Building a smart system to detect phishing websites is very important and essential and does not bear even a small error rate with a high accuracy rate, especially in the process of classifying an illegal website into a legitimate one [11,12].

Recently, many techniques have been developed to detect different types of phishing by using machine learning (ML) and deep machine learning (DML). The accuracy and precision of detection have improved. In this research, a deep machine learning technique has been developed to detect website URL phishing based on natural language processing feature extraction. The algorithm was tested by using a dataset after the pre-processing technique. Moreover, feature extraction was applied to specify features needed to discriminate between legitimate URLs and illegitimate URLs. The feature extraction was applied by using natural language processing that implemented "The Bidirectional Encoder Representations from Transformers" (BERT) model. BERT improved the earlier-mentioned unidirectionality restriction [13] by applying a "masked language model" (MLM) pre-training objective. Collobert and Weston [14] are amongst the first researchers to apply CNN-based frameworks to NLP tasks.

This work aimed to develop an intelligent convolutional neural network learning algorithm based on BERT features extraction for cyber–phishing URL website detection.

The proposed solution is an application that requests a website's URL from the user, extracts the website's features using BERT, and then classifies the website using deep machine learning. The idea of using BERT features extraction assists in extracting semantic characteristics of the data, and thus could improve overall performance.

The main contributions of this study were as follows:

1. Creation of an intelligent convolutional neural network learning method based on BERT features extraction to discover the appropriate features for cyber-phishing URL website detection.
2. It improved the accuracy of phishing website detection better than deep machine learning.
3. The proposed method was compared, examined, and evaluated as evident for its performance and effectiveness.
4. The study compared and evaluated the findings of the proposed method with other developed methods and with results from the literature that had used similar datasets.

The remainder of this paper is organised as follows: Section 2 describes the related work and provides an overview of existing work. Section 3 describes the materials and methods used in this study. Section 4 presents the proposed method, and the evaluation metrics are discussed in Section 5. Section 6 provides the results and discussion. Finally, Section 7 concludes the paper.

2. Related Works

Nowadays, deep machine learning (DML) has valuable characteristics for the current technologies used by society, such as e-commerce, online banking services, healthcare system, social networking, web searching, content management, and e-mails. Furthermore, machine learning (ML) and DML techniques are used for data analysis, business intelligence, image recognition, voice processing, and language processing. Moreover, features such as web searching are embedded in many consumers' smart products, such as smart TVs, smart phones, and smart cameras. In a previous study [10], they used the ML technique to detect phishing websites. They identified and evaluated powerful traditional machine learning (ML) solutions to phishing websites, such as random forests (RFs), ada boosting, support vector machines (SVMs), and Nave Bayes (NB). They concluded that not all phishing-related problems had been resolved by the ML technique. The research and the development of innovative approaches are ongoing as the attackers concoct smart and shrewd phishing ideas and develop new phishing methods each day. The researchers recommended that automated models should be planned based on ensemble learning and deep learning techniques in future work. Recently, many deep learning techniques have been used successfully in different application areas, such as medical [15], agriculture [16], industry [17], and engineering [18]. Therefore, several DMLs have been utilised to detect phishing websites' URLs, such as the convolutional neural network (CNN) and long short-term memory (LSTM) algorithms, whereby the accuracy obtained was 93.28% [19]. Mohamed et al. [20] developed an intelligent system that predicts phishing websites depending on the neural network technique, specifically, the self-structuring neural network. [11] generated three approaches by using different aspects for anti-phishing detection: firstly, the long short-term memory algorithm (LSTM), whereby the obtained accuracy was 98.67%; secondly, a deep-neural network (DNN) with an accuracy of 96.33%; and finally, the convolution-neural network (CNN), using 10 features with an accuracy of 97.23%. The anticipated model achieved an accuracy of 98.67% for LSTM, 96.33% for DNN, and 97.23% for CNN. A new framework was introduced using deep machine learning for phishing detection based on a deep belief network [12], whereby the accuracy gained by this framework was at 89.6%. A robust URL phishing detection based on deep machine learning was introduced by [21]. They proposed a CNN that took the URL as the input, instead of utilising the pre-determined features, such as the length of the URL. For training and evaluation stages, they predisposed more than two million URLs in a massive URL phishing detection (MUPD) dataset. The CNN algorithm utilised had attained approximately 96% accuracy on the testing dataset.

In 2022, [22] proposed a multidimensional feature of a phishing detection approach that is maintained by a speedy detection technique employing deep machine learning. In this work, the URL features were extracted in two stages. In the first stage, the character sequence features of the specified URL were extracted and implemented for speedy classification by algorithms without the assistance of a third-party vendor with specific knowledge of phishing. In the second stage, an intelligent model to predict phishing incidents is based on the artificial neural network (ANN), mainly, the self-structuring neural networks. Moreover, the statistical URL features, webpage code features, and webpage text features were associated with the ANN network. Other researchers implemented a URL classification algorithm, whereby a new dataset was created. The experimental results were established by using the random forest algorithm with only NLP in IoT (Internet of Things) environments, resulting in the best performance of 99.57% accuracy rate for the detection of phishing URLs [23]. In a Google Scholar search, there are more than 87 results of research that has been conducted using deep learning algorithms for phishing URL detection from 2018 to 2022 [24]. From the literature review, it can be concluded that using deep machine learning algorithms has improved URL phishing detection. Table 1 shows a comparison study of the most effective deep learning studies for phishing detection, whereby different techniques are implemented with different datasets.

Table 1. Recent effective deep learning studies in URL phishing detection.

Study	Best Algorithm	Result	Datasets	Dataset Size
Al-Ahmadi [25]	Multi-layer-deep neural network	95.73%	Kaagle dataset	549,346
Sahingoz et al. [26]	Random Forest	97.98%	Ebbu2017 Phishing Dataset	73,575
Zouina et al. [27]	SVM with Gaussian kernel	95.80%	PhishTank	2000
Moghimi et al. [28]	SVM and decision tree	98.86%	Yahoo directory service, PhishTank	2134
Ferreira et al. [29]	ANN-MLP	98.23%	Phishing Websites Dataset of the University of California's Machine Learning and Intelligent Systems Learning Center	3000
Peng Yang et al. [30]	CNN-LSTM, DCDA, Multidimensional Features	98.61%	Dmoztools.net, PhishTank	1,965,944
M Somesha et al. [31]	CNN-LSTM and DNN	99.52%	PhishTank and Alexa database	3526

3. Materials and Methods

This section highlights the materials and methods used in this study to achieve the objectives.

3.1. Dataset Description and Representation

The dataset was collected from the public and well-known repository Kaggle.com [32]. The dataset was called the phishing site predict dataset, and it contained 549,346 entries with two columns. The website links (URLs) are represented as the first attribute, while countries are represented as the second attribute. The website's labels were classified into good or bad. Table 2 denotes a small sample of the selected dataset:

Table 2. Dataset sample records.

