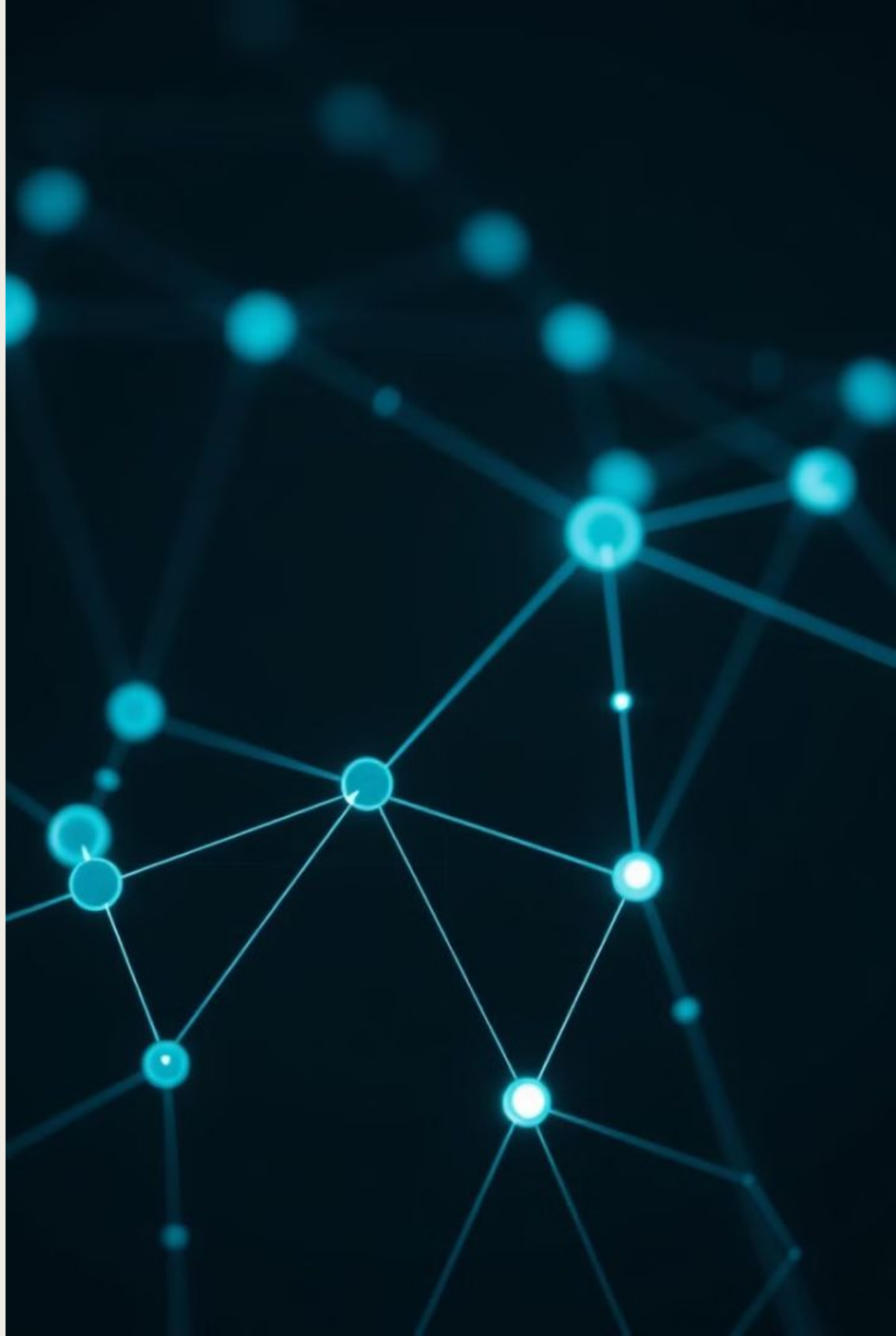
Phishing URL Detection: A Machine Learning Approach



Meet the Team

Aggrey Timbwa

Richard Macharia

Pamela Jepkorir Chebii

Cynthia Njambi

Project Objective

Develop a machine learning-based phishing URL detector to enhance cybersecurity.

Goal:

Achieve real-time, high-accuracy phishing detection to protect users and businesses.

What is Phishing?

Deceptive Tactics

- Fraudsters use email, social media, and other online channels to trick users into revealing sensitive information.

Real-World Examples

- Emails impersonating banks, government agencies, and trusted brands are common phishing techniques.

Understanding Phishing Attacks

1 Business Risks

- Financial losses
- reputational damage.

