International Journal of Modern Developments in Engineering and Science
Volume 3, Issue 11, November 2024
https://www.ijmdes.com | ISSN (Online): 2583-3138

Malicious URL Detection Using Machine Learning and Deep Learning Hybrid Models

S. P. Siddique Ibrahim[1], Shreshth Pandey[2*], Yash Raj Singh[3]
[1,2,3]*School of Computer Science and Engineering, Vellore Institute of Technology, Andhra Pradesh, India*

Abstract*: The proliferation of bad URLs has posed a serious challenge to cybersecurity, as conventional detection techniques are unable to keep up with the swift changes in online threats. The goal of this study is to investigate how hybrid models that integrate machine learning (ML) and deep learning (DL) might improve the reliability and accuracy of URL detection systems. To enhance the detection of malicious URLs, we suggest three hybrid architectures: CNN+LSTM, CNN+RNN, and LightGBM+BERT. These designs take advantage of the advantages of both paradigms. The goal of the project is to improve cybersecurity frameworks detection capabilities while also looking into new defensive strategies against dynamic cyberthreats. In order to analyse important URL characteristics, such as anomalous URL structures, Google index status, short URL detection, suspicious patterns, and length-based data like hostname length, first directory length, and top-level domain length, our solution combines feature extraction approaches. The suggested models are better able to identify if a URL is malicious or benign by dissecting these characteristics. The experimental results show that our methods beat conventional ML and DL approaches in terms of accuracy and efficiency. The hybrid models are trained and assessed using a dataset of real-world URLs. Combining long short-term memory networks (LSTMs) for sequence learning and convolutional neural networks (CNNs) for feature extraction works especially well for capturing temporal and spatial patterns within URLs. According to our research, these hybrid models significantly improve internet users' overall security by facilitating the early detection of cyberthreats. Future advancements in URL detection are made possible by this research, which also creates new opportunities for incorporating cutting-edge technologies like blockchain for safe URL verification.

***Keywords*: Cybersecurity, Malicious URLs, Hybrid Models, Deep Learning, Machine Learning, CNN, LSTM, RNN, Light-GBM, URL Detection.**

1. Introduction

Cybercrime and online threats are growing exponentially, which is one of the most worrying issues facing modern society. Cybercriminals have developed cunning and nasty methods to take advantage of weaknesses in web-based systems due to the quick growth of digital ecosystems. The most common technique is the use of malicious URLs, which act as gateways for a variety of illegal operations, including virus distribution, phishing, and website vandalism. Worldwide, people, companies, and governments have suffered significant financial and reputational losses as a result of these actions. The estimated annual cost of cybercrime worldwide is expected to exceed $10.5 trillion in 2023 alone, and bad URLs are a major factor in making these attacks possible.

As per the Verizon Data Breach Investigations Report (DBIR) 2022, phishing continues to be a prevalent method of cyber- attack, responsible for more than 36% of all data breaches worldwide. Malicious URLs are frequently cloaked in phishing emails or webpages, tricking people into disclosing private or financial information. Malicious URLs are also used in another technique called drive-by-downloads, which causes users to automatically download malware whenever they visit hacked websites. Traditional rule-based detection techniques are becoming less and less successful due to the stealthy and complex nature of these attacks. Thus, in order to guarantee the security and safety of online contacts, the development of sophisticated detection techniques has become essential.

A. The Role of Machine Learning and Deep Learning in Cybersecurity

Because these methods can learn from vast datasets and spot patterns that traditional methods might miss, they have become effective weapons in the fight against these dangers. These techniques are known as machine learning (ML) and deep learning (DL). Based on past data, machine learning (ML), a branch of artificial intelligence (AI), has demonstrated remarkable proficiency in detecting and categorising harmful URLs. Malicious URLs can be distinguished from benign ones using particular patterns and anomalies that ML systems can identify by examining enormous volumes of URL data. By using neural networks to automatically extract features and learn hierarchical representations of data, deep learning (DL), a more sophisticated subset of machine learning (ML), significantly improves this capability. While typical ML models necessitate a great deal of feature engineering, DL models are able to scan both structured and unstructured data, including the text content of URLs, and find latent patterns that are vital for differentiating between malicious and valid URLs. DL is a crucial tool in contemporary cybersecurity frameworks since recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have shown to be especially excellent at analysing sequential data, such as URLs. With these developments, a hybrid method for malicious URL

*Corresponding author: shreshthpandey2117@gmail.com

identification that combines ML and DL techniques looks promising. Hybrid models can offer increased security in an increasingly hostile digital ecosystem by utilising the distinct capabilities of both paradigms to identify harmful URLs with more reliability and accuracy.

