Malicious URL Detection using Machine Learning and LLM-based Approaches

Maheen Shoukat
Department of AI & DS, National University of Computer and Emerging Sciences, Islamabad
Email: i212719@nu.edu.pk

Syeda Aiman Azhar
Department of AI & DS, National University of Computer and Emerging Sciences, Islamabad
Email: i210290@nu.edu.pk

I. Introduction

The rapid growth of the Internet has made it an essential platform for communication, business, and social interaction. However, this convenience comes at a cost, as cybercriminals increasingly exploit URLs as conduits for illicit activities. Malicious URLs, including phishing, malware, defacement, spam, and other harmful links, pose significant threats to users and systems. Although traditional blacklisting methods effectively block known malicious domains, they fall short against newly generated URLs that continue to emerge at an alarming rate.

This report addresses the challenge of detecting and classifying malicious URLs using a combination of machine learning (ML), deep learning (DL), and large language model (LLM)-based approaches. The primary objective is to develop a robust and accurate classification model that can distinguish between five types of malicious URLs: benign, defacement, phishing, malware, and spam.

II. Data Preprocessing and Feature Engineering

To achieve the goal of accurate malicious URL classification, several crucial steps were undertaken, starting with data preparation. The assignment required merging two datasets: one containing four classes and another contributing the fifth class (spam). After merging, the combined dataset was thoroughly preprocessed to ensure data quality and consistency.

A. Data Merging

- Two datasets were merged to form a comprehensive dataset for analysis.
- Appropriate handling of key fields was implemented to maintain data integrity.
- Labels were assigned to the combined dataset to facilitate multi-class classification.

B. Data Cleaning and Preprocessing

- Missing values were handled by removing incomplete rows.
- Duplicate entries were identified and eliminated to prevent data redundancy.
- The `type` column was encoded numerically for compatibility with ML models.

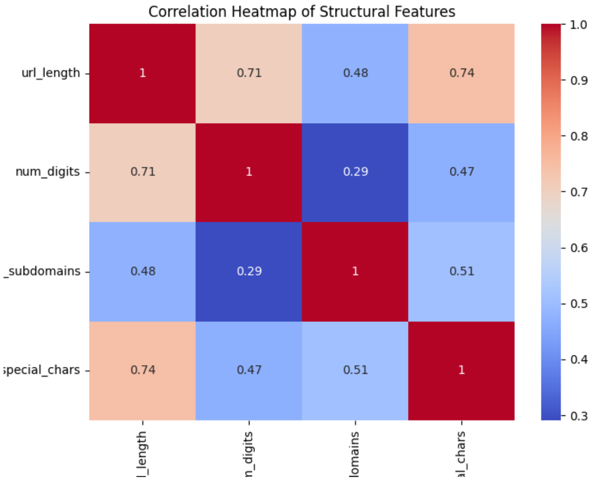

Fig. 1. Heatmap Classification of Dataset

C. Data Balancing

- A significant challenge encountered was class imbalance, which could negatively impact model performance.
- Techniques like undersampling, oversampling, and Synthetic Minority Over-sampling Technique (SMOTE) were utilized to address imbalance.
- The final balanced dataset ensured fair representation of each malicious category, significantly enhancing model generalization.

D. Feature Extraction

- Structural features were derived from URLs, including *URL length*, *number of digits*, *subdomain count*, and *special character frequency*.
- Textual features were extracted using *TF-IDF vectorization*, capturing the contextual meaning embedded in the URLs.
- Additional features involved character-level sequence embeddings to detect malicious patterns.

