Phishing website detection using novel integration of BERT and XLNet with deep learning sequential models

Kongara Srinivasa Rao[1], Dinesh Valluru[2], Satishkumar Patnala[3], Ravi Babu Devareddi[4], Tummalapalli Siva Rama Krishna[5], Andavarapu Sravani[6]
[1]Department of CSE, Faculty of Science and Technology, ICFAI Foundation for Higher Education (IFHE), Hyderabad, India
[2]Department of Information Technology, MLRITM Engineering College, Hyderabad, India
[3]Department of CSE/IT, GMR Institute of Technology, Rajam, India
[4]Department of CSE, GITAM School of Technology, GITAM University, Hyderabad, India
[5]Department of CSE, Jawaharlal Nehru Technological University Kakinada, Kakinada, India
[6]Department of CSE, GITAM School of Technology, GITAM Deemed to be University, Visakhapatnam, India

Article Info

Article history:

Received Mar 13, 2024
Revised Aug 5, 2024
Accepted Aug 11, 2024

Keywords:

BERT
GRU
LSTM
Phishing website
RNN
XLNet

ABSTRACT

Phishing websites pose a significant threat to online security, necessitating robust detection mechanisms to safeguard users' sensitive information. This study explores the efficacy of various deep learning architectures for phishing website detection. Initially, traditional sequential models, including recurrent neural networks (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU), achieve accuracies of 95%, 96%, and 96.5%, respectively, on a curated dataset. Building upon these results, hybrid architectures that combine the strengths of traditional sequential models with state-of-the-art language representation models, bidirectional encoder representations from transformers (BERT) and XLNet, are investigated. Combinations such as RNN with BERT, BERT with LSTM, BERT with GRU, RNN with XLNet, XLNet with LSTM, and XLNet with GRU are evaluated. Through experimentation, accuracies of 94.5%, 96.5%, 96.1%, 95.7%, 97.4%, and 97%, respectively, are achieved, demonstrating the effectiveness of hybrid deep learning architectures in enhancing phishing detection performance. These findings contribute to advancing the state-of-the-art in cybersecurity practices and underscore the importance of leveraging diverse model types to combat online threats effectively.

Corresponding Author:

Dinesh Valluru
Department of Information Technology, MLRITM Engineering College
Hyderabad, India
Email: dinesh.valluru15@mlritm.ac.in

1. INTRODUCTION

The proliferation of phishing websites poses a significant threat to cybersecurity, endangering the privacy and financial security of internet users worldwide. Phishing attacks employ deceptive techniques to trick users into divulging sensitive information such as passwords, credit card numbers, or personal identification details. Traditional methods of detecting phishing websites often rely on heuristics, rules, or manually curated blacklists, which may struggle to keep pace with the evolving sophistication of phishing tactics. In recent years, deep learning (DL) has emerged as a promising approach for enhancing phishing detection by leveraging the power of neural networks to automatically learn patterns and features from large-scale data. Traditional sequential models such as recurrent neural networks (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) have shown considerable success in capturing sequential dependencies within text data, making them suitable candidates for phishing detection tasks.

However, the advent of transformer-based language representation models, exemplified by bidirectional encoder representations from transformers bidirectional encoder representations (BERT) from Transformers and XLNet, has revolutionized the field of natural language processing (NLP) by offering unparalleled capabilities in capturing contextual relationships and semantic understanding in text data. These models have demonstrated remarkable performance across various NLP tasks, prompting interest in their potential application to cybersecurity domains such as phishing detection.In this paper, we proposed a novel integration of BERT and XLNet with traditional sequential models for phishing detection.

Desai *et al.* [1] proposed a phishing detection method that focuses on achieving maximum accuracy while having a small feature collection. A variety of performance criteria were used to assess the effectiveness of the machine learning (ML) models that were used. Our dataset included a collection of authentic URLs from the Alexa dataset as well as phishing URLs taken from the Phish Tank dataset. He *et al.* [2] extracted semantic and long-range dependencies from website URL strings using Tiny-BERT. These characteristics were employed in a Stacking algorithm-based classifier with four basic learners: categorical boosting (CatBoost), extreme gradient boosting (XGBoost), light gradient boosting machine (LightGBM), and gradient boosted decision trees (GBDT). Phishing websites were detected without human feature extraction. Through the stacking ensemble, basic learners helped each other reduce categorization mistakes and improve generalization, improving accuracy. Using a deep reinforcement learning (DRL)-based classifier and cutting-edge feature selection, [3] accelerated training while improved classification performance. Data from Mendeley was used to test five statistical and correlation-based feature-ranking techniques. Using the Gini index, the selection strategy decreased the number of columns in the dataset by 27%, saved more than 10% of training time, and improved classification scores. Pillai *et al.* [4] explored unusual website classifier evasion attacks and detection. To address these deficiencies, URL data extraction and ML website classification were recommended. To maintain phishing website functionality and appearance, adversarial samples targeted classification features.

