**Hybrid Approach to Detect Phishing Links**

Sumit Sharma1

Dept. of Information Technology

Maharaja Agrasen Institute of Technology (MAIT),

New Delhi, India

[Sumit.02014803122@it.mait.ac.in](mailto:Sumit.02014803122@it.mait.ac.in)

Mrs. Vandana Choudhary2

Dept. of Information Technology

Maharaja Agrasen Institute of Technology (MAIT),

New Delhi, India

[vandanachoudhary@mait.ac.in](mailto:vandanachoudhary@mait.ac.in)

Aryan Gahlaut3

Dept. of Computer Science

Delhi Technological University (DTU),

New Delhi, India

[aryangahlaut\_co22a3\_15@dtu.ac.in](mailto:aryangahlaut_co22a3_15@dtu.ac.in)

*Abstract—* Phishing continues to be a leading cybersecurity attack, where actors target individuals and organizations to exfiltrate sensitive data like login credentials and personal financial information. Conventional techniques like blacklisting and heuristic-based detection are having a hard time keeping pace with the changing adversary tactics. Within this paper, we introduce a hybrid phishing model that integrates CNN with Transformer-based models DistilBERT to improve accuracy and robustness in detection. We compare our method with a variety of machine learning models, such as XGBoost, Random Forest, Decision Tree, Logistic Regression, and a single CNN model. Using a 5,49,346 URL dataset from Kaggle, our suggested model yields 99.006% accuracy, precision, recall, and F1-score of 0.99. Experimental results show that our hybrid method outperforms traditional ML algorithms and stand-alone deep learning models and exhibits higher efficacy in detecting phishing attacks. The results prove the possibility of applying CNN and Transformers together for phishing detection and pave the way for further research involving adaptive real-time security solutions.

*Keywords—* *Phishing detection, deep learning, machine learning, CNN, Transformers, DistilBERT, cybersecurity, hybrid model, phishing attacks,* *real-time security.*

I. INTRODUCTION

Phishing has become one of the most persistent cyber attacks where the attackers attempt to steal confidential information such as login credentials, financial data, and personal information from individuals and organizations. The ever-evolving nature of phishing attacks, such as the utilization of deceptive URLs and social engineering tactics, complicates it for traditional detection methods to keep up. Blacklisting and heuristic techniques, although pervasive, are inadequate to cope with new and advance phishing techniques and thus expose the systems to potential attacks. Aiming to alleviate these weaknesses, machine learning as well as deep learning methodologies have been utilized in phishing detection because they provide increased adaptability with enhanced accuracy. Classic machine learning models like XGBoost, Random Forest [14], Decision Tree, and Logistic Regression inspect URL-based features to segregate phishing and valid links. But these models tend to be lacking in capturing rich contextual relationships present in URLs. Deeplearning models, especially Convolutional Neural Networks (CNN), have been found useful for extracting higher-level patterns, but they can also fail with complicated textual patterns found in phishing URLs. Herein, we introduce a hybrid model that combines CNN with a model based on Transformer, DistilBERT, to improve detection robustness and accuracy. The CNN model captures spatial and structural attributes from URLs, whereas DistilBERT uses natural language processing methods to examine patterns in text. To assess the performance of our method, we compare it with a classical machine learning algorithm and an isolated CNN model. With a dataset of 5,49,346 URLs obtained from Kaggle, our suggested model produces an accuracy of 99.006%, with precision, recall, and F1-score all being 0.99. The experiments show that our hybrid method is superior to traditional machine learning and isolated deep learning methods and thus, it is a viable approach for real-world phishing detection. The outcomes of this research also show the efficacy of integrating CNN and Transformer-based models as an effective method of phishing detection. This study also provides a foundation for future research in adaptive, real-time security systems that can dynamically react to new phishing attacks.

