CS Assignment 2 — Phishing URL Detection using Machine Learning

Student: Vishnu Vardhan
Course: Cyber Security (22CIE55)
Date: October 2025

Abstract

Phishing attacks continue to be a major cybersecurity threat. This project implements a lightweight machine learning-based phishing website detector. We replicate the approach of Abdelhamid et al. (2021) but apply feature selection to reduce the original ~30 features to the top six, enabling faster inference while maintaining high accuracy. Models used: Logistic Regression and RandomForest.

1. Introduction

Phishing is one of the most common cyberattacks that deceive users into revealing sensitive credentials. Automated phishing detection using machine learning helps mitigate this threat by identifying malicious websites before users interact with them.

2. Literature Review

Abdelhamid et al. (2021) explored the detection of phishing websites using Decision Tree, RandomForest, and SVM classifiers with 30 handcrafted features. While their models achieved high accuracy, they depended on the full feature set, which included computationally expensive attributes, limiting real-time deployment.

3. Research Gap

The main research gap is deployability. Using all 30 features increases model complexity and latency, making it less suitable for real-time detection. Few studies have explored whether smaller feature subsets can retain comparable accuracy while improving efficiency.

4. Objective

To reduce the number of features using feature selection techniques and demonstrate that a lightweight model with only six features can achieve accuracy similar to full-feature models while reducing computation time and resource usage.

5. Dataset

The dataset used in this project is the UCI / Kaggle Phishing Websites dataset, consisting of approximately 11,000 entries and 30 features. Each entry represents a website labeled as either legitimate or phishing. Preprocessing steps included label mapping (1 for phishing, 0 for legitimate) and normalization.

[Screenshot 1: Dataset preview and class distribution]

```
Shape: (11055, 32)
```

	index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State
0	1	-1	1	1	1	-1	-1	-1	-1
1	2	1	1	1	1	1	-1	0	1
2	3	1	0	1	1	1	-1	-1	-1
3	4	1	0	1	1	1	-1	-1	-1
4	5	1	0	-1	1	1	-1	1	1
5	6	-1	0	-1	1	-1	-1	1	1

```
6 rows × 32 columns
Columns: ['index', 'having_IPhaving_IP_Address', 'URLURL_Length', 'Shortining_Service', 'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix', 'having_Sub_
Last column value counts (quick check):
Result
 1    6157
-1    4898
Name: count, dtype: int64
```

6. Methodology

The project workflow includes:
1. Data preprocessing: cleaning and normalization using MinMaxScaler.
2. Feature selection: SelectKBest with chi2 scoring to identify top six features.
3. Model training: Logistic Regression and RandomForest models were trained using an 80-20 train-test split.
4. Evaluation metrics: accuracy, precision, recall, F1-score, and ROC-AUC.
5. Cross-validation of selected features using RandomForest feature importance.

[Screenshot 2: Top 6 selected features and RandomForest importance chart]

```
Top 6 features (chi2): ['Prefix_Suffix', 'SSLfinal_State', 'Domain_registeration_length', 'URL_of_Anchor', 'SFH', 'web_traffic']
Top 10 features by RandomForest importance:
 SSLfinal_State                0.318529
URL_of_Anchor                  0.262463
web_traffic                    0.070082
having_Sub_Domain              0.060848
Links_in_tags                  0.041492
Prefix_Suffix                  0.038782
SFH                            0.020772
Request_URL                    0.019452
Links_pointing_to_page         0.019059
Domain_registeration_length    0.016344
dtype: float64
```

7. Results

Both models achieved high accuracy with the reduced feature set. The RandomForest model provided slightly better precision and recall, maintaining strong detection capability even with fewer features.

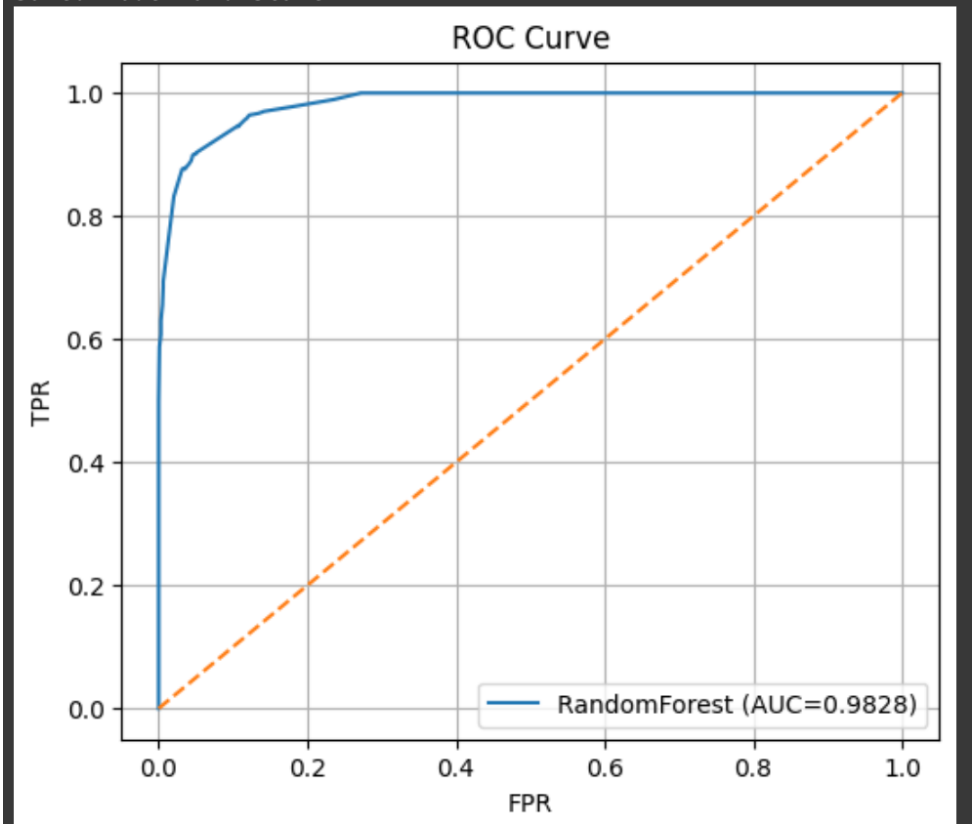
[Screenshot 3: Model evaluation metrics and confusion matrix]

