PHISHING URL DETECTION USING LSTM BASED ENSEMBLE LEARNING APPROACHES

Bireswar Banik and Abhijit Sarma

Department of Computer Science, Gauhati University, Guwahati, India

ABSTRACT

Increasing incidents of phishing attacks tempt a significant challenge for cybersecurity personals. Phishing is a deceitful venture with an intention to steal confidential information of an organization or an individual. Many works have been performed to build anti-phishing solutions over the years, but attackers are coming with new manoeuvres from time to time. Many of the existing techniques are experimented based on limited set of URLs and dependent on other software to collect domain related information of the URLs. In this paper, with an aim to build a more accurate and effective phishing attack detection system, we used the concept of ensemble learning using Long Short-Term Memory (LSTM) models. We proposed ensemble of LSTM models using bagging approach and stacking approach. For performing classification using LSTM method, no separate feature extraction is done. Ensemble models are built integrating the predictions of multiple LSTM models. Performances of proposed ensemble LSTM methods are compared with five different machine learning classification methods. To implement these machine learning algorithms, different URL based lexical features are extracted. Mutual Information based feature selection algorithm is used to select more relevant features to perform classifications. Both the bagging and the stacking approaches of ensemble learning using LSTM models outperform other machine learning techniques. The results are compared with other anti-phishing solutions implemented using deep learning methods. Our approaches have proved to be the more accurate one with a low false positive rate of less than 0.15% performed comparatively on a larger dataset.

KEYWORDS

Cyber security, Phishing attack, Machine learning, LSTM, Ensemble learning

1. INTRODUCTION

Our dependence on the Internet is growing rapidly. Many industries operate entirely on the web. Amidst the recent pandemic situation, the internet has become the key driving force. Parallel to this, criminal activities in cyberspace are also picking up speed. Phishing attacks are rampant and happen frequently every day. Attackers try to gather the personnel credentials of the users in different ways. Users may become victim of phishing attacks by clicking a link in a webpage which is redirected to a fraudulent site [1]. Attackers may try to attract users to click those links by showing greedy approaches like an exclusive offer on the products the user looking for. Attackers ask users personnel information like banking details including passwords or PINs and users enter the information by trusting the page as legitimate [2]. As they clicked on submit button, the information goes to the attacker's database.

Before implementing an attack, attackers gather different background knowledge on a particular business plan, choices of the users, products on which particular user is interested, or other geographical and economical information [2]. The attacking pages are designed in a way that it seems like a legitimate page user previously accessed [1]. Though the URLs of the malicious pages are different from the original URL, but users generally do not visually verify the URL.

Thus, users cannot detect the fact that the domain name of the URL is not the actual one the user looking for, or the subdomain length may longer than usual etc. and become the victim of phishing attacks easily.

The Anti-Phishing Working Group (APWG) publishes a phishing activity report quarterly in every year. According to the report [3], the common perpetrators of phishing attacks are Webmail, SAAS, financial institutions, and payment services. Employees of several companies have suffered from Business Email Compromise (BEC) attacks using bonus deductions, gift cards etc. As per the report, the phishers took the opportunity of pandemic situations, COVID-19 theme-based phishing attacks were started in March 2020. Online activity grows rapidly as people get trap in their homes. The attackers took this chance to target the healthcare sector, online video conferencing applications, and Business Email Compromise (BEC) attacks. This trend continues and highest number of phishing attacks ever have detected in 1st quarter of 2022. So, it has become a global challenge for the researchers to propose a significant and reliable method to prevent the users from being victim of phishing attacks.

In this paper, we use LSTM networks to detect phishing URLs, and ensemble learning approaches using LSTM models are used to improve the results. LSTM method is an improved version of Recurrent Neural Network (RNN) which establishes recurrent relationship to itself and learns from the previous results [4]. RNN can't process long sequences of previous inputs, it can't decide which information to remember whereas LSTM has the ability to remember both long and short sequences of past input data [5]. LSTM technique is proposed by Hochreiter and Schmidhuber et al. in [6] for solving the long-term dependency problem. LSTM network is composed of memory cells [4] where the cell state keeps track of values over arbitrary intervals. LSTM network has its ability to decide what information from past state require to pass to the next state and forgets the irrelevant information. We have considered this method in our work because of its efficiency and dynamicity in many fields. The important contributions of our works in this paper are stated below:

i) As per our knowledge, we have proposed the ensemble learning using only LSTM models for building phishing URL detection system for the first time.
ii) A considerably larger dataset of 247,064 URLs is taken and extensively experimented with various parameters to build an optimal and stable LSTM network. We achieve an accuracy and F-score value of greater than 99.5% using bagging and stacking approaches of ensemble learning by combining multiple LSTM models.

The rest of the paper is structured as follows: works of earlier published works are described in Section 2. Section 3 describes the methodologies of our proposed approaches. The next section explains how the experimental set-up is done. Section 5 explains the results obtained using different methods and performs the comparison of our approaches with other works. Finally, the conclusion of our works is described in Section 6.

2. EXISTING WORKS

Different works have been done for phishing URL detection by researchers. The main problem for the researchers is that the attackers may target different domains each time. So, it is hard to stay on a particular method for a long time. In this section, we discuss some of the standard works using various methods for detecting phishing URLs.

2.1. List Based Approach

In this approach, a set of URLs, domain names or IP addresses is maintained. When a user tries to access a URL, the system checks whether the URL is present in the list or not. It is of two types: Blacklisting and Whitelisting approach.

2.1.1. Blacklisting approach

Blacklisting approach is one of the traditional approaches where the researchers maintain a list of previously detected malicious domain names, URLs, or IP addresses. When a user is going to access a particular URL, the system checks whether it is blacklisted or not. If it is, it will prevent the user from accessing that URL [7]. The problem with the blacklisting approach is that if the URL is slightly modified, it may fail to detect. To detect those URLs, which are modified a bit from earlier detected phishing URLs, an approach is proposed by Prakash et al. known as phishnet [8]. Phisnet is composed of two components: The first one checks the five heuristics of URLs which predicts if there is a similarity with previously detected phishing URLs. The second component performs an approximate matching algorithm with the entities namely hostnames, IP address, brand names, and directory structure of URLs. The experiment is performed with 6000 blacklisted URLs and 18000 new phishing URLs are detected and obtained false positive and false negative rate of 3% and 5% respectively.

