International Journal of

INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep Learning Empowered Phishing URL Detection: An Exhaustive Approach

K Subashini [1], Dr. V Narmatha[2]

Submitted: 09/12/2023 **Revised:** 20/01/2024 **Accepted:** 30/01/2024

Abstract: Cybercriminals continually exploit users' vulnerabilities deceptive URLs through phishing attacks are a significant threat to both individuals and organizations. Cybercriminals regularly use phishing to trick users giving them permission to use corporate networks and digital files. Faster Recurrent Convolutional Neural Network (FRCNN) has been proposed to automatically identify phishing websites. However, there are certain drawbacks to its approach: (1) When the URL is converted into a characteristic matrix, there is a storage restriction, making it impossible to gather the embedding vector of new phrases to the actual data of sensitive characters; (2) it is also impossible to acquire the URL's long-distance dependent characteristic. Based on existing system, hybrid model Bidirectional Long Short Term Memory (Bi-LSTM) and FRCNN proposed to identify the phishing attack. The proposed system enables to obtain URL long-distance dependent characteristics by combining two current URL division approaches. Phishing websites can be quickly and accurately identified based on their URLs using the Naïve Bayes Method. According to experimental findings, this approach can produce high F1 values, recall rates and accuracy levels.

Keywords: *Machine Learning; Phishing Attack Detection; Deep Learning; Artificial Intelligence; Naïve Bayes;*

1. Introduction

Phishing Attack (PA) was a source of concern for online people. The privacy and financial security of internet users is gravely compromised. Phishers a type of scammer, build fake websites [1, 2] to appear and feel authentic and trick visitors. The fake emails are to steal the identities of legitimate people. Credit card numbers, passwords, account details, and the user's sensitive information are all collected during the transaction. Fishers frequently alter their methods of assault. Social engineering is one of the main strategies used by the fisherman [3]. It do this by getting private credentials from an impartial source.

Phishers develop fake emails and websites that, occasionally, closely approximate the websites of real companies that come from a source. Hackers are known to impersonate trustworthy sources to coerce users into updating their systems. It also request a ransom and attempt to suspend the client's account. Email spoofing was another type of phishing scam [4]. Researchers are frequently tricked into divulging private information including passwords and credit card details. Therefore, the primary goal of fishing was to acquire helpful information including credit card information password, and bank account [5]. People and businesspeople are lost faith in Internet commerce as this kind of scam was grows dramatically. Due to consumers' lack of faith in Internet transactions, Internet companies started to receive negative customer feedback [6]. Attacks are still possible [9] even while computers use encryption software to safeguard the data it save. In this study, machine learning (ML) was used to identify fishing.

The blacklist identification merely executes straightforward dataset query activities, and while the detection rate is quick, and easy, it is constrained by the need to regularly gather samples from phishing websites& maintain the blacklist dataset [7]. The URL must first be obtained from content-based detection techniques to assess the legitimacy of the website under test depending on its resemblance to an existing website or ML method. The client's risk is increased by the requirement for obtaining online content in this approach. It also necessitates a significant amount of manual characteristic engineering. Several of the characteristics require a confirmation from pertinent professionals [8]. The caliber of the manually derived features has a significant impact on its performance. The relatively set qualities of the detection mechanism make it simple for phishing attackers to go around it.A majority of currently used techniques dependent on URL characteristics identification use NNs to mechanically gather URL characteristics to assess the reliability of online pages. The steps of URL segmentation is shown in Figure 1.

In essence, the URL was a string of letters, special characters and numbers. An FV can be the NN to identify what was converted to u->X ϵ $R^{L,K}$its matrix representation $x_i(i=1,2,3,\ldots L)$, $x_i$which was a k-dimensional matrix and X

[1] *Research Scholar, Department of Computer and Information Science, Annamalai University, Tamil Nadu, India*
Author Email: subaphdscholar@gmail.com
[2] *Assistant Professor, Department of Computer and Information Science, Annamalai University, Tamil Nadu, India*
Author Email: balaji.narmatha8@gmail.com

ϵ R^Ka set of adjacent elements that that indicate the words and characters of the URL.

In conclusion, a single neural network and two-word segmentation techniques are primarily used in the method to identify phishing websites based on URL characteristics. The following are some of this method's drawbacks: When segmenting URLs depending on special characters, and words, can cause the number of words to expand significantly, changing the features of the information collection proportionally, and making unable to obtain the insertion vectors of fresh introduced words during testing; A sensitive words "login," "password," "registered," etc. will lose some of their validity if the URL is divided based on characters; additionally, using a single NN like CNN extracts the local aspects to the URL and is unable to extract its sequence features. It suggest a strategy based on delicate word grouping to address the issues with the aforementioned methods. The sensitive words are then distinguished from the other symbols by being taken as a whole after the non-sensitive words have been divided into character groups. This makes it simple to recognize the key elements in the URL, assisting the NN classifier in gathering additional relevant data. To extract more comprehensive information from URLs, neural network classifiers use bidirectional long-short memory networks and CNN.

Internet usage has increased significantly during the last ten years. Understanding that Internet users' security and privacy are not always guaranteed is essential given that the Internet connects billions of people globally. Businesses lose a lot of money yearly due to the growing prevalence of cybercrime. More than 80% of security incidents that have been reported involve phishing attempts. Data leakage was validated in at least 50% of the 3,841 phishing events that were reported up until May 2021, Verizon's 2021 (DBIR) [9]. In 2021, there were 11% more phishing-related data breaches than there were the previous year.

