

Deep learning based phishing website detection

N. Subhashini, Amogh Banerjee, Abhi Kumar, S. Muthulakshmi, S. Revathi

School of Electronics Engineering, Vellore Institute of Technology, Chennai, India

Article Info

Article history:

Received Jul 12, 2023

Revised Aug 22, 2023

Accepted Aug 30, 2023

Keywords:

Deep learning

Detection online security

Long short-term memory

Gated recurrent unit

Phishing website

ABSTRACT

Phishing attacks use fraudulent websites that trick people into disclosing sensitive information. More effective and precise methods are required to identify phishing websites so that people and organisations can be protected from the damaging effects of these online threats. The aim of this work is to develop a model that can identify phishing uniform resource locator (URLs) more accurately than current approaches while requiring less training time, testing time, and storage space. This research work proposes a novel method for identifying phishing websites using a long short-term memory (LSTM) gated recurrent unit (GRU) algorithm to detect phishing URLs. The accuracy of the suggested method is 98.89%, which is significantly better than the findings of earlier studies. The model also showed a need for shorter training and testing time, and a reduced amount of storage space.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

N. Subhashini

School of Electronics Engineering, Vellore Institute of Technology

Chennai, India

Email: subhashini.n@vit.ac.in

1. INTRODUCTION

In today's digital environment, where people and businesses are more frequently exposed to a number of cyber threats, identifying phishing websites is a crucial problem. In a phishing attack, an attacker on the internet sets up a fake website or email that looks legitimate with the intention of tricking unsuspecting victims into divulging sensitive information. Because the fraudulent website may look remarkably similar to the real ones, this kind of phishing attack can be challenging to detect [1]. Users can safeguard themselves by being watchful to ensure that the uniform resource locator (URL) of the website they are visiting is correct and by never clicking on links in shady emails. Employing two-factor authentication and informing users about the dangers of phishing are measures that businesses can take to stop phishing attacks. The complexity of phishing attacks is also rising, making it challenging for people and organizations to recognize and stop them [2]. To make their victims more susceptible to the attack, attackers may use social engineering techniques to instill a feeling of urgency or panic in them. Therefore, it is essential to develop efficient phishing website detection techniques in order to reduce the risks connected with phishing attacks.

It is possible to create novel approaches that can help protect people and organisations from the damaging effects of phishing attacks by utilising the abilities of artificial intelligence (AI). In this work, long short term memory (LSTM) and gated recurrent unit (GRU) are two deep learning models that are combined to form the LSTM-GRU model. These models have the ability to process sequential data, like URLs, by main-training an internal state that captures information from previous inputs. By leveraging the ability to learn both short-term temporal dependencies and longer-term dependencies, the model is able to capture complex patterns in the input data. By using a sizable dataset of authentic and phishing URLs to train the model, it can learn to recognise common features such as unusual domain names, suspicious keywords, and other anomalies. In this proposed work, the LSTM-GRU model has demonstrated to be highly accurate at

identifying phishing websites, making it a useful tool for enhancing online security and protecting users from possible cyber threats.

A multidimensional approach using deep learning was proposed which uses a convolutional neural network (CNN) and bidirectional LSTM [3]. Such a method helps detect phishing websites without prior knowledge of website features by making use of static features which helps avoid the security risks which arise when dealing with dynamic features. Yazhmozi *et al.* [4] proposed a deep learning solution which uses both LSTM and CNN for detecting phishing websites. Singh *et al.* [5] proposed a system which uses CNN to detect phishing website. Their system tokenises the URLs into sequences of length 30 and converts them into matrices by passing them through a GloVe embedding layer of dimension 100 and then passing them through the CNN network, followed by the dense network which uses a SoftMax activation to classify the URL.