2.1.2. Whitelisting approach

In whitelisting approach, a list of safe URLs is maintained. An approach called Automated Individual White List (AIWL) is proposed by Cao et al. [9], maintains a list of user's familiar login interfaces of websites. This will alert users when they attempt to access a page that is not whitelisted. Naïve Bayes classifier is used to maintain the list automatically. Jain and Gupta [10] proposed the technique of maintaining a whitelist containing domain names and IP addresses. If a particular URL is not listed, its classification is done depending on three different parameters related to the presence of the hyperlinks on the web page. This approach obtained accuracy of 89.38% using a dataset of 1525 URLs only. Listing approach as it is not effective in case of zero-hour attacks as malicious URLs may have a shorter lifespan. Attackers may target the URL, marked as safe a few hours ago.

2.2. Visual Similarity Approach

This approach compares the visual similarity between a legitimate webpage and a phishing webpage. Medvet et al. in the paper [11] perform phishing detection using three features on visual similarity namely text content, its style, image embedded. For this, they have considered only 41 malicious pages with their corresponding legitimate pages. They extract a signature from a suspicious webpage and perform matching with the corresponding legitimate page using wavelet transformation. Jain and Gupta in their paper [12] detects the phishing URLs based on the properties like text contained in the pages, text formatting style, position and size of images present in the page, matching in CSS, HTML tags etc.

2.3. Heuristic Based Approach

This technique detects the phishing URLs following a set of rules learned from the earlier results and experiences. Popular web browsers like Internet Explorer, Mozilla Firefox use a heuristic based technique for malicious site detection [7]. Jeeva and Rajsingh [13] proposed phishing URL detections based on heuristics defined from 14 features from URL. The experiments are

performed using associative rule mining techniques, apriori and predictive apriori algorithms for 1400 URLs only and obtained overall accuracy of 93% using apriori algorithm.

2.4. Machine Learning Approach

Many of the researchers have implemented different machine learning techniques for phishing URL detection. Heuristics approaches also used in machine learning techniques. Methods like decision tree, support vector machine, logistics regression, random forest, etc. are used by different researchers [14,15]. D. Sahoo et al. performed a detailed survey of malicious URL detection using machine learning methods in [16]. They presented different types of features used for phishing and legitimate URL classification like lexical features, host-based features, content-based features, etc. Yang et al. [17] proposed phishing website detection by integrating random forest and Convolutional Neural Network (CNN) methods. Random Forest method is used for classification of URLs based on features automatically extracted by CNN. Experiments are performed on two different datasets of 47210 and 83857 numbers of URLs.

Deep learning approach using recurrent neural network for classifying phishing URLs is proposed by Bahnsen et al. [4]. They built the LSTM model where character sequence of URL is taken as input. The model predicts whether the URL is phishing or not. Chen et al. [18] also used LSTM RNN method to detect phishing sites. They performed the experiments using 10 features. Wang et al. [19] proposed a method known as PDRCNN to detect phishing sites which first used bidirectional LSTM network to extract global features and then convolutional neural network to extract local features from URLs. They combined the features extracted by these two methods and use sigmoid function to classify the URLs. Priya et al. [20] implemented a weight-based ensemble learning approach known as DeepEEviNNet taking Radial Basis Function (RBF), Generalized RBF, Probabilistic Neural Network (PNN) and Heteroscedastic PNN as base classifiers. Dampster Shafer Theory (DST) is used to find the optimal weight for each classifier. Experiments are performed in the dataset consisting of 4654 phishing URLs and 5839 legitimate URLs only.

2.5. Summary of Existing Approaches

We survey various existing works performed to detect phishing URLs using different methods based on various features. It is observed that the problem with the host-based features is that the researchers have to depend on third-party software for retrieving feature values. For extracting the source code of the webpage in real-time, lot of time and space is needed. Most of the sites detected as phishing are immediately blocked, so accessing their content is also difficult. Many researchers have used a limited set of URLs, for implementing phishing URL detection method using content-based and host-based features. So, in this paper, our proposed system is implemented using ensemble learning on LSTM methods taking only URL as input where no manual extraction of feature is required. The experiments are performed on a dataset of more than 2.5 lakh URLs.

3. PROPOSED METHODOLOGIES

In this section, we present the working models of our works. We have proposed LSTM ensemble method using bagging and stacking approach. Ensemble learning methods are composed of two parts. First, building of base learner models using LSTM. Second, combining of models based on their predictions. Performances of ensemble learning approaches are compared with other machine learning techniques which are implemented using lexical features of URLs. Features that are used to implement other machine learning techniques are also discussed here. Mutual

Information based feature selection method is used for ranking and selecting relevant lexical features.

3.1. Building of Base Learner Models using LSTM

In this work, base models are created using LSTM networks. The structure of our base learner models using LSTM is shown in Figure. 1. This deep LSTM network is built using multiple hidden layers. The input layer is a sequential layer. Between input layer and output layer of a model, we have added two LSTM hidden layers. Each hidden layer is composed of multiple memory cells. These layers are followed by multiple fully connected dense layers. Multiple LSTM layers and dense layers are added with an aim to build more accurate, stable and expressive deep network to classify URLs.

For implementing LSTM method, URLs are taken as input. No manual extraction of features is involved. LSTM algorithm learns automatically from the sequence of characters present in the URL. First, URLs are processed to convert into tokens. URLs are converted into list of tokens using tokenization process which convert each unique character of a sequence to a specified number. Lists of tokens are given as input in the LSTM base models. The output layer predicts the final output i.e., whether the URL is phishing or not. Experiments are performed to evaluate optimized number of LSTM units, number of neurons in each dense layers and number of dense layers added in each network. Models are tuned with different number of epochs.