ID	URL	Label
92669	www.doggie-school.com/~lmafamor/www.paypal.com.au/webscr.php (accessed on 27 April 2022)	Bad
66744	www.premiumcentral.us/~lmafamor/www.paypal.com.au/webscr.php (accessed on 27 April 2022)	Bad
78923	members.tripod.com/~mindcrime_2/freemud.html (accessed on 27 April 2022)	Good
45367	www.thesmileforsuccess.com/~lmafamor/www.paypal.com.au/webscr.php (accessed on 27 April 2022)	Bad
43256	www.performancepcplus.com/majormud/index.htm (accessed on 27 April 2022)	Good
78903	www.myred19.com/~lmafamor/www.paypal.com.au/webscr.php (accessed on 27 April 2022)	Bad
224568	www.performancepcplus.com/majormud/index.htm (accessed on 27 April 2022)	Good
278908	www.gameport.com/mudproducts/index.html (accessed on 27 April 2022)	Good

3.2. BERT

The bidirectional encoder representations from transformers (BERT) are a recent language representation model introduced by [33]. The BERT is used for natural language processing (NLP) that supports feature extraction. NLP is an area of computer science concerned with the capability of machines to comprehend text and spoken words in the same way that humans can. In NLP, computational linguistics—rule-based human language modelling—is integrated with statistical, machine learning, and deep learning models. The NLP algorithm was applied to the unique data column and extracted a huge number of useful data features. The BERT model is utilised to generate pre-trained deep bidirectional representations from unclassed text through mutually training on dual direction left and right context in all layers [34]. The pre-trained BERT model is fine-tuned by adding one output layer to generate a variety of models for a wide-ranging study, such as natural language processing tasks [33]. The BERT can improve the unidirectionality restriction. Therefore, BERT could succeed in learning contextual embeddings for words.

3.3. Deep Machine Learning

A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers between the input and the output layers. The neural network is a specimen of a non-linear prediction method that has been used in many fields, such as phishing, biological and document classification, image processing pattern, and speech and handwriting recognition [20]. DNN can model a very complex non-linear relationship between the input and the output layers [35]. A multilayer perceptron (MLP) has three or more layers. It uses a non-linear activation function (mainly a hyperbolic tangent or logistic function) that allows it to classify data that are not linearly separable. Every node in a layer connects to each node in the following layer, making the network fully connected [36]. The neural network′s weights are adjusted to make the MLP function. The backpropagation algorithm, a well-known algorithm, is used to carry out the procedure. The neural network is trained using a known dataset while continuously changing the weights until an acceptable error is attained (if possible). An unknown dataset can then be used to test the neural network. DNNs come in a wide variety of topologies or structures. Convolutional neural network (CNN) technology is employed in this study. One or more convolutional layers make up a convolutional neural network (CNN), and a pooling layer typically follows each convolutional layer. As desired by the neural network creator, these two layers can be repeated. The output of these repeated layers is then fed into a flattened layer, which can be followed by numerous fully connected layers until it reaches the output layer. To map the input to the convolution layer′s size, the convolution layer uses a so-called kernel [37]. The pooling layer uses a pooling method like max-pooling, min-pooling, or average-pooling to reduce the size of the feature map in the convolution layer.

3.4. URL Phishing Components

Regardless of the technology, URL phishing, as in other types of phishing techniques, generally consists of three basic components:

1. Medium of phishing: the most used medium is the internet,
2. Transmission vector of attack: a website as a medium for transmitting attacks between attacker and the victim,
3. Attack technical approaches: include social engineering and browser vulnerabilities, such as browser vulnerabilities, cloud environment, click jacking, etc.

Figure 2 describes the link joining of the three parts.

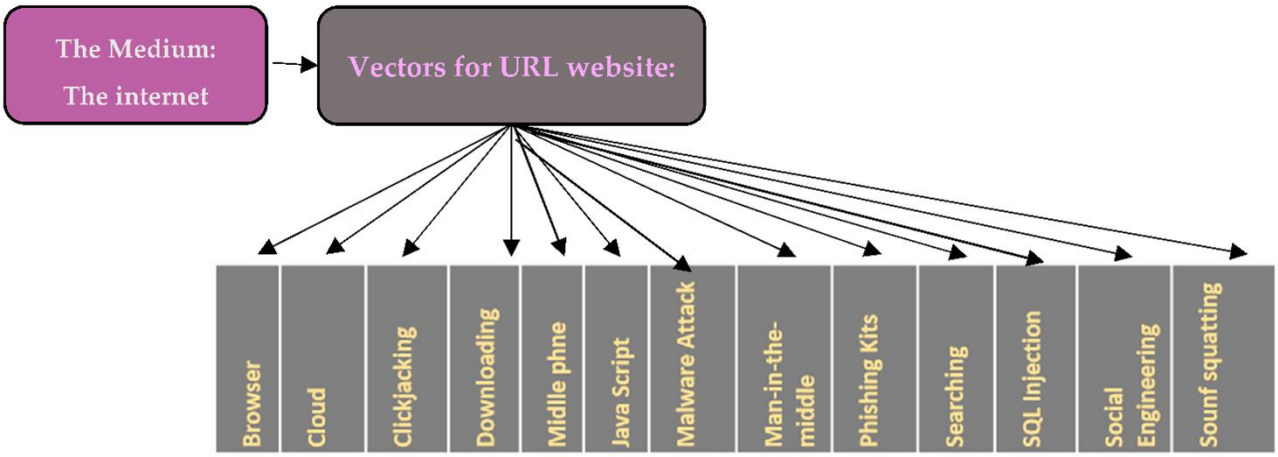


Figure 2. The join link between the URL phishing components.

4. Proposed Method

To achieve the goals of this study, a methodology of four stages was followed. Figure 3 shows the sequence of these stages and the block diagram for the proposed model is shown in Figure 4.

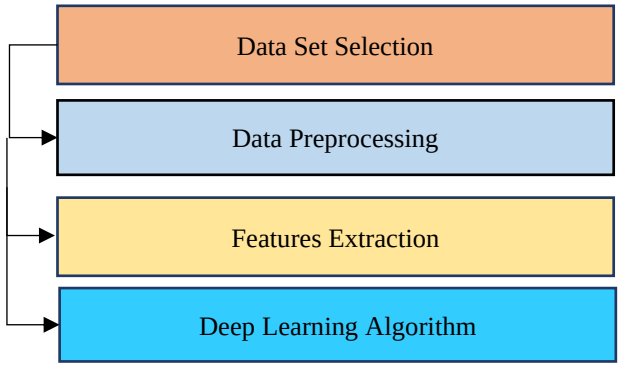


Figure 3. Stages of the proposed method.

Stage 1. URL phishing dataset

At this stage, the collected phishing site predict dataset was prepared. Table 2 shows the details of the data attributes (two columns). In addition, a sample of the data was represented in rows. The rows and columns of the table were as follows:

(1) The URL column: contained unique URLs.
(2) The Label column: contained the corresponding URL detection—good or bad (phishing).

Stage 2. Data preparation

Data pre-processing involves transforming raw data into well-formed datasets so that data mining analytics can be applied. Raw data are often incomplete and have inconsistent formatting. The adequacy or inadequacy of data preparation has a direct correlation with the success of any project that involves data analytics.

All the data pre-processing is implemented by using Python. The selected dataset pre-processing of this model is summarised as follows:

1. Read dataset.
2. Check for duplicated records and remove them. Initially, there were 549,346 records before removing the duplicated records, resulting in 472,272 records.
3. Check for the null column value and remove the corresponding record, whereby there were no null records found and the number of records remains the same, at 472,272.
4. Extract netloc (domain name): a function to extract the different structure paths of a full-path URL (parse_url), in which a new column (parsed_url) contains all the

URL parts. Then, this component column was fragmented into multiple columns, whereby each column contained one part of the URL structure. The finding was that the number of records remained the same.

5. Remove noise (null netloc): Netloc is the domain name, thus, if it is phishing, the URL would be a phishing URL regardless of remainder parts. Therefore, the important feature is the netloc feature, and it is needed to check if it has noise data, i.e., null values. After removing the null values (noise removal), the number of records was 472,259.

