Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Ensemble Deep Learning Based Detection System to Identify Phishing Attacks

Patcha Niharika [1*], Vithya Ganesan [2], V. Anjana Devi [3]

[1] *Student of M. Tech, Department of CSE, Koneru Lakshmaiah Education Foundation, AP, India. Email: patchaniharika@gmail.com*
[2] *Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India. Email: vithyamtech@gmail.com*
[3] *Associate Professor, Department of CSE, Rajalakshmi Institute of Technology, Chennai, Tamil Nadu. Email: anjanadevi.aby06@gmail.com*

ARTICLE INFO

Received: 22 Dec 2024

Revised: 17 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Phishing websites pose a significant threat in the cyber security realm, resulting in substantial financial losses. Despite ongoing updates to confrontation methods, the effectiveness of these approaches remains unsatisfactory. The proliferation of phishing websites in recent years has raised concerns, highlighting the urgent need for more advanced phishing detection technology. Machine learning and deep learning support defensive mechanism to protect from phishing attacks. A recent study introduced DeepPhish, a deep neural network to generate phishing URLs, demonstrating the evolving sophistication of phishing tactics. The expansion of phishing beyond traditional channels like email, SMS, and pop-ups to include mobile platforms and social networks has made detection more challenging. Phishing attacks encompass QR code phishing, spear phishing, and spoofing on mobile applications. Furthermore, malicious actors often host phishing sites on HTTPS and SSL-certified domains to predict them as legitimate. This diversification of phishing tactics presents new obstacles detection. Despite the elusive nature of phishes, security experts and researchers made significant efforts to combat phishing website threats. To address this issue, we propose an ensemble deep learning-based detection system to identifying both AI-generated and human-crafted phishing URLs. By incorporating URL HTML Encoding for enhanced lexical analysis, our system can classify URLs in real-time and compare them with existing detection methods. In this proposed model enhanced phishing web page detection model by CNN and Bi-Directional LSTM to identify phishing attacks.

Keywords: Cyber security, DeepPhish, Deep Learning, Convolution Neural Network (CNN), Secure Socket Layer.

INTRODUCTION

Internet is an indispensable infrastructure and offers convenience to human and society. However, it comes with certain security challenges as phishing, malicious attacks, and intruder's privacy breaches and it is a serious threat to users' financial stability. Phishing involves social and technical deception to pilfer the personal identity information and fiscal account credentials from patron. Phishing, it is a cyber attacks, leading to privacy violations, identity theft, and financial harm. In 2017, Kaspersky Lab's statistics revealed that nearly 30% of user computers encountered a Malware-class web attack, with almost 200 million unique URLs being flagged as malicious by web antivirus software. Moreover, the proportion of phishing scams surged from 47.5% to nearly 54% of all phishing incidents, underscoring the escalating threat posed by phishing in the online realm.

Phishing has expanded beyond traditional methods like e-mail, SMS, and pop-ups, utilizing the mobile Internet and social networks for tactics such as QR code phishing, spear phishing, and spoofing mobile applications. Furthermore, deceptive phishing attacks are now being hosted on websites with HTTPS and SSL certificates, leading users to believe they are safe. This evolving trend in phishing presents new obstacles for detection. Phishing website detection has been a focal point for security experts and researchers, who have invested significant resources in combating the elusive and harmful nature of phishes.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The identification of phishing websites heavily relies on the use of blacklists and white lists. These lists are now integrated into popular web browsers provide users with protection against phishing attacks. Google, in particular, offers an up-to-date blacklist of malicious websites. By Google APIs, end user can verify the safety of URL links. By phishing website detection through blacklists and white lists is characterized by its simplicity, fast operation, and low rate of false positives. Nevertheless, data shows that between 47% and 83% of phishing websites are blacklisted within Twelve hours, with 63% of these sites having a lifespan of just Two hours. This indicates a significant lag in updating blacklists compared to the rapid generation of phishing URL. Blacklists and white lists and machine learning methods are utilized in identifying phishing websites. Malicious URLs or phishing pages possess unique attributes that differentiate them from legitimate websites which make machine learning a valuable tool in this process. The predominant machine learning approaches for detecting phishing websites involve extracting statistical features from the URL.

RELATED WORK

A machine learning-based method for detecting phishing attacks technique to integrates various strategies, such as natural language processing (NLP), lemmatization, topic modeling, and advanced learning methods like re sampling and cross-validation, alongside fine-tuning hyper parameters. Remarkably, a small fraction of extracted features, which is less than 0.02% of the feature set, outperformed datasets that underwent dimensionality reduction. This highlights the value of these features in differentiating messages more effectively. Document Term Matrix (DTM) classification algorithms is followed by Latent Allocation (LDA) to handle issues like the "curse of dimensionality," sparsity, and text context inclusion. Performance with an F1 is 99.95% using the XGBoost algorithm, surpassing existing phishing detection methods, especially those focusing only on email body content and ignoring other attributes like headers, IP information, or link counts.[1]