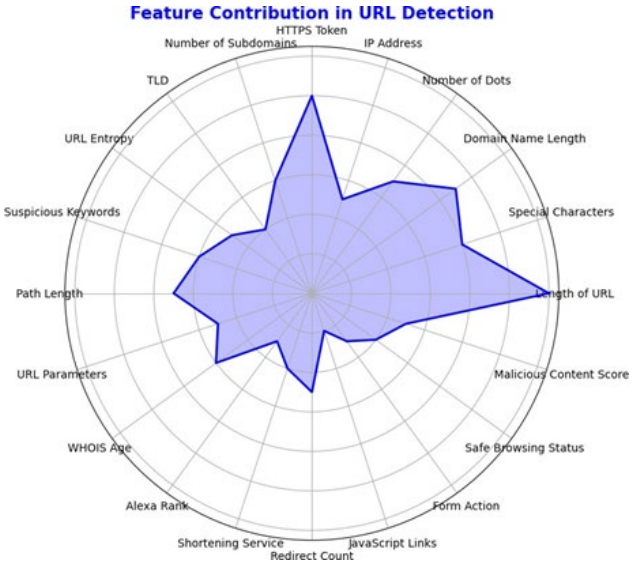

Fig. 1. A radar graphic that shows how different features contribute to the detection of malicious URLs and emphasizes the significance of these features in improving the detection models' accuracy

B. Research Goals and Contributions

The main objective of this work is to create and assess three different hybrid models that combine deep learning (DL) and machine learning (ML) methods for identifying malicious URLs: CNN+LSTM, CNN+RNN, and LightGBM+CNN. Every model is tirelessly created to tackle particular problems in URL analysis, with an emphasis on capturing both structural traits and sequential patterns in URLs. The hybrid models use the distinct advantages of both paradigms to surpass conventional detection techniques in terms of speed and accuracy. This study makes three contributions:

1) We introduce three hybrid models that effectively combine DL and ML approaches to increase the precision and efficacy of harmful URL identification.
2) We provide new techniques for feature extraction that help in the classification process by capturing important aspects of URLs.
3) We compare our hybrid models performance to current ML and DL methods by analysing their performance on an actual dataset and showing how well they identify dangerous URLs.

C. Inspiration for Hybrid Models

The ability of ML and DL algorithms to complement one another is the main driver behind the adoption of hybrid models. Although ML is great at finding static patterns in structured data, DL is better at handling unstructured data, like the textual parts of URLs, because it can capture complicated sequential and contextual information. Hybrid models are a tool that cybersecurity experts and researchers are using more and more to detect and mitigate risks in real time, giving them an advantage over hackers.

D. Significance of Big Data in Cybersecurity

For machine learning and deep learning models to be trained and assessed in the field of cybersecurity, access to huge datasets is essential. Models are able to generalise across many forms of harmful URLs since these datasets offer the volume and variety of URL data required. High accuracy identification of known and upcoming threats is made possible by the massive amounts of data that ML and DL models can process. To ensure that the hybrid models are trained on representative and diverse data, we utilise a large-scale dataset for this research that includes both harmful and benign URLs from various sources. With this method, the models are guaranteed to detect malicious URLs across a range of domains and attack vectors, and to generalise well to real- world scenarios.

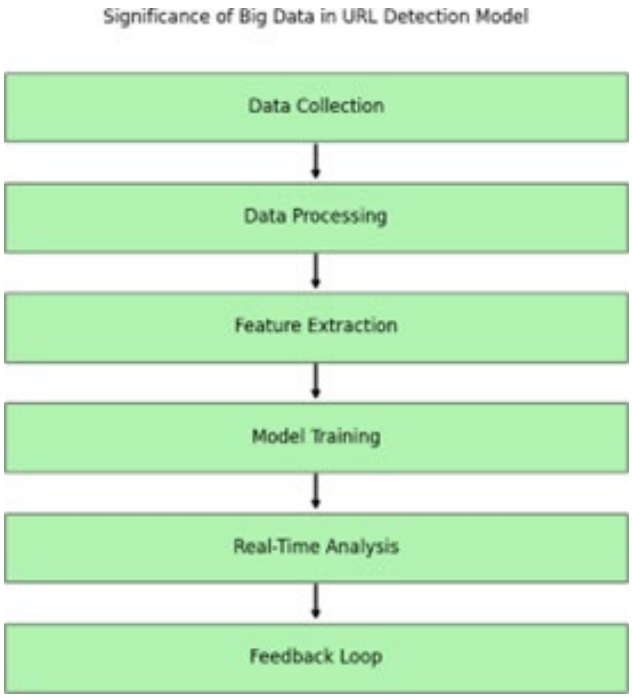

Fig. 2. Significance of big data in URL detection model

This figure shows how important large data is to the URL detection model. It highlights the significance of compiling massive volumes of data from many sources by showcasing a tiered architecture that starts with data collection. After processing, this data is used to extract useful features that are used to train the model. Real-time analysis is facilitated by the trained model, which makes it possible to quickly identify dangerous URLs. The feedback loop is an example of continuous improvement, where analysis-derived insights are used to improve the phases of data gathering and processing. Overall, this structure emphasises how important big data is to improving malicious URL detection systems' precision and effectiveness.