Finally, at the end of data pre-processing, the number of records was 472,259.

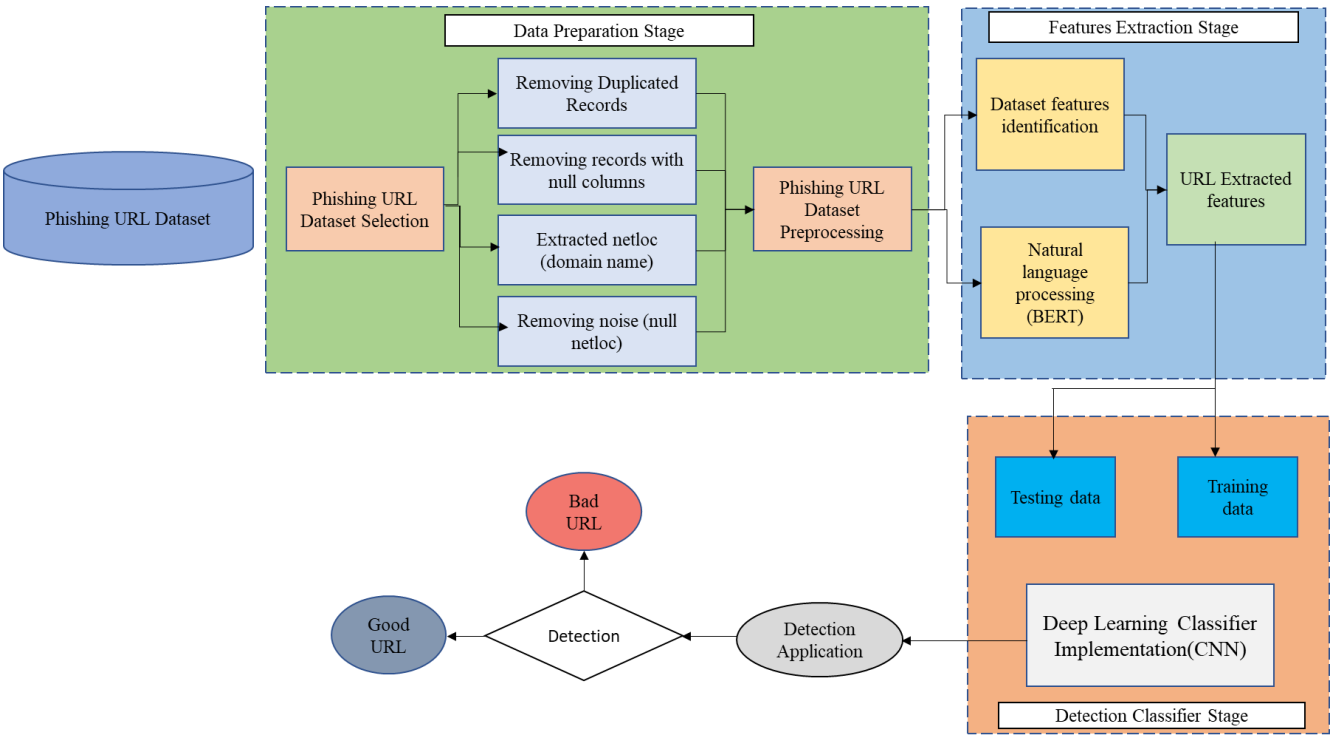


Figure 4. Block diagram of the proposed method.

Stage 3. Features extraction

The dataset contained only one data column, which contained the URL text. The BERT is used for natural language processing (NLP) that supports feature extraction. In addition, the NLP algorithm was applied to the unique data column and extracted a huge number of useful data features.

Table 3 shows the summarised extracted features.

Table 3. Dataset extracted features.

#	Feature Name	Data Type	Meaning
1	Length	Integer	URL length in character
2	Tld	Text	URL domain name extension
3	is_ip	Boolean	True if URL contains IP, false otherwise
4	domain_hyphens	Integer	The count of appearing "–"at the domain part
5	domain_underscores	Integer	The count of appearing "_" at the domain part
6	path_hyphens	Integer	The count of appearing "–"at the path part
7	path_underscores	Integer	The count of appearing "_" at the path part
8	Slashes	Integer	The count of appearing "/"
9	full_stops	Integer	The count of appearing "."
10	num_subdomains	Integer	The count of subdomains, in general this column equlas to full_stops +1
11	domain_tokens	T ext	Domain text only without digits or symbols
12	path_tokens	Text	Path text only without digits or symbols

From Request for Comments (RFC) 1808 [38], every URL should follow a specific format: <scheme>://<netloc>/<path>;<params>?<query>#<fragment>

- Scheme: the protocol name, usually http or https.
- Netloc: contains the network location—which includes the domain itself (and sub-domain if present), the port number, along with optional credentials in the form of user-name:password. Together, it may take the form of username:password@domain.com:80.
- Path: contains information on how the specified resource needs to be accessed.
- Params: element which adds fine-tuning to the path (optional).
- Query: another element, adding fine-grained access to the path under consideration (optional).
- Fragment: contains bits of information about the resource being accessed within the path (optional).

The masked language model (MLM) randomly generated masks with some of the tokens from the input, and the target was to expect the initial vocabulary ID of the masked word established only from its perspective. The MLM objective supported the depiction of combining the left and the right perspectives, which allowed a deep bidirectional transformer to be pre-trained, which was different from left-to-right language model pre-training. Moreover, the BERT model uses a "next sentence prediction" job that jointly pre-trains text-pair representations. For this study, a sentence-transformers/distillery-base-mean-tokens model was used. It mapped sentences and paragraphs to a 768-dimensional dense vector space, which can be utilised for tasks such as clustering or semantic search. Therefore, the result was a dataset with 768 dimensions for each row (a total of 472,259 rows).

Stage 4. Deep learning algorithm

The phishing site predict dataset contained only text data, and the extracted features were summarised, where the 11th and 12th features were text attributes, as presented in Table 2. Therefore, a supervised classification deep learning algorithm was tailored to deal with NLP. The convolutional neural network (CNN) was selected as a valuable deep learning algorithm. CNN is basically a neural-based approach which represents a feature function that is applied to constituent words or n-grams to extract higher-level features. The resulting abstract features have been effectively used for sentiment analysis, machine translation, and answering questions and other tasks. The main target of their method was to transform words into a vector representation via a look-up table, which resulted in a primitive word-embedding approach that learned weights during the training of the network, as shown in Figure 5 where w0, w1, ... wN−1 denotes the sentence weights, and N is the number of input sentences' words.

To perform sentence modelling with a basic CNN, sentences are first tokenised into words, which are further transformed into a word embedding matrix (i.e., an input embedding text layer) of N dimension. Then, convolutional filters are applied to this input embedding layer, which consists of applying a filter of all possible window sizes to produce what is called a feature map.

This is then followed by an ongoing max-pooling operation, which applies a max operation on each filter to obtain a fixed length output and reduce the dimensionality of the output. That procedure produces the final sentence representation.

The description of CNN layers is summarized in Figure 5 as:

Sentence matrix or input layer: In the previous example, each sentence is divided using parsing to separate words, and each word has five dimensions of data.

Convolution layer: Convolutional layers apply a convolution operation to the input, passing the result to the next layer.

