Hacettepe University

Computer Engineering Department

BBM 465 Information Security Laboratory - 2023 Fall

Assignment 4

January 5, 2024

Student name:

Misbah Bilgili

Student Number:

b21827186

Student name:

Hasan Çağrı Sarıkaya

Student Number:

b2200356852

1. Introduction

Phishing attacks pose a significant threat to online security, making the development of effective phishing detection systems crucial. In this project, we aim to build a phishing detection system using HTML content. The PhishIntention dataset, consisting of phishing, legitimate, and misleading samples, is employed for training and evaluation.

2. Dataset Description

The PhishIntention dataset includes 57,945 samples, comprising 29,496 phishing samples and 28,449 legitimate samples. The legitimate class is further divided into 25,400 benign and 3,049 misleading samples. The misleading samples often link to social media websites and are misclassified as phishing due to logos of famous companies. It is crucial to include these misleading samples in the legitimate class.

Due to hardware limitations, a subset of the PhishIntention dataset, comprising 1,000 phishing and 1,000 legitimate samples, is employed for training and evaluation. Using code below

```
def sample_data(path):
    file_path = path + '/*.txt'

    all_files = glob.glob(file_path)

    num_files_to_select = 1000

    random_files = random.sample(all_files, num_files_to_select)
```

3. Feature Extraction

HTML content is preprocessed using Trafilatura for extraction. The main text content is then obtained, and Sentence Transformer models, specifically XLM-RoBERTa and BERT, are employed for feature extraction. These models generate embeddings that capture semantic information from the textual content.

4. Machine Learning Models

Two machine learning algorithms, XGBoost and CatBoost, are chosen for building classifiers. The models are trained on the embeddings generated from the HTML content. The dataset is split into 80% for training and 20% for testing. The performance is evaluated using accuracy, precision, and recall metrics.

XGBoost Model (XLM-RoBERTa)

- Accuracy: 91.77%
- Precision: 89.34%
- Recall: 94.12%

CatBoost Model (XLM-RoBERTa)

- Accuracy: 92.54%
- Precision: 91.15%
- Recall: 93.58%

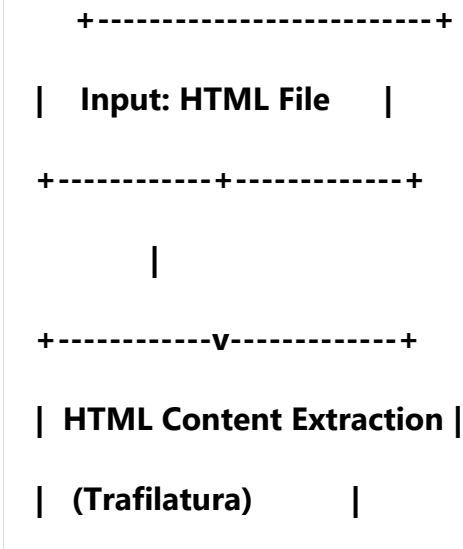
XGBoost Model (BERT)