95 percent of these attacks had a financial motivation, costing thousands of dollars every minute in losses. Around the world, several programs and seminars are held to inform users and raise awareness about phishing fraud. An internet site can be found using URL which is a special identification. It has several components, including the protocol, domain name, port, path, query, etc. A phishing website can be distinguished from a legitimate one using specific elements of its URL. Nevertheless; it is not usually feasible to determine the reliability of a website simply by looking at the URL. ML algorithms seem to be a reliable and effective approach for identifying these characteristics and determining if a specific URL was safe or PW.

In this analysis, investigate learning strategies to deal with these issue. Our approach is built with the technique of aggregation evaluation, which generates rules manually to find layout similarities between websites and subsequently identify phishing pages. It employs aspects of the website layout to develop a similarity predictor before employing it to find phishing pages.

- To assess the comparability of website layouts and recognize phishing pages, it provide a learning-based mechanism.

- To create and test our technique, actual-world website examples from phishtank.com and alexa.com were used.

The rest of the paper is organized in the following manner. Section 2 provides a comprehensive review related to our research and presents limitations of an existing systems. Section 3 presented a unifying methodology. The results of the performance evaluation were provided in Section 4. The study is concluded in Section 5.

2. Related Works

PA is specifically designed to confuse clients and manifests in many forms. In order to mitigate the risks associated with phishing attacks, there exists a range of approaches and technologies that may be used for the identification and detection of such fraudulent activities. Categorization was used as a method for identifying instances of the website phishing [10, 11].

Machine learning and other methods were used to develop detection mechanisms for these assaults. However, the accuracy of detection could still be improved. An ensemble classification algorithm is proposed in this paper [12] for detecting phishing websites (PWs).

By establishing a friendly relationship via microblog messages on various social networking sites, social phishers continue to adapt their novel trapping techniques of taking usernames and passwords. Getting rid of phishing attacks in social networking sites is an important measure for mitigating latent fraudulent phishing mechanisms. Various anti-phishing mechanisms developed for social networking sites are reviewed and analyzed in this paper [13].

In the context of cyber-attacks, phishing refers to the act of obtaining personal information or credentials from a human victim by using fake communications to entice them to provide it. In order to avoid becoming a victim of these attacks, visitors to websites can identify the phishing websites and avoid becoming a victim themselves. There are no one-size-fits-all solutions to mitigate all vulnerabilities, so many different approaches are being employed to mitigate all vulnerabilities. A voting technique based on weights was proposed in this paper [14] to combine multiple base models into an ensemble model. The results were compared before and after the application of feature selection methods and standardization on the dataset.

Wrapper and embedded methods apply set intersection operations to the final feature subsets to find the best features. According to this case study, using a hybrid method to extract features from the dataset for experimentation, only 37.84% of the features were retrieved [15].

An automatic feature extraction model that adjusts the complexity of the model as a function of its complexity is presented in this article for spam detection. Feature separation and pooling is handled by convolutional and pooling layers in the proposed model, as well as base classifiers for differentiating legitimate texts from invalid ones [16].

Several advanced techniques, strategies, and tools are used in phishing attacks, like mobile applications. The use of deep learning algorithms has provided promising results in phishing detection approaches to mitigate and avoid these attacks [17].

Many anti-phishing software packages provide interfaces to make it easy to report emails. Cyber security training programs encourage users to report phishing emails that they believe are suspicious. In this study, it investigates how perceived self-reported pressure are associated with anti-phishing behaviour [18].

2.1 Methods for Categorizing Website Phishing

Several strategies are provided [19] to shield online consumers from phishing attacks. It are among the most challenging to identify and stop because the perpetrators of email spoofing and URL phishing attempt frequently change their methods. It was advised to block fraudulent emails and fake URLs to stop these kinds of phishing. Lexical analysis was employed to foreseeably locate dangerous URLs [20]. ML methods were applied to characteristics to categorize the false URLs. These attributes that are meant to cut down on these URLs were studied. Principal Component Analysis (PCA) and the Random Forest (RF) methods are also utilized to identify PAs. The primary components of the parameters were extracted using PCA. After that, data were analyzed to determine the kind of phishing.

To the databases, the Random Forest method was deployed. Phishing was discovered after categorizing the URL [21]. This method is quite precise. Approximately 96% of phishing emails can be classified using this simple programme. It can manage large dataset quantities as well. Using categorised datasets with labels, phishing attempts were found. The classification approach made use of various characteristic categories, including text-based, email-based, and URL-based characteristics. All URL-based characteristics for classification, including IP addresses, were used by BNN, KNN, SVM, and ML classifiers. The Classification methods were used to find emails. The hacker forges the email to get data. With a supervised learning technique like Naive Bayes, these fake emails were found Emails were first categorized using techniques, after which spam and legitimate emails were divided. In ML techniques, hyperlink properties were also exploited for research [22].

This study found that the application of Convolutional Neural Networks for classification also produced favorable results. Traditional machine learning techniques including Nave Bayes, Support Vector Machines, and Decision Trees were used in the majority of the related research. By removing unimportant features, the Random Forest seeks to decrease the number of characteristics. It chooses a characteristic at random to assess its impact and, if it has a negative impact, replaces it with another [23]. This study used the BiLSTM-FRCNN_ Naive Bayes Model, which outperformed other well-liked ML techniques, to detect and classify phishing.

