Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.036626
Article

Modelling an Efficient URL Phishing Detection Approach Based on a Dense Network Model

A. Aldo Tenis* **and R. Santhosh**

Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India

*Corresponding Author: A. Aldo Tenis. Email: karpagam.publication@gmail.com
Received: 07 October 2022; Accepted: 03 May 2023; Published: 28 July 2023

Abstract: The social engineering cyber-attack is where culprits mislead the users by getting the login details which provides the information to the evil server called phishing. The deep learning approaches and the machine learning are compared in the proposed system for presenting the methodology that can detect phishing websites via Uniform Resource Locator (URLs) analysis. The legal class is composed of the home pages with no inclusion of login forms in most of the present modern solutions, which deals with the detection of phishing. Contrarily, the URLs in both classes from the login page due, considering the representation of a real case scenario and the demonstration for obtaining the rate of false-positive with the existing approaches during the legal login pages provides the test having URLs. In addition, some model reduces the accuracy rather than training the base model and testing the latest URLs. In addition, a feature analysis is performed on the present phishing domains to identify various approaches to using the phishers in the campaign. A new dataset called the MUPD dataset is used for evaluation. Lastly, a prediction model, the Dense forward-backwards Long Short Term Memory (LSTM) model ($d - FBLSTM$), is presented for combining the forward and backward propagation of LSMT to obtain the accuracy of 98.5% on the initiated login URL dataset.

Keywords: Cyber-attack; URL; phishing attack; attention model; prediction accuracy

1 Introduction

The present digital transformation and web services have increased rapidly in the last few years. The change is encouraged by the companies using the online service provided, such as SaaS (Software as a Service) or e-commerce and e-banking [1]. In today's world, limitations are spread about the model of working from home because the pandemic of COVID-19 that gives additional millions of labours, teachers and students who developed their remote activities [2], which leads to a substantial extra workload the services like platforms for students, email, company portals or VPNs. Hence, more targets are potentially exposed to phishing attacks, which mimic legal websites to steal users' payment

data or credentials [3,4]. Phishing is one of the important considerable threats that depends on social engineering, which is concluded by recent studies [5,6] when the pandemic of COVID-19 altogether with websites for executing attacks and spam emails.

The phishing sites are identified via HTTP protocol and need to be in better condition. The APWG reports that below 25% hosting of phishing websites in the 3rd quarter of 2017 [7] are hosted in the HTTPS protocol, and this amount is enhanced to 83% in the 1st quarter of 2021 [8]. End-to-end communication is secure by providing websites that send the fake impression safely to the users when the online transaction is made [9]. Further, the considerable enhancement in the attacks of phishing is reported by the AntiPhishing Working Group (APWG) [10] from the websites. During the pandemic of COVID-19, a cause behind the enhancement is the resort of people to online services.

The list-based technique is one of the most famous solutions to detect phishing by analyzing the URL requested over the database of phishing. Google SafeBrowsing OpenPhish, PhishTank, and SmartScreen [11] are some examples. The request is blocked if the requested URL matches the record, and a warning is displayed to the user before visiting the website. Moreover, if the phishing URL is not reported the last time may get fail despite the abilities of the list-dependent technique [12]. Also, this needs contiguous attempts to have the new phishing data for updating the database. OpenPhish eliminates all the URLs from the report after seven days [13]. On the other hand, many phishing URLs were eliminated and observed by Bell and Komisarczuk after the fifth day from Phishtank. Attackers are allowed by this problem to reuse a similar URL during the removal from various lists [14].

The phishing URLs with automatic detection depending on machine learning are attracted to the study because of the drawbacks of blacklist-based methods [15,16]. The techniques are segregated into four classes for the detection based on the data types required, that is, the URL text, the content page, features related visually, and the data related to networks [17]. Methodologies depend on the content of the page and the features that are visually needed to visit the website for collecting the source code for rendering the time-consuming work. In studies, other restrictions to the availability are identified, which rely on the networking and the information based on 3rd parties like rankings of the search engine or WHOIS. Since the advantages are implied, like the fast calculation due to the unloading of websites, phishing detection is focused on the proposed system via URLs to overcome the restrictions. Also, since the features are extracted from the URLs alone, it is independent of language and the third party.

The homepage URL from famous websites as the legal is used by the existing URL datasets [18,19]. Moreover, the difficulty is needed to determine when the website form of login is phishing or legal. The dataset available publicly is different from the reflecting conditions from the perspective and the best of the author's knowledge which present few real issues in detecting the phishing URL. It is obtained from the latest pages of phishing. Further, the latest machine learning suggestions achieve high accuracy with the help of outdated datasets containing the gathered URLs, typically from 2009 to 2020.

The legal login websites present the URL dataset in the proposed system for obtaining the URLs from these pages. Thus, the deep learning and machine learning approaches are evaluated to recommend more accurate methods. Then, the process of models is trained by having the legal homepages shown in the proposed system for classifying the legal URL of login, which demonstrates the hypothesis regarding the legal login URLs and the detection of phishing. In addition, the accuracy decreased on the trained models with time, and the dataset from 2016 is shown in the proposed system to evaluate the gathered data in 2020. Lastly, the present phishing encounters are overviewed

to explain the techniques and tricks of the attacker. The below summarization provides the important contribution of the proposed system. They are,

- The prior dataset MUPD is provided in the proposed system, which is available publicly for research. It is composed of legitimate and phishing URLs and websites. The dataset source is PhishTank, which is similar to the existing works.
- Three pipelines are implemented and evaluated using MUPD to detect URL phishing. The attention mechanism is provided in the forward and backward LSTM to highlight the significance. The attention model intends to provide feature learning over sequential data compared to data vectors. The attained feature information is accurate and reasonable.
- The empirical process of detecting URL phishing, which struggles to classify the URLs login, is demonstrated during the training on homepage URLs of legal URLs and phishing.
- The suggested Dense forward-backwards LSTM model ($d-FBLSTM$) phishing detection has robustness evaluated in the proposed system across time. The model on the gathered dataset is used to train the model.