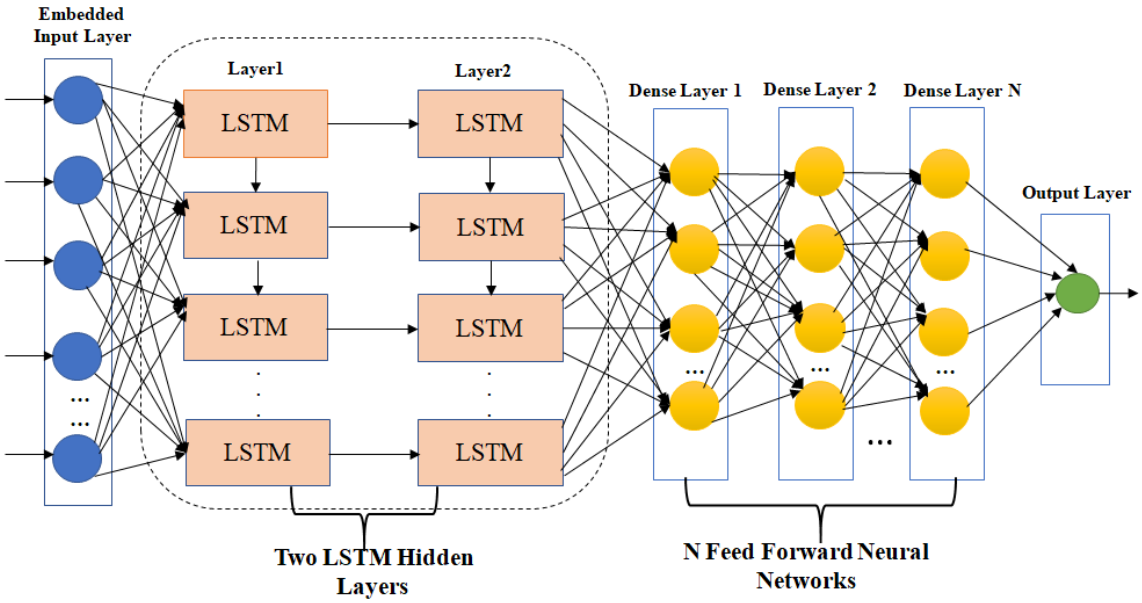


Figure 1. Structure of LSTM model of our proposed system

3.2. Ensemble of LSTM Models

Ensemble learning is a method to improve the performances of a computational problem by integrating several models of same or different machine learning classifier algorithms [14, 21]. In this paper, we have used ensemble learning with two different approaches taking LSTM methods as base learner: bagging approach and stacking approach.

3.2.1. Proposed Ensemble LSTM model using Bagging approach

In bootstrap aggregating (bagging) approach, multiple subsets of training data are created randomly where repetition of training records is allowed [14]. It helps to decrease the variance of the prediction of class labels and improves the stability and accuracy of the classification algorithms [21, 22].

The working of the LSTM ensemble method using bagging approach is shown in Figure 2. This diagram shows the approach in two parts. In first part, n LSTM models are built and fitted using n subsets of training records. In second part, a set of records is given as input to each model for testing purposes. Each model predicts the class that a particular test record belongs to. The proposed ensemble LSTM model using bagging approach predicts based on maximum votes i.e., the class that most model predicts for a particular test record is predicted as of that class by the ensemble model.

For example, a particular test record is given as input to five LSTM models, which have been earlier fitted with 5 different subsets of training records. Suppose, out of those five LSTM models, four models predict the test record as of phishing class and one model predicts as of legitimate class, the ensemble method using bagging approach will predict that record as phishing as it is predicted by majority of individual models.

3.2.2. Proposed Ensemble LSTM model using Stacking approach

Ensemble learning using stacking approach combines predictions of multiple models to generate a new model known as stacked model. It is a meta-learning model [14]. It is used to learn how to best combine the predicted values of multiple machine learning models [21, 23]. This approach is used in our work because of its capability to best combine the various models and for more accurate prediction capability than any other single model.

The stacked model does not deal with raw feature values of any dataset. In this paper, the stacked dataset is built based on the probability values predicted by the base models. A linear model is commonly used as meta model for analysing the predictions of base models [23]. In this paper, Logistic Regression (LR) algorithm is used to build the meta models as the algorithm is simple and performs well in binary classification.

To understand the working of ensemble learning using stacking approach, let us consider a sample dataset of 10,000 records of two different classes. Let the dataset is split into two parts: one part of 9000 records for training purpose and another part of 1000 records for testing purpose. Binary classifications are performed using LSTM method n times (say, n=5) and we get n single LSTM models. The working of our proposed ensemble LSTM model using stacking approach is described with this example in Figure 3.

LSTM models are fitted using the corresponding set of training records. The stacked dataset for training is created by merging the predicted probability values of n single LSTM models. Each single LSTM model gives two predicted probability values for each training record: one for belonging to a phishing class and other for belonging to a legitimate class. So, the stacked dataset for training will have 10 (= 5 × 2) feature values with 9000 records where the number of models is five. Similarly, the stacked dataset for testing also consists of 10 feature values with 1000 records. The meta model is fitted using the stacked dataset for training. For prediction of the final result, the stacked dataset for testing is used which will predict the classes of all 1000 testing records.