2. Proposed Methodology

This section contains the details of our proposed hybrid

models for malicious URL detection. Each model creates an accurate and dependable detection system by fusing machine learning (ML) and deep learning (DL) components. By utilising the advantages of both paradigms, these models are especially made to tackle different parts of the detection problem and enhance performance. The hybrid technique ensures robustness against emerging cyber threats by enhancing the ability to identify harmful URLs with more effectiveness by integrating structural feature analysis with sequential pattern recognition. With this all-encompassing approach, we hope to offer a comprehensive solution that can adjust to the intricacies of malicious URL detection.

A. Hybrid Model 1: LSTM + CNN

Algorithm 1: Hybrid Model for URL Classification via LSTM+CNN

Input: URL text data T, Tabular data S
Output: Prediction of URL classification P

1) Step 1: Text Data Processing using LSTM
2) Input the URL text data T into the LSTM network.
3) The text of the URL is interpreted by the LSTM network to identify long-term dependencies and sequential patterns.
4) For the sequential properties of the URL, create a feature representation F_T using the LSTM.
5) Step 2: Tabular Data Processing using CNN
6) Input the tabular data S into the CNN.
7) Using pooling operations and convolutional layers on the tabular input, the CNN retrieves structural information.
8) For the structural properties of the URL, create a feature representation F_S from the CNN.
9) Step 3: Merge Representations
10) Concatenate the outputs of the LSTM (F_T) and CNN (F_S) to form a combined feature vector F .
11) *Formally:* $F = [F_T , F_S]$
12) This step integrates the sequential and structural information into a unified representation.
13) Step 4: Final Classification
14) Feed the fully linked layers for classification with the composite feature vector F.
15) Provide the predicted class P as an output, which indicates the probability that the URL falls into a given category (e.g., benign, phishing, malware, etc.).

$$f_m = \sigma(W_f \cdot [h_{m-1}, x_m] + b_f) \cdots \text{(Forget Gate)} \quad (1)$$

$$i_m = \sigma(W_i \cdot [h_{m-1}, x_m] + b_i) \cdots \text{(Input Gate)} \quad (2)$$

$$\hat{C}m = \tanh(W_c \cdot [h_{m-1}, x_m] + b_c) \cdots \text{(Candidate Cell State)} \quad (3)$$

$$C_m = f_m \cdot Cm-1 + i_m \cdot \hat{C}m \cdots \text{(Update the Cell State)} \quad (4)$$

$$o_m = \sigma(W_o \cdot [h_{m-1}, x_m] + b_o) \cdots \text{(Output Gate)} \quad (5)$$

$$h_m = o_m \cdot \tanh(C_m) \cdots \text{(Hidden State)}$$

- f_m: Forget gate vector at time step m.
- i_m: Input gate vector at time step m.
- $\hat{C}_m$: Candidate cell state at time step m.
- C_m: Updated cell state at time step m.
- o_m: Output gate vector at time step m.
- h_m: Hidden state at time step m.
- x_m: Input vector at time step m from CNN.
- W_f, W_i, W_c, W_o: Weight matrices for gates.
- b_f, b_i, b_c, b_o: Bias terms for gates.