The study in explores a novel method for analyzing phishing web pages by representing URL, HTML content, and the DOM (Document Object Model) structure by sequence of Character. The approach utilizes learning automatically to acquire web page representations, which are processed by a hybrid deep learning. By combining convolution neural networks (CNN) and bidirectional long short-term memory (BiLSTM) networks which emphasize important features in the classification process. Experiments demonstrate the traditional phishing detection methods to achieve 99.05% of accuracy and a false positive rate as 0.25%. The results show that leveraging comprehensive webpage feature extraction through deep learning network to enhance phishing web page detection significantly.[2]

In addressing the malicious URL detection, a neural model is explored to extract both semantic and visual characteristics from malicious URLs. This research introduces techniques for enhancing feature extraction from email body texts. The feature extraction process utilizes distributional representations to extract features, which are then processed with machine learning classification algorithms. A key aspect of this work is the use of a visualization algorithm to create a image of gray that captures characteristics of the URL. Additionally, lexical and character features are extracted and transformed into word and character embedding vectors [3]. Vectors are combined by a neural network to incorporating Capsule Networks (CapsNet) and Independent Recurrent Neural Networks (IndRNN) to capture both visual and semantic information. An attention mechanism is added at the final layer to focus on the most relevant features, resulting in improved classification accuracy and malicious URL detection.[4]

Phishing attacks exploit human vulnerabilities, targeting end-users who are often the weakest link in security defenses. As these attacks are complex, no single solution can effectively address all vulnerabilities. This research reviews various mitigation strategies for phishing, including detection, offense defense, correction, and prevention methods. Understanding the role of phishing detection techniques within the broader mitigation framework is critical in combating these threats [5].

PhishHaven is an innovative system designed to identify phishing URLs generated by DeepPhish, an AI-driven phishing technique. PhishHaven employs lexical feature extraction and analysis to identify URLs in real time. The system introduces a new feature—URL HTML encoding—to enhance its capabilities.[6] Additionally, it includes the URL Hit method for detecting shortened URLs and implements ensemble-based machine learning through multithreading during both training and testing. By using unbiased voting during the decision-making process, PhishHaven assigns final phishing or legitimate labels to URLs efficiently.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Another methodology focuses on automatically learning representations from various features using hybrid deep learning network it treats URL, HTML content, and DOM structure as sequences of characters and applies representation learning techniques to generate their representations. The representations are passed into a hybrid deep learning network composed of CNN and BiLSTM networks to extract both local and global features, enhanced by an attention mechanism. [7]. AI-based systems have become integral to modern phishing detection strategies. Despite their success, these systems often face challenges such as high false alarm rates and limited understanding of phishing techniques.[8][9], In, four meta-learner models—AdaBoost-Extra Tree (ABET), Bagging-Extra Tree (BET), Rotation Forest-Extra Tree (RoFBET), and LogitBoost-Extra Tree (LBET)—were introduced using an extra-tree base classifier to improve accuracy and positive rate [10]

Finally, the THEMIS model, based on an enhanced recurrent convolution neural network (RCNN), effectively captures email characteristics at multiple levels, including the header, body, character, and word levels.[11]. The model evaluated on an imbalanced dataset, reflecting real-world proportions of phishing and legitimate methodology [12].

PROPOSED ARCHITECTURE

The proposed CNN-Bi LSTM comprises of different layers such as cleansing of data, customization of relevant feature and classification. The dataset is a collection free from impersonate users and phishing uniform resource locator from the Internet. The URL contents undergoes following steps such as length reduction of replicated data, consistent coding, and utilizing embedding layer to mitigate data scatterings. In the feature extraction, the CNN captures spatial features, whereas the LSTM captures contextual dependencies and clean-up by soft max. On the second module, multi dimensional features, extract URL probabilistic features, web-page code and semantic text features. The output of the CNN feature extraction is subsequently fed into the BiLSTM for classification. Although the second phase exhibits higher accuracy, incurs a higher time and cost. In real-time detection, DCDA, enhances the soft-max classifier. A threshold is employed to prioritize the accuracy and detection. Additionally, the URL is in accessible and the output of the soft-max is directly utilized for detection.

Architectural Diagram

Filters of different sizes (7x7, 5x5, 3x3) are utilized to traverse each Feature Map (FM) vertically and horizontally. In text processing, a rectangular filter is often employed to cover the entire FM, analyzing the encoded text along a single axis. Additionally, a max-pooling layer serves a dual purpose within the convolution network. Firstly, it accelerates network operations by reducing the number of calculations required at the subsequent convolution layer through FM reduction. Secondly, it enhances the overall performance of the network by selecting the most significant features, leading to improved results post-training. In earlier Convolution Neural Networks (CNNs) is a sub sampling technique utilized to find the mean ratio between filter values and increased computing time regardless of network metrics.

A convolution layer is intricately linked to the input layer size, the filter size, and the stride (step) applied to the convolution filter. Following convolution, the image is always smaller than the original, unless padding is added to maintain the image size. Padding involves appending additional outer pixels to the image. Several machine learning libraries persist now not only incorporate padding automatically to aggregate the layer size but also aligns optimal the filters. Typically, the final layers consist of fully-connected layers with conventional neurons to formation of Multi Layer Perception network (MuLP). The ultimately it reaches the MuLp network with feature map matrix inputs to enabling the execution of the final pattern classification.