Wang *et al.* [6] presented a model with recurrent CNN. It converts URL data into a 2-D tensor and feeds the tensor into a cleverly designed deep learning neural network in order to identify the original URL. Ali and Ahmed [7] employed a deep learning-based strategy together with feature selection and weighting based on genetic algorithms (GA). To categories URLs, they created a deep neural network (DNN) and trained it using the dataset. Yang *et al.* [8] developed a method which uses CNN and random forest (RF) ensemble learning for phishing website detection. They created a more advanced version of CNN and took features from several CNN levels. Three RFs make up an RF ensemble that receives the features retrieved from CNN. Several RF classifiers are given features retrieved from various layers. Zheng *et al.* [9] suggested an approach that considers both word-level and character-level URL characteristics. The deep pyramid CNN is used to extract long-range text dependencies. They performed a model ablation analysis in which they found that combining word-level and character-level features gives better performance than using a single embedded feature.

Tang and Mahmoud [10] developed a framework which consists of data collection, recurrent neural network-gated recurrent unit (RNN-GRU) as machine learning model, browser extension and cloud application. Do *et al.* [11] examined 81 papers that were carefully chosen for their rigorous literature reviews on taxonomy of deep learning algorithms for phishing detection. They performed an empirical analysis in which they included the deep learning models DNN, multi layer perceptron (MLP), CNN, RNN, LSTM, GRU, and auto encoder (AE). It was discovered that compared to DNN and MLP, CNN, LSTM, and GRU are less prone to overfitting. Al-Ahmadi *et al.* [12] proposed a generative adversarial network (GAN) which uses CNN and LSTM to detect phishing websites. Their GAN consists of a CNN discriminator and an LSTM generator. Lin *et al.* [13] developed a type of method in which the presence of sensitive inputs in web pages is checked along with the character-level representation of URLs to determine whether they are genuine or phishing. They used the faster-RCNN model to detect sensitive inputs from screenshots of webpages using object detection. The classifier light gradient boosting machine (GBM) was used for classifying the websites. Yang *et al.* [14] proposed the use of three modules make up their framework, the multidimensional features module, the CNN-LSTM module, and the dynamic category decision algorithm (DCDA) module.

Gupta *et al.* [15] proposed a hybrid feature-based phishing website detection method. They combined URL features and hyperlink features of phishing websites to form hybrid features. They compared the models RF, decision tree, support vector machine (SVM), logistic regression (LR) and XG-boost (XGB). Abuadba *et al.* [16] proposed a method which includes a horizontal feature space along with the vertical feature space for phishing website detection. Their system first collects information about a webpage from multiple trusted services, performs correlation and feature extraction, and then classifies the webpage as phishing or legitimate using LR. Aljofey *et al.* [17] used XGB for classification and XGB outperformed RF, LR, Naïve Bayes (NB), ensemble of RF, and AdaBoost in their experiment. Almomani *et al.* [18] used 16 machine learning models for their experiments, which are RF, bagging, decision tree, extra tree classifier, gradient boosting, support vector classifier (SVC), k-nearest neighbors (kNN), AdaBoost, linear SVC, logistic regression cross-validation (CV), ridge classifier CV, perceptron, BernoulliNB, passive aggressive classifier, stochastic gradient descent (SGD), and GaussianNB.

Mughaid *et al.* [19] applied 7 supervised classification algorithms to training data. Out of those 7 methods, the best accuracy was obtained by boosted decision tree. Kalabarige *et al.* [20] developed a multilayer stacked ensemble learning model for phishing website detection. The first layer contains the classifiers XGB, LR, RF, MLP, and kNN. The second layer contains XGB, RF and MLP. The third and final layer is a meta layer which contains XGB.

From the study of existing works, it has been found that deep learning approaches are suitable for the task of phishing website identification because of their high performance [21]. Moreover, using multiple models having the ability to learn different types of features together helps to improve the detection accuracy. But this makes these models complex and increases their training time and testing time. Though feature engineering and feature selection help extract, select and use relevant types of features for website classification, this also increases the time taken in the process of solving the problem. In the cases where the

hypertext markup language (HTML) features of websites are used, models may fail to detect phishing websites if embedded objects are used to hide the content. Therefore a deep learning approach based solution is needed that can automatically extract and learn various types of information while also being less sophisticated and able to effectively identify phishing websites simply by looking at their URLs. This work focusses on building such a model.