A multimodal representation technique combining textual and image-based elements to identify fraudulent websites [5]. Features were extracted by two convolutional neural network (CNN) models and integrated for decision-making. This model enhanced Matthews correlation coefficient (MCC) performance by 4% and decreased false positives by 1.6%. An architecture by [6] used special characters to separate URLs into four parts: protocol type, domain, sub-domain, and top-level domain (TLD). Features were utilized to create a vocabulary database. A modified FastText word embedding method produced numeric feature vectors. These vectors and pre-processed URL examples taught a BiLSTM classifier. Phishing URLs were detected with excellent accuracy and little processing cost. Zonta and Sathiyanarayanan [7] examined evolving threats and risk-reduction strategies using malicious detection to defend web browsers. It stressed the necessity of proactive detection in building cybersecurity frameworks that can anticipate and defeat future cyber attackers' methods as well as react to existing attacks. Nowroozi *et al.* [8] used ML classifiers for detection. Our false negative rate was 0.0037. K-means was used to cluster data and examine decision tree-based models' sensitivity to limited knowledge attacks, including zeroth order optimization. Menon and Anandhu [9] used ML-based URL detection. A new collection of URL attributes and behaviors plus a ML algorithm comprised the recommended detection approach. The testing results showed that the proposed URL attributes and behavior might improve harmful URL recognition. Stoleriu *et al.* [10] suggested a method that makes use of ML techniques and threat intelligence data to identify dangerous short URLs. The most efficient method was random forest (RF), which produced a high level of accuracy. The technology was put into practice and its efficacy was confirmed in practical applications.

Malicious URLs were found in [11] in a number of apps. The dataset, which included more than 6lakh URLs for the implementation, was divided into four categories: phishing, benign, defacement, and malware. Three ML models namely LightGBM, XGBoost, and randomized forest were used to identify and categorize dangerous URLs. Arora *et al.* [12] conducted a current evaluation of the important algorithms used to identify fraudulent URLs, with RF demonstrating the greatest accuracy. Additionally, a number of concerns with developing alternative algorithms for identifying dangerous URLs were covered. ML and DL techniques were used in [13] to identify dangerous URLs that might lead to cyberattacks. Many ML and DL algorithms were used, such as RF and Naïve Bayes for ML and RNN and LSTM for DL. Wang *et al.* [14] introduced PhishBERT, a pretrained deep transformer network for phishing URL detection. It was fine-tuned on benign and malicious URL data using supervised adversarial methods, outperforming current state-of-the-art methods in efficiency, robustness, and accuracy. Multiple binary classifiers were trained to differentiate one category from the remainder [15]. One vs all classification can accurately predict harmful URLs even with minimal datasets comprising simply domain and path. This technique shows potential in producing accurate findings with low computer resources, outperforming conventional approaches. Lexical URL categorization was studied in [16]. XGBoost, support vector machine (SVM), and artificial neural network (ANN) detected malicious URLs with 88%, 87%, and 88% accuracy. phishing URL imbalances affected detection. SVMs, RFs, DTs, KNNs, and Bayesian optimization classified URLs in [17]. DRLSH, BPLSH, and random instance selection were used for efficiency. SVMs fared well with extended training durations, while RFs had great accuracy, recall, and F1 scores. In URL categorization, instance selection greatly affected model performance.

CyberLen, a DL-based malicious URL detection system, was suggested in [18]. Position embedding decreased token ambiguity, whereas factorization machine (FM) learnt latent lexical feature interactions. TCN detected URL token long-distance dependencies. CapsNet and IndRNN combined to collect multi-modal data and merge texture and text [19]. The network filtered deep characteristics using an attention strategy to improve harmful URL categorization accuracy, according to trials. Lakshmanarao *et al.* [20] applied ML techniques to identify phishing assaults. The proposal included two priority-based algorithms. The final fusion classifier was chosen from these methods. An innovative fusion classifier was used on a UCI dataset and yielded excellent results. Nanaware [21] used ML to identify phishing URLs and text processing to assess text for phishing attack indications, addressing a major security issue from counterfeit websites and URLs. Chen *et al.* [22] suggested a phishing URL detection approach based on URL content, using basic features for preliminary screening and CNN for malicious URL identification using page content, with reasonable accuracy. The hybrid methodology [23] used classical, ML, and DL to identify dangerous URLs. To identify new and existing dangerous URLs, it used shallow and DL coupled with signatures. Nadkarni and Borkar [24] used RF with hash vectorization to identify fraudulent URLs with 97.5% accuracy. They created a Flask-based real-time URL categorization web app.

2. METHOD

The proposed method is shown in Figure 1. In this work, a hybrid model was developed for lung cancer detection, employing a combination of conventional ML algorithms and autoencoders for feature extraction. The dataset consists of CT scan images obtained from Kaggle, categorized into three classes: benign, malignant, and normal, representing various lung cancer stages.