The system's architectural design is illustrated in Fig 1. It also demonstrates the overall communication pathways between the main components during the training phase of the project. The dataset undergoes appropriate pre-processing before being fed into a combination of CNN and BiLSTM models, which collectively generate accurate predictions.

Flow of Control

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The control flow diagram illustrates how the system functions as a whole to iteratively improve itself based on user feedback. This methodology incorporates the use of two deep learning algorithms, LSTM and CNN for feature extraction from websites for enhanced phishing activity prediction. The initial stages of our classification process involve feature extraction and machine learning application. LSTM-CNN network is then utilized to accurately classify suspicious and phishing websites in real-time.

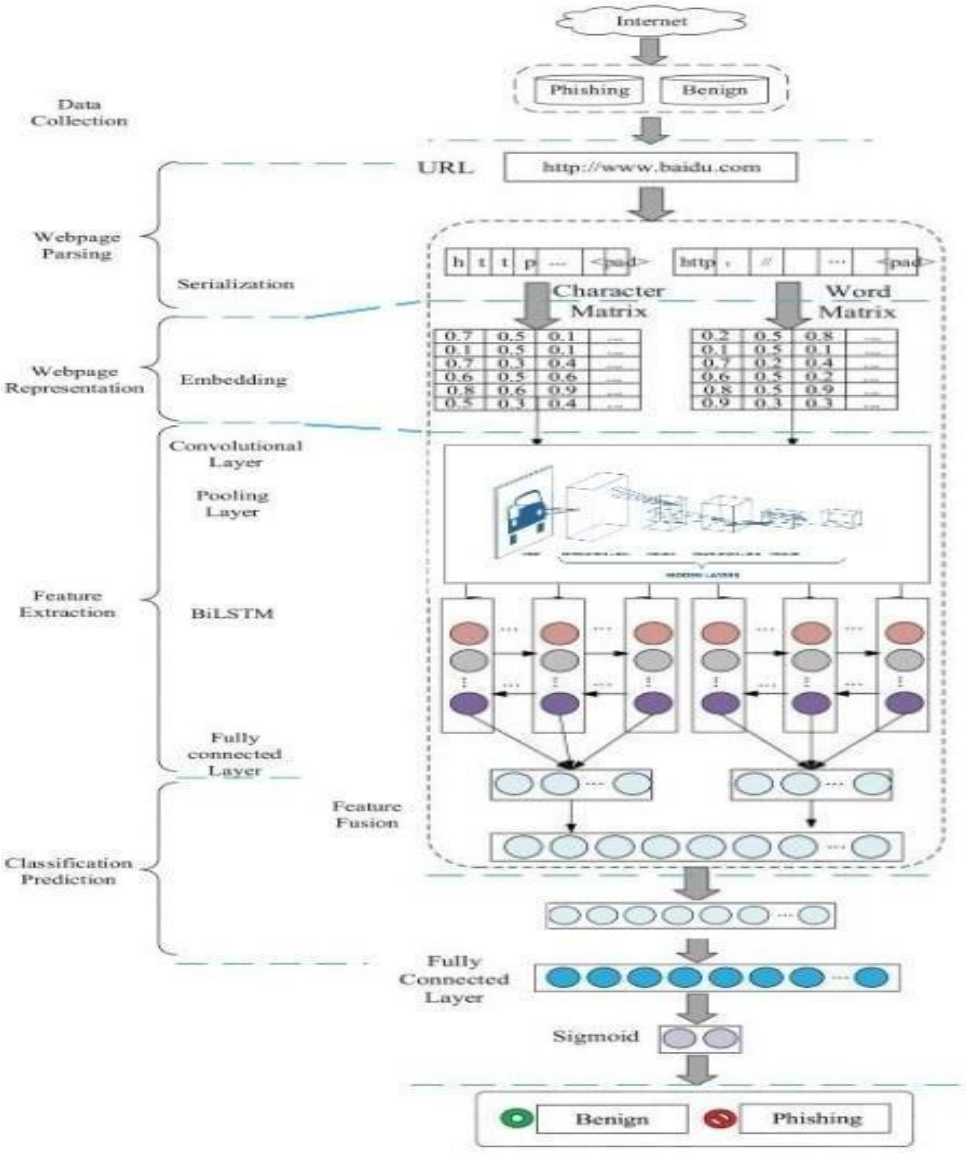

Figure 1. Architectural diagram of the proposed system

Data Set and Preprocessing

The experimental web pages utilized in this study were sourced from an actual network environment. The benign web pages were obtained from Alexa, a website maintained by Amazon that publishes global website rankings. Alexa boasts an extensive collection of URLs and comprehensive website ranking information. A selected web page from the top list of Alexa is to eliminating invalid, erroneous, and duplicate pages on a total of 162,351 normal web pages from Alexa.[13].