2.2 Limitations of Existing System

A search space dimension has typically increased as a result of using a high number of characteristics to train the classification method [24]. A classifier's effectiveness demonstrates a consistent rise to a particular criterion dimension, at that point a fall was seen, Hughe's effect—often referred to as the Curse of Dimensionality. A feature selection method should be applied to solve this issue [25].

Traditional ML algorithms are capable of producing annual results, but it are prone to overfitting and underfitting and may not necessarily produce optimal results. Techniques for ensemble ML could be utilized to solve this issue [26].

The most common method of getting in touch with potential phishing victims is via email. In addition to social media, adverts, text messages, telephone conversations, and other channels of communication, phishing websites can also be reached by unwary internet users. Therefore, the range of detection cannot be restricted to phishing emails [27-30].

Phishing may also be motivated by identity theft or human cloning in addition to financial gain. Therefore, to maintain an internet user's total safety and security, it is equally vital to concentrate on general websitesin addition to e-commerce and banking websites

3. Proposed System

3.1 Preprocessing

To turn the unprocessed URL input text into characteristics that could be predicted and analyzed, various tools and pre-processing techniques are used:

A string of characters known as a regular expression (Regex) defines a search pattern. It was employed to identify the'@' sign, the '//' redirect symbol, suffix, protocol, the prefix, subdomain, etc. of the input URL.

• Whois: The whois method can be used to retrieve the creation date and determine the domain age.

• PageRank: The PageRank algorithm uses a website's popularity—as measured by the number of visitors—to assess how important it is to the internet.

• Prefix or Suffix: To give the appearance of legitimacy, Domain names typically contain suffixes and prefixes. Usually, there is a hyphen between them.

• Long URL: Phishers may utilize long URLs to conceal the whereabouts of harmful files under numerous subdirectories.

• Domain length: A domain may have a maximum of 63 characters. Long domain names may be used by scammers to deceive users.

Our proposed work addresses can be stated as follows: We seek to gather characteristics and use machine learning techniques to comprehensively identify phishing websites thoroughly based on their layout similarity characteristics using a benign page and a group of suspicious pages as inputs shown in Figure 1.

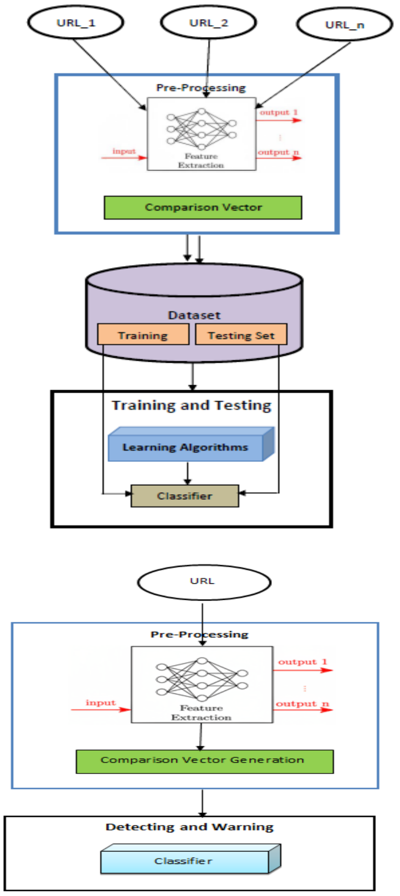

Fig 1: Overview of our strategy

3.2 System Architecture

A Uniform Resource Locator string is regarded as raw data used to collect the user's input. The lexical features and other properties are then extracted after processing. The trained model receives the analyzed input data and predicts whether the entered URL is trustworthy or not. The training data of the algorithm was obtained to the over 2000 records with 30 variables that make up the standard ML database accessible to usage by the University of California in the United States. A dataset was gathered from a global community known as "Phishtank," which works to report and confirm the existence of suspicious activity on probable phishing websites. After choosing features and data pre-processing, the dataset is utilized to develop classification techniques depending on Naïve bayes methods.

Spam sent by hackers often contains URLs to fake websites that look authentic in terms of content, page links, and graphics to steal personal information. Instead of words, itutilize graphics to make anti-phishing detection of phishing pages more difficult. The Bayesian approach of content-based phishing of visual and textual content is used to determine how similar protected and suspect URLs are to one another. The real website must be pre-processed when a prominent URL was faked and phished. Database features for the site's content, as well as an earlier method for removing phishing activities and those details, are being taken. Each word has a text classifier attached to it that separates the main text from HTML elements and performs stemming. Stems employ fundamental components as opposed to original terms. For instance, the words "algorithm", "algorithm", and "algorithm" are all pronounced as the same word. Stemming words are saved and used to build vocabulary. The histogram vector (j1, j2,...,jn) will be visualized for forums, where every component represents a single occurrence &n signifies the number of entries in the matrix.

3.3 Feature Selection

The pre-processing step of feature selection involves removing any inconsequential features from the data shown in Figure 3. The association between a feature and the target variable is assessed using the filter method known as the Chi-Square test. Equation 1 provides the Chi-Square value.

$$I_c^2 = \sum_{x=1}^{n} \frac{(P_x - E_x)^2}{E_x} \qquad (1)$$

Where E was the expected value, degree of freedom, and P was the observed. If the actual value and anticipated value are nearly identical, the Chi-square value was low. If the Chi-Square value is low, the characteristics are consequently more independent.