Ongoing max-pooling layer: Max-pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Therefore, the output after the max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

Completed joined or flatten layer: Converting the data into a one-dimensional array for inputting them to the next layer. Flatten the output of the convolutional layers to create a

single long feature vector. It is connected to the final classification model, which is called a fully connected layer.

Dropout: Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

The output layer (sigmoid): The layer in a neural network model that directly outputs a detection.

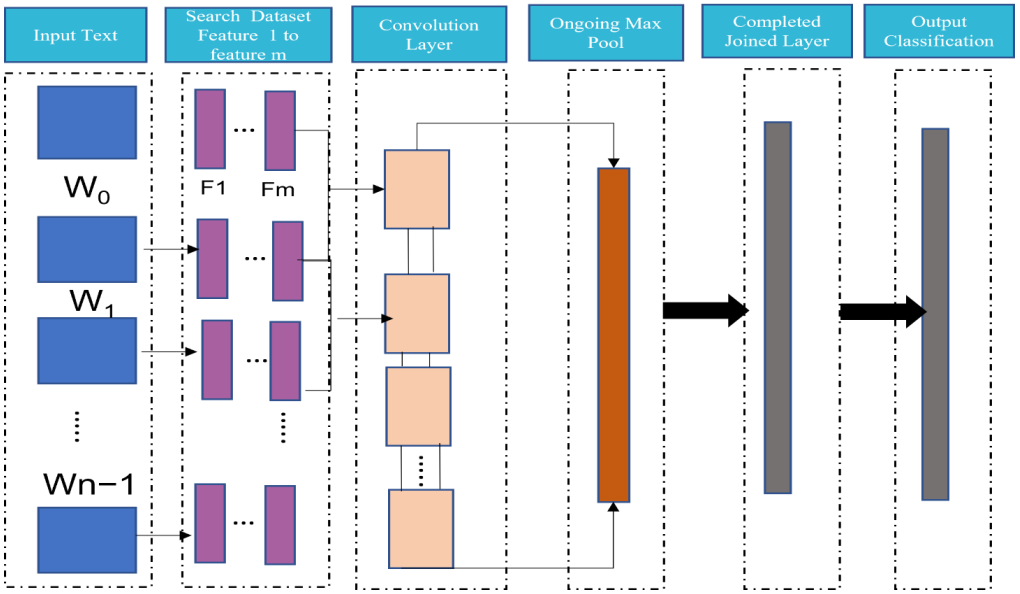


Figure 5. CNN architecture of NLP [14].

The following Figure 6 illustrates the CNN in a mathematical format.

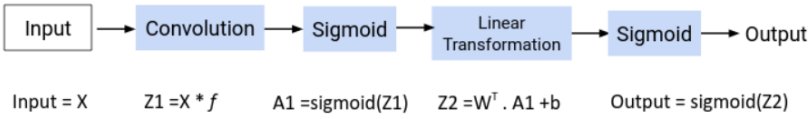


Figure 6. CNN mathematical format.

Where x represents the input data features, and Z1 is the result of conventional input data with an f function that can be a Laplace transformation function.

The convolution of f and g is written as $f * g$; it is defined as the integral of the product of the two functions after one is reversed and shifted. It is a particular kind of integral transform:

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \quad (1)$$

where τ is a dummy variable of integration. In CNN, f is the representation of data samples' features, and g is the Laplace function. A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve. A sigmoid function S has the following mathematical formula:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x) \quad (2)$$

where x is the function domain $[-\infty, \infty]$, range: (0, +1), S (0) = 0.

Finally, a linear transformation is a transformation from one feature space to another that maintains each vector space's underlying (linear) structure. A linear operator or map is another name for a linear transformation. When the range of the transformation is the same as the domain, the transformation is called an endomorphism or, if invertible, an

automorphism. The underlying field in both vector spaces must be the same. The following mathematical formula also has the defining characteristic of a linear transformation:

$$T : V \to W \text{ is that, for any vectors } v_1 \text{ , } v_2 \text{ in } V \text{ and scalars } a \text{ and } b \text{ of the underlying field,}$$

$$T(av_1 + bv_2) = aT(v_1) + bT(v_2) \quad (3)$$

Table 4 presents that the deep neural network topology for this model is a multiple CNN network with the following specification:

Table 4. Deep neural network topology parameters.

#	Parameter	Value
1	Input layer with data dimension d	768
2	One-dimension CNN network	filters = 256, kernel size = 2, activation function is relu, and the input vector dimension is (7681)
3	Batch normalization layer	1
4	One-dimension max-pooling size	2
5	Drop out layer around	0.2
6	One-dimension CNN network:	filters = 256, kernel size = 2, and activation function is relu.
7	Batch normalization layer	1
8	One-dimension max-pooling layer: with pool size	2
9	Drop out layer around	0.4
10	One-dimension CNN network:	filters = 256, kernel size = 2, and activation function is relu.
11	Batch normalization layer	1
12	One-dimension max-pooling layer: with pool size	2
13	Drop out layer around	0.4
14	Flatten layer	1
15	Dropout layer	1
16	Dense layer neurons and relu as activation function	128 neurons
17	Output layer: one neuron and activation function are relu	1

In this work, the standardised train and test datasets were divided into train size: 377,807 (80%) and test size: 94,452 (20%) for the DNN classifier model.

Fit classification models

Fitting classification models is the process of training the proposed classifier with the training dataset. As a first step, we need to reshape the data (train and test) input into a three-dimensional format to feed CNN. Then, the early stopping criteria have been defined to stop the training process according to this stopping condition: "Stop training when a monitored metric has stopped improving". The classification model is trained for epochs equal to 50 or according to early stopping. The classifier with BERT data has 8 epochs only for early stopping, which took more than 54 h to complete.

5. Evaluation Metrics

To evaluate the proposed models, evaluation metrics need to be defined [39–41].

Accuracy: is defined as the number of correct predictions divided by the total number of predictions.

F1-Score: is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into "positive" or "negative".

True Positive (TP): the total number of times the classifier classifies a URL as phishing, and it is correctly a phishing URL.

True Negative (TN): the total number of times the classifier classifies a URL as not phishing or legitimate, and it is correctly not a phishing URL.

False Positive (FP): the total number of times the classifier classifies a URL as a phishing URL, but it is not a phishing URL.

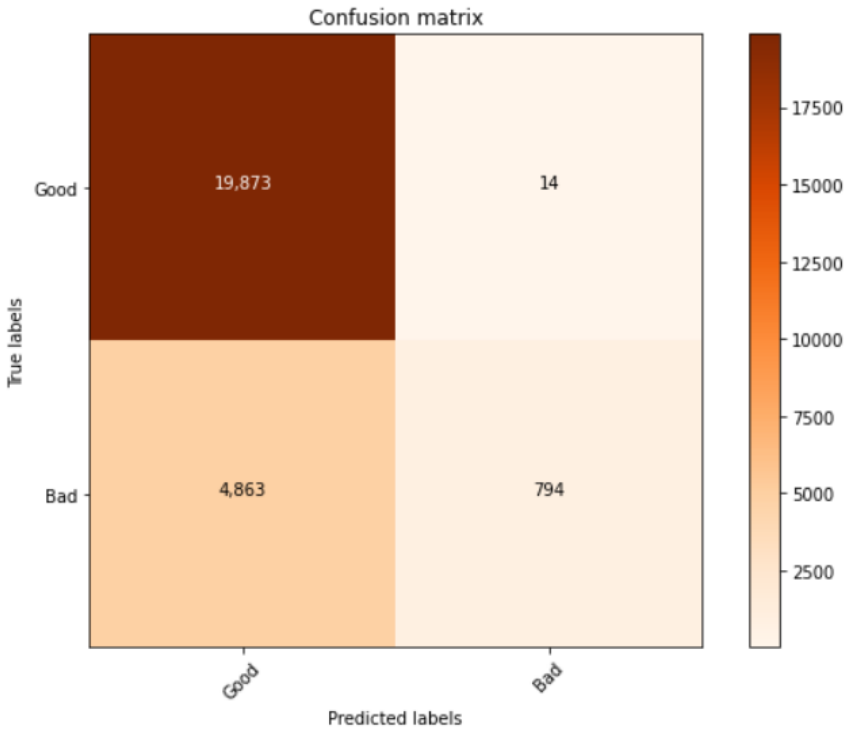
False Negative (FN): the total number of times the classifier classifies a URL as not a phishing URL, but it is a phishing URL.