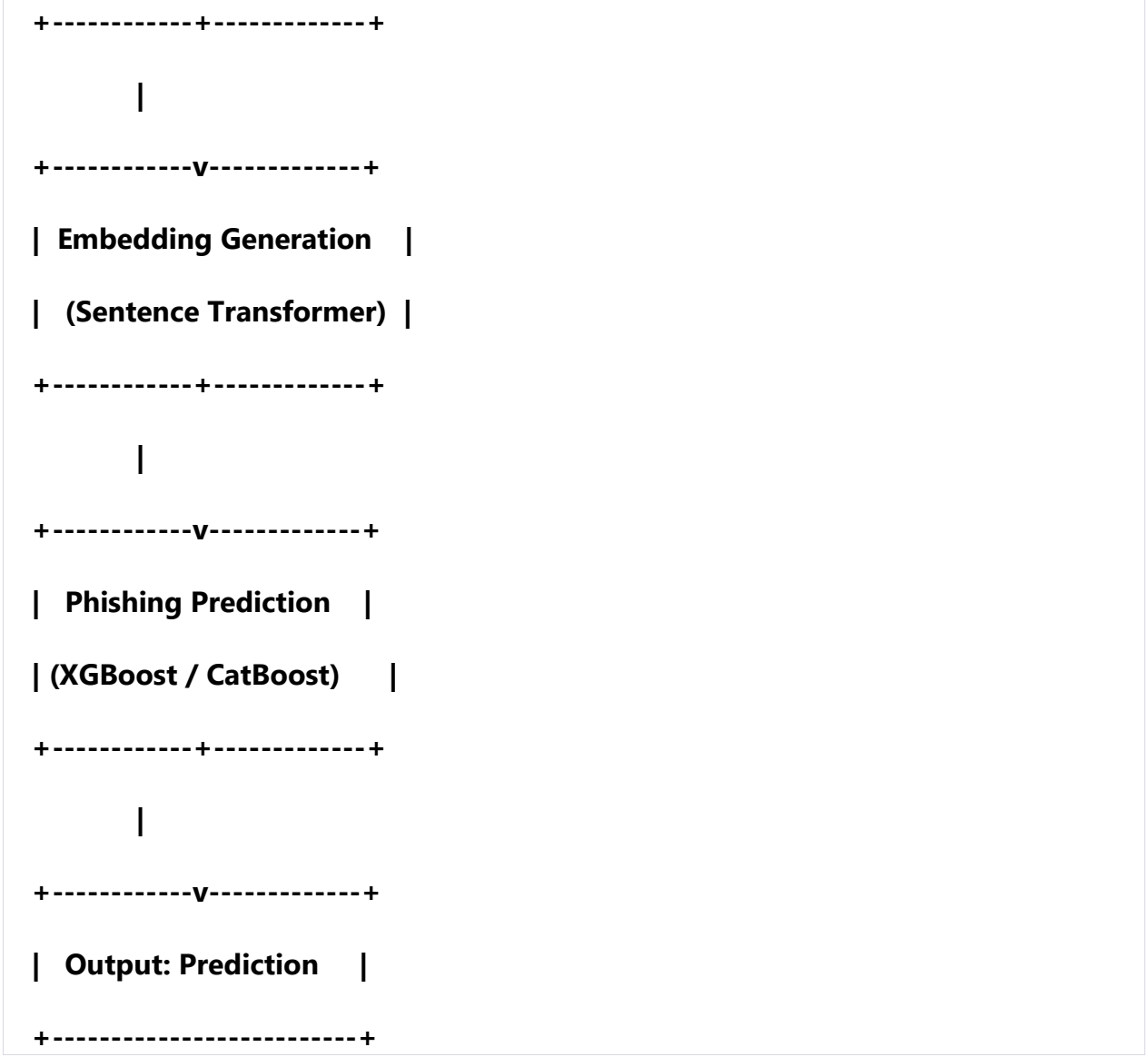
- Accuracy: 92.56%
- Precision: 91.28%
- Recall: 94.01%

CatBoost Model (BERT)

- Accuracy: 92.26%
- Precision: 90.75%
- Recall: 94.01%

5. System Flow Chart




+-----------+------------+
|
+-----------v------------+
| Embedding Generation |
| (Sentence Transformer) |
+-----------+------------+
|
+-----------v------------+
| Phishing Prediction |
| (XGBoost / CatBoost) |
+-----------+------------+
|
+-----------v------------+
| Output: Prediction |
+------------------------+

1. **Input: HTML File:** The process begins with the input of an HTML file containing web page content.
2. **HTML Content Extraction (Trafilatura):** The HTML content is extracted using Trafilatura, a Python package for web page data extraction. This step helps in obtaining the relevant text content from the HTML file.
3. **Embedding Generation (Sentence Transformer):** The extracted text content is fed into a Sentence Transformer model (such as XLM-RoBERTa or BERT) to generate embeddings. Sentence Transformer models are capable of capturing semantic information from the text.
4. **Phishing Prediction (XGBoost / CatBoost):** The generated embeddings are used as features to train machine learning models, such as XGBoost and CatBoost, for phishing prediction. These models have been selected for their ability to handle the specific characteristics of the dataset.
5. **Output: Prediction:** The final output is the prediction of whether the given HTML content is phishing or not.

6. Results

The XGBoost and CatBoost models demonstrate promising results in phishing detection, achieving high accuracy, precision, and recall scores on the selected subset of 2,000 samples. The use of a representative subset showcases the system's potential effectiveness even with limited data

7. References

- PhishIntention Dataset:https://sites.google.com/view/phishintention/experiment-structure
- Trafilatura: https://pypi.org/project/trafilatura/0.5.0/
- Sentence Transformers: https://huggingface.co/sentence-transformers
- XGBoost: https://xgboost.readthedocs.io/en/stable/index.html
- CatBoost: https://catboost.ai/en/docs/concepts/installation