The work is organized as follows: Section 2 provides a comprehensive analysis of various prevailing approaches and discusses the pros and cons of the anticipated model. In Section 3, the anticipated model for phishing URL prediction is done with $d-FBLSTM$. The numerical results of the proposed model are discussed and compared with various other approaches. The research conclusion is provided in Section 5.

2 Literature Works

The optimal solution is found using the simple structure of the model attempted by Gupta et al. [20] to eliminate the expansion of the network. The learning rate is updated to enhance the model's accuracy in the proposed system. Firstly, the three-layer neural network is constructed in the proposed system having one neuron in the hidden layer. In the training phase, the neurons in the hidden layer increase gradually, yet the features are extracted manually. The directory of Yahoo! and the directory of starting point provide the legal websites which are attained. On the other hand, MillerSmilesd and PhishTank provide phishing websites. The model has a better generalization, indicated in the results for the higher detection rate and the noise data.

Two techniques are evaluated by Wang et al. [21] to detect the phishing of URLs. The combination of statistical and linguistic analysis of URLs is the primary technique having the random forest classifier. The LSTM network learns the representation directly in the second technique from the character series of URL rather than the feature extraction manually. The PhishTank provides the URLs for phishing to be gathered, and the Common Crawl provides the legal URLs. The LSTM technique obtains the F1 score and greater accuracy over the other classifier. The GANs are used to propose the detection method of phishing URLs by Sabahno et al. [22] in the data space having the oversampling task. The statistics of string patterns are learned by training the GANs for the URLs in the minority class and creating the synthetic URLs.

The representation of synthetic samples is chosen in the proposed system with the help of Euclidean distance-based selection and k-means clustering. The repositories of Common Crawl and PhishTank provide the dataset to be gathered. A deep belief network (DBN) is used by Bu et al. [23] to detect phishing. Two features are used in the proposed system for detection: (i) interaction features and (ii) original features. The interacting features explained the interactions between websites. An original feature represents the URL's direct features, like the age of the domain. There are two layers in the

model, which have the layer that provides the limitations of Boltzmann machines (RBMs) stacked layer. The SVM is used as the binary classifier in the proposed system for classifying DBN features, and the dataset is achieved from the ISP via the real IP flows. The detection model attains a higher rate of TPD and the lower rate of FP. Somesha et al. [24] evaluate deep learning models for detecting phishing URLs, such as I-RNN, RNN, CNN, LSTM, and CNN-LSTM, as the hybrid network. The automatic feature extraction help to train the legal URL and the phishing at the character level. The DMOZ directory and Alexa provide the legal URLs which are gathered. On the other hand, OpenPhish and PhishTank provide the phishing URLs to be gathered. The CNN-LSTM and LSTM networks outperform to distinguish the URL as either phishing or legal compared to other chosen models.

The stacked RBMs are used to detect the malicious URL to select the feature for classification with the deep neural networks proposed by Atimorathanna et al. [25]. Malevolent URLs and the upgraded persistent attack URLs are gathered from the domain of the evil list. On the other hand, phishing and spamming of URLs are gathered from the Machine Learning Repository of UCI. DMOZ directory provides the legal URLs to be gathered. The model reduces the rate of FP, and the accuracy related to detection is enhanced when compared with other chosen methodologies. A tool is deployed as the Google Chrome browser extension to provide the user with a safe browsing experience proposed by Maurya et al. [26]. The URLs are analyzed and classified by the suggested tool with the help of two deep learning techniques, such as LSTM networks and artificial neural networks (ANNs). The URLs provide the features which are extracted by the models automatically. On the other hand, manual feature engineering is needed by other existing approaches that are costs related to computation and time-consuming work. The PhishTank, Twitter Streaming API, and search engines achieve the dataset.

The detection model of phishing websites depends on URL, features of HTML, and the third-party services proposed by Sundaram et al. [27]. Alexa provides the legal webpages to obtain, and PhishTank provides the phishing webpages to obtain. The stacked AE (SAE) having a Softmax classifier is used for detecting phishing websites. The proposed model's optimal width of hidden layers is determined by correlating correlation coefficients among the SAE weight matrices. The model over the other chosen models obtains better performance. Moreover, the extraction of features is done manually.

A deep learning technique is proposed by Abiodun et al. [28] to detect phishing, known as PUCNN, based on the website URL only. CNN is used for the extraction of character-level feature representations of URLs. A large-scale dataset is proposed, known as MUPD, with more than two million gathered URLs from DomCop and PhishTank. The suggested CNN obtains better accuracy than previous modern models [29]. It performs well than different machine learning models depending on the generally used features of URLs from the dataset of MUPD. Many works are increased, which use deep learning models like the RNNs and CNNs that are compared with the conventional classification algorithms [30]. However, the third-party services have relied on the proposed system combined with the system. In addition, some research utilized the model of GAN. Hence, the model is suggested for detecting phishing websites more efficiently to predefine the difficulties [30–36]. The deep learning model is used to perform the task of detection in the proposed system for the feature representation at the character level. The CNN and LSTM networks are comprised of the model based on the website URLs only.

3 Problem Formulation

Communication and networking technology is developing rapidly and has been subjected to multiple cyberattacks. A serious cyberattack is phishing which misleads the users into disclosing confidential personal data. Since many attacks are launched, the many cyberattack category is considered. Effective phishing detection approaches are needed, which are reflected critically. In the past few years, many proposed phishing detection approaches rely on third-party services and the features concerning the web pages that need crawling the webpage's content. A phishing detection technique is proposed that lies on the URL of the website over the features, which are third-party services and content-based. The greater accuracy is obtained by the URL-based technique demonstrated by [37] to classify the phishing URLs that are not seen. The extracted features are used by the model, which performs the URL strings inspection that is the same as the content-based detection systems, which enables the classification with the deep learning detection systems [38].

4 Methodology

This section provides a detailed analysis of predicting the phishing URL using the available MUPD online dataset. The prediction is made with the proposed $d - FBLSTM$ model. The performance of the anticipated model is discussed and compared with various approaches like Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Naive Bayes (NB) and Multi-Layer Perceptron's (MLP).