Precision: It is implied as to the measure of the correctly identified positive cases from all the predicted positive cases. Therefore, it is useful when the cost of False Positives is high (this will be used to calculate F1-score).

Recall: It is the measure of the correctly identified positive cases from all the actual positive cases. It is important when the cost of False Negatives is high (this will be used to calculate the F1-score).

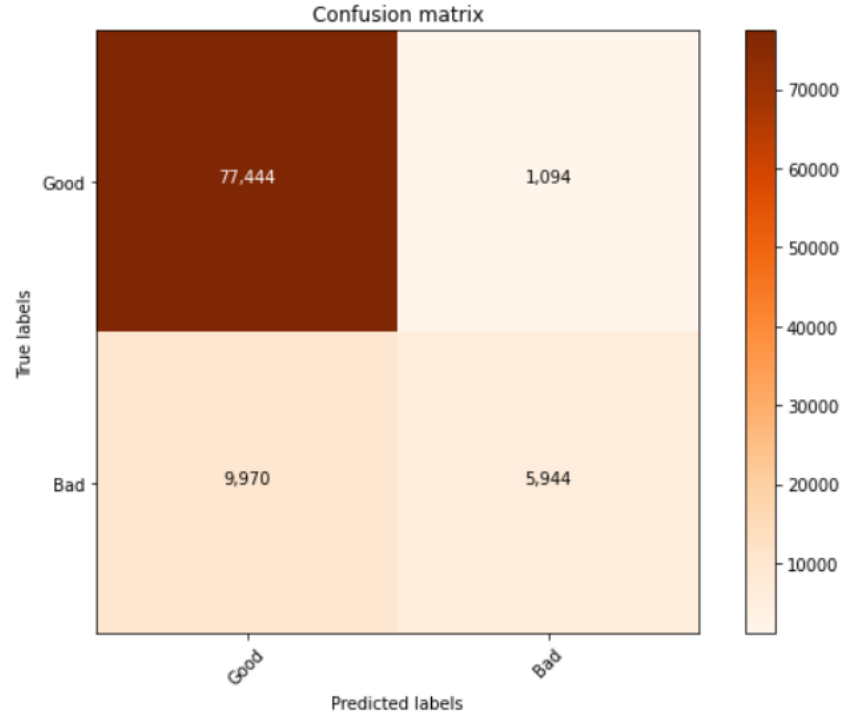
The performance of the proposed method for URL classification is compared to common classifiers which are implemented using the NLP feature extraction, which utilised dynamic feature selection with principal component analysis (PCA). The PCA is a linear dimensionality reduction technique that transforms a set of correlated characteristics in high-dimensional space into a set of uncorrelated features in low-dimensional space. Classifier 1 (C1) is where the classifier is trained over TF-IDF 100 features (Term Frequency—Inverse Document Frequency) [42,43] after dimensionality reduction from 1000 features to 100 features using the features selection method in TF-IDF algorithm. The second classifier (C2.a) is trained over 10 dynamic features. The third classifier (C2.b) is trained over only four dynamic features after applying PCA dimensionality reduction over the 10 original dynamic features to obtain the highest four variation features. Finally, the proposed classifier is trained by BERT.

The confusion metrics for all methods used in this study were calculated and summarised as shown in Figure 7a–d. The confusion matrix presented the relationship between the predicted label and the true label during the testing stage. The C1 model confusion matrix showed the huge difference between the true labels and predicted labels, which was the major reason for achieving less accuracy. The differences between the true labels and predicted labels were reduced in models C1, C2.a, C2.b, and the proposed method, which consequently increased the prediction accuracy.

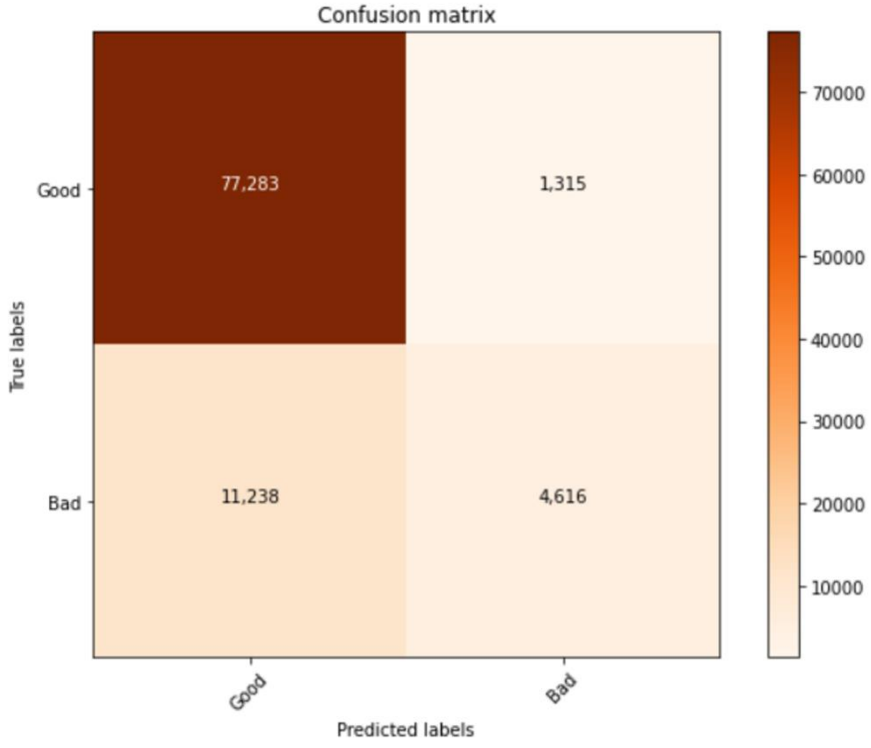


(**a**)