II. RELATED WORK

1. *Hybrid Approach*

Phishing detection via machine learning and deep learning goes beyond standard list-based systems, which rely on pre-defined whitelists and blacklists. Within list-based systems, a whitelist contains known safe URLs, while a blacklist contains known bad URLs. When a user submits a URL, the system checks it against these lists to determine access. Though efficient in detecting well-known attacks, the major disadvantage of the technique is the ongoing updating process in order to keep pace with fresh, new phishing attacks. Phishers always create new malicious domain names, thus static lists are less useful in dynamic environments.

To address such constraints, the suggested hybrid approach incorporates machine learning and deep learning for dynamic detection of phishing URLs [1] [2]. Rather than being based on predefined lists, our system extracts patterns from vast data sets utilizing feature-based ML models (XGBoost, Random Forest, etc.) [5] [7] as well as deep learning methods like CNNs and Transformers (DistilBERT) [1]. This allows the system to learn and generalize over familiar threats as well as discover new phishing attempts even if the system has not previously seen them.

The proposed system enhances security through the exploitation of the benefits of both structural pattern recognition (CNNs) [2] and text-based NLP models (DistilBERT), enhancing its adaptability, scalability, and resistance to evolving phishing patterns. The hybrid system offers an effective detection framework that does not require constant updates by humans compared to list-based systems.

1. *Deep Learning Techniques*

Elsadig et al. utilized the Kaggle Phishing Site Predict dataset, leveraging the BERT's ability to extract features as well as the machine learning classifiers in enhancing their phishing attack detection. Their work [1] leveraged the deep contextual ability of BERT in extracting semantic patterns from URLs without being subject to rule-based and blacklist constraints [3]. Their solution was, however, plagued by limitations such as high computational expenses, reliance on large labelled datasets for fine-tuning, and vulnerability to adversarial attacks with the ability to manipulate token-level representations to circumvent detection.

Inspired by their work [1] [3], we propose an improved hybrid model by integrating CNNs with DistilBERT to improve phishing detection. Although we employ the same Kaggle dataset, our approach avoids the computational inefficiencies of the full BERT models by using CNNs to capture spatial URL patterns to supplement DistilBERT's transformer-based feature learning. The combination yields a leaner yet efficient model without sacrificing robustness while reducing computational overhead. More importantly, the combination of CNNs and Transformers makes the model more dynamic to learn to cope with evolving phishing patterns, a more scalable and efficient solution than existing deep learning-based solutions.

1. *Machine Learning Techniques*

Phishing classification in machine learning algorithms also relies on feature classification such as URL pattern, domain name, and web pages. Various studies used various approaches to improving accuracy. Traditional approaches such as CANTINA[5] used Tf-IDF and heuristics with 90% accuracy. PhishWHO suggested a 3-Tier Identity Matching System with 96.10% accuracy. Other approaches used URL attributes, transport layer security, and nonlinear regression methods. Random Forest, SVM[8], and Naïve Bayes were compared often[7][8], with Random Forest generally performing better. The paper [5] concludes that feature selection and encoding of additional features such as the 58 features obtained by using URL analysis can improve accuracy and efficiency even more.

From these earlier works [4] [5], we have contrasted our suggested model with conventional machine learning methods, such as Random Forest, XGBoost, Decision Tree, Logistic Regression [6], and CNN only. Although these models were good when they were employed individually, our findings show that a hybrid model that uses DistilBERT and CNN yields the best accuracy. This model uses the contextual perception of DistilBERT for feature extraction [1] [8] and the pattern recognition ability of CNN and outperforms conventional methods in phishing URL detection.

III. METHODOLOGY

1. *Proposed Approach*

The proposed work was broken down into 8 phases:-

1. Data acquisitions
2. Pre processing
3. Building custom dataset
4. Data splitting
5. Creating data-loader
6. Model building
7. Training the model
8. Testing the model
9. *Data Acquisitions*

The performance of deep learning models relies significantly on the size and quality of the dataset. For our research, we obtained a complete dataset from Kaggle that contained 5,49,346 URLs. In contrast to other studies that used a balanced split, our dataset simulates a real-world setting with 1,53,817 phishing URLs and 3,95,529 legitimate URLs.