On the other hand, the phishing web pages were sourced from PhishTank.com. PhishTank is a widely recognized website dedicated to collecting and providing an up-to-date and authoritative list of phishing webpages. Any individual can submit a suspected phishing webpage to PhishTank, which then verifies its authenticity and

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

determines if it contains fraudulent elements before publishing it. Considering the ephemeral nature of phishing webpages, we collected a total of 157,528 phishing web pages listed on PhishTank between February 2021 and March 2019.[14],[15]. The validation parameter is set into 1:1 ratio. Fig 2 explains the overview of proposed system.

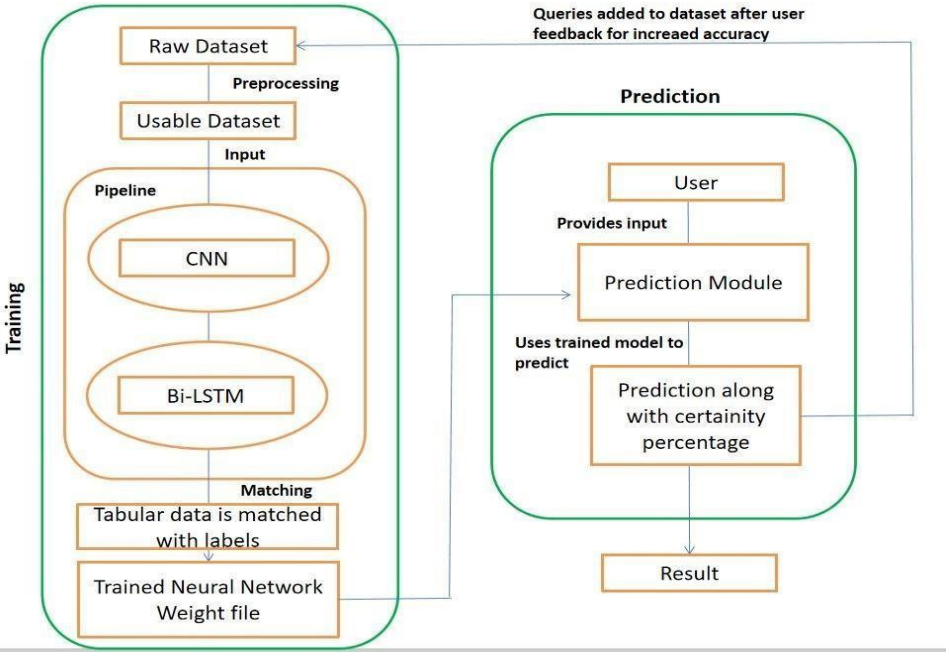

Figure 2. Overview of the proposed system

The primary focus of numerous studies has been on the automated acquisition of webpage attributes from URLs. This is primarily due to the fact that URLs are inherently composed of character sequences, making them amenable to vectorization through preprocessing techniques to process characters and words in the URL.

To ensure uniform normalization of individual or sole character sequences with predefined static length through the use of spoofing or padding to form a customized character set. From this sequence set, a character vocabulary is created by selecting a total of 96 letters, numbers, and special characters that appear frequently. This vocabulary includes a special symbol "<UNK>" which is used to replace infrequent characters, as well as a placeholder "<PAD>". Each character in the vocabulary is assigned a unique number. Subsequently, each URL character sequence in the sequence set is encoded. This encoding involves replacing each original character with its corresponding number from the vocabulary. As a result, a one-dimensional digital vector called the One-Hot code is obtained for each URL character sequence. These encoded sequences collectively constitute the character-level corpus of URLs. Fig 3 illustrates the process of constructing a character-level corpus. In this process, each URL is considered as a sequence of characters