2 Cybercrime Sophistication

- Advanced tactics
- Social engineering.

3 Protection Necessity

- Tools to identify phishing URLs for real-time detection.

Business Objectives

High-Accuracy Detection

User-Friendly Deployment

Real-Time Classification

Expected Impact

1 Financial Security

Reduce financial losses due to phishing scams.

2 User Trust

Build confidence in online interactions.

Benefits to Stakeholders

Business Users

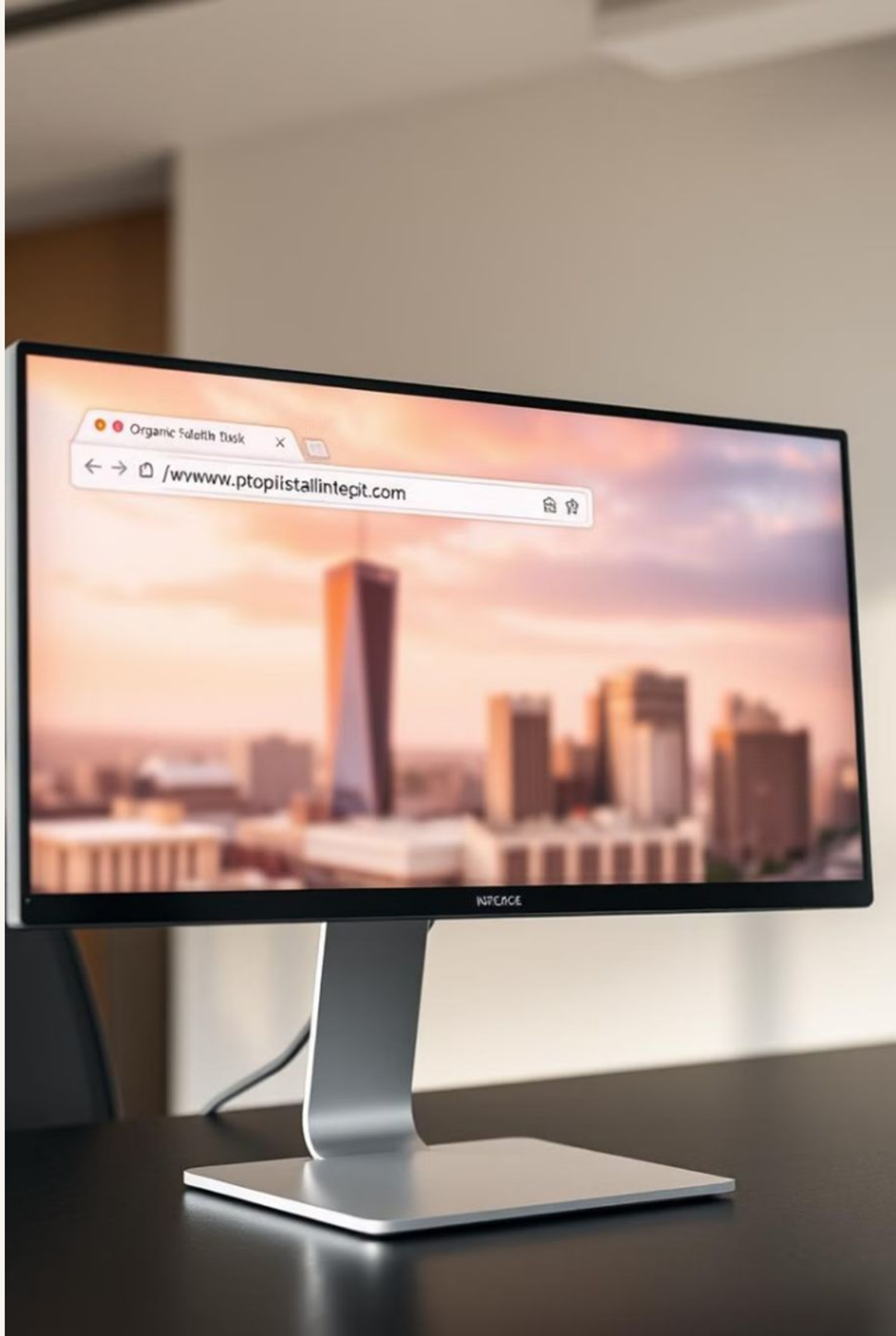
Helps secure business operations by preventing phishing scams, protecting financial data, and ensuring safe online transactions. Example is banking services.

