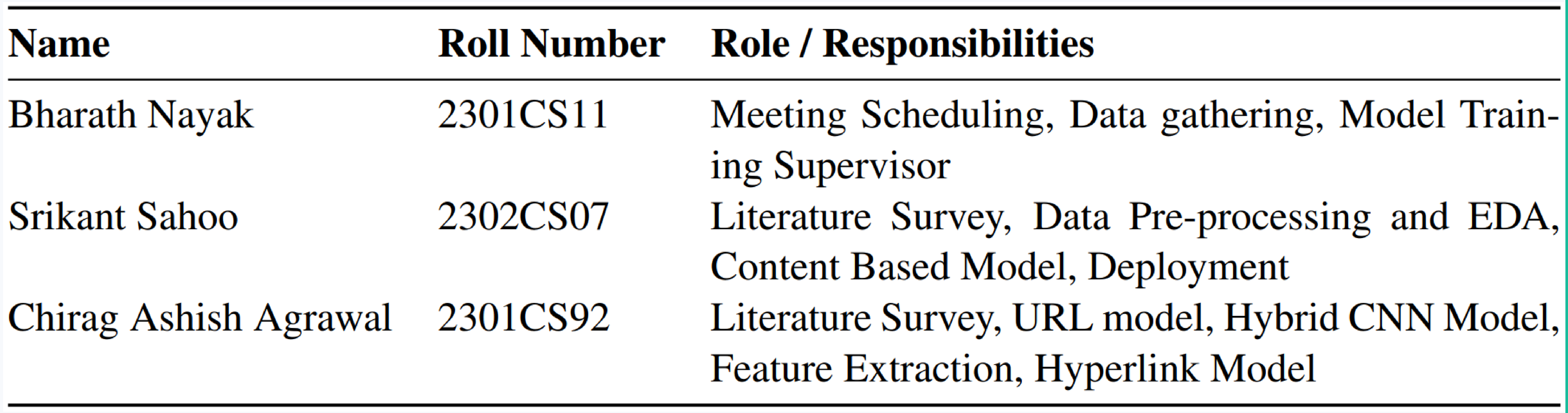
Phishing Website Detection Using NLP Techniques

Group ID : 25HR07

Roles and Responsibilities

Team Members and Their Tasks



Name	Roll Number	Role / Responsibilities
Bharath Nayak	2301CS11	Meeting Scheduling, Data gathering, Model Training Supervisor
Srikant Sahoo	2302CS07	Literature Survey, Data Pre-processing and EDA, Content Based Model, Deployment
Chirag Ashish Agrawal	2301CS92	Literature Survey, URL model, Hybrid CNN Model, Feature Extraction, Hyperlink Model

Abstract

Overview of our project findings

Phishing websites remain a major security risk, requiring accurate automated detection. We explore multi-level webpage features to improve classification: (1) **URL-level features**, where we use **200-character embedded URL sequences** along with **21 heuristic phishing indicators**, achieving **95 percent accuracy** using a **two-head CNN with fully connected layers**; (2) **structural hyperlink features** derived from webpage HTML, modeled using traditional machine learning methods, reaching **91 percent accuracy**; (3) a hybrid fusion model that combines URL embeddings, heuristic features, and hyperlink structure, improving performance to **96 percent accuracy**; and (4) full **HTML DOM content embeddings**, where sequence models including LSTM, BiLSTM, GRU, and BiGRU are evaluated, with **BiGRU** achieving the highest accuracy of **97 percent**. These results demonstrate that integrating linguistic, structural, and content-level webpage features significantly strengthens phishing detection performance.

Introduction

Problem: Phishing is a social engineering attack to steal sensitive information or deliver malware.It is widespread and causes billions of dollars in financial loss annually.

Limitations: Blacklists and signature-based systems are reactive. They fail to detect new or evolving phishing websites.

Research Gap: Most existing detection models rely mainly on only one type of feature (URL or HTML or hyperlink structure). This limits accuracy because phishing signals appear at multiple levels of the webpage.

Goal: Develop a proactive detection model that integrates: 1.URL features 2.Structural hyperlink features 3.Full HTML content feature-set to achieve higher accuracy and robustness.

Scope: This project focuses on combining multiple heterogeneous feature sets for stronger detection.