2. METHOD

2.1. Dataset

From literatures, we found that models perform better when trained on balanced datasets. So, we created our balanced URL dataset by collecting 29163 phishing URLs from PhishTank and 29163 legitimate URLs by scraping the webpages of 469 domains which were available out of the top 500 domains listed in Moz [22], [23]. The dataset has two columns, 'url' and 'label'. The column 'url' contains the URL of a website and the column 'label' contains a value of 0 representing a legitimate website or 1 representing a phishing website.

2.2. The proposed method

The input URL is first tokenised to convert it into a list of characters. Then the obtained character list is encoded by a character-to-integer mapping. To ensure that all sequences are of the same length, the length is fixed as 120 because 95% of the URLs present in our dataset have length less than or equal to 120. Sequences having length greater than 120 are truncated and those having length less than 120 are padded with zeros at the end. The resulting sequence is then fed to the LSTM-GRU network. The LSTM layer is used to fetch longterm dependencies and the GRU layer is used to fetch short-term dependencies and more general features of the URL. The extracted features are passed through multiple dense layers in the neural network. The completely linked output layer, which makes up the last layer of the network, assesses the URL as legitimate or phishing. The block diagram of the proposed method is shown in Figure 1.

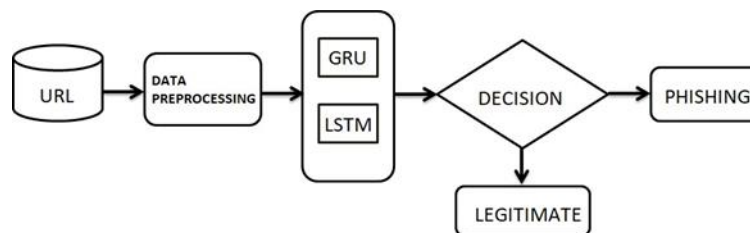


Figure 1. Block Diagram of the Proposed Method

2.3. Model architecture

The first layer is an input layer that takes in input sequences of length 120, corresponding to the length of the URL sequences after padding. The second layer is an embedding layer that maps each character in the input sequences to a dense vector of length 64. The third and fourth layers are GRU and LSTM layers with 64 units, respectively, which are used to learn the sequential patterns in the input data. The outputs of these layers are then concatenated along the column axis to form a vector of length 128. The remaining layers consist of fully connected dense layers with 64 and 32 units, respectively, which are used to perform classification based on the learned features. The output layer has a single unit and a sigmoid activation function, which produces a probability score indicating the likelihood that the input sequence belongs to the phishing website class. Model optimisation is carried out using the Adam optimizer after training with binary cross-entropy loss with a learning rate of 0.003. Finally, the model predicts whether a given input URL sequence is a phishing website or not. The architecture of the LSTM- GRU model is shown in Figure 2.

2.3.1. Input layer

This input layer is used as the entry point into the neural network. The shape of this layer has been specified as 120. This means that the input layer expects input sequences of length 120, which corresponds to the length of the URL sequences after preprocessing. By specifying the input shape as 120, we ensure that the neural network expects all input sequences to have the same length, regardless of their original length.