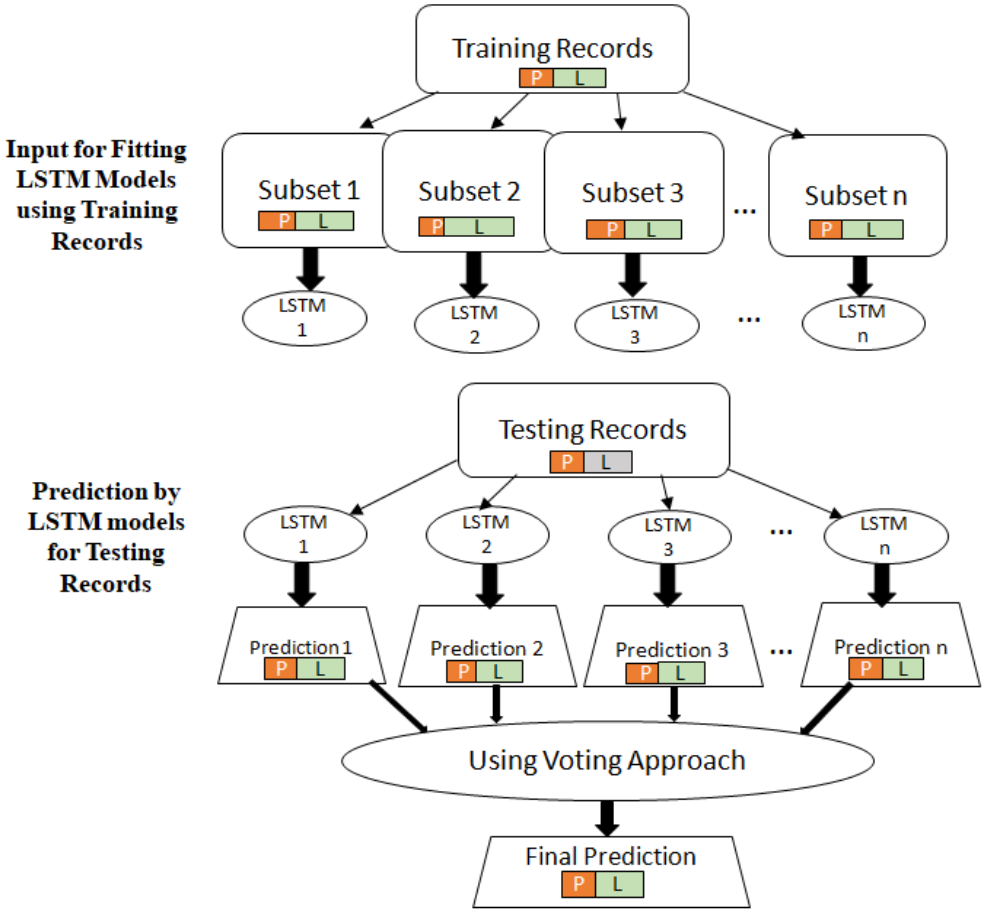


Figure 2. Working model of proposed LSTM ensemble networks using bagging approach

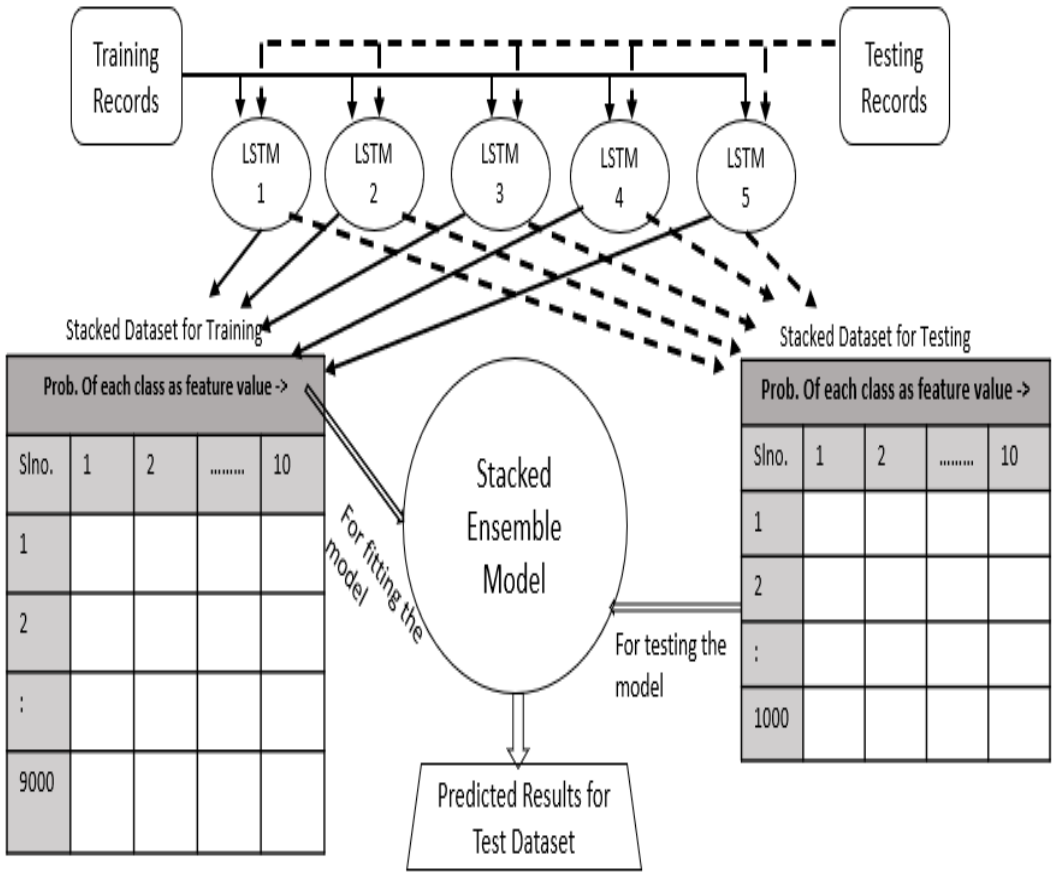


Figure 3. Working model of proposed LSTM ensemble networks using stacking approach

3.3. Lexical Feature Selection for Implementing other ML Algorithms

In this paper, the performances of proposed ensemble LSTM approaches are compared with five different machine learning techniques. These techniques are implemented using the same dataset based on the lexical properties of URLs. For this, 12 lexical features of URLs are extracted. The features commonly used by various researchers [4,13,14,18-20] are considered. Mutual information based feature selection algorithm is used to select the most relevant features among them. This method ranks the features on their importance to classify using Information Gain (IG) which is also called as Expected Mutual Information [24]. It is calculated using two-sided metrics where X and Y are two random variables, X represents any feature value and Y represents the class labels [24,25]:

$$I(X;Y)=\sum_x \sum_y \boldsymbol{P(x,y)log}\frac{\boldsymbol{P(x,y)}}{\boldsymbol{P(x)P(y)}} \quad (1)$$

Where discrete value of X and Y is represented by x and y. The marginality distribution functions of X and Y are denoted by P(x) and P(y). The joint probability distribution of X and Y are denoted by P(x,y). The features are ranked on their higher IG value. This value shows the importance of a feature to predict a class. The 12 features considered in this work are described in Table 1 rank wise with their IG values. Experiments were performed taking different number of features each time. It is found that the features with lower rank are not contributing much to

determine an URL as phishing or legitimate. So, we have considered the first seven features having higher IG values from Table 1 for further evaluation.