Literature Review

Key studies and findings on phishing detection techniques.

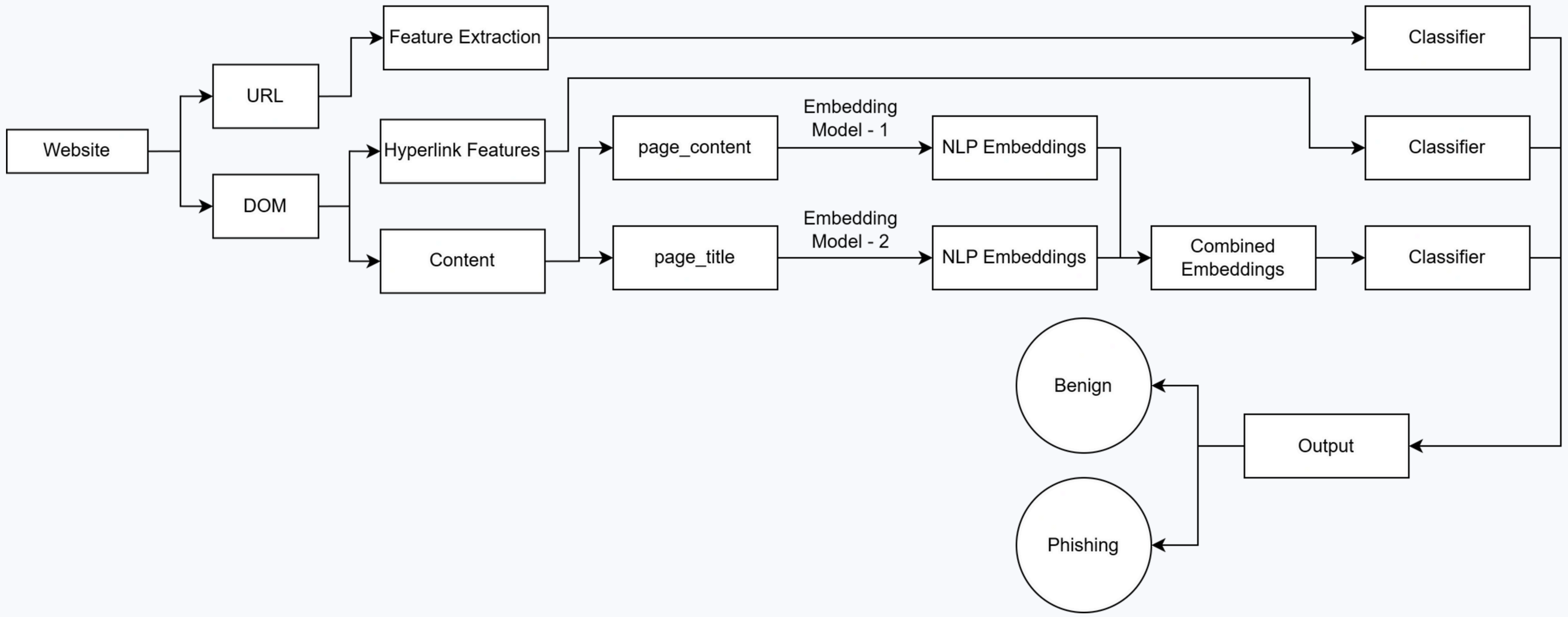
1. Kalla and Kuraku (2023) compare seven machine learning models, including Logistic Regression, Random Forest, and Linear SVC, to detect phishing URLs, evaluating their accuracy, precision, and recall.
2. Rao et al. (2022) propose a phishing detection model using various word embeddings (like TF-IDF, Word2Vec, and FastText) on website source code with classifiers like RF, SVM, XGBoost.
3. Aljofey et al. (2022) propose a phishing detection model using a hybrid feature set of URL character sequences, HTML hyperlink information, and text content (TF-IDF), with an XGBoost classifier.
4. Jain and Gupta (2018) propose a client-side phishing detection model by applying machine learning to 12 novel features extracted exclusively from HTML hyperlinks.

Methodology

Data Collection

For this study, we utilized a public dataset from Mendeley Data (source), which contains the URLs and HTML DOM for 80,000 websites. This dataset is composed of 30,000 phishing and 50,000 legitimate sites. To optimize our feature extraction pipeline, we pre-processed the entire 80,000-site corpus by parsing each HTML file into a Beautiful Soup object and serializing these objects into pickle files for efficient, repeatable access.