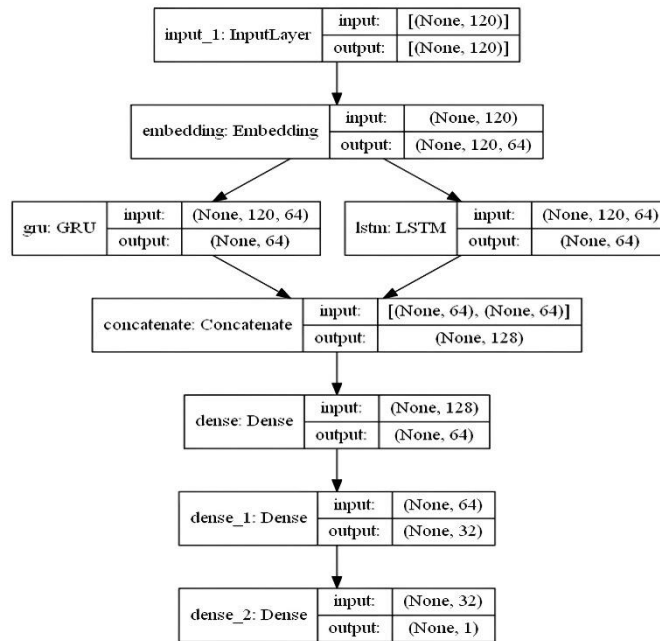


Figure 2. Architecture of the LSTM-GRU model

2.3.2. Embedding layer

The purpose of the embedding layer here is to transform the input sequence into dense vectors of fixed size that can be easily processed by the neural network. The data is converted into a continuous vector representation. Each character in the input sequence is represented by a corresponding vector in the embedding layer. This mapping is learned during the training process of the model, where the embedding layer weights are updated to minimise the loss function.

2.3.3. GRU layer

A GRU layer with 64 units is added to the model. The GRU layer has a simple gating mechanism. The ‘units’ parameter specifies the number of GRU cells in the layer. Each GRU cell maintains a hidden state that is updated at each timestep of the input sequence, and the number of cells determines the capacity of the layer to learn temporal patterns in the data. The choice of 64 as the value of units is a hyperparameter that can be tuned based on the size and complexity of the input data, as well as the desired level of expressiveness in the learned representations. In this case, 64 was chosen to match the size of the embedding vectors in the previous embedding layer and the LSTM layer, which can help facilitate the learning of more complex features in the input data.

2.3.4. LSTM layer

An LSTM layer with 64 units is added to the model. The LSTM layer is useful for processing sequential data such as text or time series. Therefore, it is suitable for processing URLs. In this layer, the number of units has been chosen as 64 to specify the number of LSTM cells in the layer. This was chosen based on the size and complexity of the input data, as well as the desired level of expressiveness in the learned representations. With a higher value of units, the LSTM layer would have more capacity to learn complex temporal patterns in the input data, but this would also increase the number of model parameters and the risk of overfitting. Conversely, with a lower value of units, the LSTM layer would have less capacity but would also have fewer parameters and be less likely to overfit. After experimenting with different values, it was found that 64 was a good balance that gave good performance without overfitting the model.

2.3.5. Concatenation

The model enhances its learning by combining the outputs of GRU and LSTM layers. GRU is recognized for its simplicity and efficiency in capturing short-term temporal dependencies. In contrast, the LSTM layer is adept at modeling longer-term dependencies and capturing complex patterns in the input data. Through this concatenation, the model gains a nuanced understanding of both short-term and long-term aspects, leading to more expressive representations of the input data.

2.3.6. Output layer

The last layer of the neural network is a dense layer with a single neuron and a sigmoid activation function. This layer is used for our binary classification task, where the goal is to predict one of two possible outcomes. The sigmoid function takes a real-valued input and maps it to a value between 0 and 1, which can be interpreted as a probability.

3. RESULTS AND DISCUSSION

3.1. Experimental setup

In the ratios of 60:20:20, the dataset was split into training, validation, and testing sets. The deep learning models used in the experiments were developed, trained, and evaluated using the Keras package [24]. On a system with an Intel i5 10th generation processor, 8 GB of DDR4 RAM, and a 4 GB NVIDIA Geforce GTX 1650 GPU, the experiments were carried out.

3.2. Performance metrics

The evaluation of the proposed model was done using seven metrics [25]. These are, specificity, F1-score, recall, false negative rate (FNR), false positive rate (FPR), and accuracy. The percentage of URLs that were correctly classified out of all URLs is known as accuracy. A model with a high accuracy is capable of correctly classifying URLs as phishing or legitimate.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

Here, TP is the number of true positives, TN is the number of true negatives, FP is the number of false positive and FN is the number of false negatives. The percentage of accurate positive predictions compared to all positive predictions is known as precision. It is a measure of the amount of URLs correctly classified as phishing out of all URLs that were predicted to be phishing.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall, also known as sensitivity, is the percentage of true positive predictions among all real positive cases. Out of all the URLs that are truly phishing, it calculates the percentage of URLs that were correctly identified as such.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