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

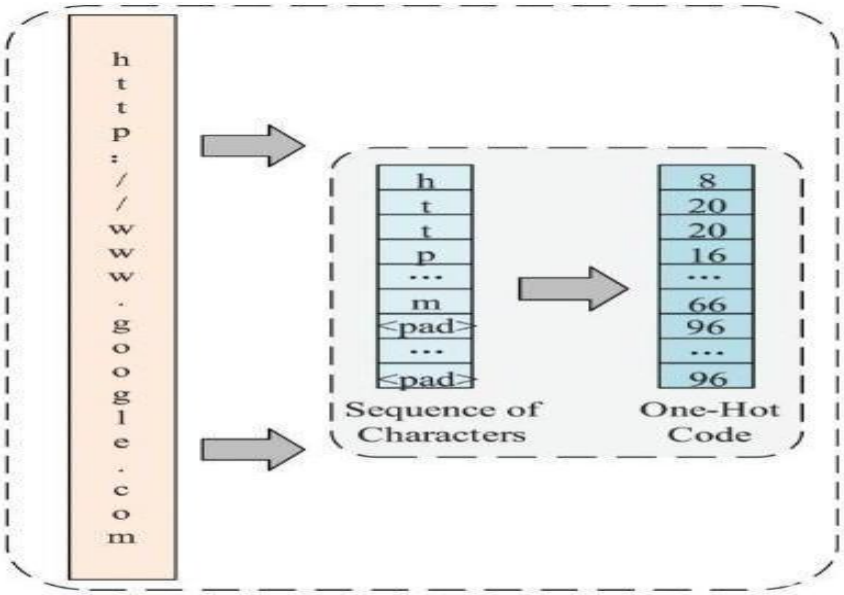

Figure 3. URL Preprocessing

Training Module

The CNN-BiLSTM algorithm consists of three main parts: embedded URL embedding, characteristic extraction and classification. Fig 3 a visual representation of the training process connected with this algorithm. The Word Embedding representation model concern with normalization of URL code sequences with predefined length regardless of elite features and filing features. one-hot code sequence obtains normalization of given URL code. Consequently, conversion of dense lettering into bedding matrix done at the embedded layer generates sparse one hot matrix. When dealing with characteristic extraction levels, local, deep correlated features of the embedded matrix are extracted using folding layers and maximum pooling of CNNs. The results of the pooling operation are fed to the Neural Network LSTM and record the context of the URL sequence.

In the classification stage, LSTM neural network fed to the activation unit. Well-defined dropout or loss function and the aggregate unit outputs the probability of URL belongs to a phishing website. To optimize the overall performance of the set of rules, the loss characteristic is constantly adjusted via way of means of incremental weights the neural community throughout incremental phases of the model. One normally used optimization method is the stochastic gradient descent (SGD) set of rules, which calculates the gradient for every pattern and updates the parameters hence throughout the education method. However, common parameter updates via way of means of SGD can result in excessive oscillations withinside the loss characteristic and can save you the attainment of the minimal value. To cope with this issue, the Adam (adaptive second estimation) set of rules is applied as a development over SGD. Adam calculates unbiased adaptive mastering prices for distinct parameters via means of estimating the primary and 2d moments of the gradient. Compared to other optimization algorithms, Adam mitigates problems such as the disappearance of the learning rate, slow convergence, and significant fluctuations in the loss function and it is explained in fig 4.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

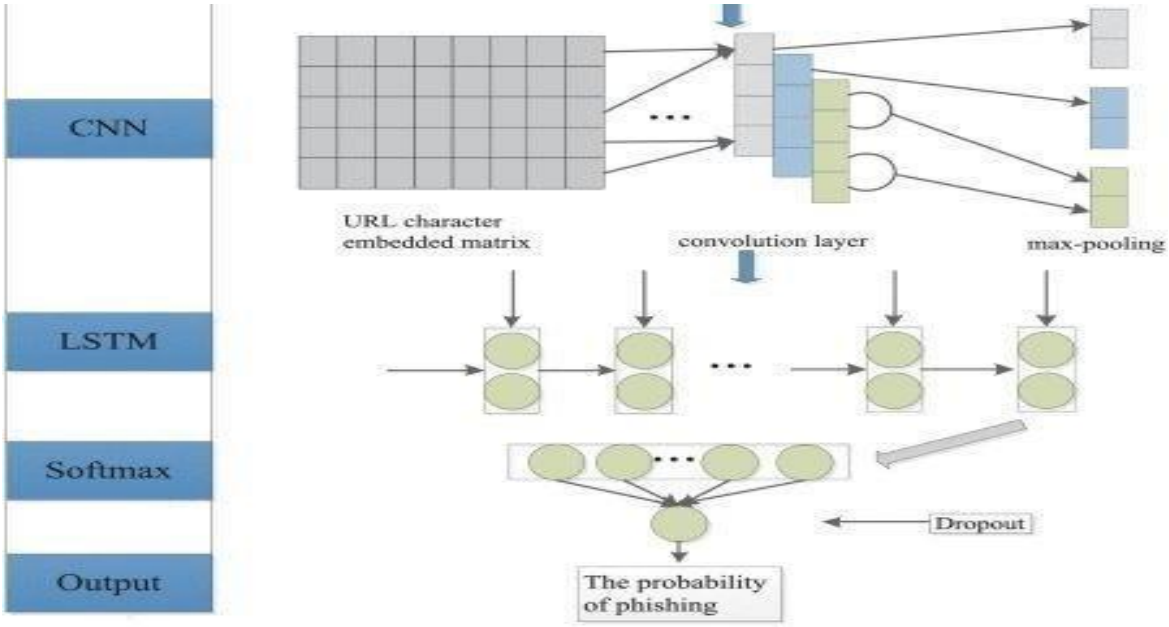

Figure 4. Training Process

Prediction Module

The training module produces a .h5 file which contains the structure and weights of the neural network that has been implemented. The model is then used in this module where the user can give an input for which a prediction is made by the.h5file. The URL contained information and varied length sizes. The user query has a tube pre-shaped in a way that the trained model understands, thus some manipulation is again required. Fig 5 depicts the various prediction flow of the detection of phishing systems.