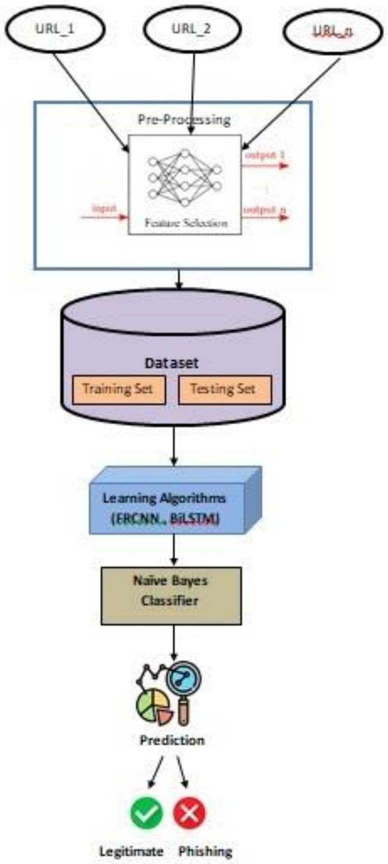

Fig 2: System Architecture

Chi-Square value versus the characteristic set is displayed in Figure 4.

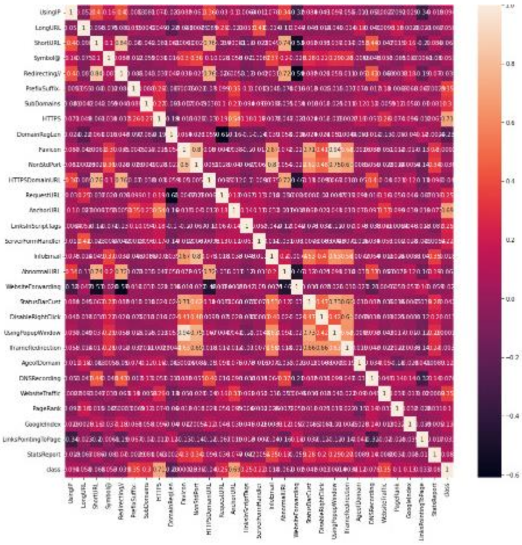

Fig 3: Feature Selection

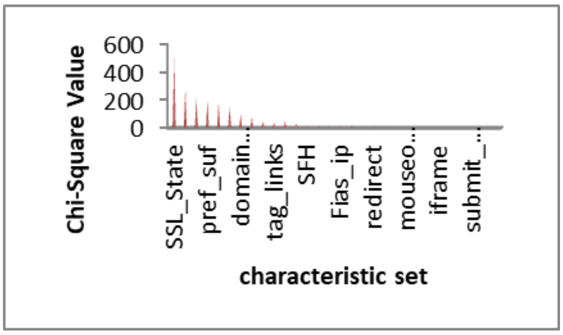

Fig 4: Chi-Square value vs characteristic

Table 1: The Optimal values of parameters for BiLSTM-FRCNN

Parameter	Optional value	Default value
Learning rate	[0.01, 0.005, 0.0001, 0.02]	0.002
Batch size	[32, 64, 128]	32
Epochs	[6,10,20]	10
Optimizer	['adam', 'RMSprop', 'Adagrad']	adam'
Dropout rate	0.5	0.5
Loss function	['Binary_crossentropy']	Binary_crossentropy'

The categorization model's training set consisted of the top 15 characteristics with the highest Chi-square values.Figure6 and Table 1 the approach framework for categorizing URLs using the hybrid FRCNN-BiLSTM algorithm proposed in this article. The configuration of the particular method and detector model variables is as the following:

Algorithm 1: FRCNN - BiLSTM

Step 1.Segment the URL using the sensitive word segmentation technique.

Step 2. It is determined that each URL has 300 total letters and keywords (table 1)based on the URL database and sensitive vocabulary. Excess characters will be terminated after the URL if it is longer than 300 characters. A pad >tag was utilized as the new word at the end of the URL the length is less than 300. The unknown character mark (unk) is used to identify an unknown character in a URL.

Step 3. The total number of sensitive phrase and special characters are 121 when the URL database and sensitive vocabulary are analyzed. As illustrated in Table 2, an MT should be created to assign special codes to delicate words and symbols.

Step 4: Word embedding matrix is used to convert the digitally encoded URL into a two-dimensional dense characteristic vector matrix. As indicated in equation 1, where i x was a one-dimensional 1, 2, 3,...300 matrices, the URL is first converted into a 300 * 1 matrix X according to the MT of letters and delicate words. Eventually as indicated to equation 2, the matrix X is converted into a 300 * 32 two-dimensional column vector, xi dense matrix that contains semantic data. In this case, xi was a 32-dimensional column vector.

$$I = (i_1, i_2, \dots, i_{300}) \qquad (2)$$

$$\acute{I} = (i_1, \acute{i}_2, \dots, i_{300}) \qquad (3)$$

Step 5. The convolution neural network receives the characteristic vector matrix as an input, and the convolution kernel automatically extracts the local characteristics from the characteristic matrix. Character vector's dimension of 32, a total of 200 convolution kernels, and a sliding step size of 1. Equation 3 displays the formula for your convolution kernel's URL embedding matrix, where "xi" denotes a vector representation of letters or sensitive words. In Equation 4, where W_f and bf are the activation variable Relu, the new characteristic $A = \pi r^2$ produced by the convolution process is set. To create the feature map cf, the According to formula 5, CK iterates through the entire embedding matrix, bias term, and weight matrix.