The imbalance was maintained on purpose to simulate more closely the natural ratio of web-based phishing attacks. We further apply label encoding to this data for ease of processing during model training. This partition serves as a strong reference point for training and evaluating the model under conditions that closely resemble real-world deployment scenarios.

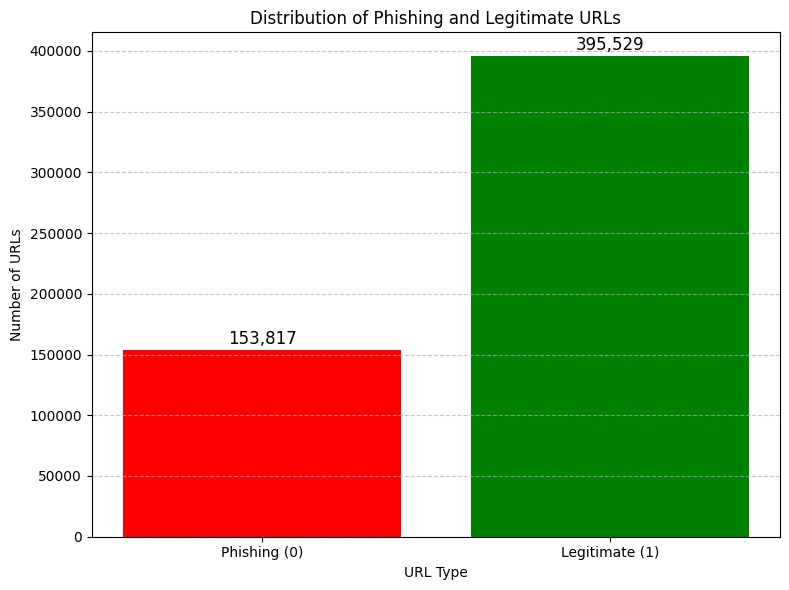


Fig. 1. Data distribution plot

1. *Pre-Processing*

Deep learning models, especially transformer-based models like DistilBERT cannot directly use our dataset of 549,346 URLs as input without preprocessing. Label encoding, in which every URL is identified as either phishing (0) or legitimate (1), makes sure the target variable is appropriate for binary classification.

1. *Building Custom Dataset*

We developed a unique dataset class called PhishingDataset to efficiently handle preprocessing and prepare URL inputs for our hybrid deep learning model. This class passes each URL through the DistilBERT tokenizer using a fixed max length of 64 tokens. The tokenizer automatically truncates URLs longer than 64 tokens and pads shorter ones with [PAD] tokens to guarantee consistency across all inputs. Each item in the dataset returns a flattened tensor of input ids, attention mask and the corresponding label. Consequently, the input shape (549,346 × 64) remains consistent across the dataset.

1. *Data Splitting*

To prepare our data for training and evaluation, we used an 80-20 train-test split. The train test split function divides the dataset at random, allocating 80% of the URL label pairs to the training set and 20% to the test set. This ensures that while learning from a large portion of the data, the model is validated on a unique, invisible subset of data. By setting random state = 42, reproducibility is guaranteed, the split remains constant throughout code execution.

1. *Creating Data Loader*

We create DataLoader objects to efficiently handle data batches during training and testing. The train\_loader is set up with a batch size of 512 and shuffling enabled to ensure better generalization by feeding the model randomized data every epoch, whereas the test\_loader handles evaluation batches without shuffling. We also load the pre-trained DistilBERT model (distilbert-base-uncased) from Hugging Face, which converts tokenized URLs into embeddings that are contextually relevant. This model is then moved to the GPU (cuda) to speed up the training and inference processes.

1. *Model Building*

Our hybrid model combines the output of both Disitilbert model and CNN model.

