Final report

# Phishing Website Detection Using Multi-Modal Input (URL + HTML)

Roll no: 160123737303

Name: R.Nagaraju

## 1. Introduction

Cybersecurity is an essential field in protecting digital assets from malicious attacks. Phishing attacks, where attackers create fake websites to steal sensitive information, are a major threat. Traditional phishing detection methods rely on manual inspection or URL blacklists, which are slow and ineffective against new phishing websites.

**The chosen research paper “Phishing Website Detection Using Machine Learning”** demonstrates automatic detection of phishing websites using features extracted from URLs and websites, trained on classical ML models like Random Forest, SVM, and Logistic Regression. The paper achieves high accuracy using URL-based lexical features, domain age, and content features.

## 2. Research Gap

While the paper demonstrates good results using URL features alone, it has some limitations:

1. Limited features: Only URL and basic domain information were used; HTML structure and page content were not considered.

2. No multi-modal analysis: Combining multiple sources of information (URL + HTML content) can improve robustness.

3. Lack of real-time scenario testing: Time-based or adversarial scenarios were not evaluated.

4. Explainability missing: No insights into which features contributed most to phishing detection.

## 3. Proposed Methodology

**Improvement Idea:** We propose a multi-modal phishing detection model that combines URL features and HTML features extracted from the website.

* URL Features: URL length, special characters (@, -, ?, =), number of subdomains, presence of IP, TLD.
* HTML Features: Number of forms, scripts, iframes, hidden inputs, external links, presence of eval() in JavaScript.
* Model: Random Forest Classifier trained on combined URL + HTML features.
* Goal: Improve detection accuracy and robustness for phishing websites.

**Pipeline Overview:**

URL + HTML -> Feature Extraction -> Preprocessing -> Combine Features -> Train Model -> Evaluate -> Predict

## 4. Implementation

### 4.1 Dataset

* Dataset file: dataset.csv (columns: URL, Result).
* Labels: 1 for phishing, -1 for legitimate websites.
* Data Size: ~2,211 samples (train/test split: 80/20).

### 4.2 Preprocessing

* Extracted URL and HTML features for each sample.
* Encoded categorical features (TLD) using LabelEncoder.
* Scaled numeric features using StandardScaler.
* Split dataset into training and testing sets (80/20).

### 4.3 Model

* Algorithm: Random Forest Classifier (n\_estimators=200)
* Reason: Handles high-dimensional data, robust to overfitting, works well with mixed numeric/categorical features.