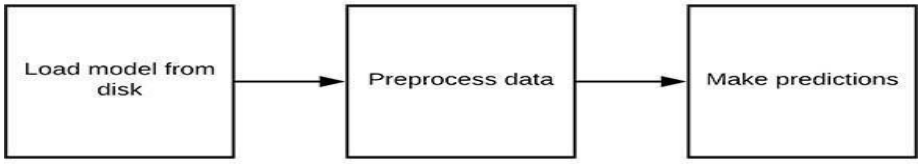

Figure 5. Prediction Flow

The data has to be pre-processed to make it available for training the model. Multiple URLs need to be gathered and stored in Microsoft Excel as comma-separated values, with each URL occupying one column and its corresponding category label in another column. Once the prediction is generated, the user is requested to provide feedback regarding the accuracy of the prediction. In the event of an incorrect decision, the URL is saved as raw data for future training purposes, aiming to enhance the system's precision.

PARAMETER AND RESULTS

Parameter Selection

Number of EPOCHS

Initially, it's vital to make changes to the parameters of the CNN-BiLSTM algorithm. The carried out test well-known shows that the common duration of valid internet site samples withinside the dataset DATA is 34.7, at the same time as the common duration of phishing internet site samples is 87.3. Moreover, the general common duration of all of the statistics is calculated to be 61.5. Additionally, it's miles located that the duration of URLs exceeding 98.3% is underneath 400.Consequently, set up the constant duration of the URL as L=400. The cross-validation with inside the CNN-BiLSTM is depicted in Fig 6, illustrating the common progression.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Selection of Optimizer

Adam optimization is an extension of stochastic gradient descent in deep learning and natural language processing. Its implementation is relatively straight forward.

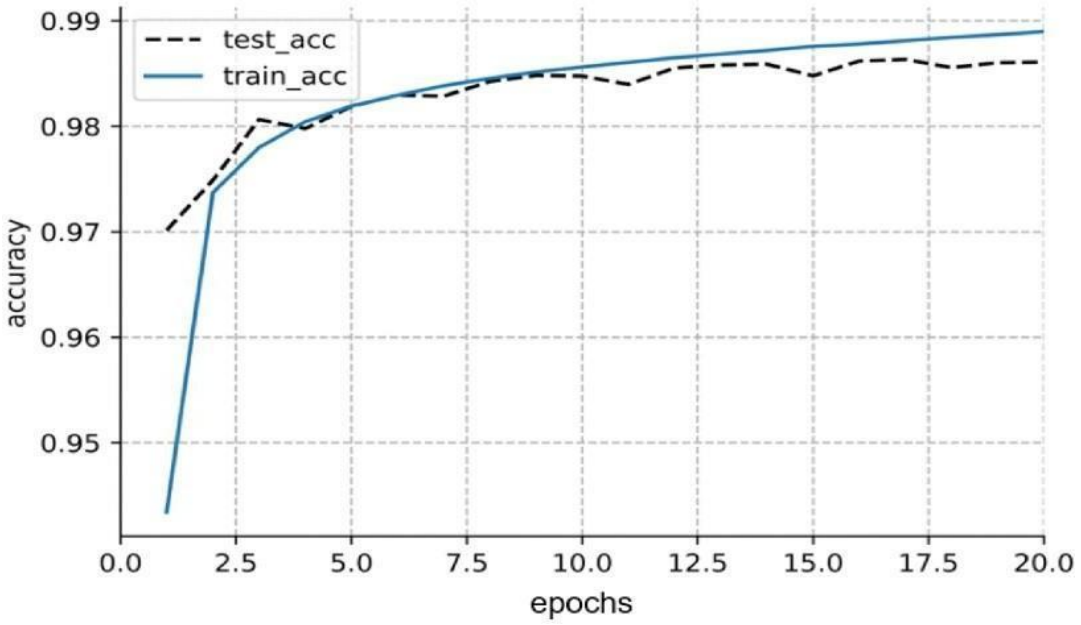

Figure 6. CNN-BiLSTM average training curve

Root Mean Square Propagation (RMSProp) algorithm address the challenges posed by online and non-stationary problems encountered in natural language and computer vision tasks and maintaining learning rates that are adjusted according to the average magnitudes of recent gradients for each weight. By considering how quickly the weights are changing, RMSProp effectively adapts its learning rates, making it particularly suitable for handling noisy and dynamic environments and shown in Fig 7.