$$I'_x = (i_1, i_{x+1}, \dots, i_{x+h-1}) \quad (4)$$

$$c_x^f = \sigma(W_f * I_x + b_f) \quad (5)$$

$$c^f = (c_1^f, c_2^f, \dots, c_{300-h+1}^f) \quad (6)$$

Step 6. Increase the PW of f c (the pooling step size is, pooling window size is 2,) to get more representative characteristics. The new charateristic map m_if after pooling was displayed in for equation 6 when it was set to ith pW and Equation 7 shows how to get the new feature map when the pooling window has traversed the entire cf.

$$m_x^f = max(c_1^f, c_2^f, \dots, c_{x+pl-1}^f) \quad (7)$$

$$m^f = max\left(c_1^f, c_2^f, \dots, c_{300-h+1/pl}^f\right) \quad (8)$$

Step 7. To create the sequence vector represented in formula 8 (where s=[(L-h+1)/pl)], stack the fresh feature maps that are produced after I is pooled by all convolution kernels.n is the quantity of convolution kernels, mp is the feature vector made up of all convolution check URL words inserted into the same region of the matrix after convolution and pooling function mp_x R(n*1), and n was the number of characters in the convolution..

$$MP = (mp_1, mp_2, \dots, mp_x) \qquad (9)$$

Step 8. Consider MP as the sequence data on the time axis that the BiLSTM receives as input, and mpx as the input of the BiLSTM at the x-th instant. The forward LSTM uses the forget gate, output gate and input gate to memorize data. The value h_F is kept as a record of the output of information before the time x=s through this instant. The forget gate output and input gate are the final three gate that the reverse LSTM uses to recall the information. To extract long-distance variable characteristics in two distinct directions of the URL $h=h_F \oplus h_R$, the output at this point is saved as h_R.The LSTM's final moment output is then combined as F in two distinct directions.

$$p_x = e^z / \sum_x^k e^z \ (10)$$

3.4 Classification model

To achieve better outcomes, it makes use of weighted classifiers, parallelization, and tree pruning. Finally, Naïve Bayes calculates the probability of each class given the observed features. The class with the highest probability is then assigned as the predicted class.

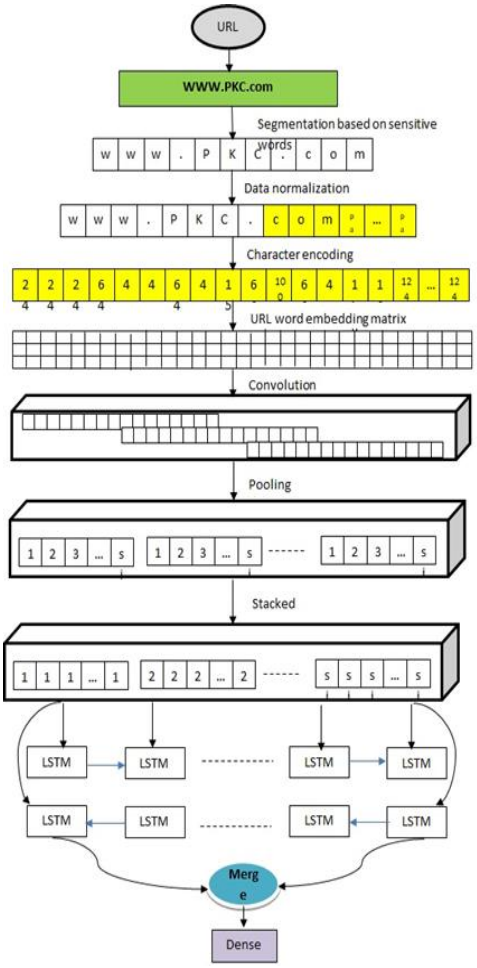

Fig 5: Method architecture for identifying URL groups dependent on the FRCNN-BiLSTM

An overfitting problem is known to be solved by reducing variation using bagging techniques like Random Forest, however, it may also provide significant bias or underfit results. Boosting algorithms fix each learner's mistakes, eliminating the issue of underfitting, but it can also result in overfitting. Combining boosting and bagging will produce an acceptable bias-variance trade-off. When utilized for PD-like cases, to enhance the accuracy of the Naïve Bayes method because forecasting precision is far more crucial than method interpretability.

Algorithm 3: Naïve Bayes

```
1. Load the testing dataset (features).
2. For each instance in the testing dataset:
   a. Initialize variables for storing the probability of the instance belonging to each class.
   b. For each class C in classes:
      i. Initialize the probability P(instance|C) with the prior probability P(C).
      ii. For each feature F in the instance:
         1. Multiply P(instance|C) by the corresponding feature probability P(F|C).
      iii. Store P(instance|C) in the probability variables for each class.
   c. Choose the class with the highest probability as the predicted class for the instance.
3. Return the predicted classes for all instances.
```

The algorithm leads to fast training times, especially when compared to more complex models like deep neural networks.

Table 2: Confusion Matrix

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

The CM that was depicted in Table 2 shows the method's performance on test information that the true values were available, is created using these parameters.