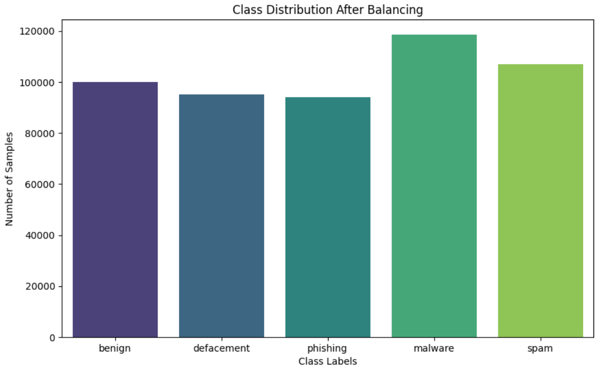

Fig. 2. Class Distribution Graph for EDA

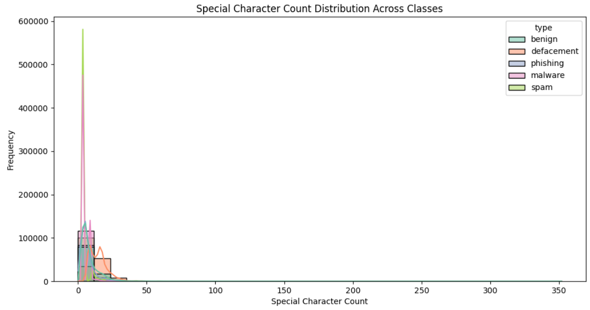

Fig. 3. Special Character Distribution and Count Across Dataset

E. *Exploratory Data Analysis (EDA)*

- Exploratory analysis was conducted to extract insights from the dataset.
- Visualizations were generated to understand class distributions and feature correlations.
- Patterns between URL structures, suspicious keywords, and malicious behavior were identified, guiding feature selection for model training.

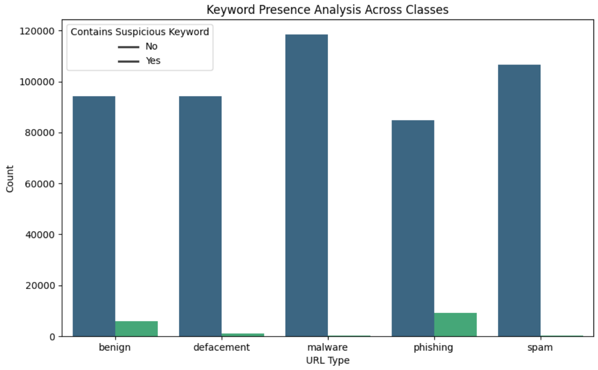

Fig. 4. Analysis of the Keywords Present Across the Dataset

III. Model Training and Performance Analysis

The dataset was trained and evaluated using three types of models: **Traditional Machine Learning models (Random Forest and XGBoost)**, **Deep Learning models (LSTM)**, and **Large Language Model (LLM) based models (DistilBERT)**. Each model was assessed based on its performance in terms of *Precision, Recall, F1-Score, and Accuracy*.

A. *Traditional Machine Learning Models*

The traditional machine learning models applied in this analysis were Random Forest and XGBoost, both of which utilized URL embeddings generated using BERT. Random Forest achieved an accuracy of 77%, with a macro average F1-score of 78% and a weighted average F1-score of 77%. This model performed well for most categories, but it encountered significant difficulties when classifying malware (Class 3) and spam (Class 4). The poor performance on these categories can be attributed to the high similarity in structural features between them, which resulted in misclassification.

XGBoost, on the other hand, demonstrated slightly superior performance compared to Random Forest. It achieved an accuracy of 78%, a macro average F1-score of 80%, and a weighted average F1-score of 78%. The enhanced performance of XGBoost can be primarily attributed to its robust ability to handle imbalanced data and capture complex patterns effectively. The model exhibited higher precision and recall for most categories, particularly benign, defacement, and phishing classes. However, similar to Random Forest, XGBoost struggled with classifying the malware and spam categories due to their overlapping structural characteristics. The improved performance of XGBoost over Random Forest is largely due to its boosting mechanism, which sequentially improves weak learners and provides better generalization capabilities.

B. *Deep Learning Model (LSTM)*

The Deep Learning model employed for this task was a Long Short-Term Memory (LSTM) network designed to capture sequential patterns present within URLs. The LSTM model achieved an accuracy of 84%, which is notably higher than the traditional models. This improvement in performance can be explained by the LSTM's ability to process sequential data effectively, allowing it to detect patterns within URLs that are more difficult to capture through traditional feature extraction techniques.