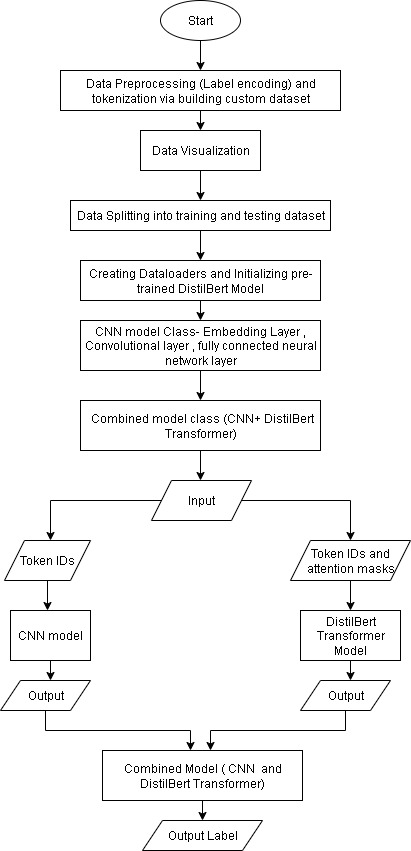


Fig. 2. Model Architecture

First, the pretrained Distilbert transformer model takes input as input\_id and attention\_mask.

Secondly, CNN model is given input IDs as input. An embedding layer converts each token IDs to dense vectors of dimension 128. Therefore, embedding layer is used before convolutional layers and shape of tensor thus becomes [512,64,128]. This tensor is permuted to [512,128,64] to get it feed into convolutional layers. Each layer has 128 input channels and 100 filters with stride of 1. First layer has filter size (3) and shape [128,3] , second having filter size (4) and shape [128,4] and third having filter size of (5) and shape [128,5]. Each layer works in parallel and takes input tensor as input and outputs feature maps of shapes [512,100,62] ,[512,100,61], [512,100,60] respectively. Relu activation function is then used on all these outputs. After it, the max pool is then used to extract   
most important feature, reducing each of them to [512,100]. The output from these layers is then concatenated to form tensor of shape [512,300] and passed to the fully fused neural network layer. The output dimension of CNN model is 128. Dropout value of 0.5 is used to prevent overfitting.

Third, The Combined model class combines CNN and transformer model output tensors. It is initialized with dropout = 0.3 to prevent overfitting and output dimension = 2 (for binary classification). Output tensors are concatenated and then fed directly into one layer and the given output value is then taken and used to predict label.

1. *Training The Model*

This dataset is representative of a real-world distribution with 1,53,817 phishing URLs and 3,95,529 legitimate URLs. It follows an 80-20 train-test split with a fixed random state (random\_state=42). Consequently, 4,39,476 URLs are allocated for training and 1,09,870 for testing. In the training phase, the DistilBERT model, CNN model, and the fusion model are all set to training mode.

For every epoch, a batch of data are loaded where input\_ids, attention\_mask, and labels are given to the GPU. Input\_ids and attention\_mask are first processed by running them through the DistilBERT model in order to retrieve deep semantic embeddings based on the representation of the token which gets the meaning of the whole sequence of inputs. At the same time, the same input\_ids are fed through the CNN model to learn local features that learn sequential and character-level patterns within the URLs. These two representations are then combined in the joint model to produce final predictions.

Cross-entropy loss function is employed to calculate the difference between predicted and actual labels, and this loss is minimized by using backpropagation and AdamW optimizer with learning rate of (5e-5). This loop of training is repeated for some number of 3 epochs.