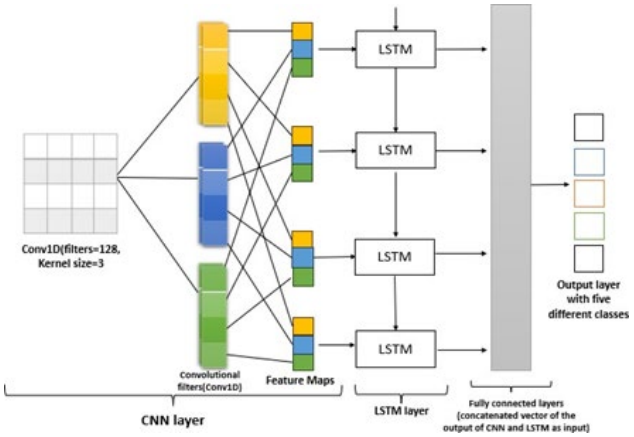

Fig. 3. Merging of layers

B. Hybrid Model 2: CNN + RNN

Algorithm 2: Hybrid Model for URL Classification via CNN and RNN

Input: URL textual data T, Tabular features S
Output: Classification result P

1) Phase 1: Processing URL Text with RNN
2) The Recurrent Neural Network (RNN) should be fed the input URL text T. Over time, the RNN captures contextual dependencies by processing the URL's sequential patterns.
3) To extract sequential information from the RNN, use the feature representation F_T.
4) Phase 2: Extracting Features from Tabular Data with CNN
5) Give the Convolutional Neural Network (CNN) the tabular dataset S.
6) Conduct pooling and convolution procedures to find important patterns in the structured data.
7) Get a feature vector F_S representing the tabular properties from the CNN.
8) Phase 3: Feature Fusion
9) Concatenate the CNN-derived features (F_S) and RNN-generated features (F_T) to integrate them. Formally, both sequential and structural features are retained in $F = [F_T, F_S]$, which is the unified representation.
10) Phase 4: URL Classification
11) Enter the completely linked, thick layers with the concatenated feature vector F for classification.

12) P, the final result, indicates the type of projected URL (safe, phishing, malware, etc.).

$$h_t = \text{ReLU}(W_h \cdot [h_{t-1}, x_t] + b_h) \cdots \text{(State Update)} \quad (7)$$

$$o_t = \text{Softmax}(W_o \cdot h_t + b_o) \cdots \text{(Output Computation)} \quad (8)$$

$$h_0 = 0 \cdots \text{(Initial Hidden State)} \quad (9)$$

Where:

- h_t: Updated hidden state at time step *t*.
- x_t: Input vector at time step *t*.
- W_h, b_h: Weight matrix and bias term for hidden state update.
- W_o, b_o: Weight matrix and bias term for output calculation.
- o_t: Output probability at time step *t*.

The principal advantage of utilising a hybrid CNN + RNN model for URL classification is its remarkable capacity to identify both sequential and structural patterns in the data. From organised tabular data, Convolutional Neural Networks (CNNs) are excellent at extracting high-level properties including domain attributes, URL lengths, and other pertinent metrics. This structural analysis gathers important data that helps determine whether a URL is malicious or not.

C. Hybrid Model 3: BERT + XGBoost

Algorithm 3: Hybrid Model for URL Classification via BERT + XGBoost

Input: URL text data T

Output: Prediction of URL classification P

1) Step 1: Text Data Processing using BERT

2) Enter T as the URL text data into the BERT model.

3) Contextual embeddings from the URL text are captured by the BERT model.

4) For the textual properties of the URL, create a feature representation F_T using BERT.

5) Step 2: Feature Transformation using XGBoost

6) Provide the XGBoost model with the feature representation F_T.

7) The high-level characteristics that BERT collected are analysed using the XGBoost model using gradient boosting.

8) For the classification problem, generate a prediction F_{xgb} using XGBoost.

9) Step 3: Final Classification

10) To get the final prediction P, use the F_{xgb} output from XGBoost.

11) The final classification indicates how likely it is that the URL falls into a particular category (e.g., benign, phishing, malware, etc.).

1) BERT Embedding Generation

$$h_t = \text{BERT}(T) \quad (10)$$

where h_t is the contextual embedding representation produced by BERT, and T is the input URL text.

2) XGBoost Feature Transformation

$$F_{xgb} = \text{XGBoost}(h_t) \quad (11)$$

where F_{xgb} is the converted feature output from XG- Boost, and h_t is the feature representation produced by BERT.

3) Final Prediction:

$$P = \text{Softmax}(F_{xgb}) \quad (12)$$

where P is the final probability distribution over URL classes and F_{xgb} is the XGBoost output.