The model demonstrated an excellent ability to distinguish between structurally distinct URLs, such as phishing and benign categories. This success is largely due to the sequential nature of URLs, which can be effectively modeled by LSTM's recurrent architecture. However, despite its superior performance, the LSTM model still exhibited limitations in differentiating between highly similar categories like malware and spam, which shared overlapping structural features. Further optimization and the incorporation of additional contextual features may be necessary to improve its performance in these cases.

C. LLM-Based Model (DistilBERT)

The LLM-based model used in this analysis was DistilBERT, a distilled version of BERT optimized for efficiency and speed. The model achieved an accuracy of 80.4%, which is relatively high but still slightly lower than the performance of the LSTM model. DistilBERT demonstrated significant potential in capturing contextual patterns that are often missed by traditional feature extraction techniques.

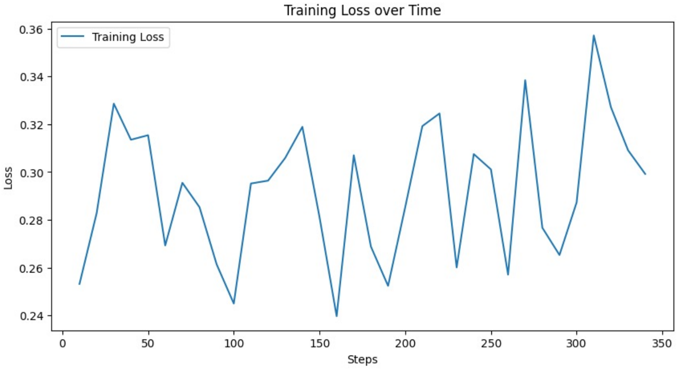

Fig. 5. Distil BERT Loss over Time Graphical Representation

The model performed particularly well when classifying defacement and phishing categories, where contextual understanding played a key role in distinguishing between them. However, it struggled to differentiate between malware and spam categories, likely due to structural similarities present in the dataset. The slightly lower accuracy of DistilBERT compared to the LSTM model can be attributed to its reduced complexity, which trades off some performance for improved speed and efficiency. Nonetheless, the model remains a viable option for URL classification, particularly when computational efficiency is a priority.

D. Model Comparison

A comparative analysis of the models reveals that the LSTM model achieved the highest accuracy of 84%, making it the most effective model for this task. Traditional models like Random Forest and XGBoost performed reasonably well, with XGBoost slightly outperforming Random Forest due to its ability to handle complex patterns and imbalanced data effectively. However, both traditional models failed to match the performance of the LSTM model.

The LLM-based approach using DistilBERT demonstrated considerable potential in identifying contextual patterns within URLs but was slightly outperformed by the LSTM model. This observation suggests that while LLM-based models can be useful for URL classification, their performance may be further enhanced by combining them with models that excel at capturing sequential data.

E. Challenges and Potential Improvements

Despite the success of the LSTM model, several challenges were encountered throughout the analysis. One of the primary challenges was class imbalance. Although techniques such as undersampling, oversampling, and SMOTE were employed to address this issue, the model still struggled to achieve high performance for the malware and spam categories. This limitation was further compounded by the similarity between the URL structures of these categories, making it difficult to distinguish between them accurately.

Traditional models relied heavily on structural features, which limited their ability to generalize effectively. Additionally, the training time required for LLM-based models was significantly higher due to the complexity of embeddings, which added computational overhead to the training process.

To address these challenges, several potential improvements can be proposed. First, a hybrid model that combines the strengths of LSTM and DistilBERT could enhance classification performance by leveraging the advantages of both models. Enhanced feature engineering that includes additional structural and contextual features may also improve performance, particularly for the malware and spam categories. Further, model optimization through fine-tuning hyperparameters and employing techniques such as transfer learning may contribute to improved accuracy. Additionally, data augmentation techniques that generate synthetic URLs with slight modifications could help enhance the model's generalization ability, particularly for categories where training data is limited.