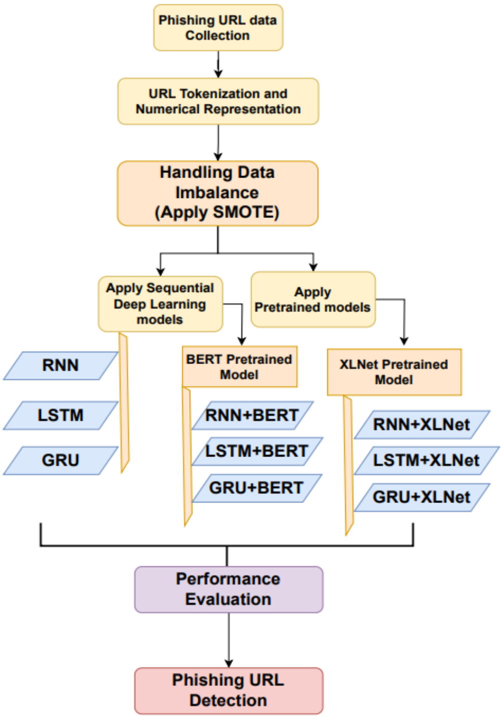


Figure 1. Proposed model for phishing website detection

In this paper, a comprehensive methodology was proposed for phishing website detection, leveraging a combination of traditional sequential models and state-of-the-art language representation models such as BERT and XLNet. The methodology comprised several interconnected steps aimed at systematically investigating the efficacy of different DL architectures and integration strategies in enhancing phishing detection performance. A dataset from Kaggle was used for experiments. Firstly, the challenge of imbalanced data was addressed by employing techniques such as synthetic minority over-sampling technique (SMOTE) to ensure a balanced representation of phishing and legitimate URLs in the dataset. This step was crucial for mitigating biases and improving the robustness of the models. Subsequently, the effectiveness of traditional sequential models, including RNN, LSTM, and GRU, in capturing sequential dependencies within URL data was explored. These models were trained on the preprocessed and balanced dataset, and their performance was evaluated using standard metrics to establish baseline performance.

Next, the integration of BERT and XLNet, two transformer-based language representation models renowned for their ability to capture contextual relationships and semantic understanding in text data, was investigated. Pre-trained BERT and XLNet models were fine-tuned on a large corpus of text data, and various integration approaches with the traditional sequential models were explored, such as combining output representations and embeddings. This hybridization aimed to leverage both the contextual understanding of language models and the sequential dependencies captured by traditional models.

Following model integration, the hybrid models were trained and evaluated on a split dataset, comprising training, validation, and test sets. Each model underwent training using appropriate optimization algorithms and hyperparameters, with training progress monitored using validation data to prevent overfitting. The trained models were then evaluated on the test set using standard evaluation metrics, and statistical significance tests were conducted to compare their performance against baseline models and standalone implementations of BERT and XLNet. Finally, the results were analyzed and interpreted to identify the most effective models and integration strategies for phishing detection. Insights into the learned representations and decision-making processes of the hybrid models were gained. Any limitations or challenges encountered during the experimentation process were discussed, and avenues for future research were proposed to address them.

2.1. Phishing URL collection

The dataset [25] contains 549,346 samples with two columns. One column serves as a two-category prediction label, distinguishing between "Good" sites, which are not phishing and contain no harmful URLs, and "Bad" sites, which are phishing sites with harmful URLs. Out of the total, 156,420 URLs were classified as bad (phishing), while 392,920 were categorized as good.

2.2. Class imbalance with SMOTE

Class imbalance refers to the scenario where one class (often the minority class) is significantly underrepresented compared to the other class(es) in a classification problem. In the context of phishing website detection, class imbalance occurs when the number of phishing URLs is much lower than the number of legitimate URLs in the dataset. Addressing class imbalance is crucial because ML models trained on imbalanced datasets tend to exhibit biases towards the majority class, leading to poor performance in detecting instances of the minority class. One common technique used to alleviate class imbalance is SMOTE.

2.3. Deep learning sequential algorithms

In this study, we leverage three foundational DL algorithms namely RNN, LSTM, and GRU for the task of phishing website detection. RNNs capture temporal dependencies in sequential data like URLs to analyze it. We used RNNs as a baseline model for phishing detection performance measurements. LSTMs are adept at modeling sequences with long-range dependencies, making them well-suited for the nuanced patterns often found in phishing URLs. We also examined GRU networks, a simpler LSTM, for phishing website identification. GRUs can capture temporal relationships while being computationally efficient, offering an alternative to LSTM-based models.

2.4. Pretrained models for phishing website detection

In this paper, we delved into the utilization of pretrained language representation models, specifically BERT and XLNet, for the task of phishing website detection. A transformer-based paradigm called BERT captured bidirectional contextual linkages in text data, revolutionizing NLP. Pretrained on massive quantities of text input, BERT encoded left and right contexts to build complex text sequence representations. On a dataset of phishing and authentic URLs, we fine-tuned pretrained BERT models for phishing detection using contextual and semantic knowledge. XLNet, another transformer-based language representation architecture, enhanced BERT. XLNet used a permutation-based training aim to capture bidirectional context and use autoregressive model benefits.