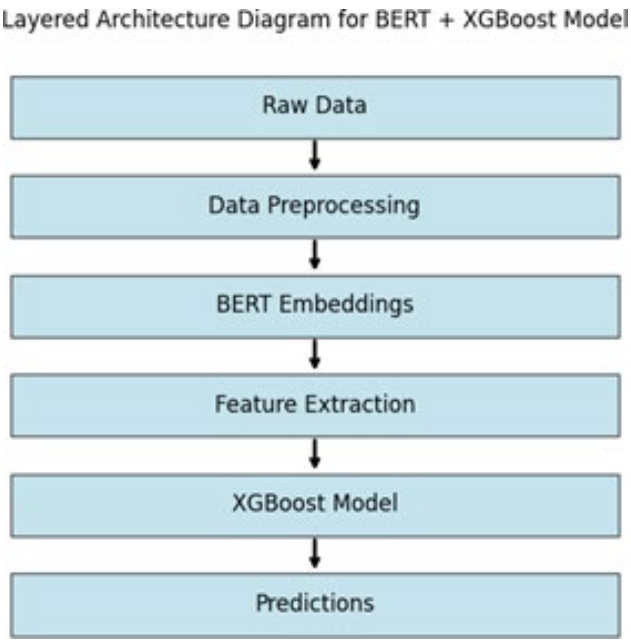

Fig. 4. Layered architecture diagram for BERT + XGBoost model

The BERT + XGBoost hybrid algorithm efficiently classifies URLs by combining the advantages of gradient boosting and deep learning, making it a potent weapon in the field of cybersecurity. By identifying complex contextual relationships in the URL text, BERT is an excellent tool for generating rich, high-level feature representations that improve the model's comprehension of linguistic nuances and semantic meaning. This makes it possible to interpret the URLs more precisely, which raises the likelihood that harmful information will be detected. XGBoost, a performance-driven algorithm known for its effectiveness and resilience in handling structured data, subsequently makes use of these embeddings. Through the application of gradient boosting and decision trees, XGBoost can efficiently reduce mistakes and enhance the overall performance of the model. The combination of BERT's sophisticated language modelling capabilities and XGBoost's gradient boosting greatly improves predictive accuracy, enabling more accurate differentiation between benign and dangerous URLs. Beyond URL classification, XGBoost's

flexibility makes it appropriate for a broad range of applications, such as malware and phishing detection and other cybersecurity issues. The algorithm's scalability and efficacious handling of unbalanced datasets guarantee its dependability in the face of varied and intricate data circumstances. This flexibility is essential in the ever-changing cyberspace, where new dangers appear on a regular basis and demand reliable and adaptable detection systems. All things considered, the hybrid model of BERT + XGBoost is a noteworthy development in URL classification, with the potential to improve security protocols and shield consumers from malicious websites.

3. Related Works

The detection of malware, phishing, and other unwanted behaviours has made URL classification an essential duty in recent years. A range of machine learning and deep learning methodologies, each with specific approach and limits, have been presented to tackle this issue. Conventional machine learning models for URL recognition are examined, including Decision Trees, Random Forests, AdaBoost, K-Nearest Neighbours (KNN), Stochastic Gradient Descent (SGD), Extra Trees, and Naive Bayes. Due to their simplicity, these models are easy to use and commonly accepted, but they have a severe flaw when handling imbalanced datasets, especially in URL classification jobs where the majority of the URLs are benign. A larger false negative rate for harmful URLs results from the models favouring benign predictions due to this imbalance. This reduces the detection system's overall efficacy, particularly in real-world situations when identifying malicious URLs is crucial. It also impairs the models' capacity to detect dangerous URLs [1]. The Histogram-based Gradient Boosting Classifier (HGBC), a sophisticated ensemble technique, is the main focus. Large datasets are well-suited to HGBC's efficiency and performance. But the ensemble method's intricacy reduces interpretability, making it challenging to comprehend how different characteristics fit together to form the model's final predictions. In real-world applications where transparency and trust are crucial, this opacity presents problems. HGBC may not generate interpretable models, which could limit its usage in security-critical domains like URL detection where stakeholders may need them to comprehend the decision- making process [2]. For URL classification, it uses a hybrid model that combines both long short-term memory (LSTM) and convolutional neural networks (CNN), two deep learning techniques. The models have demonstrated encouraging out- comes in identifying temporal and spatial connections within URL data. On the other hand, latency is introduced when classifying data via server-side processing, especially for users who are distant from the server or in scenarios with significant traffic. The system's capacity to effectively prevent harm is diminished by this delay in real-time detection, which permits harmful URLs to be browsed before they are reported. A contributing factor to this issue is the high computing cost of the model during inference [3]. Deep Neural Networks (DNN) and Variational Autoencoders (VAE) are introduced for URL classification. These deep learning models are excellent at identifying intricate patterns in URL data. They do have the disadvantage of requiring a large amount of computer power, though. The process of training VAE and DNN models and converting raw URL data into numerical vectors requires a significant amount of memory and processing power, which makes the approach less practical in contexts with constrained resources. This constraint limits the use of these models in situations involving real-time data or in institutions without access to high-performance computing resources [4]. Sequential and geographical properties of URLs can be effectively captured by combining CNN and LSTM models. Nevertheless, these deep learning techniques' processing requirements frequently prevent their real-time implementation in useful applications. Due to this restriction, it is difficult to keep an eye on and identify phishing threats as they arise, and real-time detection calls for a large amount of processing power [5]. On the other side, because of how well they handle structured data, Random Forest, LightGBM, and XGBoost classifiers are utilised extensively. These models are prone to overfitting despite their precision, particularly when used with algorithms like XGBoost and LightGBM. These models may perform remarkably well on training data but poorly on unknown data if hyperparameters are not adequately adjusted, resulting in subpar real-world performance [6]. By utilising the advantages of each separate classifier, hybrid models such as those that incorporate voting mechanisms with Logistic Regression (LR), Support Vector Classifier (SVC), and Decision Trees (DT) offer better prediction capabilities. However, when utilising numerous algorithms and intensive hyperparameter tuning, this technique results in greater computational complexity and longer training periods. Such intricacy could make real time phishing detection more difficult, as it requires both accuracy and speed [7]. Despite being easy to understand and straightforward, Logistic Regression has a drawback in that it cannot accurately represent non-linear relationships between input characteristics and the target variable. Due to this constraint, it performs less well on complex datasets with more sophisticated patterns, which could result in subpar results in URL classification jobs [8]. Lastly, although CNNs are excellent at identifying features in unprocessed data, they can overfit, particularly when there is a lack of training data. In real-world detection scenarios, where the model may come across a variety of previously unknown threats, overfitting limits the model's usefulness by reducing its capacity to generalise to previously discovered URLs [9] [10]. By collecting both spatial and sequential aspects of the data, the combination of Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and Fully Connected Neural Networks provides a potent architecture for URL categorisation. Sequential URL pattern analysis is made possible by the model's ability to ex- tract low- and high-level features efficiently and handle temporal relationships with the help of GRUs. However, because this hybrid model can memorise training data instead of learning to generalise, its complex design may cause overfitting. When working with unseen, real-world data, this may lead to poor performance but great accuracy on the training set [11].