4.1 Dataset Acquisition

The MUPD dataset has 2,353,933 phishing URLs, and the 2,220,853 legitimate URLs are used in the MUPD in the experiments. The PhishTank is the source for the phishing URLs that is the same as the reviewing many works in the relation works. The phishing websites are considered by the MUPD dataset only that are checked on PhishTank as phishing. The legal websites have the source, the top 4 million domains listed from the DomComp. The index page is in the dataset of MUPD for every top 4 million domains, and the internal URL is random. Further, the publication of the MUPD dataset is done through the easy training process, and the proposed system is evaluated.

After the pre-processing step, the final dataset has 1,140,599 legal URLs and 1,167,201 phishing URLs. The pre-processing steps are done to create the published dataset that is sampling to promise a balanced dataset. The duplicate data needs to be removed; the dataset is split into three subsets: training, validation, and testing. Various URLs are in the gathered datasets from many recursive URLs or similar hosts. Consider an instance and a similar phishing website that provides many pages that are often reported as the pages which are phishing. The collection process is used as the domain of top-level results for different reasons, like the redirection of HTTP in the recursive hosts. Hence, the suggested MUPD dataset provides the URLs having the duplicate URLs and the repeated hosts are eliminated. The evaluation decision is enhanced, and the models are prevented from memorizing the host is focused in the step.

In binary classification, the balanced dataset is preferred when the measure of accuracy is used. The representation of phishing URLs is done only on one-third of the dataset, even though the dataset was balanced before eliminating the duplicate URLs during the removal of duplicate URLs. A random sampling of 1,200,000 legal URLs is used To fix this problem. A balanced dataset of 1,167,201 phishing URLs and 1,140,599 legal URLs is achieved through this step. The splitting of the dataset into three subsets is the unexpected way. They are (i) 0.6 training, (ii) 0.2 validation and (iii) 0.2 testing.

4.2 Pre-Processing

The process of encoding is the important stage for this step which is suggested. Primarily, the length of URL is fixed at 255 characters, stated in HTTP standard protocol RFC2616 is a limit on the length of the URL that is "servers needs to be serious regarding the lengths of URL which is above 255 bytes due to the few older client or the implementations of proxy are not supported the lengths properly". The extra zeros are inserted at the end of the length shorter than 255 characters and if the URL length exceeds 255 characters respectively. Then, since the neural network uses several vectors for performing any mathematical function, every character in URL is encoded in the one-hot vector, which has 0 and 1. 10 numeric digits and 33 special characters are permitted in URLs like /, &, -, ? and =, and 26 characters of the alphabet are included in the used characters. Lastly, the tensor is composed of the encoded characters, which are given as the input for the model.

4.3 Prediction Method

The convolutional layer is required for capturing the traffic data's local features after the mentioned processing functions. The convolution layer is the major part of CNN that convolves the input or the feature maps with the convolutional kernels for creating the various feature maps. The shallower convolutional layers having the receptive field are narrow for extracting the local data. On the other hand, deep layers capture the global data having a larger visual field. Therefore, as the number of convolutional layers maximizes and scales, the convolutional features gradually become coarser. The convolutional layer has the input formulated as the size $H \times W \times 1$ tensor. Here, the width and height of the yielded data using normalization processing are represented as W and H. Consider that some N layers of units as input are described using the convolution layer. The convolutional output will be $(N - m + 1)$ united if the width filter w is used in the proposed model. Eq. (1) shows the process of convolutional calculation.

$$x_{i,k}^{l,j} = f\left(b_j + \sum\nolimits_{a=1}^{m} w_{a,k}^{j} r_{i+(k-1)*s+a-1}^{l-1,j}\right. \tag{1}$$

Here, the range of the section is s and $x_{i,k}^{l,j}$ is one of the units of the j feature map of the k^{th} section in the l^{th} layer. The non-linear mapping is denoted as f and utilizes the tanh $(\cdot)$ as the tangent function.

4.3.1 Dense Forward-Backwards LSTM Model ($d - FBLSTM$)

$d - FBLSTM$ (See Fig. 1) uses the content data for the feature learning for the time series data comprised of URL bytes. The time series feature is learned using the $d - FBLSTM$ in the data packet. Every data packet has the traffic bytes consecutively input to $d - FBLSTM$ that, finally, achieves the pocket vector. The LSTM (Long Short-Term Memory) is the enhanced version of $d - FBLSTM$. The coarse-grained features are extracted using the $d - FBLSTM$ model using the connection of the backward LSTMS and the forward LSTM. The input gate i designs the LSTM, the f is denoted by the forget gate, and the o is the output gate for controlling the overwriting of the data using a comparison of the inner memory cell C during the arrival of new data. It needs to be judged if this is useful based on the related rules when the data enter the LSTM network. The information that meets the authentication algorithm has remained, and the inconsistent data is forgotten via the forget gate. The hidden states of the $d - FBLSTM$ layer as $h = (h_0, \ldots, h_t)$ and the input sequence are presented by $x = (x_0, \ldots, x_t)$ at time t is provided below. The input x_t at the present moment is the input to forget selectively in the cell state C_t, and the forget gate will take the hidden layer h_{t-1} output at the previous moment that is presented below.