3. RESULTS AND DISCUSSION

3.1. Applying SMOTE with phishing website dataset

To rectify the class imbalance within the dataset, we employed the SMOTE. This process involved generating synthetic instances for the minority class, phishing URLs, to equalize the representation of both classes in the dataset. After the application of SMOTE, the number of bad URL samples are raised to more than 3,00,000. In the final dataset we considered 3,00000 samples bad URLs and 3,00000 samples of good URLs. This rebalancing of the dataset enables our ML models to learn from a more representative set of examples, enhancing their ability to accurately detect phishing URLs.

3.2. Applying hashing vectorizer

In this preprocessing step, we applied the hashing vectorizer technique to convert the textual features, such as URLs or webpage content, into numerical representations. Hashing vectorizer was employed to transform the variable-length text data into fixed-size numerical vectors, facilitating the subsequent application of DL models. By utilizing this method, we aimed to prepare the input data for the upcoming DL models, namely RNN, LSTM, and GRU.

3.3. Applying traditional deep learning sequential algorithms

We applied RNN, LSTM and GRU to the balanced dataset for phishing website detection. For the RNN model, we employed a simple architecture consisting of an embedding layer, followed by a single RNN layer and a dense output layer with sigmoid activation. Training was conducted on the balanced dataset, with phishing and legitimate URLs equally represented. The model utilized Adam optimizer and binary cross-entropy loss during training. Upon evaluation on a separate test set, the RNN model achieved an accuracy of 94.5%. Next, we utilized a standard LSTM architecture comprising an embedding layer, a single LSTM layer, and a dense output layer with sigmoid activation. Similar to the RNN model, training was performed on the balanced dataset using Adam optimizer and binary cross-entropy loss. The LSTM model exhibited improved performance, achieving an accuracy of 96.5% upon evaluation on the test set. Similarly, for the GRU model, we employed a comparable architecture to the LSTM model, including an embedding layer, a single GRU layer, and a dense output layer with sigmoid activation. Training of the GRU model was conducted on the balanced dataset using Adam optimizer and binary cross-entropy loss. Upon evaluation, the GRU model achieved an accuracy of 96.1% on the test set. Table 1 and Figure 2 shows F1 score and accuracy of DL sequential models.

Table 1. Result with traditional sequential modles

Model	Accuracy	F1-score
RNN	94.5%	93%
LSTM	96.5%	95%
GRU	96.1%	94.8%

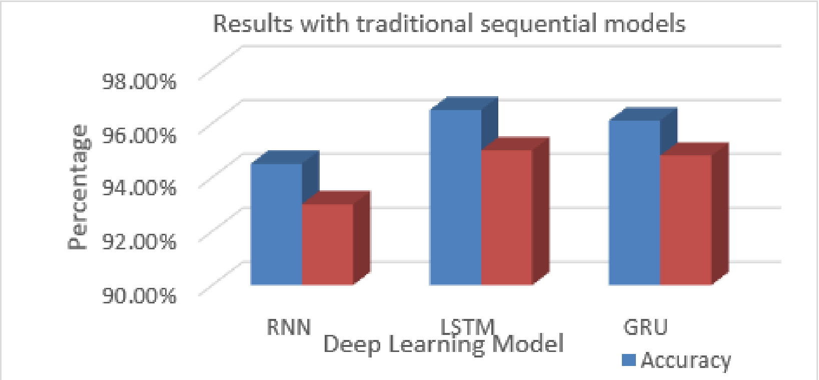


Figure 2. Performance of DL sequential models

3.4. Initializing the BERT model

Firstly, a tokenizer is initialized using the BERT model's tokenizer. This tokenizer is crucial for converting textual features of URLs or webpage content into a format that the BERT model can comprehend.

This model, pre-trained on extensive text data, is adept at classifying sequences into two categories. In the phishing website detection context, it discerns whether URLs or webpage content are phishing attempts or legitimate.

3.5. Initializing the XLNet model

XLNet model initialization is similar to BERT, involving loading pre-trained weights and setting up the model architecture. Pre-trained weights for XLNet can be obtained from a publicly available pre-trained XLNet model. The architecture of XLNet also consists of multiple transformer layers with self-attention mechanisms, but with a permutation language modeling objective. During initialization, adjustments done made to fine-tune the model for the target task, such as modifying hyperparameters or adding task-specific layers for classification or regression.

3.6. Implementing hybrid architecture with BERT

In this step, we explored hybrid architectures that integrate the BERT model with traditional sequential models. By combining the contextual understanding of BERT with the ability of sequential models to capture temporal dependencies, we aimed to enhance the performance of phishing website detection. We investigated three hybrid architectures. Figure 3 shows the results of BERT with all DL models.