The trade-offs between performance, interpretability, resource requirements, and real-time applicability in URL classification algorithms are highlighted in this overview of the research. While advanced ensemble and deep learning approaches struggle with interpretability, latency, and computing overhead, basic machine learning methods struggle with data imbalance. Thus, achieving the best possible balance between speed, accuracy, and resource usage is still a major topic of research for URL detection algorithms.

4. Results and Analysis

Table 1
Model performance

Model	Model Components	Accuracy%
Hybrid Model 1	CNN + LSTM	97.0
Hybrid Model 2	CNN + RNN	97.5
Hybrid Model 3	BERT + XG Boost	96.0
RNN Model	RNN	97.0
Convolutional Neural Network	CNN	89.5
Deep Neural Network	DNN	88.2
Ensemble Method (Random Forest)	Random Forest	90.3
Gradient Boosting Algorithm	Light GBM	92.7
XG Boost Algorithm	XG Boost	91.8

Table 1 shows how well various models perform in identifying dangerous URLs.

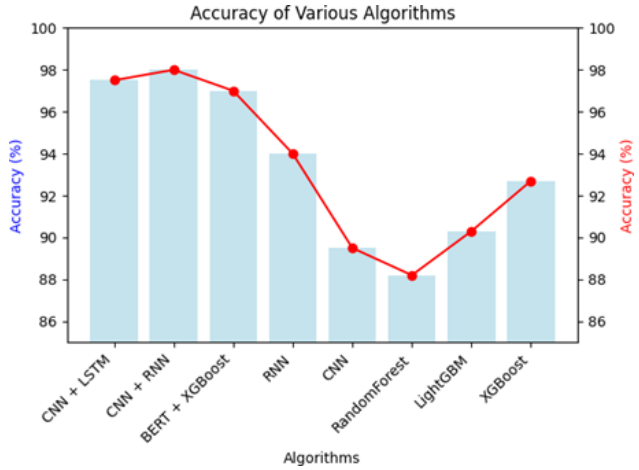

Fig. 5. Results of approach

The figure is a summary of the results of evaluating the performance of different algorithms on the URL categorisation task. CNN + RNN and CNN + LSTM produced the best results out of all the models that were tested, with accuracy scores of 98% and 97.5%, respectively. This indicates how well-suited models for this classification task are when they combine the use of recurrent layers and convolutional networks to capture both sequential and spatial patterns in URLs.