Figure 7. *Cont.*

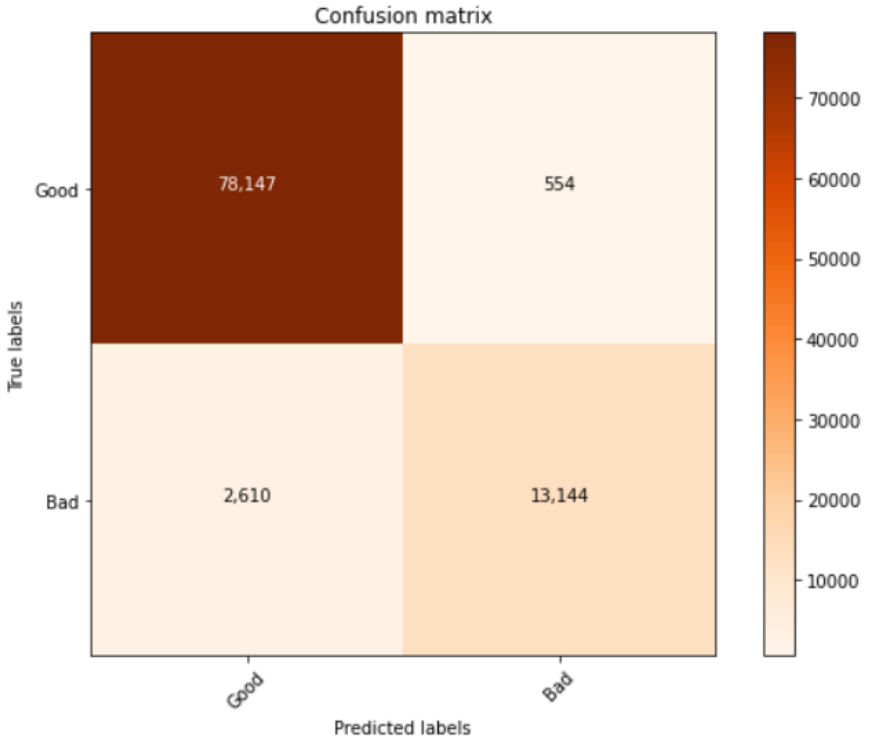


(**b**)



(**c**)

Figure 7. *Cont.*



(**d**)

Figure 7. (**a**) Confusion matrix C1; (**b**) Confusion matrix C2.a; (**c**) Confusion matrix C2.b. (**d**) Confusion matrix proposed method.

The classification model with BERT features with high accuracy approximately equalled 96.70%. The obtained TP and TN were high, whereas FP and FN were acceptable. The algorithm must fit for less TP and TN errors to increase the accuracy and precision. The confuse matrix represents the true labels for the good and bad URLs, which increases by the increasing trained sample. The results obtained from C1, C2.a, C2.b and proposed method are compared in terms of TP, FP, TN, and FN as shown in Figure 8.

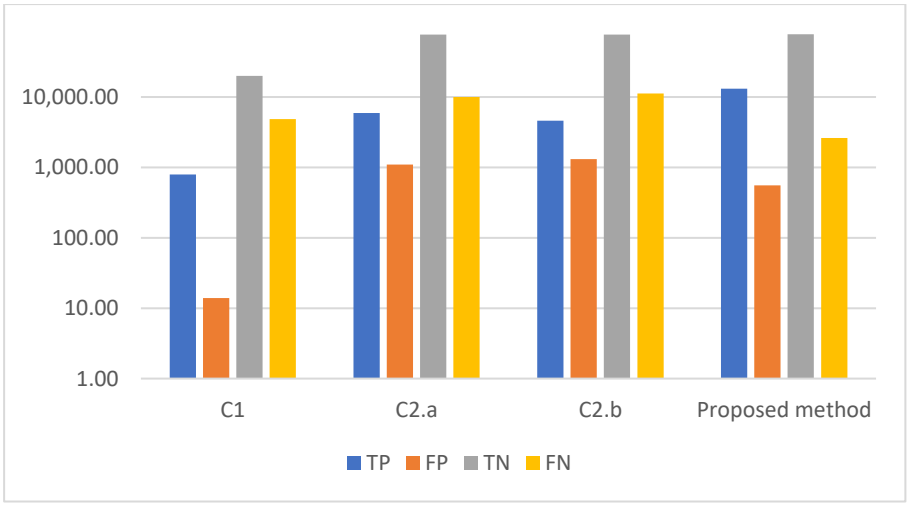


Figure 8. Comparison with other models.

6. Results Discussion

In this study, as a comparison with the proposed method, three scenarios, C1, C2.a, and C2.b, were implemented, respectively. The proposed method and all the scenarios

utilised the phishing website URL data to evaluate the effectiveness of the model. Firstly, in C1, we utilised CNN in a direct way to the phishing website URL data. Then, C2.a utilised the feature selection method to extract and reduce the data features, resulting in 10 features depending on the URL length, and implemented the CNN algorithm. This model used principal component analysis (PCA), which is a linear dimensionality reduction technique that transforms a set of correlated characteristics in high-dimensional space into a set of uncorrelated features in low-dimensional space. Lastly, C2.b resulted from the applied feature selection to the same phishing website URLs' data, resulting in four features applied to CNN.

Table 5 delivers the comparison of F1—score of results. As mentioned earlier, TP and TN indicate malicious and legitimate URLs, respectively. According to the TP, TN, FP, and FN, mutual precision and recall are determined as well. From these values, F1—measure is calculated. It implies the recovering capability of the URL indicator. As for the result, it is noticeable that the proposed method is superior to other methods.

Table 5. The classifier result evaluation.

Model	TP	FP	TN	FN	Accuracy	F1-Score	Precision
C1	794	14	19,873	4863	81.36%	57.58%	81.36%
C2.a	5944	1094	77,444	9970	88.28%	72.56%	88.28%
C2.b	4616	1315	77,283	11,238	86.70%	67.43%	86.70%
The proposed method	13,144	554	78,147	2610	96.66%	93.63%	96.66%

Figure 9 show in the first model, C1, we implemented the term frequency–inverse document frequency (TF-IDF) for feature extraction after dataset pre-processing. The TF-IDF feature extraction yielded 127,719 records with dimensionality reduction from 1000 features to 100 features by using features selection. With this dataset, the CNN classifier generated a phishing detection accuracy of 81.36% and a precision of 81.36%, with an F1 score of 57.58. In this model, C1, the classifier generated 794 true positives (TP), which was higher than the 74 false positives (FP) generated. On the other hand, this model generated 19,873 true positive errors and 4863 false positive errors. In the second model, C2, the NLP feature extraction utilised dynamic feature selection with principal component analysis (PCA). This model selected two dynamic features which had higher variance percentages, such as the "URL length feature" at 25% and the "the domain hyphens number feature" at 14.65%. The CNN classifier C2.a utilised the dataset generated by the PCA with the "URL length" feature, whereby the total number of dataset records was 472,259, with 10 features after reduction. The results showed an improvement in the phishing detection, which was 88.29% of accuracy, an F1-score of 72.56, and 88.28% of precision. This model had a true positive (TP) of 5944, which was higher than model C1. The FP was 1094, the TN was 77,444, and the FN was 9970. Meanwhile, C2.b utilised the PCA feature selection with the second feature, ''the domain hyphens number", whereby the total dataset records were 472,259 with 4 features after reduction. In addition, with C2.b, the accuracy had also improved to 8.71%, but less than with C2.a. C2.b generated an F1-score of 67.433 and a precision of 86.71%. TP, FP, TN, and FN were 4616, 1315, 77,283, and 11,238, respectively. Finally, the proposed method was implemented after applying the BERT NLP method. The BERT method showed a great enhancement in the URL phishing detection accuracy at 96.66%, with an F1-score of 93.63% and precision of 96.66%. The TP, FP, TN, and FN were 13,144, 554, 78,147, and 2610, respectively.

The result showed fewer false negatives. This model utilised 768 features and 472,259 records. The results obtained by the proposed method were the best amongst the three other classifiers, C1, C2.a, and C2.b. All the models divided the datasets into 80% for training and 20% for testing. The results also indicated that NLP played a great role in enhancing the accuracy of the detection. Likewise, the dynamic features programming language control (PLC), with a higher number of features selected, improved the accuracy. In addition, the proposed method was compared with other recent studies in URL phishing detection

by using similar datasets [25,42,43]. Table 6 clearly present the results obtained by the proposed method and other previous studies' results. These studies were chosen for comparison in this study because of the similarity of the phishing data they used. They all use URLs for phishing detection, using different methods and the same data that we used.