4 Results and Discussion

4.1. Experimental Data

To expand the source of URL data, the information collection utilized in this study contains open-source examples from different platforms, phishing URLs obtained from Malware Patrol, PhishTank and and legal URLs obtained from Alexa& DMOZ. An anti-phishing website called PhishThank allows users to report, confirm, and exchange phishing information. MalwarePatrol gives users the option to download phishing URLs. The biggest international directory community is DMOZ, which is run and developed volunteers to all across the world. It seeks to top-notch URLs where users can receive legitimate URL data sets. The world's top website rankings are published on the website Alexa. It currently has a big number of URLs and comprehensive website ranking data. A legal URL data set is compiled from the top-ranked websites. An information collection to 206,200 labelled URL examples in total, of that105,100 were phishing samples and 101,100 were genuine samples after deduplicating the data. Tables 3 and 4 comparison of proposed and existing systems

Table 3: Performance Evaluation metrics for BiLSTM-FRCNN_Naive Bayes

Parameters for measuring performance	BiLSTM-FRCNN	BiLSTM-FRCNN_ Naive Bayes (Proposed Approach)
Accuracy	**97.81**	**98.2**
Precision	97.3	97.5
Recall	98.3	98.4
F-Score	98.04	98.4

Table 4: Comparison of performance metrics of Proposed Model with Existing Model

Approach	**Accuracy**	**TPR**	**F-Score**
Nepal et al [31]	94.3	93.3	94.5
Devalla et al [32]	95.2	96.2	95.6
Benavides et al [33]	97.3	98	98
Ren et al [34]	98	98.3	98.1
Proposed Approach	**98.2**	**98.6**	**98.4**

Each sample database is iterated through the process ten times to confirm that it may be model's ability to identify using the mean of the 10 test outcomes using Naïve Bayes, items are rated that were predicted as an evaluation set. The precision rate of the algorithm developed in this study on both the testing set and the training dataset is shown in Figure 8 and 9 as a mean-variance curve and loss graph respectively. The graphical representation shows that throughout the training phase, the model's parameters converge properly. The testing and training performance of the model tends to be steady during 30 learning rounds.

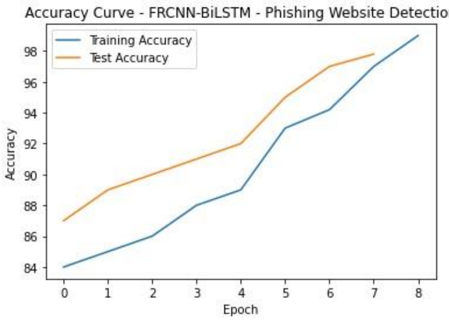

Fig 6: FRCNN-BiLSTM performance curve on test and training data.

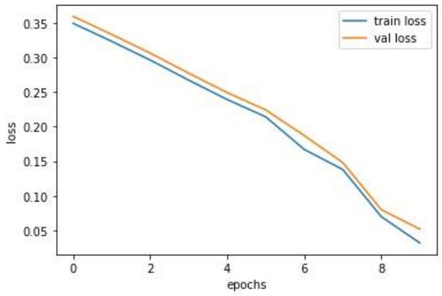

Fig 7: Loss graph of FRCNN-BiLSTM

Figures 6 – 11 shows the performance measures of proposed system based on training dataset and verification set. While running our proposed model, it gives 98.2% accuracy rate which is higher than other existing models. At the same time, TPR at 98.6% and F-Score as 98.4%. A loss graph during testing and training is a graphical representation that illustrates the change in a model's loss function over the course of both the training and testing phases in a machine learning or deep learning task. The loss function quantifies the discrepancy between the model's predictions and the actual target values.