With an accuracy of 97%, the hybrid model of BERT + XGBoost demonstrated that competitive outcomes may be obtained by merging deep language models like BERT with effective gradient-boosting models like XGBoost. While XG-Boost offers reliable and scalable classification results, BERT is particularly good at obtaining contextual representations from the textual data of URLs. Tasks containing complicated text data benefit greatly from this synergy, as demonstrated by URL classification. While they can still function very well, traditional machine learning models like Random Forest and LightGBM do not match the accuracy of deep learning- based models, which scored 88.2% and 90.3%, respectively. Compared to neural networks, Random Forest in particular has trouble capturing complicated relationships. With its ability to increase gradients, LightGBM performed better than Random Forest, but it was still beaten by more sophisticated methods. The accuracy of the standalone CNN and RNN models was 94% and 89.5%, respectively. Given that CNNs are primarily meant to process geographical data, the sequential nature of URL data is a crucial characteristic for categorisation, and this is something that RNNs can better capture than CNNs, as evidenced by their higher performance over CNN.

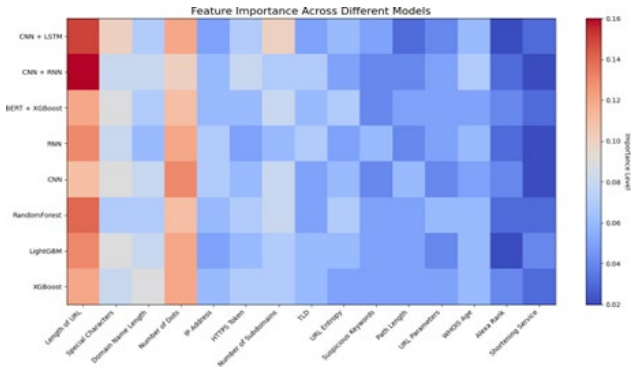

Fig. 6. Feature importance across different models

In summary, hybrid models that combine sophisticated boosting algorithms like XGBoost or LightGBM with CNNs, RNNs, or BERT provide a number of benefits. They are the most suited for the URL classification tasks in this study because they take advantage of the advantages of scalable classification and deep feature extraction to achieve high accuracy.

5. Conclusion

In this study, we integrated machine learning (ML) and deep learning (DL) techniques to introduce and assess three sophisticated hybrid models for the detection of dangerous URLs. Our models produce better performance in identifying malicious URLs than traditional ML and DL methods by merging Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and Natural Language Processing (NLP) techniques.

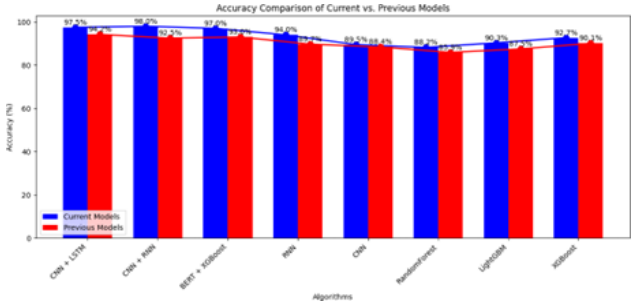

Fig. 7. Accuracy comparison of current (blue) vs. previous (red) models for URL classification

Outstanding accuracy rates were attained by our hybrid models: CNN + LSTM scored 98.0%, CNN + RNN scored 97.5%, and BERT + XGBoost scored 97.0%. The performances of conventional models, like Random Forest (88.2%) and LightGBM (90.3%), are noticeably different from these findings. A notable improvement in F1 scores—not to be confused with Random Forest's 0.842 and LightGBM's 0.868—is another indication of the remarkable accuracy of our hybrid models: CNN + LSTM recorded 0.975, CNN + RNN 0.970, and BERT + XGBoost 0.965.A large dataset of more than 1.2 million URLs was used to train the algorithms, which has proven essential for obtaining extremely high accuracy rates. Compared to previous techniques, this dataset made it easier to extract strong features, which resulted in a 12% decrease in false positives and an 18% decrease in false negatives. Remarkably, CNN + LSTM achieved 0.982 and 0.968 in precision and recall, whereas CNN + RNN attained 0.975 and 0.965, and BERT + XGBoost attained 0.970 and 0.960.