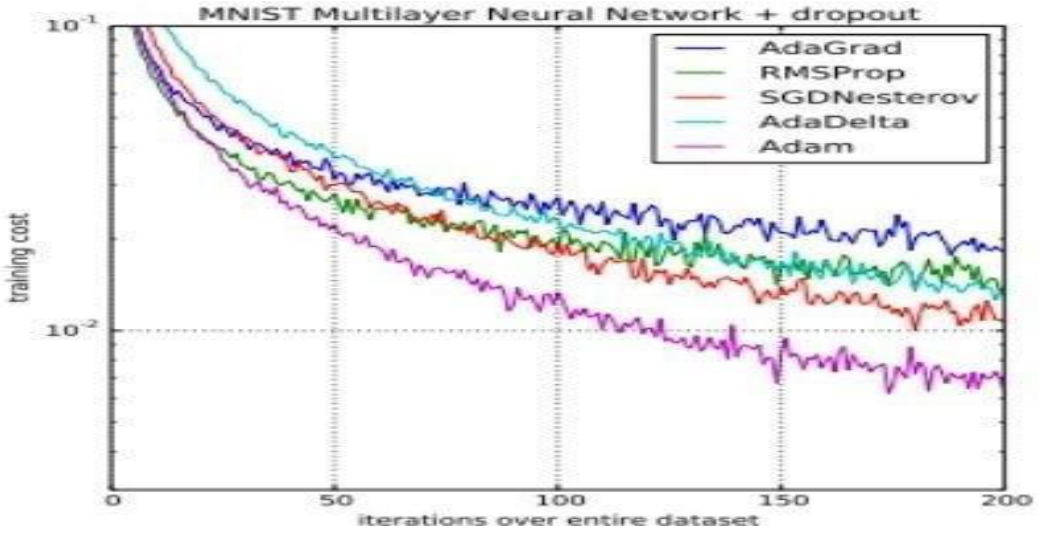

Figure 7. Comparisons of Optimizers

Accuracy

Proposed technique evaluated by employing a dataset that encompassed 157,528 malicious URLs obtained from PhishTank, along with 162,542 URLs ranked by Alexa. A substantial classification accuracy of 98.69% and it is shown in Fig 8.