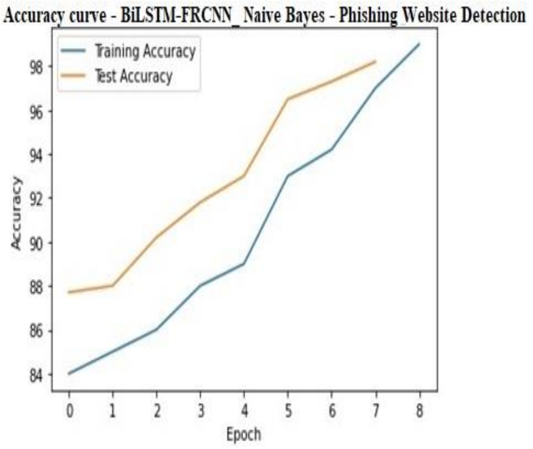

Fig 8 Accuracy of proposed system

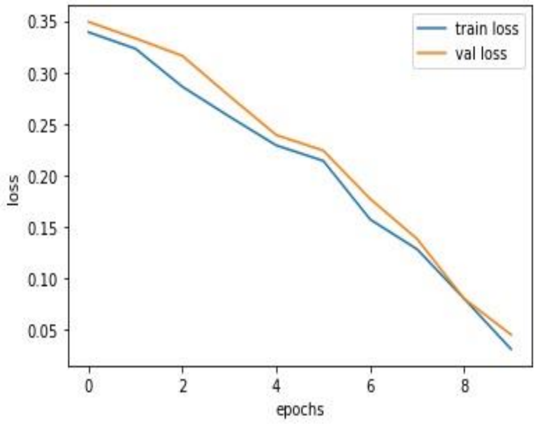

Fig 9: Loss graph during testing and training

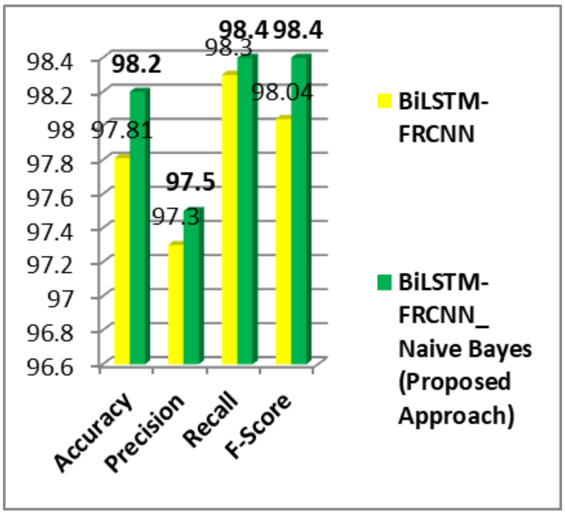

Fig 10: Comparison of existing and proposed system

Table 5: Overall Performance of Proposed Approach

FACTORS	PROPOSED APPROACH VALUES
Accuracy	98.2
False Negative Rate (FNR)	0.015
False Alarm Rate (FAR)	0.019

FNR and FAR are important metrics as given in table 5 which are used to evaluate the performance of binary classification models, particularly in scenarios where class imbalances exist. The authenticity of each website is estimated separately by the two algorithms, and then an average is obtained to get the final result. A technique generates three possible outcomes: Half (0.5), Zero (0) for a secure website for a website that was a 50% chance of malicious, and One (1) for a website that was phishing and should be avoided. These metrics provide insights into the model's ability to correctly classify positive and negative instances and help assess its effectiveness in real-world applications.

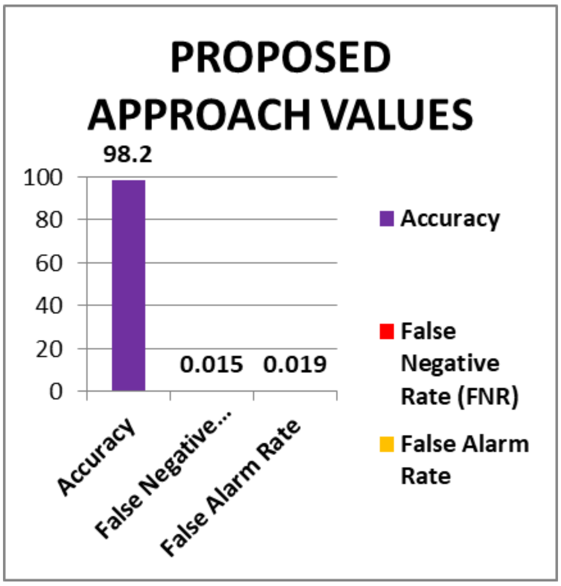

Fig 11: Overall approach values

5. Conclusion

An application of deep learning techniques to identify and detect phishing attacks within URLs presents a promising and effective approach in the realm of cyber security. The algorithm's training set included more than 2000 records from the University of California in the United States with 30 variables. The results obtained from this research demonstrate that FRCNN-BiLSTM-Naïve Bayes methods have the potential to significantly enhance the accuracy and efficiency of phishing attack detection when compared to other detection methods. XGBoost is used to optimizing the distributed Gradient library that is designed for efficient and training of Machine learning model. Stacking is used to predict multiple nodes of new model which improve performance of the proposed system. Finally, receiving emails is classified using the Naïve Bayes classifier, which alerts the user to potential dangers based on a collection of common traits. Naïve Bayes calculates the probability of each class given the observed features. The class with the highest probability is then assigned as the predicted class. In future, phishing detection using optimization techniques holds the potential to enhance the accuracy, efficiency, and adaptability of anti-phishing systems.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Kalabarige, L. R., Rao, R. S., Abraham, A., &Gabralla, L. A. (2022). Multilayer stacked ensemble learning model to detect phishing websites. *IEEE Access*, *10*, 79543-79552.

[2] Alsariera, Y. A., Balogun, A. O., Adeyemo, V. E., Tarawneh, O. H., &Mojeed, H. A. (2022). Intelligent tree-based ensemble approaches for phishing website detection. *J. Eng. Sci. Technol*, *17*, 563-582.

[3] Atre, M., Jha, B., and Rao, A. (2022). Detecting Cloud-Based Phishing Attacks by Combining Deep Learning Models. *arXiv preprint arXiv:2204.02446*.

[4] Puri, N., Saggar, P., Kaur, A., and Garg, P. (2022, July). Application of ensemble Machine Learning models for phishing detection on web networks. In *2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT)* (pp. 296-303). IEEE.

[5] Hota, H. S., Shrivas, A. K., &Hota, R. (2018). An ensemble model for detecting phishing attacks with proposed remove-replace feature selection technique. *Procedia computer science*, *132*, 900-907.

[6] Bidabadi, F. S., and Wang, S. (2022). A new weighted ensemble model for phishing detection based on feature selection. *arXiv preprint arXiv:2212.11125*.

[7] Ramanathan, S., Mirkovic, J., and Yu, M. (2020, January). Blag: Improving the accuracy of blacklists. In *NDSS*.

[8] Hossain, F., Islam, L., and Uddin, M. N. (2022, September). PhishRescue: A Stacked Ensemble Model to Identify Phishing Website Using Lexical Features. In *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)* (pp. 342-347). IEEE.