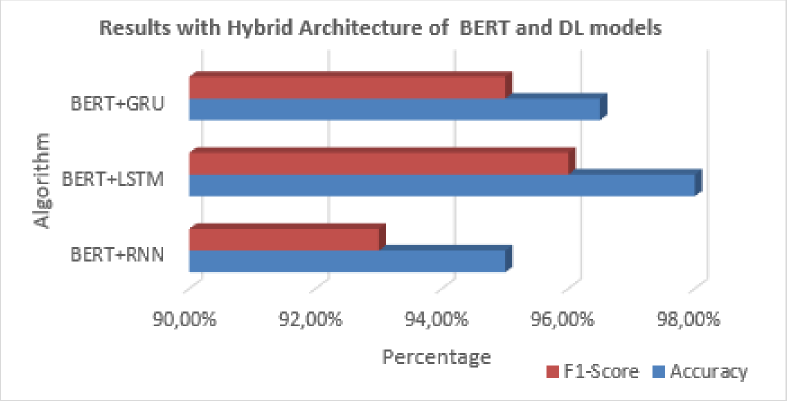


Figure 3. Performance of BERT+DL sequential models

3.6.1. Implementing hybrid BERT and RNN

In this phase, we combined the output representations from the BERT model with the hidden states of a RNN. This hybrid model leverages both BERT's contextual understanding and RNN's ability to capture sequential patterns in the data. The model BERT+RNN achieved an accuracy of 95% and an F1-score of 93%.

3.6.2. Implementing hybrid BERT and LSTM

Similar to BERT+RNN, we combined the output representations from the BERT model with the hidden states of a LSTM network. The LSTM network enhances the BERT model by capturing long-term dependencies in the data. The model BERT+LSTM attained an accuracy of 98% and an F1-score of 96%.

3.6.3. Implementing hybrid BERT and GRU

In this hybrid architecture, we combined the output representations from the BERT model with the hidden states of a GRU. GRU offers a simplified architecture compared to LSTM while still capturing sequential information effectively. The BERT+GRU model achieved an accuracy of 96.5% and an F1-score of 95%.

3.7. Implementing hybrid architecture with XLNet

In this step, we explored hybrid architectures that integrate the XLNet model with traditional sequential models. These architectures combine XLNet's permutation language modeling objective with the sequential modeling capabilities of RNN, LSTM, and GRU. Figure 4 shows the results of BERT with all DL models.

3.7.1. Implementing hybrid XLNet and RNN

In this step, we combined the output representations from the XLNet model with the hidden states of a RNN. This hybrid model capitalizes on XLNet's language modeling abilities and RNN's sequential modeling capabilities. The model, XLNet+RNN, achieved an accuracy of 95.6% and an F1-score of 95%.

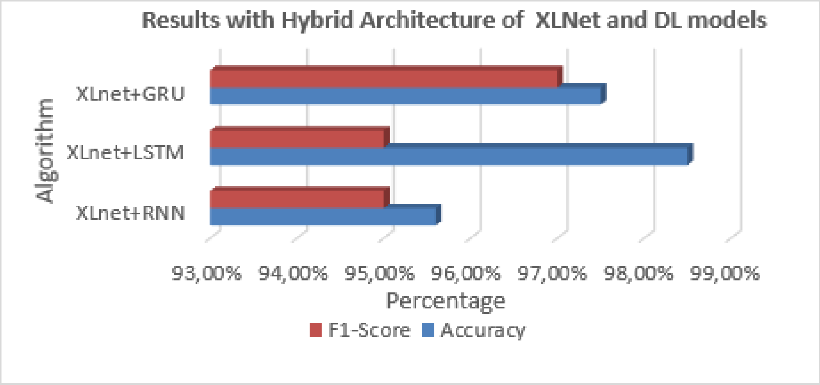


Figure 4. Performance of XLNet+DL sequential models

3.7.2. Implementing hybrid XLNet and LSTM

Similar to XLNet+RNN, we combinde the output representations from the XLNet model with the hidden states of a LSTM network. LSTM's ability to capture long-term dependencies complements XLNet's permutation language modeling objective. The XLNet+LSTM model demonstrated an accuracy of 98.5% and an F1-score of 95%.

3.7.3. Implementing hybrid XLNet and GRU

In this hybrid architecture, we combined the output representations from the XLNet model with the hidden states of a GRU network. GRU offers a simpler architecture compared to LSTM while still capturing sequential information effectively, enhancing the capabilities of XLNet for phishing website detection. XLNet+GRU model achieved an accuracy of 97.5% and an F1-score of 97%.

3.8. Comparison of traditional DL models with hybrid models

Figure 5 shows the accuracy and F1 score comparison of traditional DL models with hybrid BERT+DL and XLNet+DL models. From Figure 5, it is observed that the efficacy of various models in phishing website detection. Traditional sequential models, including RNN, LSTM, and GRU, demonstrated respectable performances, with LSTM leading the pack with a remarkable accuracy of 96.50% and an F1-score of 95%.