$$f_t = sigmoid\ \left(W_{xf} x_t + W_{hf} h_{t-1} + b_f\right) \tag{2}$$

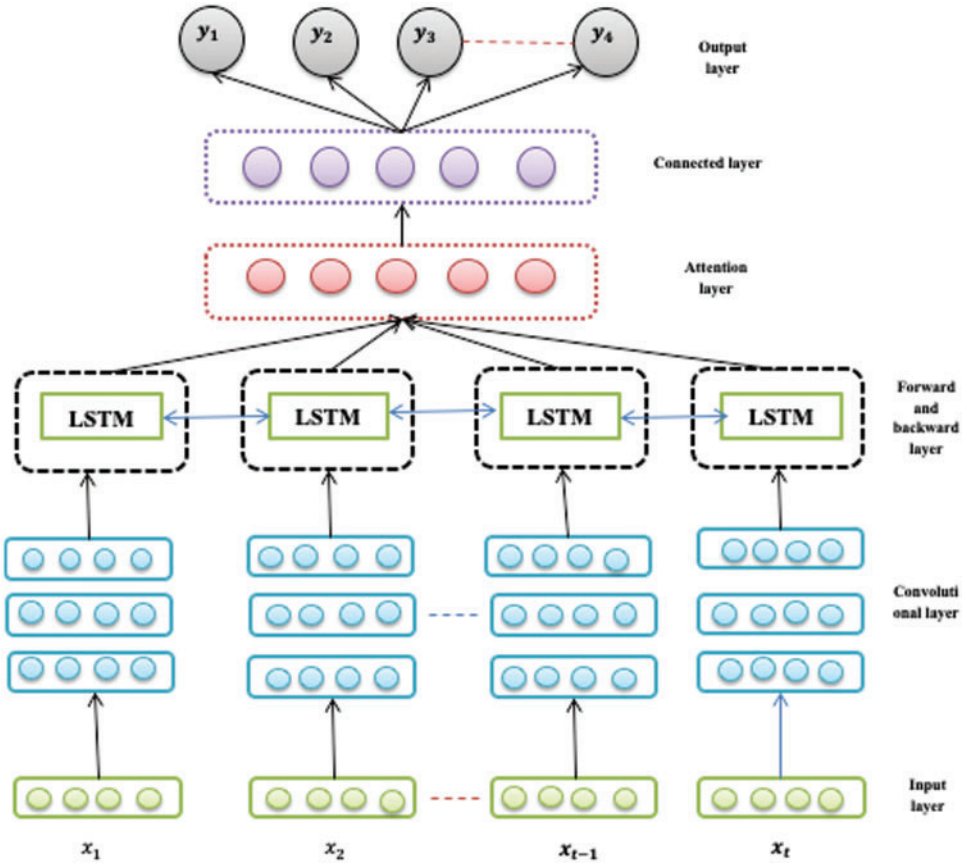

Figure 1: A prediction model

Tanh creates a new vector of the candidate. The input gate is cooperated, having the tanh function to control the extra new data. The input gate creates the value for every item in $\widetilde{C}_t$ to control from 0 to 1 the amount of new data is added, which is provided in Eqs. (3) and (4):

$$C_t = sigmoid\ \left(f_t.\ C_{t-1} + i_t.\ \widetilde{C}_t\right) \tag{3}$$

$$i_t = sigmoid\ \left(W_{xi}x_t + W_{hi}h_{t-1} + b_t\right) \tag{4}$$

$$\widetilde{C}_t = \tanh\left(W_c x_t + W_c h_{t-1} + b_c\right)$$

The output gate that is provided below is used to control the amount of present unit state filtered in Eq. (5):

$$o_t = sigmoid\ \left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \tag{5}$$

The h_t is the hidden state for the $d - FBLSTM$ model at time t, which is the packet vector created from every packet and is defined as the concatenation $\overleftarrow{h}_t$ and $\overrightarrow{h}_t$ provided below.

$$h_t = \overleftarrow{h}_t + \overrightarrow{h}_t \tag{6}$$

$$\overrightarrow{h}_t = tanh\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \tag{7}$$

$$\overleftarrow{h}_t = \tanh\left(W_{w\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \tag{8}$$

Here, the point-wise product is denoted by '0', and the heterogeneous time series data has the input that is presented as x. The hidden states of the backward and forward LSTM layers are represented by $\overleftarrow{h}_t$ and $\overrightarrow{h}_t$ at time t. W is the connection weights of all the matrices between two units, and the bias vectors are represented as b.

4.3.2 Attention Mechanism

Every data has a packet vector, which is generated by $d-FBLSTM$. The packet vectors are formed in the order of communication between two parties in the network stream to generate the packet vectors. The attention layer helps the relationship in the packet vectors to learn. The mechanism of attention is utilized for adjusting the packet vectors' probability. Hence the proposed $d - FBLSTM$ system provides more focus on the main features. Primarily, the $d - FBLSTM$ model extracts the packet vectors h_t are utilized for attaining the implicit representation via the non-linear transformation presented below.

$$o_t = sigmoid\ (W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{9}$$

The importance of packet vectors is measured next depending on the same representation utn having the context vector u_w and achieving the coefficient of important weight α_t, which is normalized. The random initialization matrix is presented as uw, and is concentrated on the main data than ut. The coefficient of weight for the mentioned coarse-grained features is presented below.

$$\alpha_t = \frac{\exp\left(u_t^T\ u_w\right)}{\sum \exp\left(u_t^T u_w\right)} \tag{10}$$

Lastly, through the weighted sum of i_t depends on α_t, the fine-grained feature is computed, and the s is presented below.

$$s = \sum \alpha_t h_t \tag{11}$$

Here, s is the fine-grained feature vector created from the attention mechanism utilized to recognize the malicious traffic with the softmax classifier presented below.

$$y = softmax\ (W_h s + b_h) \tag{12}$$

Here, the classifiers' weight matrix is presented as W_h that can map s to the new vector having the h as length. The number of categories of the traffics of the network is presented as h.

4.3.3 Training Process

The proposed network is trained and presents the backwards and forward passes. The proposed model performs a forward propagation-based attention layer and $d - FBLSTM$ layer, with every layer representing the structures and those that played various roles in the complete model. The $d - FBLSTM$ layer provides forward propagation to the attention layer. The previous model has the processing to obtain the input of the present model. The final result of recognition is achieved after the completion of the forward propagation. Here, X is defined as the input dataset. The testing division and the training dataset are presented as $x_1, x_2,\ and\ x_3$. Each sample is transformed into the format X''

after the operation of one hot and the operation of normalization acceptable to the proposed model. The state is presented as the size of the cell state vector is set. The algorithm for the abnormal URL detection depends on the presented in algorithm 1. The proposed model has the objective function, which is the cross-entropy-based cost function. The models' training aims to reduce the expected cross-entropy and the exact outputs for complete activities. Eq. (13) shows the below formula:

$$C = -\sum_i \sum_j y_i^j \ln a_i^j + \left(1 - y_i^j\right) \ln\left(1 - a_i^j\right) \tag{13}$$

Here, the index of the phishing URL is i, and the traffic category is j. The phishing URL has the actual category, and the predicted category is y. The back-propagation algorithm calculates the Adam. The model is trained; having the adam is the backward propagation. The back-propagation is done for error differentials having the forward-backwards algorithm. The error differentials are calculated using the Back-Propagation Through Time (BPTT). The Back Propagation Through Time (BPTT) algorithm is used in the proposed system to attain the objective function derivatives related to all weights and reduce the objective function using the stochastic gradient descent.