Architecture



Methodology

URL–Based Features

This phase focused on extracting features solely from the URL string. The data was processed to generate a total of 221 features for each URL.

a. Heuristic Features

These are human-readable, rule-based features designed to capture common phishing "tricks”. The 21 features included:

Length-Based: url_length, hostname_length, path_length, tld_length.

Character-Count: count_hyphen, count_at, count_dot, count_digits, etc.

Boolean Flags: is_ip_address, is_https, is_shortened.

Keyword-Based: has_suspicious_keyword (e.g., "login", "secure", "paypal").

Obfuscation-Based: tld_in_subdomain, abnormal_subdomain.

b. Character-Sequence Features

This feature set was designed for deep learning models.

A fixed vocabulary of 78 characters (a-z, A-Z, 0-9, and symbols) was defined.

A Keras Tokenizer was used to map each character to a unique integer index.

Each URL string was converted into a sequence of integers.

The sequences were then padded (with 0s) or truncated to a fixed length of 200, resulting in a **[1 x 200] vector for every URL**.

Methodology

Hyperlink-Based Features

This phase involved parsing the HTML of a webpage to extract features related to its outgoing links and resource requests.

a. Feature Extraction

The raw features extracted included:
total_links, internal_links, external_links, null_links, internal_css, external_css, internal_errors, external_errors, internal_favicon, external_favicon

b. Preprocessing & Feature Engineering

Two experiments were conducted:

1. **Raw Counts**: Using the raw counts (e.g., 10 internal, 50 external).
2. **Percentage-Based**: New features like pct_internal_links (internal_links / (internal_links + external_links)) were engineered to provide context and prevent the model from being skewed by large, legitimate sites.

Both experiments resulted into similar results.

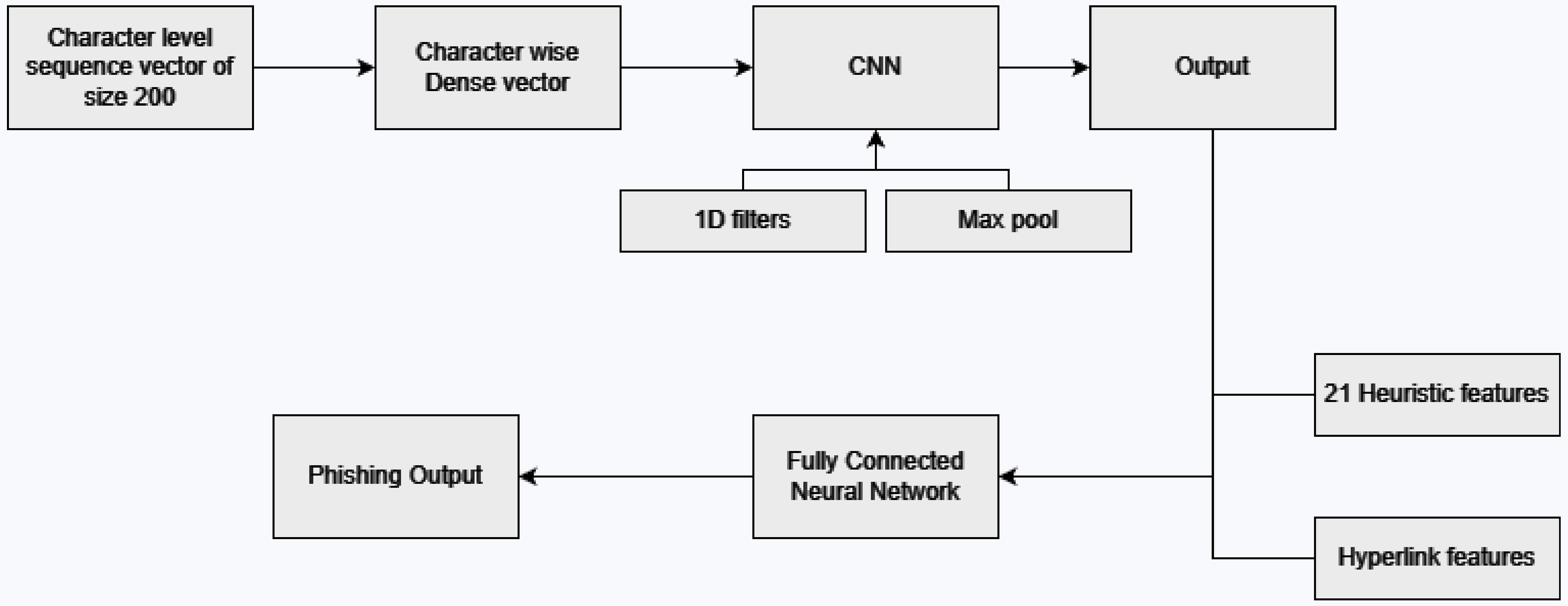
Hybrid CNN model for URL and Hyperlink features