[9] Garikapati, P., Balamurugan, K., and Latchoumi, T. P. (2022). K-means partitioning approach to predict the error observations in small datasets. *International Journal of Computer Aided Engineering and Technology*, *17*(4), 412-430.

[10] Livara, A., and Hernandez, R. (2022, January). An Empirical Analysis of Machine Learning Techniques in Phishing E-mail detection. In *2022 International Conference for Advancement in Technology (ICONAT)* (pp. 1-6). IEEE.

[11] Ansari, M. F., Panigrahi, A., Jakka, G., Pati, A., and Bhattacharya, K. (2022, November). Prevention of Phishing attacks using AI Algorithm. In *2022 2nd Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)* (pp. 1-5). IEEE.

[12] Pandey, M. K., Singh, M. K., Pal, S., and Tiwari, B. B. (2022). Prediction of Phishing Websites Using Stacked Ensemble Method and Hybrid Features Selection Method. *SN Computer Science*, *3*(6), 488.

[13] Ali, M. M., Qaseem, M. S., and Rahman, M. A. U. (2020). A survey on deceptive phishing attacks in social networking environments. In *Proceedings of the Third International Conference on Computational Intelligence and Informatics: ICCII 2018* (pp. 443-452). Springer Singapore.

[14] ShiraniBidabadi, F., and Wang, S. (2022). A new weighted ensemble model for phishing detection based on feature selection. *arXiv e-prints*, arXiv-2212.

[15] Bhowmik, P., and Bhowmik, P. C. (2022, October). A Machine Learning Approach for Phishing Websites Prediction with Novel Feature Selection Framework. In *Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021* (pp. 357-370). Singapore: Springer Nature Singapore.

[16] Shaaban, M. A., Hassan, Y. F., &Guirguis, S. K. (2022). Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. *Complex and Intelligent Systems*, 1-13.

[17] Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., and Shukla, S. (2022). Applications of deep learning for phishing detection: a systematic literature review. *Knowledge and Information Systems*, 1-44.

[18] Kwak, Y., Lee, S., Damiano, A., &Vishwanath, A. (2020). Why do users not report spear phishing emails?. *Telematics and Informatics*, *48*, 101343.

[19] Mithra Raj, M., and Arul Jothi, J. A. (2022, October). Website Phishing Detection Using Machine Learning Classification Algorithms. In *Applied Informatics: 5th International Conference, ICAI 2022, Arequipa, Peru, October 27–29, 2022, Proceedings* (pp. 219-233). Cham: Springer International Publishing.

[20] Shmalko, M., Abuadbba, A., Gaire, R., Wu, T., Paik, H. Y., and Nepal, S. (2022). Profiler: Profile-Based Model to Detect Phishing Emails. *arXiv preprint arXiv:2208.08745*.

[21] Rao, R. S., Umarekar, A., &Pais, A. R. (2022). Application of word embedding and machine learning in detecting phishing websites. *Telecommunication Systems*, *79*(1), 33-45.

[22] M. Z. Gashti, "Detection of Spam Email by Combining Harmony Search Algorithm and Decision Tree", *Engineering, Technology & Applied Science Research*, vol. 7, no. 3, pp. 1713–1718, Jun. 2017.

[23] A. Darem, "Anti-Phishing Awareness Delivery Methods", Engineering, Technology & Applied Science Research, vol. 11, no. 6, pp. 7944–7949, Dec. 2021.

[24] Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyedeji, S., and Porras, J. (2023). Mitigation Strategies against the Phishing Attacks: A Systematic Literature Review. Computers and Security, 103387.

[25] Greco, F., Desolda, G., and Esposito, A. (2023). Explaining phishing attacks: An XAI approach to enhance user awareness and trust. In Proc. of the Italian Conference on CyberSecurity (ITASEC '23).

[26] [26] Roy, S. S., Naragam, K. V., and Nilizadeh, S. (2023). Generating Phishing Attacks using ChatGPT. arXiv preprint arXiv:2305.05133.

[27] Asiri, S., Xiao, Y., Alzahrani, S., Li, S., and Li, T. (2023). A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks. IEEE Access.

[28] Butt, U. A., Amin, R., Aldabbas, H., Mohan, S., Alouffi, B., and Ahmadian, A. (2023). Cloud-based email phishing attack using machine and deep learning algorithm. Complex and Intelligent Systems, 9(3), 3043-3070.

[29] Al Ogaili, R. R. N., and Manickam, S. (2023). A Critical Review: A New Taxonomy for Phishing Attacks Based on Phishing Techniques Used. Wasit Journal for Pure sciences, 2(2), 251-269.

[30] Alabdan, R. (2020). Phishing attacks survey: Types, vectors, and technical approaches. Future internet, 12(10), 168.

[31] Nepal, S., Gurung, H., and Nepal, R. (2022). Phishing URL Detection Using CNN-LSTM and Random Forest Classifier.

[32] Devalla, V., Raghavan, S. S., Maste, S., Kotian, J. D., and Annapurna, D. (2022). murli: A tool for detection of malicious urls and injection attacks. *Procedia Computer Science*, *215*, 662-676.

[33] A. O. Aljahdali, F. Thabit, H. Aldissi, and W. Nagro, "Dynamic Keystroke Technique for a Secure Authentication System based on Deep Belief Nets*", Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10906–10915, Jun. 2023.

[34] Ren, F., Jiang, Z., and Liu, J. (2019, December). A bi-directional lstm model with attention for malicious url detection. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 300-305). IEEE.