The harmonic mean of recall and precision is known as the F1-score. A high F1-score offers a fair compromise between recall and precision.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

The specificity metric shows the percentage of legitimate URLs that are correctly categorised as negative cases (i.e., actual negative cases).

$$Specificity = \frac{TN}{TN+FP} \quad (5)$$

FPR represents the percentage of legitimate URLs that are misclassified as phishing.

$$FPR = \frac{FP}{FP+TN} \quad (6)$$

FNR represents the percentage of phishing URLs) that are misclassified as legitimate.

$$FNR = \frac{FN}{FN+TP} \quad (7)$$

Other important metrics which we have used for the evaluation of models are training time, testing time and storage space. The training time and testing time may vary based on elements like the model's architecture and the computing power available. Longer training and testing times can produce models that

are more accurate, but they might also demand more time and resources, which can be a barrier to real-time phishing website detection. The storage space is also a significant with regard to the model's scalability and simplicity of deployment. For devices with limited resources or cloud environments with little available storage space, a model that requires a lot of space may not be suitable.

3.3. Model evaluation and comparison

We took the architecture of deep learning models from [1]–[4] and implemented them. Each of these models is capable of learning from URLs and detecting phishing URLs with high accuracy. For simplicity, we have named these models as CNN+BiLSTM, CNN+LSTM, BiLSTM+Parallel CNN, and multibranch CNN respectively. A comparison of the performance of the proposed model with each of these models was done.

Table 1 shows the comparison of the proposed LSTM-GRU model with the others. It can be observed that our proposed model achieved the highest accuracy and F1-score of all the models. In terms of precision and specificity, the proposed model outperformed all the models except CNN+LSTM. On the other hand, it outperformed CNN+LSTM in terms of recall. The proposed model outperformed CNN+BiLSTM, BiLSTM+Parallel CNN and multibranch CNN in terms of FPR but not CNN+LSTM in terms of FNR. Figures 3 to 7 shows a bar graph representation of the comparison of accuracy, precision, F1 score, specificity, false positive rate comparison between the models.

Table 1. Performance comparison of deep learning models

Model	Accuracy	Precision	Recall	F1-score	Specificity	FPR	FNR
LSTM-GRU	0.988857	0.991723	0.985942	0.988824	0.991771	0.008229	0.014058
CNN+BiLSTM	0.987314	0.982352	0.992457	0.987378	0.982170	0.017830	0.007543
CNN+LSTM	0.985513	0.995972	0.974970	0.985359	0.996057	0.003943	0.025030
BiLSTM+Parallel CNN	0.985085	0.977875	0.992628	0.985197	0.977542	0.022458	0.007372
Multibranch CNN	0.986113	0.985115	0.987142	0.986128	0.985085	0.014915	0.012858

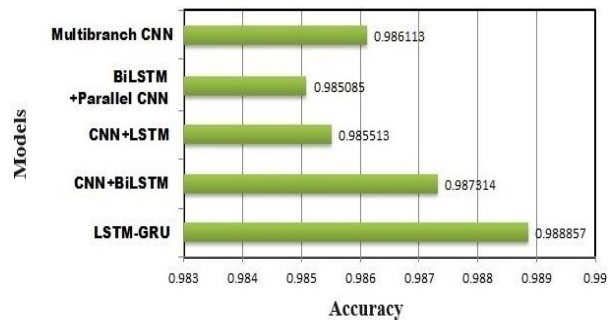


Figure 3. Comparison of accuracy

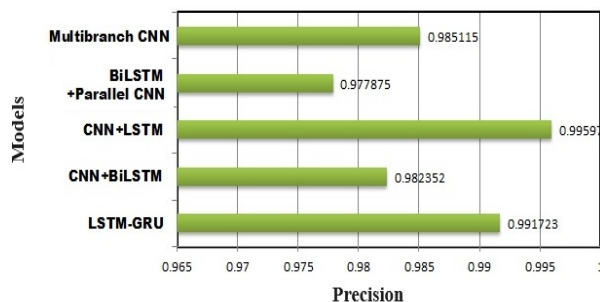


Figure 4. Comparison of precision

The comparison of training time, testing time, and storage space is shown in Table 2. It is observed that the developed LSTM-GRU model is quicker to train than CNN+LSTM and BiLSTM+Parallel CNN. When testing the models, it is the fastest of all models except Multibranch CNN. Moreover, the proposed model takes the least amount of storage space.