Model Architecture (PyTorch): A "two-headed" hybrid model was built.

Head 1 (Sequential Head): Took the 200-char-sequence vector (c_1...c_200) and fed it into an Embedding layer followed by either a 1D-CNN or a GRU layer. This is isolation gave an accuracy of 95 %.

Head 2 (Heuristic Head): Took all other 34 heuristic features (21 lexical + 13 hyperlink) and fed them into a standard multi-layer ANN (FC) head along with the result from convolutional layers.

This model achieves an accuracy of 96.32 %.


Character level sequence vector of size 200
Character wise Dense vector
CNN
Output
1D filters
Max pool
21 Heuristic features
Hyperlink features
Fully Connected Neural Network
Phishing Output

Methodology

HTML Content (DOM) Features

This phase was designed to extract semantic meaning from the page's text. The HTML <body> and <title> tags were parsed using BeautifulSoup.

a. Text Cleaning and Preprocessing

The raw text was heavily preprocessed:

1. Converted to **lowercase**.
2. Removed all URLs, mentions, hashtags, and non-alphabetic characters using regex.
3. **Tokenized** (split into individual words).
4. **Stopword Removal**: Common English stopwords (e.g., "the", "a", "is") were removed.
5. **Lemmatization**: To reduce words to their root form, a WordNetLemmatizer was used. This was enhanced with POS-tagging (Part-of-Speech).
6. **Vectorization (Tokenization)**: A Keras Tokenizer was fit on the entire corpus (all titles and content) to build a master vocabulary.

b. Embedding

1. **Pre-trained GloVe embeddings** (glove.6B.100d.txt) were loaded.
2. An **embedding matrix** was built, mapping words in the vocabulary to their 100-dimensional GloVe vectors.
3. The page_title text was converted to a 10-word integer sequence and the page_content to a 100-word integer sequence.

Results

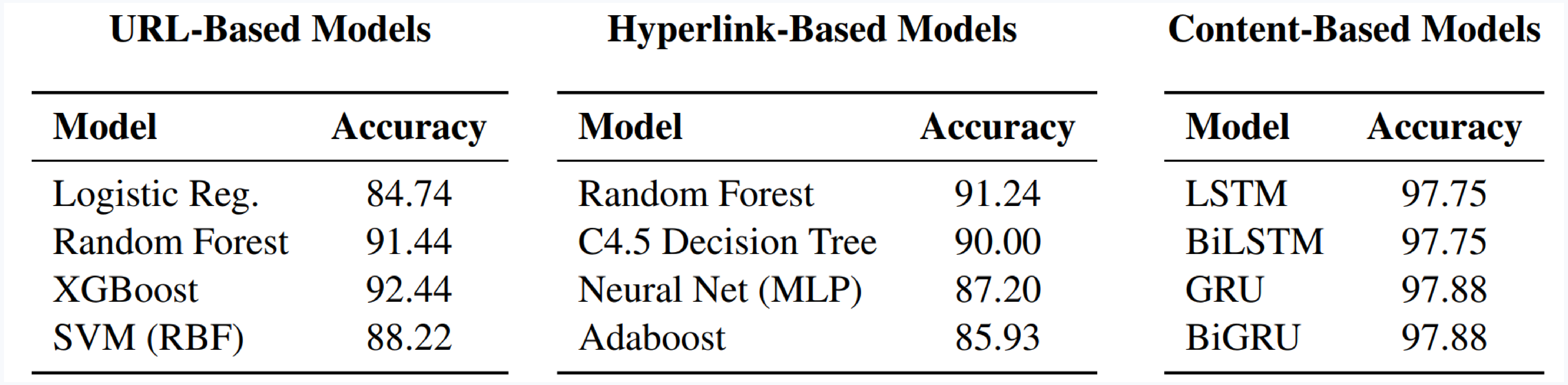
Model Evaluation and Comparison

We experimented with a variety of classical and deep learning architectures across different feature domains.