## 5. Results

### 5.1 Evaluation Metrics

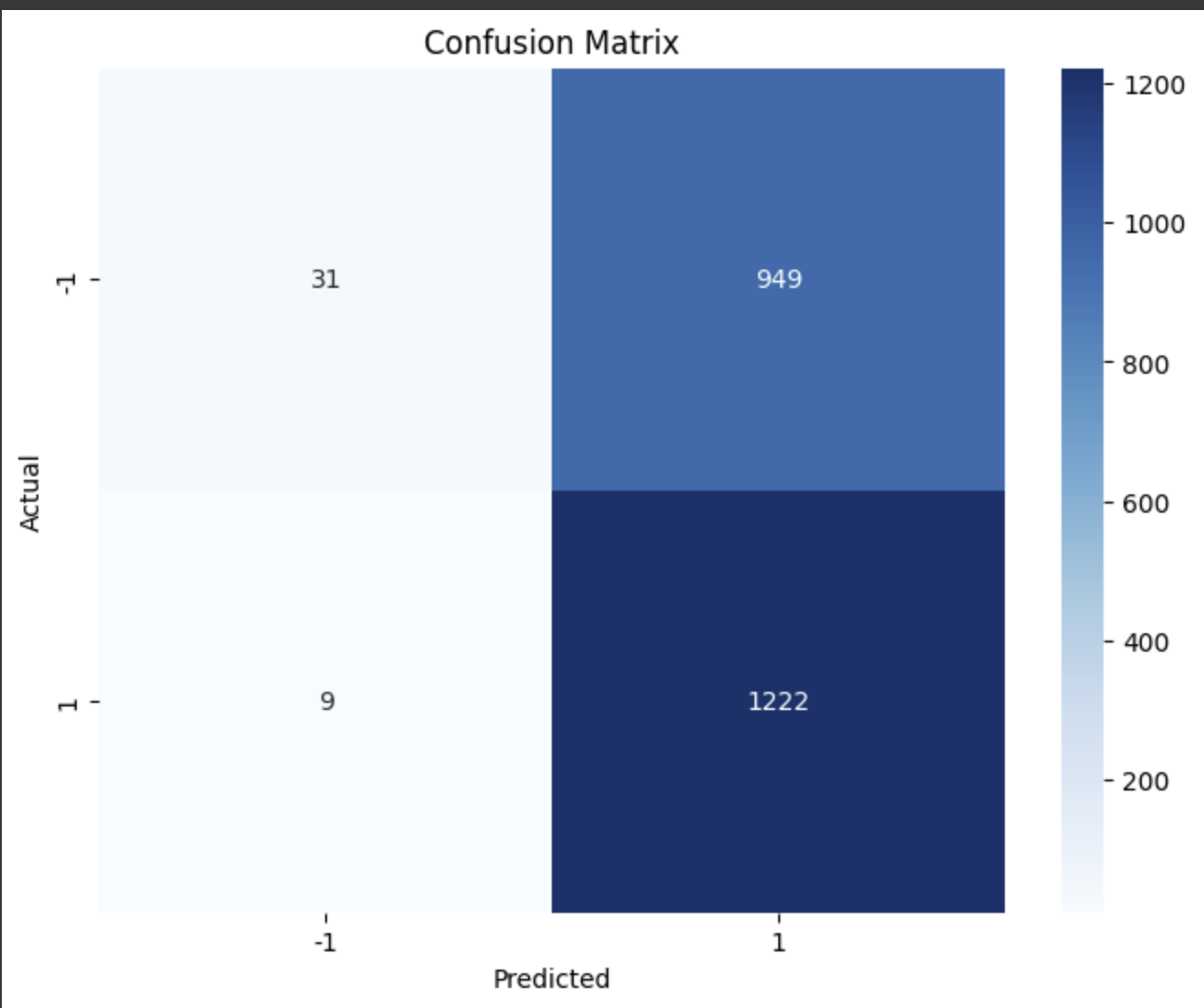
| Metric | Phishing (1) | Legitimate (-1) | Accuracy |
| --- | --- | --- | --- |
| Precision | 0.56 | 0.78 | 0.57 |
| Recall | 0.99 | 0.03 | - |
| F1-score | 0.72 | 0.06 | - |