The proposed model exhibits higher FPR, specificity, accuracy, and precision. Additionally, it requires less storage space. Accuracy, precision, specificity, and FPR are all crucial criteria to take into account while detecting phishing URLs. This is because the repercussions of incorrectly classifying a phishing URL as secure (false negative) or a safe URL as phishing (false positive) can be severe, such as financial loss or compromise of sensitive data. Furthermore, storage space is crucial in the context of phishing URL detection because it might not always be possible to build a large model. The suggested model would therefore be a superior option for this assignment.

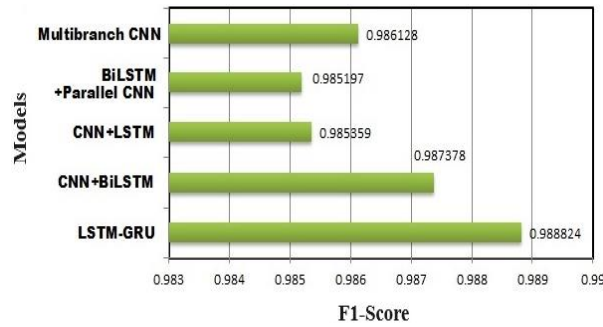


Figure 5. Comparison of F1 score

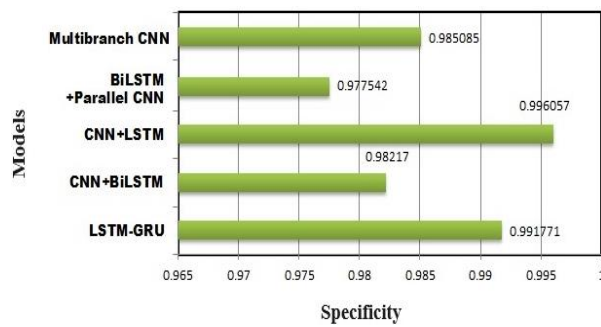


Figure 6. Comparison of specificity

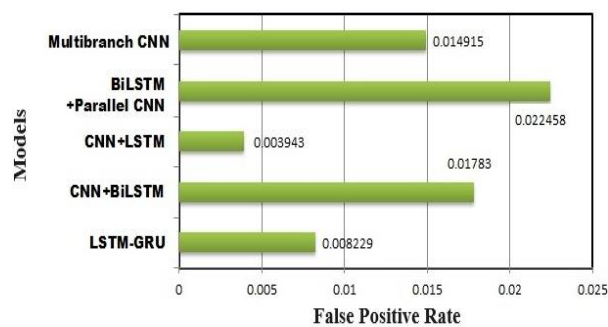


Figure 7. Comparison of false positive rate

Table 2. Comparison of training time, testing time and storage space

Model	Training time (h:mm:ss)	Testing time (seconds)	Storage space
LSTM-GRU	0:03:44	2.855061	960.2 KB
CNN+BiLSTM	0:02:08	2.987972	1.6 MB
CNN+LSTM	0:05:36	3.561903	11.3 MB
BiLSTM+Parallel CNN	0:04:35	3.728533	1.8 MB
Multibranch CNN	0:01:54	1.677524	3.0 MB