Table 1. Detail of URL based lexical features for classification using machine learning methods

Feature ID	Name of the feature	Description	IG Value (%)
F1	Ratio of path length to total length of URL	This feature evaluates the ratio of total path length to the URL length. This ratio seems to be higher in phishing URLs.	21.39
F2	URL Length	This feature takes the length of the URL as average length of phishing URLs is observed much greater than legitimate URLs.	19.90
F3	Ratio of number of special characters to total length of URL	Number of special characters are comparatively more in phishing URLs. So, ratio of number of special characters to total length is considered as a feature.	19.09
F4	Number of suspicious keywords	This feature counts presence of words like recovery, validation, config, secure, verify, unblock, payment, login, submit, logon, signin, suspend, webscr, security, xdomain, wp-include, webhostapp, etc. which are commonly present in phishing URLs.	15.56
F5	Number of suspicious characters	This feature counts presence of no. of characters like '$', '!', '%', '*', '^', '@' etc. as their frequency is more in phishing URLs.	6.91
F6	Number of question Marks	This feature counts the number question marks present in URL.	6.20
F7	Query present	This feature checks the presence of any query in URL.	5.99
F8	Presence of @	It checks whether '@' symbol is present in the URL or not.	1.42
F9	Presence of http at middle	It checks whether the term 'http' is present in the URL or not.	1.14
F10	Presence of symbol in last character	It checks whether the last character of the URL is a symbol or not (except slash (/) symbol).	1.04
F11	Occurring of redirection (//)	It checks the presence of '//' in between the URL or not which indicates the redirection of URL .	0.87
F12	Presence of IP address	It finds whether any IP address in mentioned in the URL or not.	0.48

4. EXPERIMENTAL SET UP

In this section, we discuss how the set-up is done for performing the experiments for detecting phishing sites. First, how the raw data are collected and preparation of processed datasets are discussed. Then, the performance metrices used for evaluating the performances are presented.

4.1. Collection of Data

Phishing and legitimate URLs collected for preparing our datasets to perform phishing URL detection. In our paper, legitimate URLs are collected from DMOZ directory [26]. DMOZ is an open directory for world wide web links which was maintained under Open Directory Project.

Phishing URLs are collected from Phishtank [27]. Phishtank is community-based phishing verification sites and widely used by the researchers for preparing their datasets. Our dataset consists a total of 247064 URLS where 149991 URLs are legitimate and 97073 URLs are phishing.

4.2. Performance Metrics

In this paper, for measuring the performances of each technique, four parameters namely true positive (TP), true negative (TN), false negative (FN), and false positive (FP) values are evaluated. Where TP denotes the total number of phishing URLs correctly classified as phishing URLs, TN denotes the total number of legitimate URLs correctly classified as legitimate URLs, FP is the total number of legitimate URLs misclassified as phishing URLs and FN is the total number of phishing URLs misclassified as legitimate URLs. Using these values following metrices are evaluated formeasuring the performance of different methods [4,12,18]:

$$\text{True Positive Rate(TPR)}=\frac{TP}{TP+FN}*100\% \quad (2)$$

$$\text{True Negative Rate(TPR)} =\frac{TN}{TN+FP}*100\% \quad (3)$$

$$\text{FalsePositive Rate(TPR)= 100\% - TNR} \quad (4)$$

$$\text{False Negative Rate (TPR)= 100\% - TPR} \quad (5)$$

$$\text{Accuracy (ACC)}=\frac{TP+TN}{TP+FP+TN+FN}*100\% \quad (6)$$

$$\text{Precision (PRE)}=\frac{TP}{TP+FP}*100\% \quad (7)$$

$$\text{Recall (REC)= TPR} \quad (8)$$

$$FSC=2*(\frac{PRE*REC}{PRE+REC}) \quad (9)$$

5. RESULTS AND DISCUSSION

This section describes the experiments performed using the various methods with different parameters. The performances of these experiments are evaluated using equation (2) to (9). The first part of our work is building the base models. LSTM models are tuned with different number of LSTM units, number of dense neurons in each dense layer and number of dense layers. All these experiments are performed by trained and tested the whole dataset three times. The best results in terms of accuracy are shown in the tables Table 2 to Table 4. The LSTM models with optimal parameters are considered for building base models. Then, ensemble models are built using both the bagging and stacking approaches. Each ensemble model is created by merging n (where n=3,5,7) single LSTM models of same number of epochs. That means, an ensemble model of m (say) number of epochs means the particular model is built by combining n single base LSTM models of m epochs. Here, the value of m is 3,5,10,15 and 20. In the next part, the results obtained using ensemble methods are compared with various machine learning methods. The experiments are performed using different training-testing splitting ratios of the dataset. Finally, a comparison of our approaches with other related works on phishing URL detection is done.

5.1. Performance Tuning of Proposed LSTM Model

The LSTM models are tuned using various numbers of LSTM units. Table 2 shows the performances obtained using seven different number of LSTM units as 5, 10, 20, 50, 75, 100 and 150. These experiments are performed taking 90% of the dataset as training and the rest 10% of the dataset as testing. The performances increase as the number of units increases. But it does not improve significantly beyond the number of LSTM units as 50, whereas the computation time increases a lot.