For URL-based analysis, models focused on lexical and structural characteristics extracted from web addresses.

Hyperlink-based models analyzed the distribution and nature of internal and external links within webpages to capture behavioral cues.

Content-based models leveraged natural language representations of webpage text and titles to understand semantic intent.



URL-Based Models

Model	Accuracy
Logistic Reg.	84.74
Random Forest	91.44
XGBoost	92.44
SVM (RBF)	88.22

Hyperlink-Based Models

Model	Accuracy
Random Forest	91.24
C4.5 Decision Tree	90.00
Neural Net (MLP)	87.20
Adaboost	85.93

Content-Based Models

Model	Accuracy
LSTM	97.75
BiLSTM	97.75
GRU	97.88
BiGRU	97.88

Results

Comprehensive findings on phishing detection accuracy presented

URL-Based Models

The 221 URL features were used to train 5 classic ML models. XGBoost was the top performer with 92.4% accuracy and the lowest false negatives (674), outperforming Random Forest (91.4%) and SVC (88.2%).

Hyperlink-Based Models

An 8-model comparison was run on the hyperlink features (without percentages). Random Forest was the clear winner with 91.2% accuracy and an F1-Score of 88.88%. Naive Bayes performed very poorly (57.5% accuracy), indicating the features are not conditionally independent. A separate deep learning model (a simple ANN) was trained on the features with percentages. This achieved 87.2% accuracy.

HTML Content-Based Models

A comparison on the 110-feature concatenated vectors showed that Random Forest was the top performer with an accuracy of 93.7 % accuracy and an F1-Score of 91.7 %.

Four “two-headed” RNNs were built (BiLSTM, LSTM, GRU, BiGRU) using keras. A shared Embedding layer (using the GloVe matrix) fed into two parallel RNNs, which were then concatenated and passed to a final classifier. All models performed well, BiGRU giving the highest accuracy of 97.8 %. Surprisingly, BiLSTM and LSTM performed exactly the same which implies that the task is dominated by local or forward dependencies

Conclusion

Our approach's impact on phishing detection.

This project successfully designed, implemented, and evaluated several high-performance models for phishing detection. We demonstrated that by using modern NLP and machine learning techniques, it is possible to proactively detect phishing sites with high accuracy, surpassing the limitations of traditional blacklist-based systems.

A key finding of this report is the remarkable standalone performance of deep learning models. The Content-Based BiGRU (97.8%) and the URL-Based CNN (95%) both performed at a level comparable to, or even exceeding, the final hybrid model (96.32%). 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.

References

1. A. Aleroud and L. Zhou, “Phishing environments, techniques, and countermeasures: A survey,” Computers amp; Security, vol. 68, p. 160–196, Jul. 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2017.04.006
2. D. Kalla and S. Kuraku, “Phishing website url’s detection using nlp and machine learning techniques,” Journal on Artificial Intelligence, vol. 5, no. 0, p. 145–162, 2023. [Online]. Available: http://dx.doi.org/10.32604/jai.2023.043366
3. R. S. Rao, A. Umarekar, and A. R. Pais, “Application of word embedding and machine learning in detecting phishing websites,” Telecommunication Systems, vol. 79, no. 1, p. 33–45, Nov. 2021. [Online]. Available: http://dx.doi.org/10.1007/s11235-021-00850-6
4. A. Aljofey, Q. Jiang, A. Rasool, H. Chen, W. Liu, Q. Qu, and Y. Wang, “An effective detection approach for phishing websites using url and html features,” Scientific Reports, vol. 12, no. 1, May 2022. [Online]. Available: http://dx.doi.org/10.1038/s41598-022-10841-5
5. . K. Jain and B. B. Gupta, “A machine learning based approach for phishing detection using hyperlinks information,” Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, p. 2015–2028, Apr. 2018. [Online]. Available: http://dx.doi.org/10.1007/s12652-018-0798-z