Individual Users (General Public)

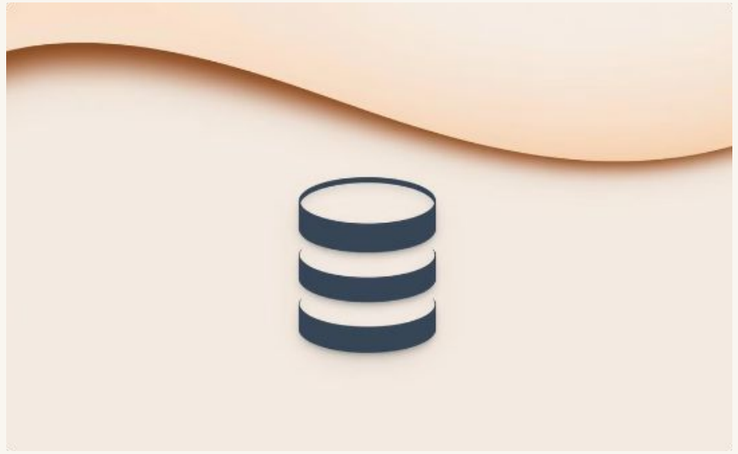
Provides an easy way to avoid phishing attacks and safeguard personal information while browsing the internet.

Cybersecurity Teams

Supports proactive threat prevention by identifying phishing URLs and integrating into existing security measures.



Data Foundation



Dataset

Mendeley Phishing URL Dataset.

Size

6,568,184 URLs, reduced to 10,000 for analysis.

Preparation

Feature extraction, cleaning, and encoding were performed.

Data Preparation

1

Data Collection

- Gather diverse labelled URLs from Mendeley Dataset .