Taking LSTM units as 50, we have trained the LSTM models with different number of neurons in dense layer of the LSTM network. Table 3 presents the performances obtained using various number of neurons in a single dense layer of LSTM network. Accuracy of 99.52% is obtained using the number of neurons in dense layer as only 50 which does not improve further with increase in the number of neurons in dense layers. So, the rest of the experiments in this work are performed using the number of LSTM unit as 50 only with 50 neurons in added dense layers.

All these experiments were performed taking a single dense layer in each LSTM network. But the results may vary with increase of number of layers. So, we have performed experiments as shown in Table 4 taking various number of dense layers. The best result is found taking number of dense layers as three. So, our base models are built using the LSTM structure consisting of number of LSTM units as 50 with three dense layers between LSTM layers where the number of neurons in each dense layer is also 50.

Table 2.Performances using different numbers of LSTM units

No. of LSTM units	ACC	TNR	REC	PRE	FSC
5	99.24	99.42	98.97	99.11	99.04
10	99.39	99.67	98.96	99.49	99.22
20	99.47	99.69	99.13	99.53	99.33
50	99.51	99.71	99.20	99.55	99.38
75	99.51	99.68	99.26	99.51	99.38
100	99.49	99.94	98.81	99.91	99.36
150	99.50	99.65	99.27	99.47	99.37

Table 3. Performances using different numbers of neurons in dense layers of LSTM network

No. of neurons in dense layers	ACC	TNR	REC	PRE	FSC
10	99.45	99.69	99.07	99.53	99.30
25	99.51	99.71	99.20	99.55	99.37
50	99.52	99.71	99.22	99.56	99.39
75	99.52	99.84	99.03	99.75	99.39
100	99.52	99.71	99.23	99.56	99.39
200	99.51	99.93	98.87	99.90	99.38
500	99.51	99.75	99.16	99.61	99.38

Table 4.Performances using different numbers of dense layers between LSTM layers

No. of dense layers	ACC	TNR	REC	PRE	FSC
1	99.52	99.71	99.22	99.56	99.39
2	99.52	99.63	99.36	99.43	99.40
3	99.56	99.87	99.08	99.80	99.44
4	99.53	99.81	99.08	99.71	99.40
5	99.54	99.89	99.01	99.84	99.42

5.2. Performance of Ensemble LSTM Model using Bagging Approach

Ensemble LSTM network using bagging approach is implemented as shown in Figure 2. Table 5 describes the results for seven individual LSTM models and the ensemble models using bagging approach. These single LSTM models are tuned with number of epochs as 10. The dataset is split into 90% training-10% testing ratio. Ensemble models are created by merging n (where n=3,5,7) single LSTM models. The top n single models in terms of accuracy are selected out of seven models to evaluate the results of ensemble method by merging n models.

The figures in Figure 4 present the comparison of accuracy obtained by the average of n LSTM models and ensemble LSTM models using the bagging approach by merging those n models using 5 different number of epochs. The accuracy and F-score obtained by the average of n models (where n=3, 5, 7) is less than the accuracy and F-score obtained by the ensemble LSTM model using bagging approach by merging those n models.

5.3. Performance of Ensemble LSTM Model using Stacking Approach

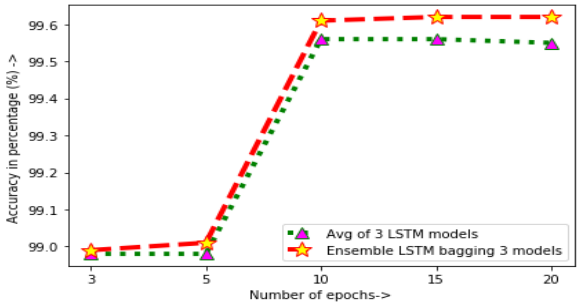
The stacking ensemble method combines the multiple LSTM models and builds a new meta-model based on the probabilities of detecting a URL as phishing or legitimate by those models as shown in Figure 3. The individual LSTM models fitted with particular subset of whole dataset used in ensemble learning using bagging approach are also used for stacking approach. The individual LSTM models trained with similar subsets of dataset are considered for both bagging and stacking approaches with an objective to perform a better comparison between these two approaches.

The figures in Figure 5 presents the TPR, TNR and accuracy values obtained by using stacking approach by merging n (where n=3,5,7) single LSTM models. Like in bagging here also, the accuracy increases from epoch 3 to 10. After that with increase in number of epochs, the performances do not increase much.

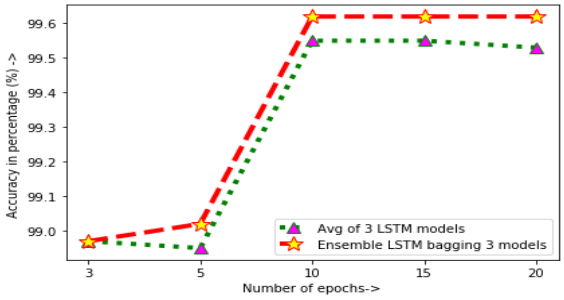
Using both the approaches, we found that the results obtained by merging five single LSTM models are comparatively better. Performance does not improve significantly with merging a greater number of individual models. So, for further comparison, the results obtained of our proposed approaches by merging 5 single LSTM models trained using only 10 epochs are considered. It is observed that the accuracy and F-score value obtained by ensemble learning using stacking approach is slightly higher than the ensemble learning using bagging approach.