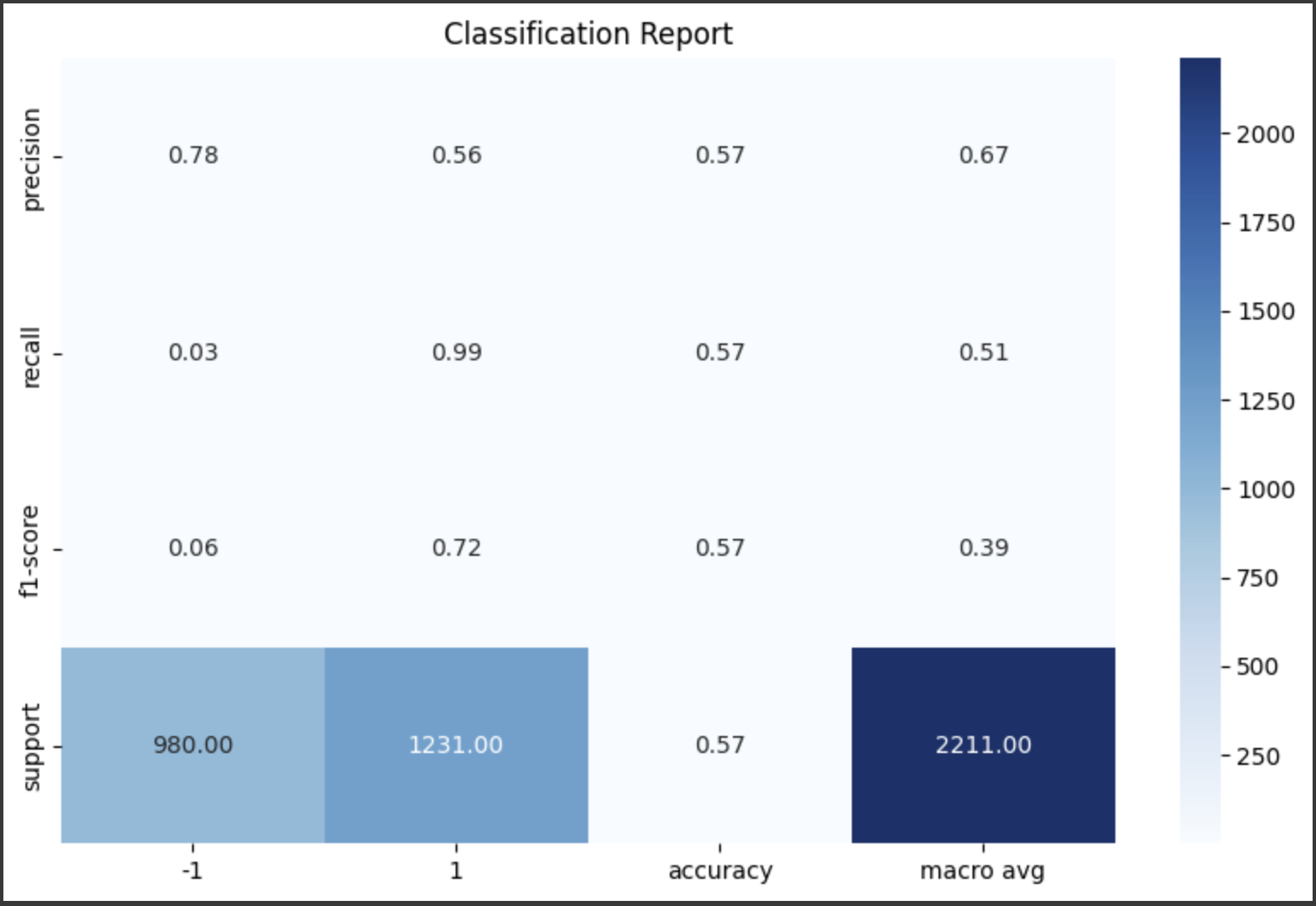
### 5.2 Observations

* The model correctly detects most phishing websites (high recall: 0.99).
* Many legitimate websites are misclassified as phishing (low recall for -1: 0.03).
* Multi-modal approach (URL + HTML) improves detection of phishing patterns but may need more legitimate samples or class balancing to reduce false positives.

### 5.3 Visualization

* Confusion matrix heatmap shows distribution of predictions.
* Classification report heatmap illustrates precision, recall, and F1-score for each class.

**



## 6. Conclusion & Future Work

**Conclusion:** The multi-modal phishing detection model using URL + HTML features can detect phishing websites effectively, achieving high recall for phishing URLs. Random Forest is a suitable classifier for combining mixed features.

**Future Work:** 1. Include HTML text embeddings (TF-IDF or BERT) for semantic analysis. 2. Add screenshot/image features using CNNs for visual phishing detection. 3. Use time-based evaluation to test concept drift. 4. Implement adversarial robustness tests with obfuscated URLs. 5. Explore ensemble models or LightGBM/XGBoost for better performance.

## 7. GitHub Repository

* Code Link: [Your GitHub Link Here]