5 Evaluation Metrics

The below measurements related to performance are used for determining the suggested system and other models such as recall, accuracy, F-measure and precision. The number of legal URLs is accurately labelled as legal, plus the count of phishing URLs that are accurately labelled as phishing is termed as the accuracy of the total number of samples of the test set. The calculation of accuracy is provided in Eq. (14):

$$Accuracy,\ A = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}$$

The number of phishing URLs is accurately labelled as phishing, and then the total count of the labelled URLs as phishing is termed precision. Eq. (15) shows the calculation for precision.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

The number of phishing URLs is accurately labelled as phishing rather than the total count of exact URLs of phishing is termed as recall, which is referred to as sensitivity and TPR. Eq. (16) shows the calculation for recall.

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

The rates of precision and recall have the weighted harmonic mean termed the F-measure. Methodology having the greater F-measure is more efficient. Eq. (17) shows the calculation for F-measure.

$$F = \frac{2 * precision * recall}{precision + recall} \tag{17}$$

The phishing URLs are accurately labelled as phishing URLs as indicated by true positives (TPs), and the legal URLs are inaccurately labelled as phishing URLs as indicated by false positives (FPs); the legal URLs are correctly labelled as legal URLs as indicated by true negatives (TNs), and phishing URLs are inaccurately labelled as legal URLs is indicated by false negatives (FNs).

5.1 Comparative Analysis

The comparison of the suggested model performance having P-CNN, P-GAN and P-RCNN models is made in the proposed model for verifying the capability of the model for determining the URLs of phishing. The phishing detection technique is P-CNN based on the website URL only. CNN makes the extraction of character-level feature representation of the URS. The dataset called MUPD is used by P-CNN, which has 2,353,933 phishing and 2,220,853 legal URLs. PhishTank is the source for phishing website URLs.

Efficiency is calculated using prediction accuracy, as in Table 1. The performance of the anticipated model is compared with SVM, RF, NB and DT. The proposed model gives 98.5% accuracy, 98.8% precision, 98.3% recall and 98.7% F-measure as in Fig. 2. Legal websites are the source of the top 4 million domain list from DomCop. CNN is used in the proposed model as the important baseline model because of the different similarities to the proposed model. Primarily, this lies in the URLs, making it the same as the situation. The same dataset is used for the training, validation, and testing, raising confidence in the comparison results. The phishing website detection technique only lies on the URL and is termed RCNN. The URL information is encoded into the two-dimensional tensor, and the tensor is fed into the deep learning neural network for classifying the original URL. The constructed tensor's global features are extracted using a proposed network.

Table 1: Result analysis

Dataset	Model	Accuracy	Precision	Recall	F-measure
MUD	Support vector machine (SVM)	96%	96%	96.8%	96.4%
Synthetic phishing URL + MUPD (5000)	Random forest (RF)	96.5%	95.8%	96.8%	96.2%
Synthetic phishing URL + MUPD (10,000)	Naïve bayes (NB)	96.54%	96.8%	96.3%	96.5%
Synthetic phishing URL + MUPD (50,000)	Decision tree (DT)	97.5%	98%	97.2%	97.6%
MUPD (Proposed)		98.5%	98.8%	98.3%	98.7%

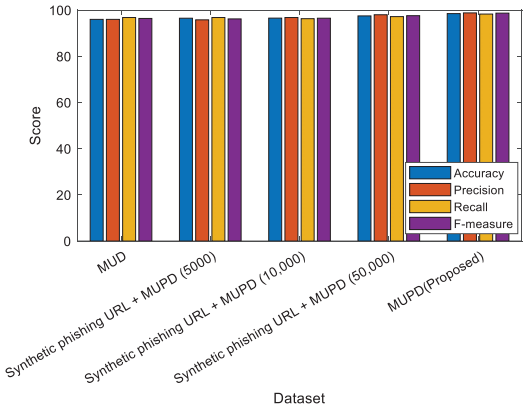

Figure 2: Result analysis

Local features are extracted by using CNN. The dataset having nearly 500,000 URLs is used by P-RCNN, obtained via PhishTank and Alexa. Due to the workflow, which depends on the URLs and is the same as the discriminator model, like the character level CNN, the second baseline model, P-RCNN, is used in the proposed $d - FBLSTM$ system.

5.2 Analysis

The test dataset evaluates the model performance and compares the two chosen baseline models and the proposed technique. The model has the results on the original test dataset presented in Table 1 and on the 10,000 tests of synthetic URLs of phishing. As seen from the confusion matrix, the phishing URLs of 236,732 are accurately classified as phishing URLs, and 4,771 legal URLs are inaccurately classified for the FPR of 21% only. The accuracy measure is considered for the comparison, and the accuracy is considered the famous measure in the proposed $d - FBLSTM$ system. Moreover, another performance measure is provided in the proposed system and the baseline models presented in Fig. 3. The suggested model is seen for detecting phishing URLs with greater precision, accuracy, F-measure, and recall scores over two baseline models. The proposed $d-FBLSTM$ model has a precision value near 988.8%, which shows the supreme performance for the suggested model with the various variations learned from the phishing features for creating the other URLs that are not learned using the intermediate layers. The accuracy of 98.5% is obtained by the proposed model, which performs well than the other two baseline models. The suggested model is efficient and achieves the greatest accuracy among other compared systems based on the outcomes. The interpretation is made from the capability for exploring the URLs over the original dataset and the decidability of the model discovering the phishing URLs. The suggested model efficiently combines the merits of both CNN and LSTM models. Also, the suggested model can give better outcomes when considering URLs only to detect phishing websites using confusion matrix as in Table 2.