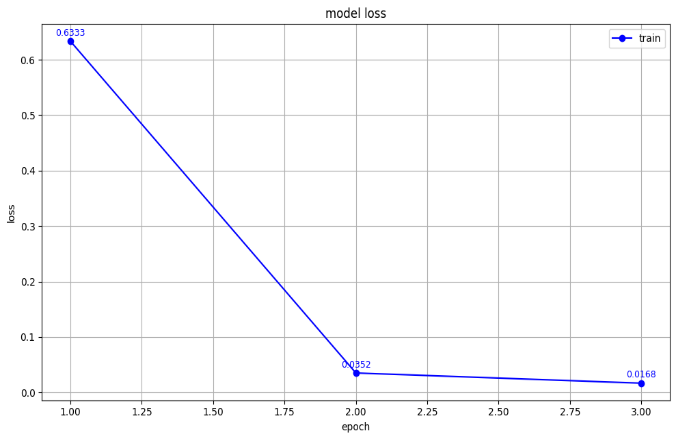


Fig. 3. Number of epochs vs. loss

1. *Testing the Model*

The trained model was validated using a test set of 1,09,870 URLs, and predictions were obtained. The chart for classification that indicates the accuracy, precision, recall, and F1 score is depicted in Fig.4. It can be observed that the precision for detecting phishing URLs was 99%, and the accuracy was 99.006%. The model shows high recall and high precision, making it a very good classifier. The confusion matrix of the classifier was visualized in Fig 5.

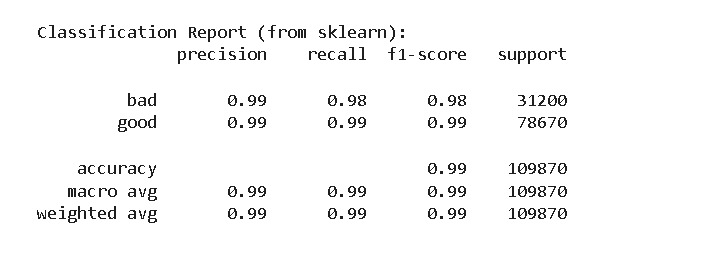


Fig. 4. Classification chart

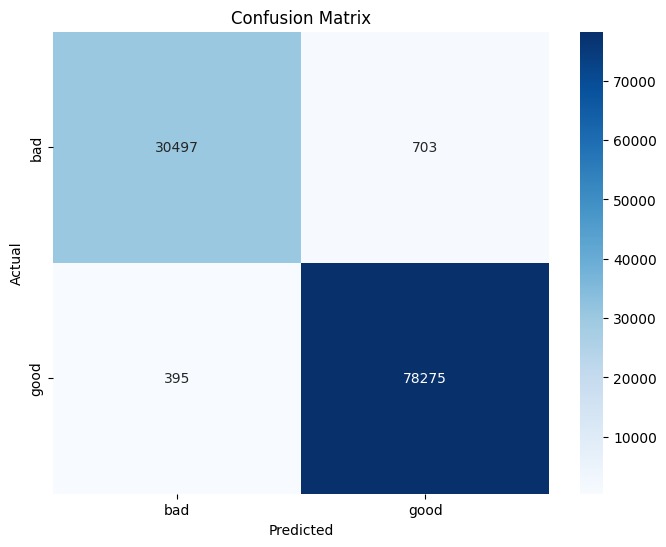


Fig. 5. Confusion Matrix

IV. RESULTS AND DISCUSSION

It was observed that our proposed model outperformed the existing models discussed in related work[2][7].

We have also compared our suggested hybrid model to some of the machine learning methods like XGBoost, Random Forest, Decision Tree, Logistic Regression, and a pure CNN model. Comparison was made using the same datasets and metrics to determine the fairness of the comparison. Results show that our model beats the above methods in all the important performance measures such as accuracy, precision, recall, and F1-score. This best illustrates the enhanced effectiveness and credibility of our model in precisely identifying phishing URLs and thus making it a very reliable solution in practical applications.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **F1 score** | **Recall** |
| *Proposed model* | *99.006%* | *0.99* | *0.99* | *0.99* |
| CNN classifier | 92.8679% | 0.9307 | 0.9263 | 0.9287 |
| Logistic regression | 90.42% | 0.9032 | 0.9355 | 0.9703 |
| Decision Tree | 91.73% | 0.9170 | 0.9440 | 0.9726 |
| Random Forest | 92.16% | 0.9185 | 0.9470 | 0.9773 |
| XGboost Classifier | 90.34% | 0.9049 | 0.9348 | 0.9667 |