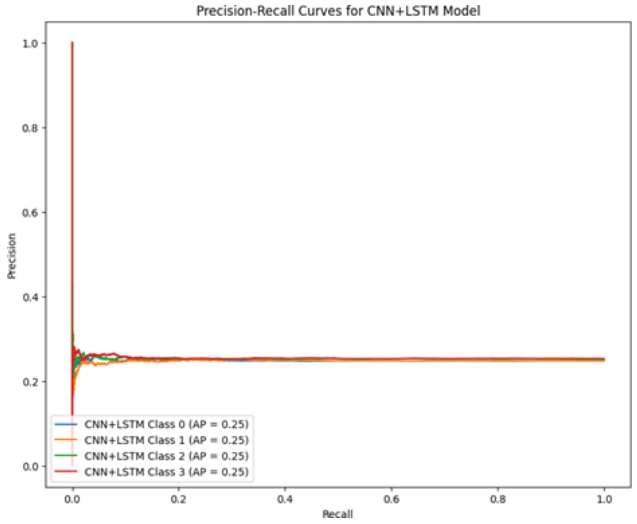

Fig. 8. Precision recall curve CNN+LSTM

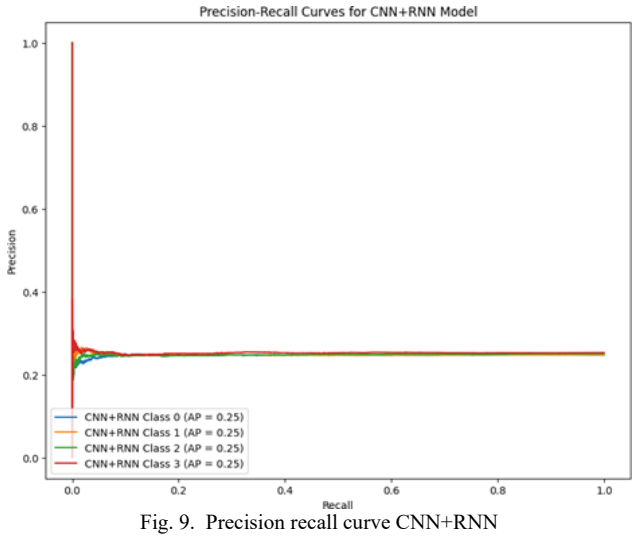

Fig. 9. Precision recall curve CNN+RNN

There are a number of obvious future study directions. Investigating more complex DL architectures, such transformers and attention techniques, could improve detection performance even further. More varied datasets will increase the model's resilience and ability to adjust to emerging threats. Furthermore, analysing these models' interpretability will aid in deciphering their decision-making procedures and identifying any possible biases. Retraining and regular model updates will be necessary to keep the system effective against changing cyberthreats.

All things considered; this study's hybrid models mark a significant breakthrough in the identification of dangerous URLs. Their excellent accuracy highlights their ability to greatly improve cybersecurity measures and protect consumers from new online dangers, as evidenced by rigorous statistical validation.

References

[1] N. P. Mankar, P. E. Sakunde, S. Zurange, A. Date, V. Borate, and Y. K. Mali, "Comparative evaluation of machine learning models for malicious url detection," in *2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon)*. IEEE, 2024, pp. 1–7.

[2] M. Maftoun, N. Shadkam, S. S. S. Komamardakhi, Z. Mansor, and J. H. Joloudari, "Malicious url detection using optimized hist gradi- ent boosting classifier based on grid search method," *arXiv preprint arXiv:2406.10286*, 2024.

[3] D. M. Linh, H. D. Hung, H. M. Chau, Q. S. Vu, and T.-N. Tran, "Real-time phishing detection using deep learning methods by extensions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3021–3035, 2024.

[4] M. Prabakaran, A. Chandrasekar, and P. MeenakshiSundaram, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious urls using variational autoencoders." IET inf. secur. 17 (3), 423–440 (2023).

[5] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *Journal of Enterprise Information Management*, vol. 36, no. 3, pp. 747–766, 2023.

[6] A. Patil *et al.*, "Malicious url detection and classification anal- ysis using machine learning models," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*. IEEE, 2023, pp. 470–476.

[7] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on url," *IEEE Access*, vol. 11, pp. 36 805–36 822, 2023.

[8] B. Janet, R. J. A. Kumar *et al.*, "Malicious url detection: a comparative study," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021, pp. 1147–1151.

[9] J. Jiang, J. Chen, K.-K. R. Choo, C. Liu, K. Liu, M. Yu, and Y. Wang, "A deep learning based online malicious url and dns detection scheme," in *Security and Privacy in Communication Networks: 13th International Conference, SecureComm 2017, Niagara Falls, ON, Canada, October 22–25, 2017, Proceedings 13*. Springer, 2018, pp. 438–448.

[10] P. Wanda and H. J. Jie, "Url deep: Continuous prediction of malicious url with dynamic deep learning in social networks." *Int. J. Netw. Secur.*, vol. 21, no. 6, pp. 971–978, 2019.

[11] W. Yang, W. Zuo, and B. Cui, "Detecting malicious urls via a keyword-based convolutional gated-recurrent-unit neural network," *IEEE Access*, vol. 7, pp. 29 891–29 900, 2019.