http://login.paypal.com-casepp-96616.11qnt.info/login.htm
http://clientform.ref13560903351.bbt.com.dllsoro.cn/clients/data/proc.jsp
http://businessbanking.53.com.session0690244.tenpost.cn/clientbase/form.asp
http://business-eb.client86825907-
form.bbt.com.tenipp.cn/clients/form/b_form.jsp

Figure 3: Sample-generated phishing URL

Table 2: Confusion matrix

	Phishing	Legitimate	Total
Phishing	236, 733 (97%)	6,635 (2.8%)	243,365
Legitimate	4,770 (2.2%)	223, 425 (97.95%)	228, 195
Total	241, 504	230, 056	471, 565

A few examples of synthetic and real phishing URL is shown in Fig. 3. In the top list, the created phishing URLs have the same features as those presented in the bottom list. The proposed system accurately obtained the general URL structure like domains, hostname, etc. Moreover, a few URL details need to be explained more clearly and understood completely. The dropout approach is used to provide the result which prevents over-fitting (See Table 3 and Fig. 4).

Table 3: Performance of ***d − FBLSTM*** over different scales in the provided dataset

Dataset split	Accuracy	Precision	Recall	F-measure
10%	98.45	97.45	97.11	92.92
20%	98.82	97.46	96.28	94.36
30%	97.07	97.69	95.21	94.40
40%	97.87	98.71	94.92	95.32
50%	98.60	98.57	96.18	96.03
60%	98.63	97.25	95.18	96.03
70%	97.52	98.11	97.42	96.25
80%	98.19	97.60	97.61	96.59
90%	98.21	97.57	96.63	96.60
100%	98.45	98.78	99.02	97.85

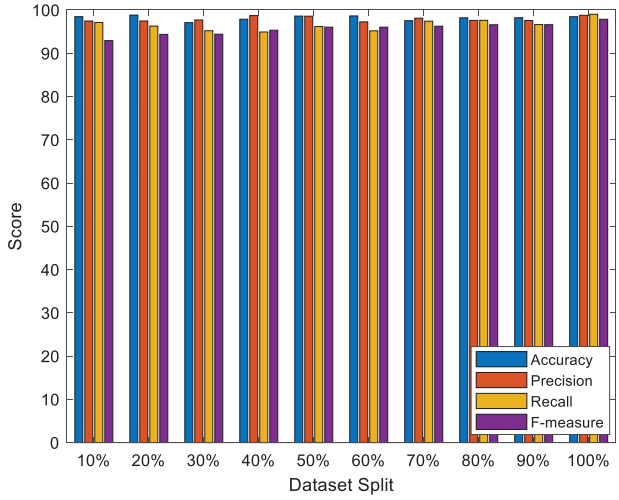

Figure 4: Performance analysis on a different scale

Moreover, the phishing URLs are successfully found by the suggested model. Various amounts of representations of phishing URLs are obtained after the completion of the training phase. 10,000, 50,000, and 100,000 synthetic phishing URLs are generated in the proposed system. Every set is divided into a ratio of 0.2 testing and 0.8 training and combined with the original testing set and training set. All the performance metric in the suggested model for the three generated sets and the original dataset is shown in Table 4. Fig. 5 depicts the anticipated model's comparison with 99.3% accuracy, 98.6% precision, 98.6% recall and 98% F-measure for 100%. Generally, data partitioning is done at 80:20, 70:30 or 90:20 for testing and training models. With a 70:30 ratio, the model gives 99% accuracy and precision and 98.2% recall and 98.4% F-measure. With an 80:20 ratio, the model gives 99.1% accuracy, 98.9% precision, 98.5% recall and 98.9% F-measure. With a 90:10 ratio, the model gives 99.2% accuracy, 98.7% precision, 98.6% recall and 98.7% F-measure.

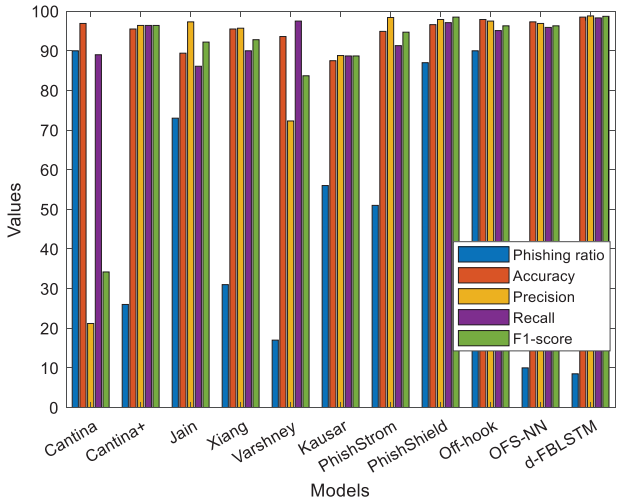

Figure 5: Performance analysis with different approaches

Table 4: Performance comparison among $d - FBLSTM$ and existing approaches

Models	Testing data		Phishing ratio	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
	Legitimate	Phishing					
Cantina	2100	20	90%	96.9	21.2	89	34.2
Cantina+	1689	945	26%	95.5	96.4	96.4	96.4
Jain	406	1125	73%	89.4	97.3	86.1	92.2
Xiang	7905	3542	31%	95.5	95.7	90	92.8
Varshney	2501	505	17%	93.6	72.3	97.5	83.7
Kausar	72	90	56%	87.5	88.8	88.7	88.7
PhishStrom	48008	48010	51%	94.9	98.4	91.3	94.7
PhishShield	251	1605	87%	96.6	97.9	97.1	98.5
Off-hook	200000	2005	90%	97.9	97.5	95.1	96.3
OFS-NN	13107	1480	10%	97.3	96.9	95.9	96.3
$d - FBLSTM$	13307	1275	8.5%	98.5	98.8	98.3	98.7

All the experiments have a considerable outcome in the model even though the low accuracy is obtained having the original MUPD dataset. The phishing websites detected in the suggested model have the greatest precision, accuracy, F-measure, and recall scores on the dataset of MUPD, plus the synthetic URLs with 50,000 counts compared with other demonstrations. The proposed model gives the enhanced classification results, and how the model does this enhancement due to the exploration of the module with other phishing URLs cannot learn (See Figs. 6 and 7).