Table 1. Comparison Table

To train the proposed hybrid DistilBERT-CNN model, experiments were conducted using the Google Colab environment on an NVIDIA T4 Tensor Core GPU (16 GB VRAM), approximately 16 GB of RAM and a dual-core Intel Xeon CPU. Due to the computational cost of transformer-based contextual embedding and convolutional feature extraction, training required substantial compute power. Total training time for 3 epochs was approximately 2 hours, averaging 40 minutes per epoch. VRAM usage was consistently high during training, frequently going near the 14–15 GB limit, which required memory-efficient batching of data and optimization. Furthermore, CPU usage during tokenization was moderately demanding.

V. CONCLUSION AND FUTURE SCOPE

The hybrid model based on Transformer-based DistilBERT with CNN reaches more than 99% accuracy within three training epochs while surpassing the performance of Random Forest, XGBoost, Decision Tree and Logistic Regression classifiers. The model demonstrates superior performance compared to previous phishing detection methods. Future work may involve exploring alternative transformer architectures and ensemble methods to further enhance performance. Improving model efficiency, reducing computational overhead and optimizing inference time will be essential for real-time deployment in environments like browsers, email systems, and enterprise security tools, especially in resource-constrained settings.

VI. REFERENCES

1. Elsadig, M., Ibrahim, A. O., Basheer, S., Alohali, M. A., Alshunaifi, S., Alqahtani, H., Alharbi, N., & Nagmeldin, W. (2022). Intelligent deep machine learning cyber phishing url detection based on BERT features extraction. Electronics, 11(22), 3647.
2. Gupta, B. B., Gaurav, A., Arya, V., Attar, R. W., Bansal, S., Alhomoud, A., & Chui, K. T. (2024). Advanced BERT and CNN-Based computational model for phishing detection in enterprise systems. Computer modeling in engineering & sciences, 141(3), 2165–2183.
3. M, Y. V., Janet, B., & Reddy, S. (2020). Anti-phishing system using LSTM and CNN. 2020 IEEE International conference for innovation in technology (INOCON).
4. Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina,a content based approach to detecting phishing web sites” Proceedings of the 16th international conference on World Wide Web - WWW 07, pp. 639-648,2007.
5. Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020). Detection of phishing websites by using machine learning-based url analysis. 2020 11th International conference on computing, Communication and networking technologies.
6. Alam, M. N., Sarma, D., Lima, F. F., Saha, I., Ulfath, R.-E.-, & Hossain, S. (2020). Phishing attacks detection using machine learning approach. 2020 Third international conference on smart systems and inventive technology (ICSSIT).
7. James, J., Sandhya L., & Thomas, C. (2013). Detection of phishing URLs using machine learning techniques. 2013 International conference on control communication and computing (ICCC).
8. Jain, A. K., & Gupta, B. B. (2018). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing.
9. Kamarudin, Siti & a Hamid, Isredza Rahmi & Mohd Foozy, Cik Feresa & Abdullah, Zubaile. (2022). Feature selection approach to detect phishing website using machine learning algorithm. AIP Conference Proceedings. 2644. 040003.
10. Zaimi, Rania & Mohamed, Hafidi & Lamia, Mahnane. (2024). A Permutaion importance based feature selection method and deep Learning model to detect phishing websites.
11. Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics*, *12*(1), 232.
12. Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2020). Intelligent phishing detection scheme using deep learning algorithms. Journal of Enterprise Information Management.
13. Ariyadasa, Subhash & Fernando, Subha & Fernando, Shantha. (2020). Detecting phishing attacks using a combined model of LSTM and CNN. International Journal of advanced and applied science. 7. 56-67.
14. Nagy, N., Aljabri, M., Shaahid, A., Ahmed, A. A., Alnasser, F., Almakramy, L., Alhadab, M., & Alfaddagh, S. (2023). Phishing urls detection using sequential and parallel ML techniques: Comparative analysis. *sensors*, *23*(7), 3467.