Journal of Information Systems Engineering and Management
2025, 10(48s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
Training
Epoch 1/25
7072/7072 [==============================] - 182s 25ms/step - loss: 0.1830 - accuracy: 0.9342 - val_loss: 0.1004 - val_accuracy: 0.9681
Epoch 2/25
7072/7072 [==============================] - 178s 25ms/step - loss: 0.0790 - accuracy: 0.9760 - val_loss: 0.0699 - val_accuracy: 0.9783
Epoch 3/25
7072/7072 [==============================] - 178s 25ms/step - loss: 0.0607 - accuracy: 0.9820 - val_loss: 0.0603 - val_accuracy: 0.9817
Epoch 4/25
7072/7072 [==============================] - 177s 25ms/step - loss: 0.0482 - accuracy: 0.9864 - val_loss: 0.0544 - val_accuracy: 0.9837
Epoch 5/25
7072/7072 [==============================] - 177s 25ms/step - loss: 0.0415 - accuracy: 0.9884 - val_loss: 0.0566 - val_accuracy: 0.9829
Epoch 6/25
7072/7072 [==============================] - 178s 25ms/step - loss: 0.0397 - accuracy: 0.9888 - val_loss: 0.0535 - val_accuracy: 0.9851
Epoch 7/25
7072/7072 [==============================] - 178s 25ms/step - loss: 0.0352 - accuracy: 0.9909 - val_loss: 0.0545 - val_accuracy: 0.9846
Epoch 8/25
7072/7072 [==============================] - 177s 25ms/step - loss: 0.0277 - accuracy: 0.9932 - val_loss: 0.0507 - val_accuracy: 0.9863
Epoch 9/25
7072/7072 [==============================] - 177s 25ms/step - loss: 0.0269 - accuracy: 0.9934 - val_loss: 0.0546 - val_accuracy: 0.9866
Epoch 10/25
7072/7072 [==============================] - 177s 25ms/step - loss: 0.0241 - accuracy: 0.9944 - val_loss: 0.0566 - val_accuracy: 0.9850
Epoch 11/25
7072/7072 [==============================] - 176s 25ms/step - loss: 0.0234 - accuracy: 0.9945 - val_loss: 0.0541 - val_accuracy: 0.9856
Epoch 12/25
7072/7072 [==============================] - 175s 25ms/step - loss: 0.0224 - accuracy: 0.9950 - val_loss: 0.0565 - val_accuracy: 0.9859
Epoch 13/25
7072/7072 [==============================] - 174s 25ms/step - loss: 0.0226 - accuracy: 0.9948 - val_loss: 0.0504 - val_accuracy: 0.9871
Epoch 14/25
7072/7072 [==============================] - 174s 25ms/step - loss: 0.0216 - accuracy: 0.9952 - val_loss: 0.0501 - val_accuracy: 0.9859
Epoch 15/25
7072/7072 [==============================] - 173s 25ms/step - loss: 0.0210 - accuracy: 0.9954 - val_loss: 0.0518 - val_accuracy: 0.9865
Epoch 16/25
7072/7072 [==============================] - 173s 25ms/step - loss: 0.0208 - accuracy: 0.9953 - val_loss: 0.0525 - val_accuracy: 0.9861
Epoch 17/25
7072/7072 [==============================] - 173s 24ms/step - loss: 0.0207 - accuracy: 0.9954 - val_loss: 0.0509 - val_accuracy: 0.9864
Epoch 18/25
7072/7072 [==============================] - 174s 25ms/step - loss: 0.0211 - accuracy: 0.9953 - val_loss: 0.0514 - val_accuracy: 0.9866
Epoch 19/25
7072/7072 [==============================] - 173s 24ms/step - loss: 0.0199 - accuracy: 0.9957 - val_loss: 0.0527 - val_accuracy: 0.9861
Epoch 20/25
7072/7072 [==============================] - 172s 24ms/step - loss: 0.0199 - accuracy: 0.9956 - val_loss: 0.0520 - val_accuracy: 0.9857
Epoch 21/25
7072/7072 [==============================] - 172s 24ms/step - loss: 0.0203 - accuracy: 0.9955 - val_loss: 0.0543 - val_accuracy: 0.9850
Epoch 22/25
7072/7072 [==============================] - 172s 24ms/step - loss: 0.0222 - accuracy: 0.9949 - val_loss: 0.0484 - val_accuracy: 0.9869
Epoch 23/25
7072/7072 [==============================] - 172s 24ms/step - loss: 0.0210 - accuracy: 0.9953 - val_loss: 0.0530 - val_accuracy: 0.9853
Epoch 24/25
7072/7072 [==============================] - 176s 25ms/step - loss: 0.0208 - accuracy: 0.9953 - val_loss: 0.0576 - val_accuracy: 0.9846
Epoch 25/25
7072/7072 [==============================] - 176s 25ms/step - loss: 0.0230 - accuracy: 0.9945 - val_loss: 0.0538 - val_accuracy: 0.9866
2358/2358 [==============================] - 22s 9ms/step - loss: 0.0532 - accuracy: 0.9870
Test score: 0.053237155079841614
Test accuracy: 0.986981213092804
```

Figure 8. Accuracy and Loss Statistics

CONCLUSION AND FUTURE WORK

Distinguishing legitimate URLs from malicious URLs is by combining the two technologies of CapsNet and IndRNN. Phish Tank and the malware domain list and more than 30,000 legitimate URLs ranked by Alexa. The suggested approach achieved an impressive classification accuracy of 99.78%. Despite the satisfactory performance of the proposed model, there is still room for improvement.

Further studies and enhancements are necessary to optimize the entire system. One possible approach to enhance the model structure involves incorporating malicious classes for multiple classifications through various modifications. In subsequent research, it may be beneficial to select newer and more advanced versions to replace certain components of the model.

REFERENCES

[1] Eder S. Gualberto, RafaelT. DeSousa, Thiago P. DeB.Vieira, JoãoPaulo C. L. DaCosta, Cláudio G. Duque, From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection", IEEE Access, 8,2020

[2] J. Feng, L. Zou, O. Ye and J. Han, "Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning," in IEEE Access, vol. 8, pp. 221214-221224, 2020, doi: 10.1109/ACCESS.2020.3043188

[3] Maria Sameen, Kyunghyun Han, Seong Oun Hwang, "PhishHaven—An EfficientReal-Time AI Phishing URLs Detection System", IEEE Access Volume:8,2020

[4] Raj M. N., Vithalpura P. J. 'A survey on phishing detection based on visual similarity of Web pages", Int.J.Sci.Res.Sci., Eng.Technol., 2018

[5] Bilal Naqvi, Kseniia Perova, Ali Farooq, Imran Makhdoom, Shola Oyedeji, Jari Porras, Mitigation strategies against the phishing attacks: A systematic literature review, Computers & Security, Volume 132, 2023, https://doi.org/10.1016/j.cose.2023.103387.

Research Article

[6] Aldakheel, E. A., Zakariah, M., Gashgari, G. A., Almarshad, F. A., & Alzahrani, A. I. (2022). A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators. Sensors, 23(9), 4403. https://doi.org/10.3390/s23094403

[7] Kaur, G., Sharma, A. A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. J Big Data 10, 5 (2023). https://doi.org/10.1186/s40537-022-00680-6

[8] V. Anjana Devi, "ECC Based Malicious Node Detection System for MANET", Journal of Theoretical and Applied Information Technology, vol.68, no.2, pp.239-248, 2014.

[9] Pijush Samui, Dookie Kim, and Viswanathan R (2014). Spatial variability of rock depth using adaptive neuro-fuzzy inference system (ANFIS) and multivariate adaptive regression spline (MARS), Environ Earth Sci, Vol. 73, pp. 4265–4272

[10] Basit, A., Zafar, M., Liu, X. et al. A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun Syst 76, 139–154 (2021). https://doi.org/10.1007/s11235-020-00733-2

[11] Atawneh, S., & Aljehani, H. (2022). Phishing Email Detection Model Using Deep Learning. Electronics, 12(20), 4261. https://doi.org/10.3390/electronics12204261

[12] Arjun K.P., Sreenarayanan N.M., Kumar K.A., Viswanathan R. Reforming the traditional business network (2021) Block chain and Machine Learning for e-Healthcare Systems, pp. 185 – 210.

[13] Kennedy, R.K.L., Villanustre, F., Khoshgoftaar, T.M. et al. Synthesizing class labels for highly imbalanced credit card fraud detection data. J Big Data 11, 38 (2024). https://doi.org/10.1186/s40537-024-00897-7

[14] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. Frontiers in Computer Science, 3, 563060. https://doi.org/10.3389/fcomp.2021.563060

[15] Sánchez-Paniagua, M., Fidalgo, E., Alegre, E., & Alaiz-Rodríguez, R. (2022). Phishing websites detection using a novel multipurpose dataset and web technologies features. Expert Systems with Applications, 207, 118010. https://doi.org/10.1016/j.eswa.2022.118010