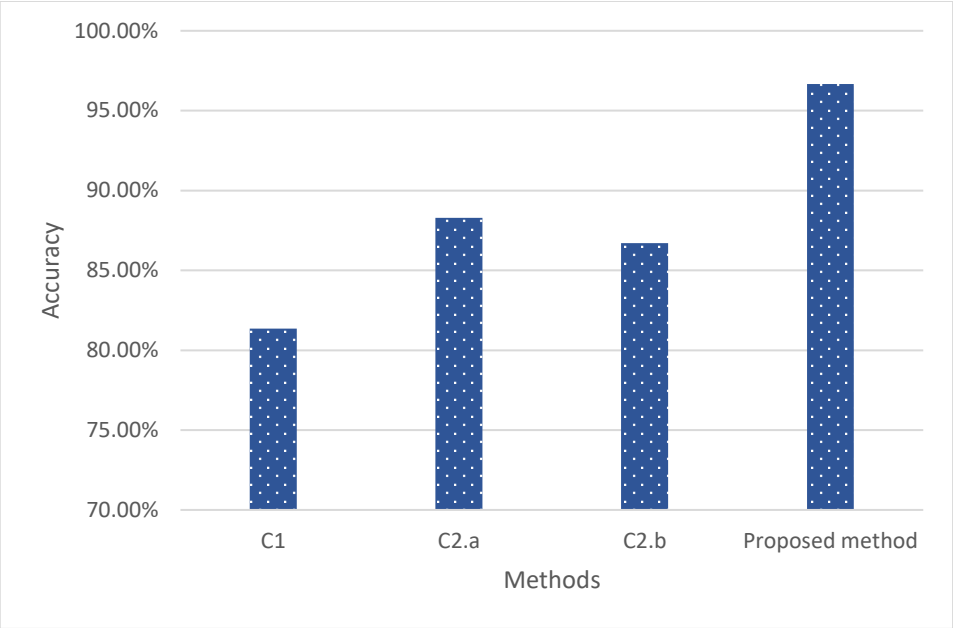


Figure 9. Comparison with other models.

Table 6. Comparison with literature works.

Study	Accuracy	Method
[25]	0.9573	Multi-layer-deep neural network
[42]	0.9750	ML+ hash vectorizer + random forest
[43]	0.9450	Random forest
Proposed method	0.9666	CNN + BERT

Figure 10 displays the outcomes of the proposed method and those of other methods earlier investigations. The proposed method had achieved a better accuracy result in most cases, except in one case study by [42], which obtained a result better than the proposed method. This is because the researcher utilised the hashing = vectorisation method, which is a non-semantic technique that is fundamentally destined to alter a group of text into a matrix of token incidences. The hashing vectoriser does not save the subsequent vocabulary in memory, takes a long time, generates light and large matrices, and does not hold any semantic meaning of the word, whereas BERT as NLP utilised a semantic technique that received pairs of sentences as input and was trained to predict whether the second sentence in the pair was the successive sentence in the original text. The URL text context, in general, is partially semantic text, which may indicate the reason why the non-semantic technique has a higher accuracy.

The results show that the accuracy of the suggested method is much higher than that of classic deep learning techniques on the same dataset. This is because typical deep learning techniques have trouble understanding the nuanced relationship between URLs and phishing. Additionally, these approaches' effectiveness is highly dependent on manual feature extraction. Comparing the suggested technique to the C1, C2.a, and C2.b methods also yields better results, further demonstrating the proposed method's advantage. The strength and capacity of the BERT methodology are primarily responsible for the proposed method's exceptional results.

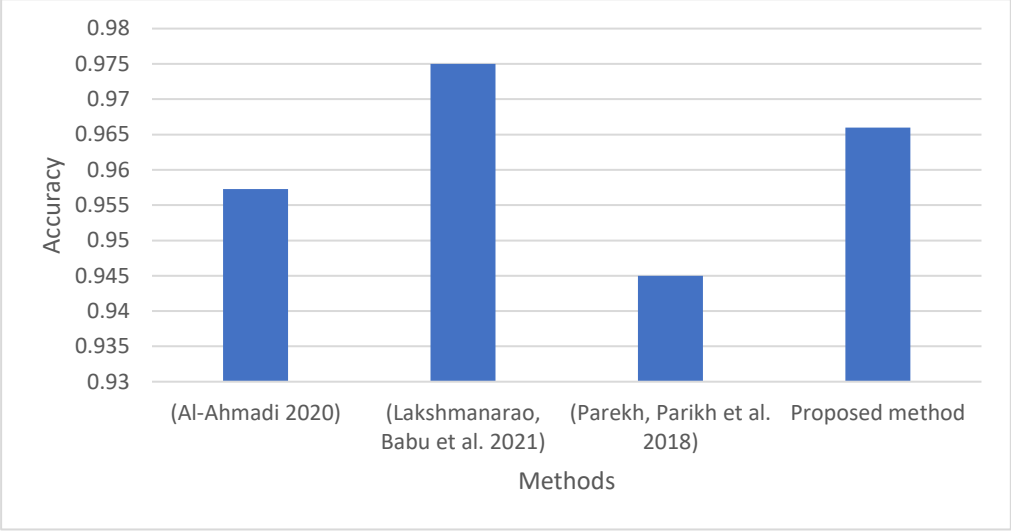


Figure 10. Comparison with previous works.

7. Conclusions

URL phishing detection is mandatory in order to overcome fraud activities. In this paper, a phishing detection solution is proposed based on BERT feature extraction and the CNN algorithm. A URL labelled dataset was used. The BERT natural language processing technique was utilised to extract the features from the URL text. Five million URLs were used from Kaggle's phishing URL tank to process, train, and test the proposed solution. During the pre-processing stage, the number of records was reduced to 472,259. The solution utilised the BERT model to extract the features and attain 768 dimensions for each row. The pre-processed data, with features extracted, were split into 80% for training and 20% for validation purposes. During the validation step, the accuracy obtained from the proposed CNN model was 96.66%. Therefore, the accuracy attained showed good performance in using a deep machine learning technique with natural language processing features extraction for URL phishing detection. Future work would enhance BERT with dynamic feature selection and CNN as a DNN classifier, which could help to optimise URL phishing detection with more accurate results.

Author Contributions: Conceptualization, M.E., S.B. and M.A.A.; methodology, M.E. and S.B.; software, M.E., S.A. and H.A.; validation, A.O.I., M.E., N.A. and S.B.; formal analysis, M.E., S.B. and M.A.A.; investigation, A.O.I., W.N. and M.E.; resources, M.E. and S.A.; data curation, M.E., H.A. and N.A.; writing—original draft preparation, M.E., A.O.I., S.B. and M.A.A.; writing—review and editing, M.E., W.N. and A.O.I.; visualization, M.E., S.A., H.A. and N.A.; supervision, M.E. and A.O.I.; project administration, M.E. and A.O.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R195), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against phishing attacks: State of the art and future challenges. *Neural Comput. Appl.* **2017**, *28*, 3629–3654. [CrossRef]
2. Ali, B.J. Impact of COVID-19 on consumer buying behavior toward online shopping in Iraq. *Econ. Stud. J.* **2020**, *18*, 267–280.
3. Huang, Y.; Qin, J.; Wen, W. Phishing URL detection via capsule-based neural network. In Proceedings of the 2019 IEEE 13th International Conference on Anti-Counterfeiting, Security, and Identification (ASID), Xiamen, China, 25–27 October 2019; pp. 22–26.