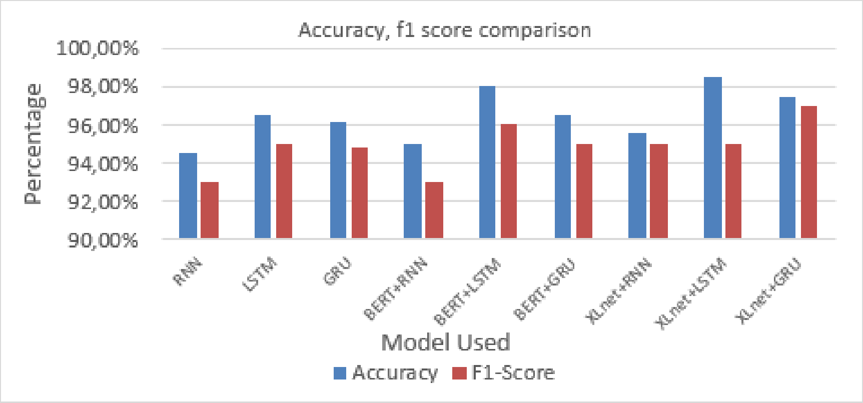


Figure 5. Accuracy, F1 score comparison of DL sequential models with hybrid models

However, the introduction of hybrid architectures, combining the power of pre-trained language models with the sequential modeling capabilities of traditional models, significantly elevated the detection accuracy. Among the hybrid models, BERT+LSTM emerged as the top performer, achieving an impressive accuracy of 98.00% and an F1-score of 96%. This underscores the advantage of integrating BERT's contextual understanding with the long-term dependencies captured by LSTM.

Similarly, XLNet+LSTM showcased remarkable results, surpassing other models with the highest accuracy of 98.50% and a commendable F1-score of 95%. The highest accuracy achieved among all the models is 98.5%, achieved by the XLNet+LSTM model. The highest F1-score achieved is 97%, obtained by the user model XLNet+GRU.

The propsed model performance compairon with existing works is shown in Table 2 and Figure 6. In comparison to existing models, the proposed method achieves a higher accuracy of 97.4%, surpassing other approaches such as ANN (88%), ML fusion (96%), and traditional ML (97%). This enhancement underscores the effectiveness of hybrid BERT and XLNet with traditional DL approches.

Table 2. Comparison with existing models

Model	Accuracy
ANN [16]	88%
ML fusion [20]	96%
Traditional ML [10]	97%
Proposed method	97.4%

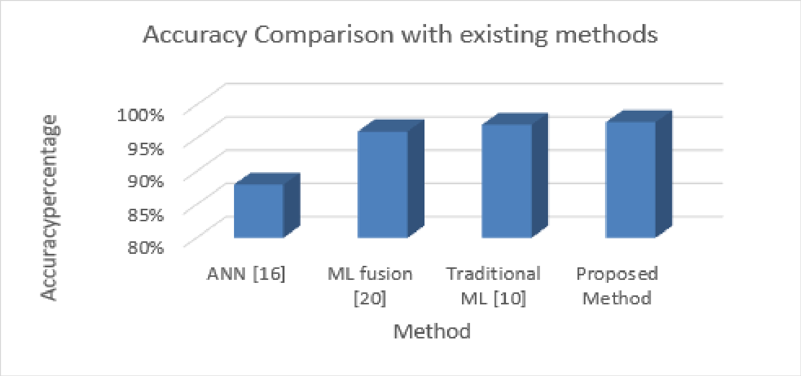
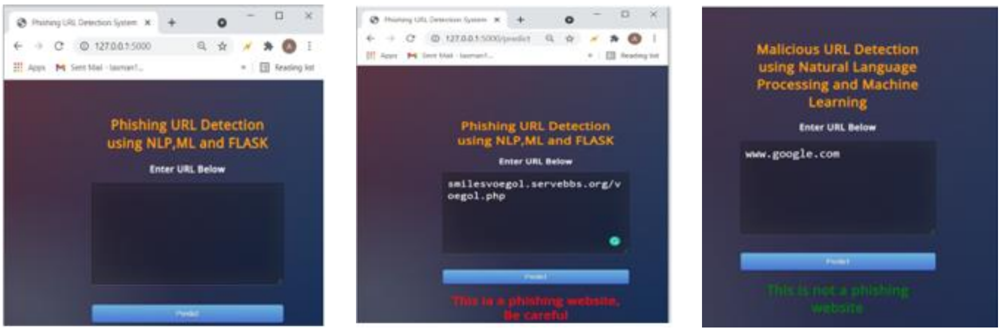


Figure 6. Accuracy comparison with existing methods

The trained model is also created as webapp for further usage of proposed mode in real time. Here, the user can enter the website address and can able check it is phishing site or not. Figure 7 shows screen shots of phishing website detection tool usage. In Figure 7(a) shows the tool where user can able to enter website address. In Figure 7(b), user already entered a phishing URL and it is shown as phishing URL. In Figure 7(c), user entered a good URL and it is shown as not a phishing website.