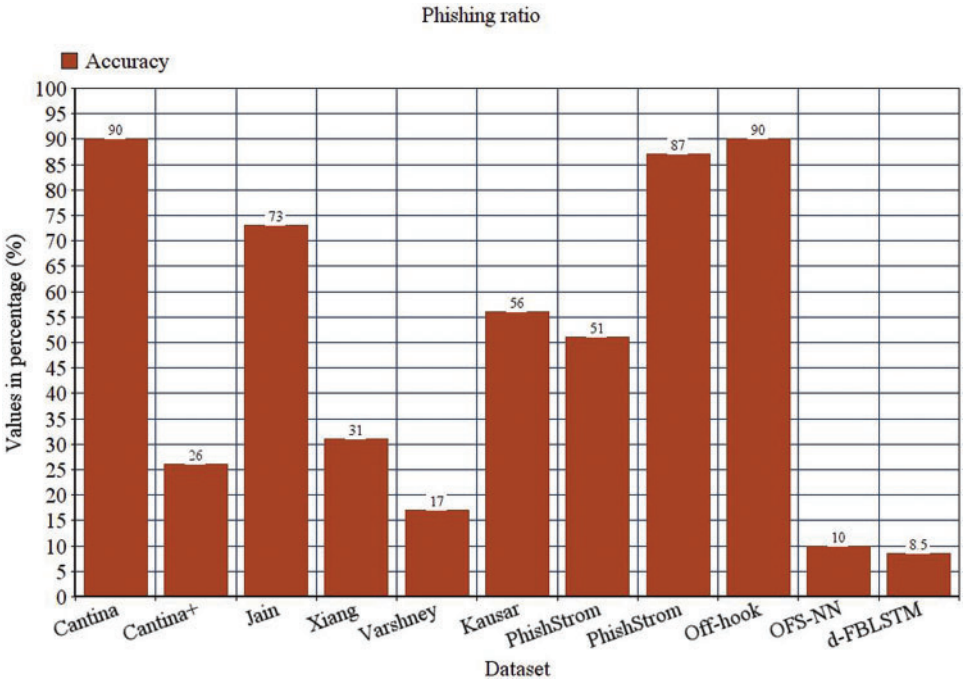

Figure 6: Phishing ratio

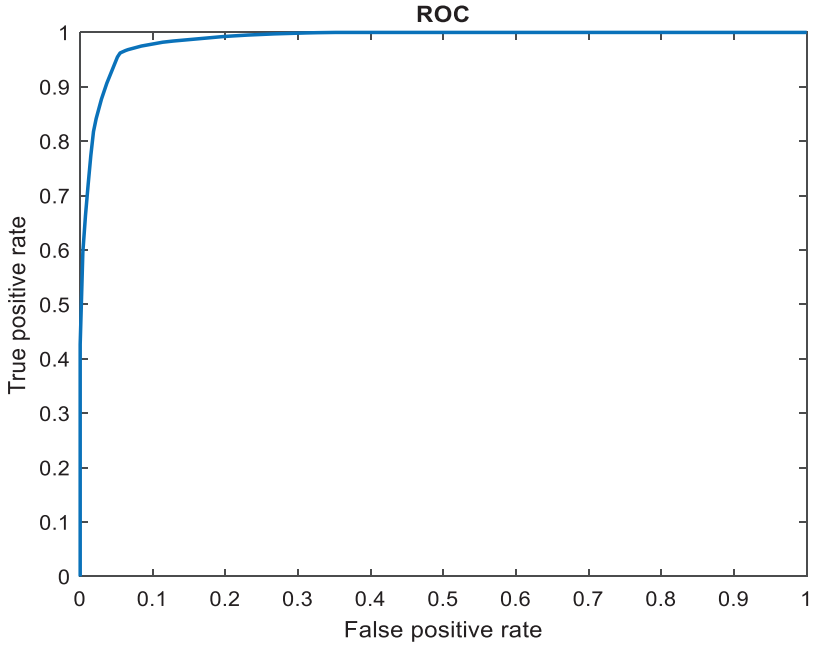

Figure 7: ROC computation

6 Conclusion

A new phishing detection model, $d-FBLSTM$, is presented in the proposed model. The proposed $d-FBLSTM$ model analyzes a URL, and the related webpage is classified as legal or phishing. The first module generates synthetic phishing URLs. URL is legal, or the secondary module decides to phish. The content of the webpage or the third-party services is independent of the suggested model based on the URL of the website for obtaining a good phishing detection rate. However, the capability for distinguishing the phishing URLs is enhanced by the process in the suggested model using the exploration of other phishing URLs that are not associated with the training dataset. The models found the phishing URLs successfully, even though some URL details have inaccurate semantic data and need to be understood fully. Various experiments are performed on the huge dataset, which has two million phishing URLs and the legal URLs, divided into training, validation, and testing datasets. The precision of 98.8% and accuracy of 98.5% are obtained in the suggested model with no dependence on third-party services and higher accuracy over other compared systems. The outcomes demonstrate the improved classification result, and the suggested model detects phishing URLs. The complexity of the model is calculated to enrich the comparison among various models in future. The suggested model has the scope to be expanded to cover the similarity-based techniques visually. Also, the character level similarity impacts are analyzed in the proposed system among different URL components for a better synthetic phishing URL representation.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] J. A. Chaudhry, S. A. Chaudhry and R. G. Rittenhouse, "Phishing attacks and defences," *International Journal of Security and its Applications*, vol. 10, no. 1, pp. 247–256, 2016.

[2] M. Hijji and G. Alam, "A multivocal literature review on growing social engineering-based cyber-attacks/threats during the COVID-19 pandemic: Challenges and prospective solutions," *IEEE Access*, vol. 9, pp. 7152–7169, 2021.

[3] R. Chen, J. Gaia and H. R. Rao, "An examination of the effect of recent phishing encounter on phishing susceptibility," *Decision Support Systems*, vol. 133, pp. 113287, 2020.