Table 5. Performances in % for different LSTM models and ensemble LSTM bagging models

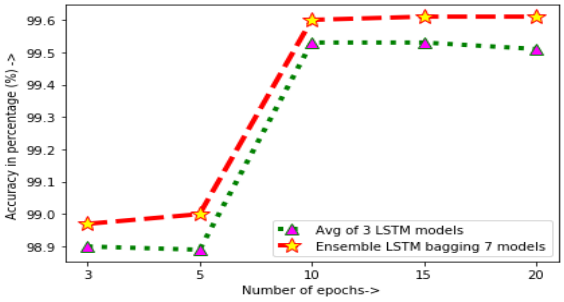
Models	ACC	REC	PRE	FSC
Model 1	99.53	99.18	99.62	99.40
Model 2	99.56	99.08	99.80	99.44
Model 3	99.55	99.00	99.87	99.43
Model 4	99.45	98.76	99.86	99.30
Model 5	99.58	99.06	99.87	99.46
Model 6	99.53	99.04	99.77	99.40
Model 7	99.49	99.02	99.70	99.36
Ensemble LSTM bagging 3 models	99.61	99.14	99.87	99.50
Ensemble LSTM bagging 5 models	99.62	99.18	99.86	99.55
Ensemble LSTM bagging 7 models	99.60	99.13	99.86	99.49



(a) Taking no. of models (n) =3

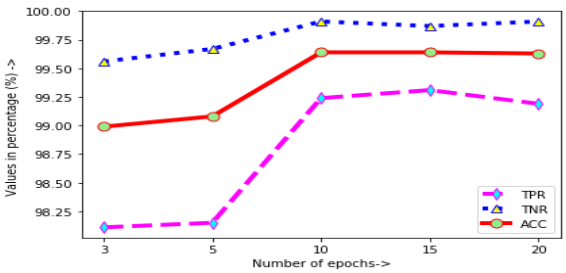


(b) Taking no. of models (n) =5

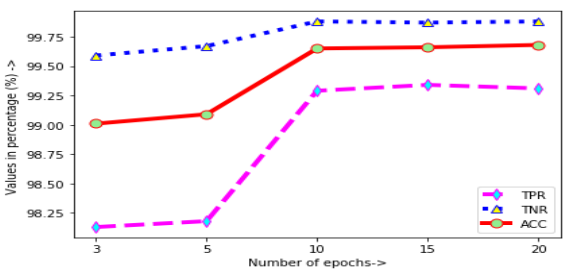


(c) Taking no. of models (n) =7

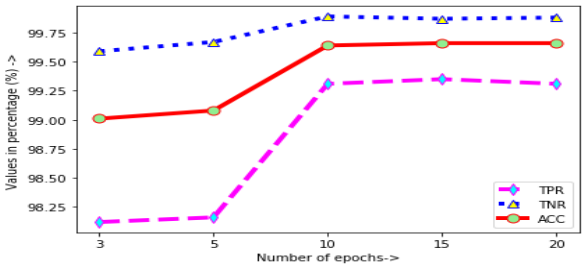
Figure 4. Epoch wise accuracy of average of n LSTM models and ensemble LSTM bagging approach of these n models



(a) Taking no. of models (n) =3



(b) Taking no. of models (n) =5



(c) Taking no. of models (n) =7

Figure 5. Epoch wise performances of ensemble LSTM stacking approach by merging n single LSTM models

5.4. Comparisons of LSTM Ensemble Approaches with other Methods

We compared the performance of LSTM ensemble methods with five different machine learning techniques. These algorithms based on seven lexical features. These features are selected out of 12 extracted features using mutual information feature selection method as depicted in **section 3.3**. The experiments are performed by splitting the dataset into three different ratios. Table 6 shows the detail comparison of performances of our proposed approaches and other different algorithms.

Table 6. Performances in % using different methods for various training-testing splitting ratios

Methods	Train %:Test %	ACC	REC	PRE	FSC	FPR
SVM	90:10	83.45	64.10	87.92	74.15	5.64
	80:20	83.38	63.79	91.37	75.13	3.91
	70:30	83.29	63.66	91.13	74.96	4.01
NB	90:10	79.39	54.63	88.87	67.66	4.46
	80:20	79.30	63.63	91.37	75.01	3.91
	70:30	79.24	54.08	88.67	71.70	4.47
DTREE	90:10	90.52	81.38	93.77	87.14	3.52
	80:20	90.10	79.24	94.74	86.30	2.85
	70:30	90.34	79.90	94.68	86.67	2.91
RF	90:10	98.56	98.73	97.66	98.73	1.54
	80:20	98.43	98.50	97.54	98.02	1.61
	70:30	98.29	98.46	97.22	97.84	1.82
MLP	90:10	82.15	62.32	89.21	73.38	4.91
	80:20	81.26	69.32	91.37	74.44	3.91
	70:30	81.03	63.07	85.69	71.99	6.71
Proposed LSTM Ensemble using Bagging approach	90:10	99.62	99.18	99.86	99.52	0.09
	80:20	99.59	99.11	99.86	99.48	0.09
	70:30	99.60	99.18	99.81	99.49	0.12
Proposed LSTM Ensemble using Stacking approach	90:10	99.65	99.29	99.81	99.55	0.12
	80:20	99.63	99.26	99.80	99.53	0.13
	70:30	99.60	99.28	99.71	99.49	0.19

The performance using the random forest method is comparatively much higher than other four machine learning algorithms. The decision tree algorithm also shows accuracy of more than 90%. But the results obtained using the LSTM ensemble method using both bagging and stacking

approach outperforms the random forest method. It is observed that the F-score value greater than 99.45% is obtained using all three training-testing splitting ratios. The performances using 90% - 10% splitting ratio is higher than the results obtained other splitting ratios. So, for final comparison, our results obtained using 90%-10% training-testing ratio using only 10 epoch is considered.

5.5. Comparison with other related Works

Results obtained using our proposed methods are compared with recent relevant works performed using various deep learning methods. Table 7 describes the methods used in those papers, the features used and the results obtained. We achieve highest accuracy of 99.62% and 99.64% using bagging and stacking approach respectively with false negative value of less than 1% in each case. This proves our proposed method as more effective and accurate than the other methods for detecting phishing sites.