4. CONCLUSION

In this work, the issue of phishing attacks was addressed and the LSTM-GRU model was proposed which combines the strengths of two powerful deep neural networks in data science to detect and classify URLs. A novel balanced URL dataset was constructed and the model was evaluated and compared with powerful deep learning models from other research papers for the task of detecting phishing URLs. The proposed model scored the maximum accuracy of 98.89% in detecting the phishing URLs. It detects URLs faster than three of the four models it was compared with, and takes the least amount of space when saved. Analysis results show the adequacy of the model for this task. Although the proposed model achieved promising results, it suffers from low recall and high FNR. This is a limitation interpreted from the results of this study. This may be due to the complexity of the underlying patterns in the data, which may require a more sophisticated modelling approach. Further investigation is needed to explore strategies to enhance how well the model is able to capture subtle patterns and reduce false negatives. Additionally, the model was trained and tested on a single dataset, and its generalisability to other datasets remains to be evaluated. Future work should focus on testing the ability of model to perform on diverse datasets and can be implemented as an application or a web browser extension for real-time detection of phishing websites to protect users from phishing attacks and other such cyber threats.




REFERENCES

- [1] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Applied Sciences*, vol. 13, no. 8, Apr. 2023, doi: 10.3390/app13084649.
- [2] P. Zhang *et al.*, "CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2021, pp. 1109–1124, doi: 10.1109/SP40001.2021.00021.
- [3] A. S. S. V. L. Pooja and M. Sridhar, "Analysis of Phishing Website Detection Using CNN and Bidirectional LSTM," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, Nov. 2020, pp. 1620–1629, doi: 10.1109/ICECA49313.2020.9297395.
- [4] V. M. Yazmoozhi, B. Janet, and S. Reddy, "Anti-phishing System using LSTM and CNN," in *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, IEEE, Nov. 2020, pp. 1–5, doi: 10.1109/INOCON50539.2020.9298298.
- [5] S. Singh, M. P. Singh, and R. Pandey, "Phishing Detection from URLs Using Deep Learning Approach," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, IEEE, Oct. 2020, pp. 1–4, doi: 10.1109/ICCCS49678.2020.9277459.
- [6] 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, Oct. 2019, doi: 10.1155/2019/2595794.
- [7] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Information Security*, vol. 13, no. 6, pp. 659–669, Nov. 2019, doi: 10.1049/iet-ifs.2019.0006.
- [8] 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, Dec. 2021, doi: 10.3390/s21248281.
- [9] F. Zheng, Q. Yan, V. C. M. Leung, F. Richard Yu, and Z. Ming, "HDP-CNN: Highway deep pyramid convolution neural network combining word-level and character-level representations for phishing website detection," *Computers & Security*, vol. 114, Mar. 2022, doi: 10.1016/j.cose.2021.102584.
- [10] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," *IEEE Access*, vol. 10, pp. 1509–1521, 2022, doi: 10.1109/ACCESS.2021.3137636.
- [11] N. Q. Do, A. Selamat, O. Krejcar, E. H. -Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.
- [12] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, "PDGAN: Phishing Detection With Generative Adversarial Networks," *IEEE Access*, vol. 10, pp. 42459–42468, 2022, doi: 10.1109/ACCESS.2022.3168235.
- [13] S.-C. Lin, P.-C. Wl, H.-Y. Chen, T. Morikawa, T. Takahashi, and T.-N. Lin, "SenseInput: An Image-Based Sensitive Input Detection Scheme for Phishing Website Detection," in *JCC 2022 - IEEE International Conference on Communications*, IEEE, May 2022, pp. 4180–4186, doi: 10.1109/ICC45855.2022.9838653.
- [14] P. Yang, G. Zhao, and P. Zeng, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.
- [15] S. Das Gupta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," *Annals of Data Science*, Mar. 2022, doi: 10.1007/s40745-022-00379-8.
- [16] A. Abuadba, S. Wang, and M. Almashor, "Towards Web Phishing Detection Limitations and Mitigation," *arXiv preprint arXiv:220400985*, 2022, [Online]. Available: <https://arxiv.org/pdf/2204.00985.pdf>.
- [17] A. Aljofey *et al.*, "An effective detection approach for phishing websites using URL and HTML features," *Scientific Reports*, vol. 12, no. 1, May 2022, doi: 10.1038/s41598-022-10841-5.
- [18] A. Almomani *et al.*, "Phishing Website Detection With Semantic Features Based on Machine Learning Classifiers," *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–24, Feb. 2022, doi: 10.4018/IJSWIS.297032.
- [19] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsouid, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Computing*, vol. 25, no. 6, pp. 3819–3828, Dec. 2022, doi: 10.1007/s10586-022-03604-4.
- [20] L. R. Kalabarige, R. S. Rao, A. Abraham, and L. A. Gabralla, "Multilayer Stacked Ensemble Learning Model to Detect Phishing Websites," *IEEE Access*, vol. 10, pp. 79543–79552, 2022, doi: 10.1109/ACCESS.2022.3194672.
- [21] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.
- [22] "PhishTank." <https://phishtank.org> (accessed Apr. 15, 2023).
- [23] "Moz Top 500 Most Popular Websites." <https://moz.com/top500> (accessed Jan. 05, 2023).
- [24] "Keras: The Python Deep Learning API." <https://keras.io/> (accessed Jan. 05, 2023).
- [25] B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3629–3654, Dec. 2017, doi: 10.1007/s00521-016-2275-y.