Phishing URL Detection System
127.0.0.1:5000
Apps
Sent Mail - lasman1...
Reading list
Phishing URL Detection using NLP,ML and FLASK
Enter URL Below
Predict
127.0.0.1:5000/predict
Malicious URL Detection using Natural Language Processing and Machine Learning

smilesvoegol.servebbs.org/voegol.php

This is a phishing website, Be careful

www.google.com

This is not a phishing website

(a) (b) (c)

Figure 7. Phishing website detection web app: (a) phishing website detection webapp; (b) user entered phishing site and webapp output it as phishing site; and (c) user entered good URL and webapp output it as not a phishing site

4. CONCLUSION

In conclusion, our study delved into the critical task of phishing website detection, aiming to detect malicious URLs. We initiated our exploration by scrutinizing the performance of traditional sequential models, including RNN, LSTM, and GRU, which yielded commendable accuracies ranging from 95% to 96.5%. Building upon these foundational results, we ventured into the realm of hybrid architectures, merging the robustness of traditional models with the contextual understanding offered by state-of-the-art language representation models, BERT and XLNet. The incorporation of these methodologies resulted in a significant enhancement in detection performance, with accuracies reaching up to 97.4%. Notably, the hybrid models, particularly those incorporating BERT and XLNet, showcased remarkable accuracies, underscoring their efficacy in phishing website detection tasks. The findings emphasize the paramount importance of embracing hybrid DL architectures to fortify cybersecurity practices effectively.

REFERENCES

[1] S. Desai, S. Salunkhe, R. Deshmukh, and S. Zalte, “Study and analysis of machine learning models for detection of phishing URLs,” *Machine Learning Applications*, pp. 85-95, 2023, doi: 10.1002/9781394173358.ch6.

[2] D. He, X. Lv, S. Zhu, S. Chan and K. -K. R. Choo, "A method for detecting phishing websites based on tiny-bert stacking," in *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2236-2243, 15 Jan.15, 2024, doi: 10.1109/JIOT.2023.3292171.

[3] A. Maci, N. Tamma and A. Coscia, "Deep reinforcement learning-based malicious URL detection with feature selection," *2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC)*, Houston, TX, USA, 2024, pp. 1-7, doi: 10.1109/ICAIC60265.2024.10433827.

[4] M. J. Pillai, S. Remya, V. Devika, S. Ramasubbareddy and Y. Cho, "Evasion attacks and defense mechanisms for machine learning-based web phishing classifiers," *IEEE Access*, vol. 12, pp. 19375-19387, 2024, doi: 10.1109/ACCESS.2023.3342840.

[5] M. Alsaedi, F. A. Ghaleb, F. Saeed, J. Ahmad and M. Alasli, "Multi-modal features representation-based convolutional neural network model for malicious website detection," *IEEE Access*, vol. 12, pp. 7271-7284, 2024, doi: 10.1109/ACCESS.2023.3348071

[6] K. Mangalam and B. Subba, "PhishDetect: a BiLSTM based phishing URL detection framework using FastText embeddings," *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bengaluru, India, 2024, pp. 637-641, doi: 10.1109/COMSNETS59351.2024.10427067.

[7] R. A. K, T. Zonta and M. Sathiyanarayanan, "A holistic review on detection of malicious browser extensions and links using deep learning," *2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC),* Houston, TX, USA, 2024, pp. 1-6, doi: 10.1109/ICAIC60265.2024.10433842.

[8] E. Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An adversarial attack analysis on malicious advertisement URL detection framework," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332-1344, 2023, doi: 10.1109/TNSM.2022.3225217.

[9] R. R. K. Menon and V. Anandhu, "Machine learning supported malicious URL detection," *2023 4th IEEE Global Conference for Advancement in Technology (GCAT),* Bangalore, India, 2023, pp. 1-5, doi: 10.1109/GCAT59970.2023.10353402.

[10] R. Stoleriu, C. Negru, B. -C. Mocanu and F. Pop, "Malicious short URLs detection technique," *2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Craiova, Romania, 2023, pp. 1-6, doi: 10.1109/RoEduNet60162.2023.10274913.

[11] U. S. D. R, A. Patil and Mohana, "Malicious URL detection and classification analysis using machine learning models," *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023, pp. 470-476, doi: 10.1109/IDCIoT56793.2023.10053422.

[12] R. Arora, R. Gupta and P. Yadav, "Mischievous URL prediction through supervised machine learning algorithms," *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)*, Bangalore, India, 2023, pp. 1-5, doi: 10.1109/InC457730.2023.10262881.

[13] P. Malaviya, S. Bhadja, V. Gajjar and Y. Kumar, "Cyber security and data mining: detecting malicious URL using data mining techniques," *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, Greater Noida, India, 2023, pp. 527-534, doi: 10.1109/ICCSAI59793.2023.10421348.

[14] Y. Wang, W. Zhu, H. Xu, Z. Qin, K. Ren and W. Ma, "A large-scale pretrained deep model for phishing URL detection," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095719.

[15] K. Saranya, K. Varunvikash, K. Karthikeyan and S. Rosanakthar, "Detecting malicious URLs using data analytics," 2023 *International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICSCSS57650.2023.10169447.

[16] C. G. Vung and Y. Y. Win, "URL classification based on lexical features by machine learning," *2023 IEEE Conference on Computer Applications (ICCA),* Yangon, Myanmar, 2023, pp. 345-350, doi: 10.1109/ICCA51723.2023.10181514.

[17] S. Abad, H. Gholamy, and M. Aslani, “Classification of malicious URLs using machine learning,” *Sensors*, vol. 23, no. 18. MDPI AG, p. 7760, Sep. 08, 2023. doi: 10.3390/s23187760.