4. Venkatesha, S.; Reddy, K.R.; Chandavarkar, B. Social engineering attacks during the COVID-19 pandemic. *SN Comput. Sci.* **2021**, *2*, 78. [CrossRef] [PubMed]
5. Available online: https://www.statista.com/statistics/420442/organizations-most-affected-byphishing/ (accessed on 28 July 2022).
6. Oest, A.; Safei, Y.; Doupé, A.; Ahn, G.-J.; Wardman, B.; Warner, G. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In Proceedings of the 2018 APWG Symposium on Electronic Crime Research (eCrime), San Diego, CA, USA, 15–17 May 2018; pp. 1–12.
7. Hong, J. The state of phishing attacks. *Commun. ACM* **2012**, *55*, 74–81. [CrossRef]
8. Akbar, N. Analysing Persuasion Principles in Phishing Emails. Master's Thesis, University of Twente, Twente, The Netherlands, 2014.
9. Jamil, A.; Asif, K.; Ghulam, Z.; Nazir, M.K.; Alam, S.M.; Ashraf, R. Mpmpa: A mitigation and prevention model for social engineering based phishing attacks on facebook. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5040–5048.
10. Odeh, A.; Keshta, I.; Abdelfattah, E. Machine learningtechniquesfor detection of website phishing: A review for promises and challenges. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Virtual, 27–30 January 2021; pp. 0813–0818.
11. Khan, M.F. Detection of Phishing Websites Using Deep Learning Techniques. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 3880–3892.
12. Yi, P.; Guan, Y.; Zou, F.; Yao, Y.; Wang, W.; Zhu, T. Web phishing detection using a deep learning framework. *Wirel. Commun. Mob. Comput.* **2018**, 4678746. [CrossRef]
13. Taylor, W.L. "Cloze procedure": A new tool for measuring readability. *J. Appl. Psychol.* **1953**, *30*, 415–433. [CrossRef]
14. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
15. Alex, S.A.; Jhanjhi, N.; Humayun, M.; Ibrahim, A.O.; Abulfaraj, A.W. Deep LSTM Model for Diabetes Prediction with Class Balancing by SMOTE. *Electronics* **2022**, *11*, 2737. [CrossRef]
16. Khan, T.; Sherazi, H.H.R.; Ali, M.; Letchmunan, S.; Butt, U.M. Deep learning-based growth prediction system: A use case of China agriculture. *Agronomy* **2021**, *11*, 1551. [CrossRef]
17. Sircar, A.; Yadav, K.; Rayavarapu, K.; Bist, N.; Oza, H. Application of machine learning and artificial intelligence in oil and gas industry. *Pet. Res.* **2021**. [CrossRef]
18. Chen, Y.; Zhang, D. Theory-guided deep-learning for electrical load forecasting (TgDLF) via ensemble long short-term memory. *Adv. Appl. Energy* **2021**, *1*, 100004. [CrossRef]
19. Adebowale, M.A.; Lwin, K.T.; Hossain, M.A. Deep learning with convolutional neural network and long short-term memory for phishing detection. In Proceedings of the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Island of Ulkulhas, Maldives, 26–28 August 2019; pp. 1–8.
20. Mohammad, R.M.; Thabtah, F.; McCluskey, L. Predicting phishing websites based on self-structuring neural network. *Neural Comput. Appl.* **2014**, *25*, 443–458. [CrossRef]
21. Al-Alyan, A.; Al-Ahmadi, S. Robust URL phishing detection based on deep learning. *KSII Trans. Internet Inf. Syst. (TIIS)* **2020**, *14*, 2752–2768.
22. Vigneshwaran, P.; Roy, A.S.; Sathvik, B.S.; Nasirulla, D.M.; Chowdary, M.L. Multidimensional features driven phishing detection based on deep learning. In Proceedings of the Integrated Emerging Methods of Artificial Intelligence & Cloud Computing, IEMAICLOUD 2021. Smart Innovation, Systems and Technologies; Springer: Berlin/Heidelberg, Germany; Volume 273. [CrossRef]
23. Bustio-Martínez, L.; Álvarez-Carmona, M.A.; Herrera-Semenets, V.; Feregrino-Uribe, C.; Cumplido, R. A lightweight data representation for phishing URLs detection in IoT environments. *Inf. Sci.* **2022**, *603*, 42–59. [CrossRef]
24. Available online: https://scholar.google.com/scholar?as_q=phishing&as_epq=Deep+learning&as_oq=&as_eq=&as_occt=title&as_sauthors=&as_publication=&as_ylo=2018&as_yhi=2022&hl=ar&as_sdt=0%2C5 (accessed on 20 July 2022).
25. Al-Ahmadi, S. PDMLP: Phishing detection using multilayer perceptron. *Int. J. Netw. Secur. Its Appl. (IJNSA)* **2020**, *12*, 59–71.
26. Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine learning based phishing detection from URLs. *Expert Syst. Appl.* **2019**, *117*, 345–357. [CrossRef]
27. Zouina, M.; Outtaj, B. A novel lightweight URL phishing detection system using SVM and similarity index. *Hum.-Cent. Comput. Inf. Sci.* **2017**, *7*, 1–13. [CrossRef]
28. Moghimi, M.; Varjani, A.Y. New rule-based phishing detection method. *Expert Syst. Appl.* **2016**, *53*, 231–242. [CrossRef]
29. Ferreira, R.P.; Martiniano, A.; Napolitano, D.; Romero, M.; Gatto, D.D.D.O.; Farias, E.B.P.; Sassi, R.J. Artificial neural network for websites classification with phishing characteristics. *Soc. Netw.* **2018**, *7*, 97–109. [CrossRef]
30. Yang, P.; Zhao, G.; Zeng, P. Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* **2019**, *7*, 15196–15209. [CrossRef]
31. Somesha, M.; Pais, A.R.; Rao, R.S.; Rathour, V.S. Efficient deep learning techniques for the detection of phishing websites. *Sādhanā* **2020**, *45*, 1–18. [CrossRef]
32. Kaggle.com, P.S.U.A.O. Available online: https://www.kaggle.com/taruntiwarihp/phishing-site-urls (accessed on 27 April 2022).

33. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805, 2018.
34. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
35. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
36. Taud, H.; Mas, J. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 451–455.
37. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
38. Fielding, R. *Relative Uniform Resource Locators*; RFC1808 (acm.org); 1995; Available online: https://dl.acm.org/doi/pdf/10.17487/RFC1808 (accessed on 18 March 2022).
39. Khan, M.R.H.; Afroz, U.S.; Masum, A.K.M.; Abujar, S.; Hossain, S.A. Sentiment analysis from bengali depression dataset using machine learning. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–5.
40. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*. [CrossRef]
41. Buckland, M.; Gey, F. The relationship between recall and precision. *J. Am. Soc. Inf. Sci.* **1994**, *45*, 12–19. [CrossRef]
42. Lakshmanarao, A.; Babu, M.R.; Krishna, M.B. Malicious URL Detection using NLP, Machine Learning and FLASK. In Proceedings of the 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 24–25 September 2021; pp. 1–4.
43. Parekh, S.; Parikh, D.; Kotak, S.; Sankhe, S. A new method for detection of phishing websites: URL detection. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 20–21 April 2018; pp. 949–952.