Table 7. Comparison of our approaches (in %) with other relevant works

Paper	Methods	Features	ACC	REC	PRE	FSC	FNR
Bahnsen et al. [4]	LSTM	Automatic	98.76	98.93	98.60	98.76	1.07
Ubing et al. [14]	Ensemble Learning using various models	30 initial features, nine final selected features	95.40	95.90	93.50	94.70	4.10
Yang et al. [17]	Embedding CNN and Random Forest	Automatic based on URL features	99.35	99.21	99.52	99.34	0.79
Chen. W. et al. [18]	LSTM	10 features extract from URL	99.14	98.91	98.74	98.82	2.12
Wang et al. [19]	Bidirectional LSTM, Recurrent CNN	URL, Nine URL based features	95.60	93.78	97.33	95.52	6.22
Priya et al. [20]	Weight based Ensemble Learning approach	16 features	96.96	96.18	96.99	96.58	3.10
Zhu et al. [28]	OFS-ON (Neural network Model)	30 features, selection with FVV index	99.30	95.9	96.90	96.40	4.10
Proposed approaches	LSTM ensemble bagging approach	Automatic	99.62	99.18	99.86	99.52	0.82
	LSTM ensemble stacking approach		99.65	99.29	99.81	99.56	0.71

6. CONCLUSION

We performed phishing URL detection using LSTM networks. The bagging and the stacking approaches of the ensemble learning using LSTM models are used to detect phishing sites. Experiments are performed to tune the LSTM network using various epochs, different number of LSTM units, number of dense layers and number of dense neurons in each layer. No manual feature extraction is carried out to implement LSTM methods. Both bagging and stacking approach of ensemble learning using LSTM outperforms all other classification methods. Our performances are compared with some other related works on phishing sites detection using deep learning techniques. Experiments are performed on a relatively larger dataset. The results demonstrate that by using very few numbers of epochs and other parameters, we achieve the highest accuracy and f-score value. It concludes that our suggested method of using ensemble

learning based LSTM approach, can be effective and able to identify phishing sites more precisely than other existing methods. In the future, the LSTM method can be combined with other deep learning methods using ensemble learning techniques to enhance the outcomes.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672–694, 2021.

[2] Q. Ma, "The process and characteristics of phishing attacks-a small international trading company case study," *Journal of Technology Research*,vol. 4, p. 1, 2013.

[3] "Phishing activity trends reports," *APWG*. [2018-2022]. Available: https://apwg.org/trendsreports/ . [Accessed: 09-May-2022].

[4] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing urls using recurrent neural networks," *2017 APWG Symposium on Electronic Crime Research (eCrime)*, 2017.

[5] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of Recurrent Neural Networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," *Telecommunication Systems*, vol. 67, no. 2, pp. 247–267, 2017.

[8] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," *2010 Proceedings IEEE INFOCOM*, 2010.

[9] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on Automated Individual White-list," *Proceedings of the 4th ACM workshop on Digital identity management - DIM '08*, 2008.

[10] 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, 2016.

[11] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," *Proceedings of the 4th international conference on Security and privacy in communication netowrks - SecureComm '08*, 2008.

[12] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, pp. 1–20, 2017.

[13] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing URL detection using association rule mining," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, 2016.

[14] A. A. Ubing, S. Kamilia, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Phishing website detection: An improved accuracy through feature selection and Ensemble Learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, 2019.

[15] E. Gandotra and D. Gupta, "Improving spoofed website detection using Machine Learning," *Cybernetics and Systems*, vol. 52, no. 2, pp. 169–190, 2020.

[16] D. Sahoo, C. Liu and S. C. H. Hoi, "Malicious URL detection using Machine Learning: a survey", *arXiv:1701.07179v2*,2017.

[17] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing website detection based on deep convolutional neural network and Random Forest Ensemble Learning," *Sensors*, vol. 21, no. 24, p. 8281, 2021.

[18] W. Chen, W. Zhang, and Y. Su, "Phishing detection research based on LSTM recurrent neural network," *Communications in Computer and Information Science*, pp. 638–645, 2018.

[19] 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.

[20] S. Priya, S. Selvakumar, and R. L. Velusamy, "Evidential theoretic deep radial and probabilistic neural ensemble approach for detecting phishing attacks," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[21] M. Farsi, "Application of ensemble RNN deep neural network to the fall detection through iot environment," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 199–211, 2021.
[22] J. Xia, S. Pan, M. Zhu, G. Cai, M. Yan, Q. Su, J. Yan, and G. Ning, "A long short-term memory ensemble approach for improving the outcome prediction in Intensive Care Unit," *Computational and Mathematical Methods in Medicine*, vol. 2019, pp. 1–10, 2019.
[23] BJ. Brownlee, "Stacking Ensemble Machine Learning with python," *Machine Learning Mastery*, 26-Apr-2021. [Online]. Available: https:// machinelearningmastery.com/ stacking-ensemble-machine-learning-with-python/. [Accessed: 19-Nov-2021].
[24] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on Imbalanced Data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 80–89, 2004.
[25] A. Al-Ani and M. Deriche, "Feature selection using a mutual informationbased measure," *Proceedings of the 16th InternationalConferenceon Pattern Recognition*,vol.4, pp.82–85,IEEE,2002.
[26] DMOZ URL gr33ndata, "GR33NDATA/DMOZ-urlclassifier: Preparing DMOZ dataset for my n-gram LM-based URL Classification Research," *GitHub*. [Online]. Available: https://github.com/gr33ndata/dmoz-urlclassifier. [Accessed: 13-Aug-2020].
[27] "Join the fight against phishing," *PhishTank*. [Online]. Available: https://www.phishtank.com/. [Accessed: 25-July-2020].
[28] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "OFS-NN: An effective phishing websites detection model based on optimal feature selection and Neural Network," *IEEE Access*, vol. 7, pp. 73271–73284, 2019.

AUTHORS

Bireswar Banik is currently a Research Scholar in the Department of Computer Science, Gauhati University, India. He graduated under Gauhati University in 2013. He received his Post Graduation degree in Computer Science from Gauhati University, India in 2015. He has qualified in UGC-NET and SLET(NE) examinations. His area of interest includes Network Security, Computer Networks and Machine learning.

Abhijit Sarma had retired as an Associate Professor from the Department of Computer Science, Gauhati University, India. He acquired his PhD degree in the area of Wireless Networks from the Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati (IITG) in 2014. Ha had presented his research findings in various peer reviewed journals and international conferences. His research interest includes Network Security, Wireless LAN and Heterogeneous Networks.