Project Report

Project Title: Phishing website detection using NLP models
Group ID: 25HR07

1 Introduction

Phishing is one of the most persistent forms of cybercrime, exploiting human psychology to steal sensitive information or deliver malicious software. To address these limitations, our project proposes a **Natural Language Processing (NLP), driven machine learning framework** for proactive phishing detection. By combining linguistic, structural, and heuristic signals, the model can discern phishing intent even in newly generated URLs and web content.

2 Dataset and Preprocessing

The dataset was sourced from **Mendeley Data**, containing 80,000 websites, 30,000 phishing and 50,000 legitimate. Each entry includes both the URL and its HTML DOM. All HTML files were parsed using *BeautifulSoup* and serialized into pickle objects for efficient reuse. Preprocessing ensured uniformity across models dealing with character-level, feature-level, and semantic representations.

3 Methodology

3.1 URL-Based Features

This stage focused on extracting both deep and heuristic patterns from website URLs, producing a total of 221 features per entry.

3.1.1 Heuristic Features

Twenty-one rule-based features captured common phishing tricks:

- **Length-based:** URL length, hostname length, path length, TLD length.
- **Character-count:** Counts of hyphens, digits, dots, and special symbols.
- **Boolean flags:** Indicators such as presence of IP address, HTTPS usage, or URL shortening.
- **Keyword and obfuscation cues:** Suspicious terms like "login", "secure", or abnormal subdomains.

3.1.2 Character-Sequence Features

URLs were represented as fixed-length 200-character sequences using a vocabulary of 78 characters tokenized via *Keras Tokenizer*. Sequences were padded or truncated to shape $[1 \times 200]$ and fed into a **Convolutional Neural Network (CNN)** to capture structural and lexical cues.

3.2 Hyperlink-Based Features

The second component analyzed hyperlink-related characteristics derived from HTML documents. Extracted features included:

- Counts of internal/external links, broken links, and CSS or favicon references.
- Ratio-based metrics (e.g., percentage of internal vs. external links) to normalize by page size.

A **Random Forest (RF)** classifier was trained, achieving 91.2% accuracy and high interpretability.

3.3 HTML Content and Textual Features

The third model captured the semantic meaning from webpage text. The `<title>` and `<body>` tags were cleaned via lowercasing, tokenization, stopword removal, and lemmatization. Pre-trained **GloVe.6B.100d** embeddings mapped words into 100-dimensional vectors. The processed title and body text were padded to 10 and 100 tokens respectively, then passed through a **Bidirectional GRU (BiGRU)** network. This model achieved 97.8% accuracy, outperforming other RNNs. Interestingly, BiLSTM and LSTM produced identical accuracy, implying primarily forward or local dependencies in the text.

3.4 Hybrid CNN Model

The final architecture integrated the CNN-based URL model, heuristic features, and hyperlink-based Random Forest outputs. Implemented in **PyTorch**, it had two primary heads:

- **Sequential Head:** 200-character embedded sequences through CNN layers.
- **Heuristic Head:** 34 tabular features (21 lexical + 13 hyperlink) through a fully connected network.

Outputs were concatenated and passed through a fusion layer, achieving **96.32% accuracy**.

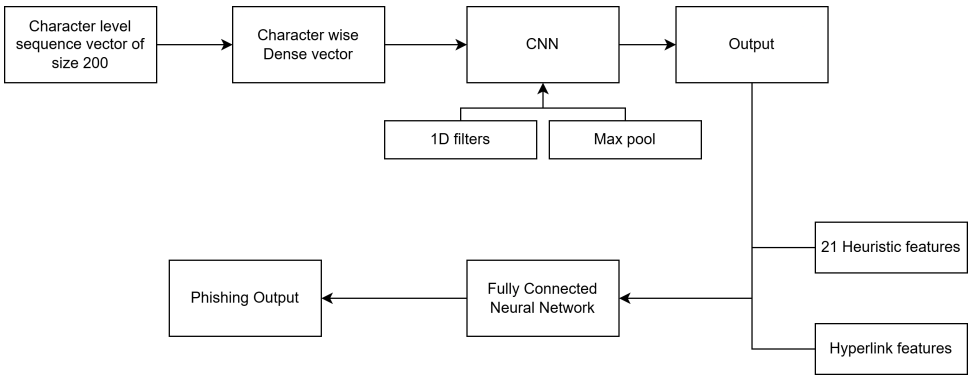

Figure 1: Hybrid CNN Model

4 Results and Discussions

URL-Based Models		Hyperlink-Based Models		Content-Based Models	
Model	**Accuracy**	**Model**	**Accuracy**	**Model**	**Accuracy**
Logistic Reg.	84.74	Random Forest	91.24	LSTM	97.75
Random Forest	91.44	C4.5 Decision Tree	90.00	BiLSTM	97.75
XGBoost	92.44	Neural Net (MLP)	87.20	GRU	97.88
SVM (RBF)	88.22	Adaboost	85.93	BiGRU	97.88

Table 1: Comparison of Model Accuracies Across Feature Categories

The URL-based CNN + Heuristic model achieved an accuracy of 95%, while the Hybrid CNN with URL and hyperlink features gave an accuracy of 96.3 %. The identical performance of BiLSTM and LSTM implies that phishing indicators are mostly local rather than requiring backward context.

5 Conclusion

We successfully developed and evaluated a hybrid model that synthesizes features from multiple paradigms: deep-language patterns from URL strings, structural features from hyperlink analysis, and content based features. Our findings demonstrate the exceptional power of modern machine learning techniques. A BiGRU model analyzing just the HTML text content achieved a 97.8% accuracy, while a 1D-CNN operating on URL character sequences alone reached 95%. Our final hybrid model, which combined the URL-CNN with the hyperlink and heuristic features, achieved a peak accuracy of 96.32%, confirming the effectiveness of our multi-paradigm approach. The most significant insight from our work is the apparent feature redundancy. The standalone performance of the BiGRU and CNN models suggests that these deep learning architectures are implicitly learning the same complex patterns that we sought to capture manually with heuristic and structural features.