[4] S. Bell and P. Komisarczuk, "An analysis of phishing blocklists: Google safe browsing, OpenPhish, and PhishTank," in *Proc. Australasian Computer Science Week Multiconf.*, Melbourne, Australia, pp. 1–11, 2020.

[5] L. Li, E. Berki, M. Helenius and S. Ovaska, "Towards a contingency approach with allowlist- and blacklist-based anti-phishing applications: What do usability tests indicate?" *Behaviour & Information Technology*, vol. 33, no. 11, pp. 1136–1147, 2014.

[6] R. S. Rao and A. R. Pais, "Jail-phish: An improved search engine based phishing detection system," *Computers & Security*, vol. 83, pp. 246–267, 2019.

[7] K. L. Chiew, E. H. Chang, C. Lin Tan, J. Abdullah and K. S. C. Yong, "Building standard offline anti-phishing dataset for benchmarking," *International Journal of Engineering & Technology*, vol. 7, no. 4, pp. 7–14, 2018.

[8] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive blocklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, San Diego, USA, pp. 1–5, 2010.

[9] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 1–11, 2016.

[10] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.

[11] Y. Li, Z. Yang, X. Chen, H. Yuan and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[12] G. Xiang, J. Hong, C. P. Rose and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing websites," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 2, pp. 1–28, 2011.

[13] M. A. Adebowale, K. T. Lwin, E. Sánchez and M. A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated images, frames and text," *Expert Systems with Applications*, vol. 115, pp. 300–313, 2019.

[14] A. Al-Alyan and S. Al-Ahmadi, "Robust URL phishing detection based on deep learning," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 7, pp. 2752–2768, 2020.

[15] X. Zhang, J. Zhao and Y. Lecun, "Character-level convolutional networks for text classification," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, vol. 28, pp. 649–657, 2015.

[16] R. S. Rao, T. Vaishnavi and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 813–825, 2020.

[17] M. W. Al-Nabki, E. Fidalgo, E. Alegre and R. Aláiz-Rodríguez, "File name classification approach to identify child sexual abuse," in *Proc. 9th ICPRAM*, Valletta, Malta, pp. 228–234, 2020.

[18] C. Peersman, C. Schulze, A. Rashid, M. Brennan and C. Fischer, "ICOP: Live forensics to reveal previously unknown criminal media on P2P networks," *Digital Investigation*, vol. 18, pp. 50–64, 2016.

[19] S. Marchal, J. Francois, R. State and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[20] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione *et al.,* "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, pp. 47–57, 2021.

[21] W. Wang, F. Zhang, X. Luo and S. Zhang, "PDRCNN: Precise phishing detection with recurrent convolutional neural networks," *Security and Communication Networks*, vol. 2019, pp. 1–15, 2019.

[22] M. Sabahno and F. Safari, "ISHO: Improved spotted hyena optimization algorithm for phishing website detection," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 34677–34696, 2022. https://doi.org/10.1007/s11042-021-10678-6

[23] S. J. Bu and S. B. Cho, "Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection," *Electronics*, vol. 10, no. 12, pp. 1492, 2021.

[24] M. Somesha, A. R. Pais, R. S. Rao and V. S. Rathour, "Efficient deep learning techniques for detecting phishing websites," *Sādhanā*, vol. 45, no. 1, pp. 1–18, 2020.

[25] D. N. Atimorathanna, T. S. Ranaweera, R. A. H. Devdunie Pabasara, J. R. Perera and K. Y. Abeywardena, "NoFish; total anti-phishing protection system," in *Proc. 2nd ICAC*, Colombo, Sri Lanka, 2020.

[26] S. Maurya, H. Singh and A. Jain, "Browser extension based hybrid anti-phishing framework using feature selection," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, pp. 1–10, 2019.

[27] K. M. Sundaram, R. Sasikumar, A. S. Meghana, A. Anuja and C. Praneetha, "Detecting phishing websites using an efficient feature-based machine learning framework," *Revista Gestão Inovação e Tecnologias*, vol. 11, no. 2, pp. 2106–2112, 2021.

[28] O. Abiodun, A. S. Sodiya and S. O. Kareem, "Linkcalculator—An efficient link-based phishing detection tool," *Acta Informatica Malaysia*, vol. 4, no. 2, pp. 37–44, 2020.

[29] S. Seo, C. Kim, H. Kim, K. Mo and P. Kang, "Comparative study of deep learning-based sentiment classification," *IEEE Access*, vol. 8, pp. 6861–6875, 2020.

[30] M. Chatterjee and A. S. Name, "Detecting phishing websites through deep reinforcement learning," in *Proc. IEEE 43rd Annual COMPSAC*, Milwaukee, USA, vol. 2, pp. 227–232, 2019.

[31] M. N. Amrutha and R. Santhosh, "A survey on security breaches of data availability and data integrity in cloud computing," *Karpagam Journal of Computer Science*, vol. 14, no. 5, pp. 225–231, 2020.

[32] S. Rajeshwari and R. Santhosh, "Security and privacy in emerging wireless networks with mobile sinks," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 2, no. 2, pp. 1–10, 2013.

[33] R. Santhosh and M. Shalini, "Security enhancement using a chaotic map and secure encryption transmission for wireless sensor networks," *International Journal of Engineering and Technology*, vol. 9, no. 2, pp. 689–694, 2017.

[34] P. Sherubha, S. P. Sasirekha, V. Manikandan, K. Gowsic and N. Mohanasundaram, "Graph-based event measurement for analyzing distributed anomalies in sensor networks," *Sådhanå*, vol. 14, no. 1, pp. 1–5, 2020.

[35] P. Sherubha, "An efficient network threat detection and classification method using ANP-MVPS algorithm in wireless sensor networks," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 1597–1606, 2019.

[36] P. Sherubha, "An efficient intrusion detection and authentication mechanism for detecting clone attack in wireless sensor networks," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 5, pp. 55–68, 2019.

[37] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, "TBE-Net: A three-branch embedding network with the part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14557–14569, 2022.

[38] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. 52, no. 8, pp. 8448–8463, 2022.