BIOGRAPHIES OF AUTHORS

N. Subhashini    is an associate professor in the School of Electronics Engineering, Vellore Institute of Technology, Chennai, India. She has over 18 years of teaching and research experience. She received her B.E degree in Electronics and Communication Engineering from the University of Madras. Master's degree in Systems Engineering and Operations Research from College of Engineering, Guindy. She was awarded a gold medal for securing the first rank in her post graduation and was also awarded a gold medal for being the best outgoing student. she has co-authored books and her research papers have been published in reputed peer-reviewed journals and presented in conferences. She guides a number of UG and PG students in their projects. Her research interests include machine learning, deep learning techniques, network and information security, optical metro/access networks, next generation architectures and services and WDM Systems. She has also organized several international conferences, workshops, and seminars in the field of communication engineering, networks and information security. She can be contacted at email: subhashini.n@vit.ac.in.






Amogh Banerjee    is a final year student pursuing his B. Tech in Electronics and Communication Engineering in Vellore Institute of Technology, Chennai. He has a strong academic track record and has consistently excelled in his studies. He possesses a keen interest in problem-solving and AI. He holds a prestigious Udacity nano degree in AI programming with Python and is also certified in Microsoft Azure fundamentals. A passionate and driven individual, he aspires to build a remarkable career and make a significant impact in his chosen field of study. He can be contacted at email: amogh.banerjee2019@vitstudent.ac.in.






Abhi Kumar    is a passionate final year student pursuing B. Tech. in Electronics and Communication Engineering in Vellore Institute of Technology, Chennai. Committed to academic excellence and personal growth, he actively engages in co-curricular and extracurricular activities and demonstrates strong leadership skills. With a keen interest in full stack web development, he aims to make a positive impact in his chosen field. He can be contacted at email: abhi.kumar2019@vitstudent.ac.in.



S. Muthulakshmi    received her graduate degree in Instrumentation and Control Engineering from Regional Engineering College, Trichy (present National Institute of Technology, Trichy) in 1997, M. Tech in Embedded Systems Technology in 2007 from SRM University, Kattankulathur and Ph.D. from VIT University in 2019. She has more than 10 years of teaching experience. Her research interests focus on embedded system design with machine learning/deep learning techniques, embedded artificial intelligence, ML/DL techniques for medical imaging applications, memristor applications, approximate computing memristor and its applications, architecture development using memristors for embedded systems applications. She is presently working as associate professor, School of Electronics Engineering, Vellore Institute of Technology, Chennai. She can be contacted at email: muthulakshmi.s@vit.ac.in.



S. Revathi    is currently an associate professor with the School of Electronic Engineering, VIT at Chennai, Chennai. She has over 23 years of teaching and research experience. Her expertise is in CMOS, microelectronic manufactures, MEMS, microfluidic devices, cybersecurity. Her Ph.D. research work is based on the design and development of a composite layer micropump for drug delivery. She can be contacted at email: revathi.s@vit.ac.in.