2

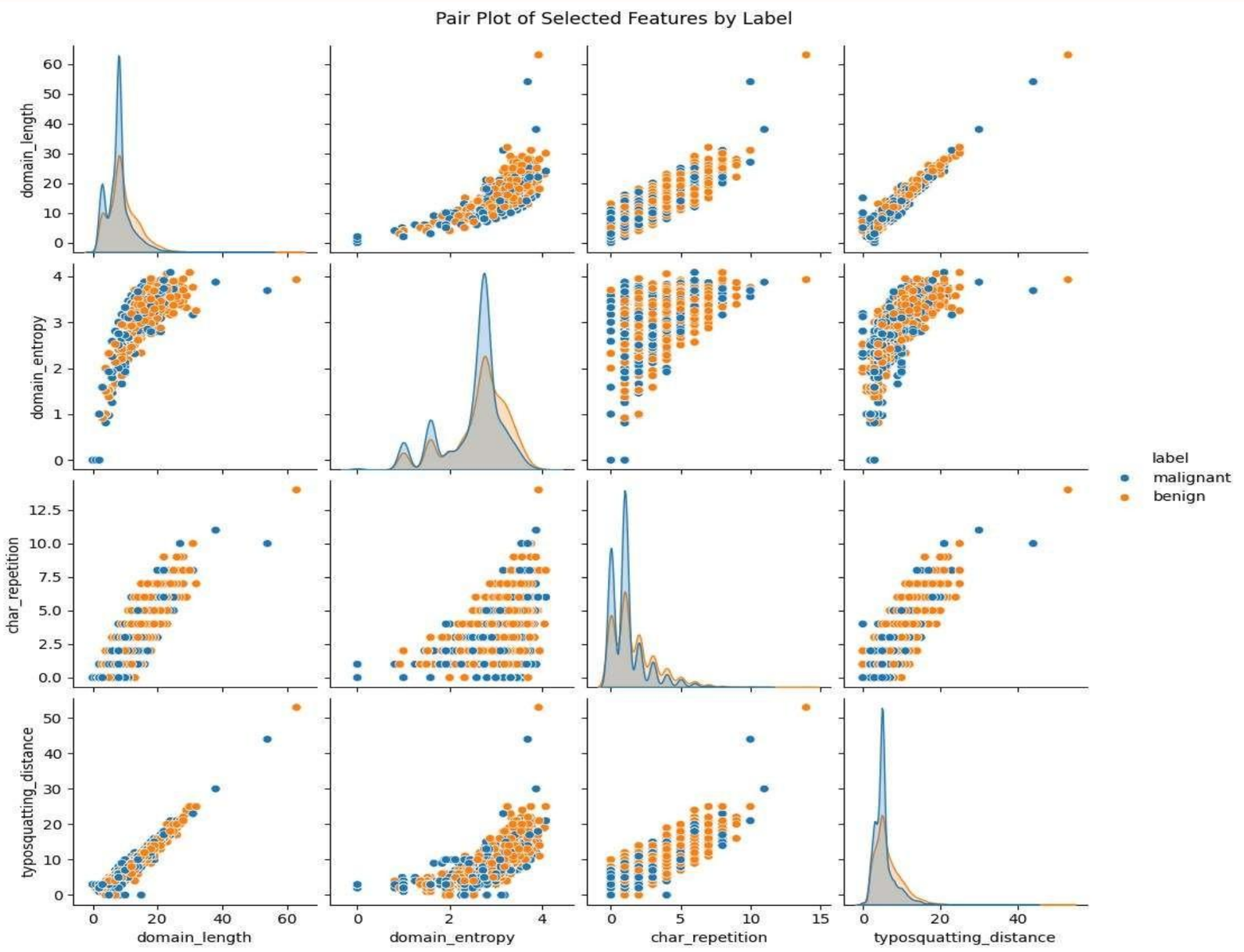
Data Loading

- Loaded the txt data into dataframes

3

Feature Extraction and Engineering

- Extract relevant URL attributes like domain age, URL length.



Select Feature Distribution

Distinct Distributions:

- Some features show clear differences between benign and malignant labels.

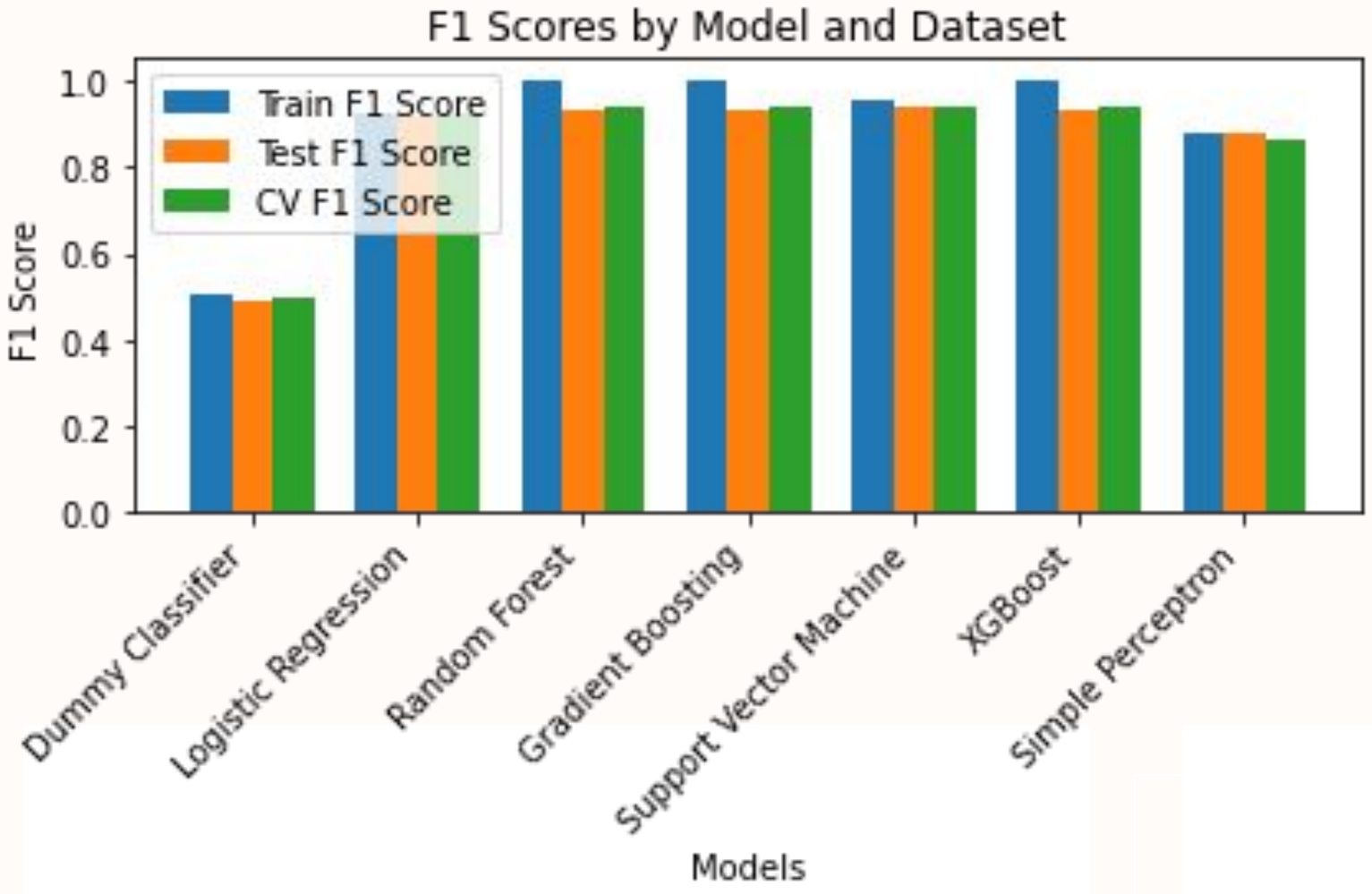
Class Separation:

- Features like domain length and typosquatting distance aid in distinguishing classes.