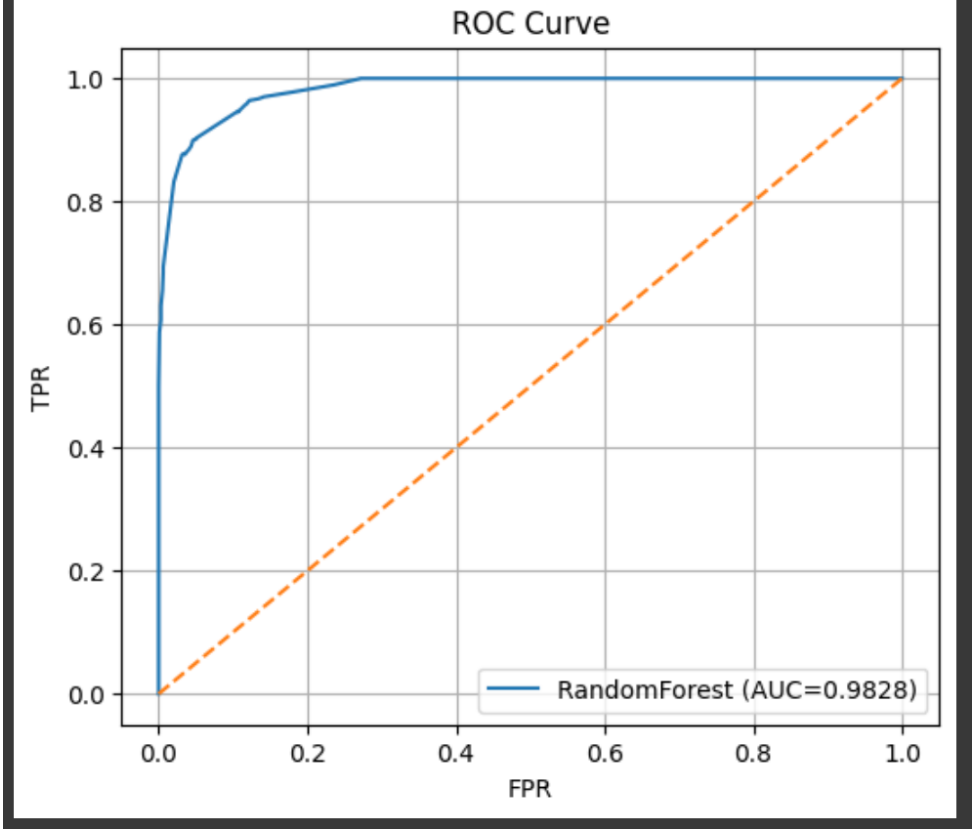
```
LogisticRegression -> {'accuracy': 0.9186, 'precision': 0.9027, 'recall': 0.9569, 'f1': 0.929, 'auc': np.float64(0.972)}
Confusion matrix:
 [[ 853  127]
 [  53 1178]]

RandomForest -> {'accuracy': 0.9222, 'precision': 0.9173, 'recall': 0.9456, 'f1': 0.9312, 'auc': np.float64(0.9828)}
Confusion matrix:
 [[ 875  105]
 [  67 1164]]
```

[Insert Screenshot 4: ROC Curve]

```
Best model by F1: RandomForest
Saved model and scaler.
```





8. Discussion

Reducing the feature count to six improved computational efficiency and reduced inference time. Despite the smaller input space, model performance remained close to that of full-feature implementations, confirming that phishing detection can be simplified without major accuracy loss. The approach is well-suited for lightweight browser extensions or edge devices.

9. Conclusion

This project demonstrates that phishing detection using a reduced feature subset is both feasible and efficient. The combination of SelectKBest and RandomForest offers a practical approach for real-world deployment, balancing speed and accuracy.

10. Future Work

Future extensions could involve collecting live URLs, integrating dynamic content features, and deploying the model through a REST API or browser plugin. Ensemble learning or neural approaches could further improve detection accuracy.

11. References

1. Abdelhamid, S., Ayesh, A., & Thabtah, F. (2021). Detection of Phishing Websites Using Machine Learning Techniques. Expert Systems with Applications, 173, 114737.
2. UCI Machine Learning Repository: Phishing Websites Dataset.
3. Kaggle: Phishing Websites Dataset (CSV version).