[18] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu and D. Zeng, "Robust detection of malicious URLs with self-paced wide & deep learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 717-730, 2022, doi: 10.1109/TDSC.2021.3121388.

[19] J. Yuan, G. Chen, S. Tian and X. Pei, "Malicious URL detection based on a parallel neural joint model," *IEEE Access*, vol. 9, pp. 9464-9472, 2021, doi: 10.1109/ACCESS.2021.3049625.

[20] A. Lakshmanarao, P. S. P. Rao and M. M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021, pp. 1164-1169, doi: 10.1109/ICAIS50930.2021.9395810.

[21] T. P. Nanaware, “Malicious URL detection using machine learning,” *International Journal of Integrated Science and Technology*, vol. 2, no. 1. pp. 29–36, 2024. doi: 10.59890/ijist.v2i1.1289.

[22] Y. Chen, Y. Zhou, Q. Dong and Q. Li, "A malicious URL detection method based on CNN," *2020 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, Shenyang, China, 2020, pp. 23-28, doi: 10.1109/TOCS50858.2020.9339761.
[23] B. Gogoi, T. Ahmed and A. Dutta, "A Hybrid approach combining blocklists, machine learning and deep learning for detection of malicious URLs," *2022 IEEE India Council International Subsections Conference (INDISCON)*, Bhubaneswar, India, 2022, pp. 1-6, doi: 10.1109/INDISCON54605.2022.9862909.
[24] N. S. Nadkarni and S. Borkar, "Detection of lung cancer in CT images using image processing," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 863-866, doi: 10.1109/ICOEI.2019.8862577.
[25] T. Tiwari, “Phishing-site-URLs dataset.” [Online]. Available: https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls (Accessed: December 12, 2023)

BIOGRAPHIES OF AUTHORS

Dr. Kongara Srinivasa Rao is having 15+ years of teaching experience, he received his Ph.D. in CSE from JNTUK Kakinada, 2021, M. Tech in CSE from SRM University, Chennai, 2006 and B. Tech in CSE from JNTU Hyderabad, 2004. He had published papers in reputed national and international journals. He had attended many workshops, conferences and presented various research papers at national and international conferences. His teaching areas are unix shell scripting, unix programming, network programming, computer networks, data base management systems, operating systems, data ware housing and data mining, data engineering, computer programming. His research interests are information retrieval systems, database management systems, natural language processing, machine learning, data mining, and computer networks. He can be contacted at email: drksrao@ifheindia.org.

Dr. Dinesh Valluru is an Associate Professor, at the Department of Information Technology at MLRITM Engineering College, Telangana, India. He received his Ph.D. from Anna University, Chennai in Computer vision. He has more than 12 years of teaching experience. He has published papers in reputed national and international journals. He had attended many workshops, and conferences, and presented various research papers at national and international conferences. His areas of interest include artificial intelligence, computer vision, and IoT. He can be contacted at email: dinesh.valluru15@mlritm.ac.in.

Satishkumar Patnala received his Ph.D. from Andhra University in the area of MANETs. He has more than 10 years of teaching experience. He had published papers in reputed national and international journals. He had attended many workshops, conferences and presented various research papers at national and international conferences. His areas of interest include computer networks, artificial intelligence, machine learning. He can be contacted at email: srtsatishsrt@gmail.com.

Dr. Ravi Babu Devareddi is having 19+ years teaching experience and presently working as Assistant Professor, Department of Computer science and Engineering in GITAM School of Technology, GITAM University, Hyderabad, 502329, India. He obtained his Ph.D. (Computer Science and Engineering) from Acharya Nagarjuna University, Guntur, received his obtained M.Tech (Computer Science and Technology) Degree from Department of Computer Science and Engineering, Andhra University in 2009, B.Tech (Computer Science and Engineering) Degree in 2005 from Andhra University. His current research interests are in internet of things, image processing, computer vision, and artificial intelligence. He can be contacted at email: ravibabu.devareddi@gmail.com.

Dr. Siva Rama Krishna did his Ph.D. in Cloud Forensics, and currently working as Assistant Professor in the Department of Computer Science and Engineering at Jawaharlal Nehru Technological University Kakinada, India. He authored 3 books and obtained 4 patents. He published 23 research papers in reputed journals. He received three awards for his research works and four awards for his contributions in teaching. His areas of research include cyber security and digital forensics, artificial intelligence, and blockchain. He obtained 10 professional certifications. He is a fellow of India School on Internet Governance and a diploma holder in Internet Governance. He can be contacted at email: srktummalapalli@gmail.com.

Andavarapu Sravani is Assistant Professor, Department of CSE in GITAM School of Technology, Andhra Pradesh, India. She is Pursuing her Ph.D. from Andhra University in Machine Learning. She has more than 10 years of teaching experience. She had published papers in reputed national and international journals. She had attended many workshops, conferences and presented various research papers at international conferences. Her areas of interest include datamining, machine learning, and IoT. She can be contacted at email: sravani61sravz@gmail.com.