Significant Features:

- Domain length, entropy, and character repetition stand out for classification.

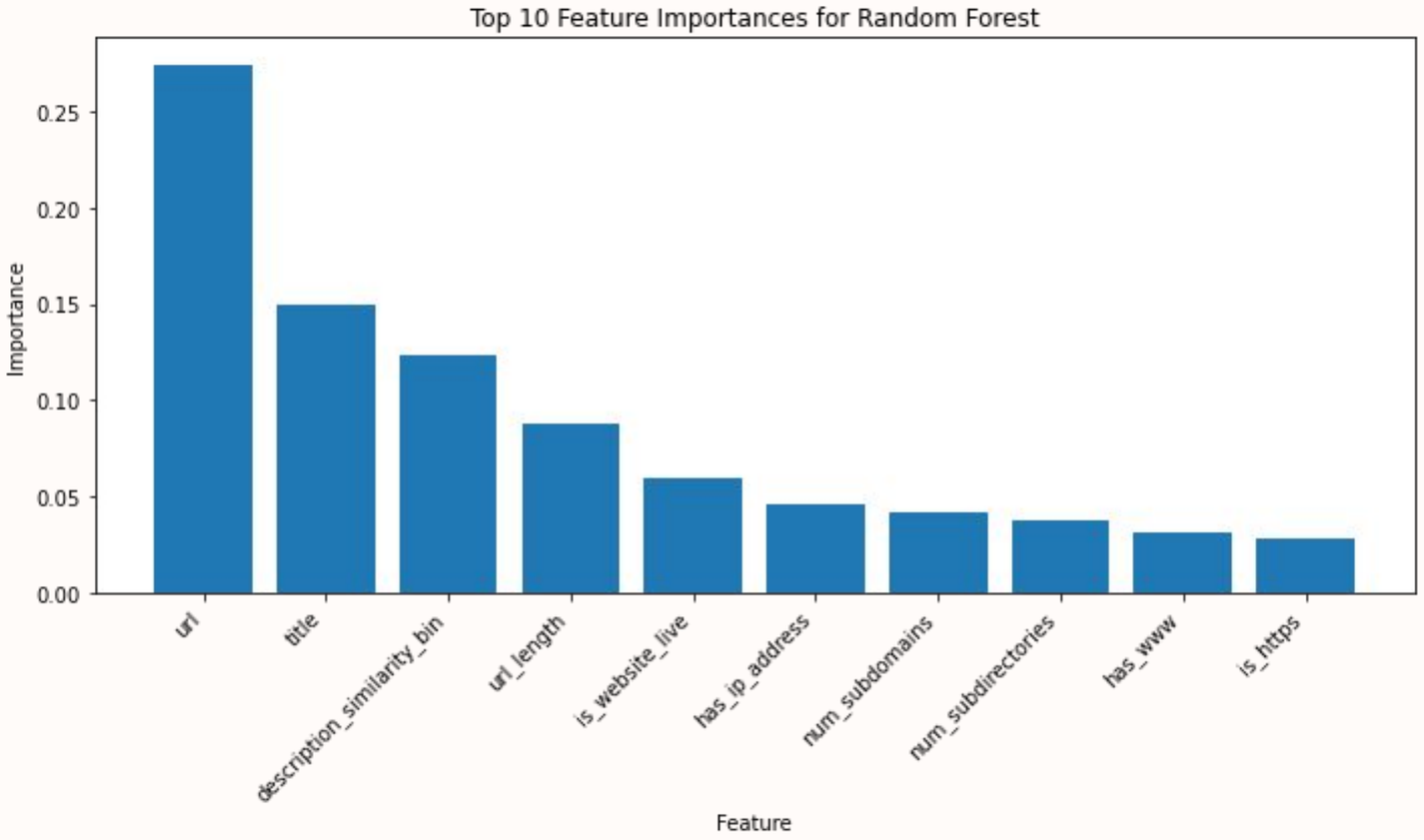
Methodology and Modeling



Select Feature Distribution

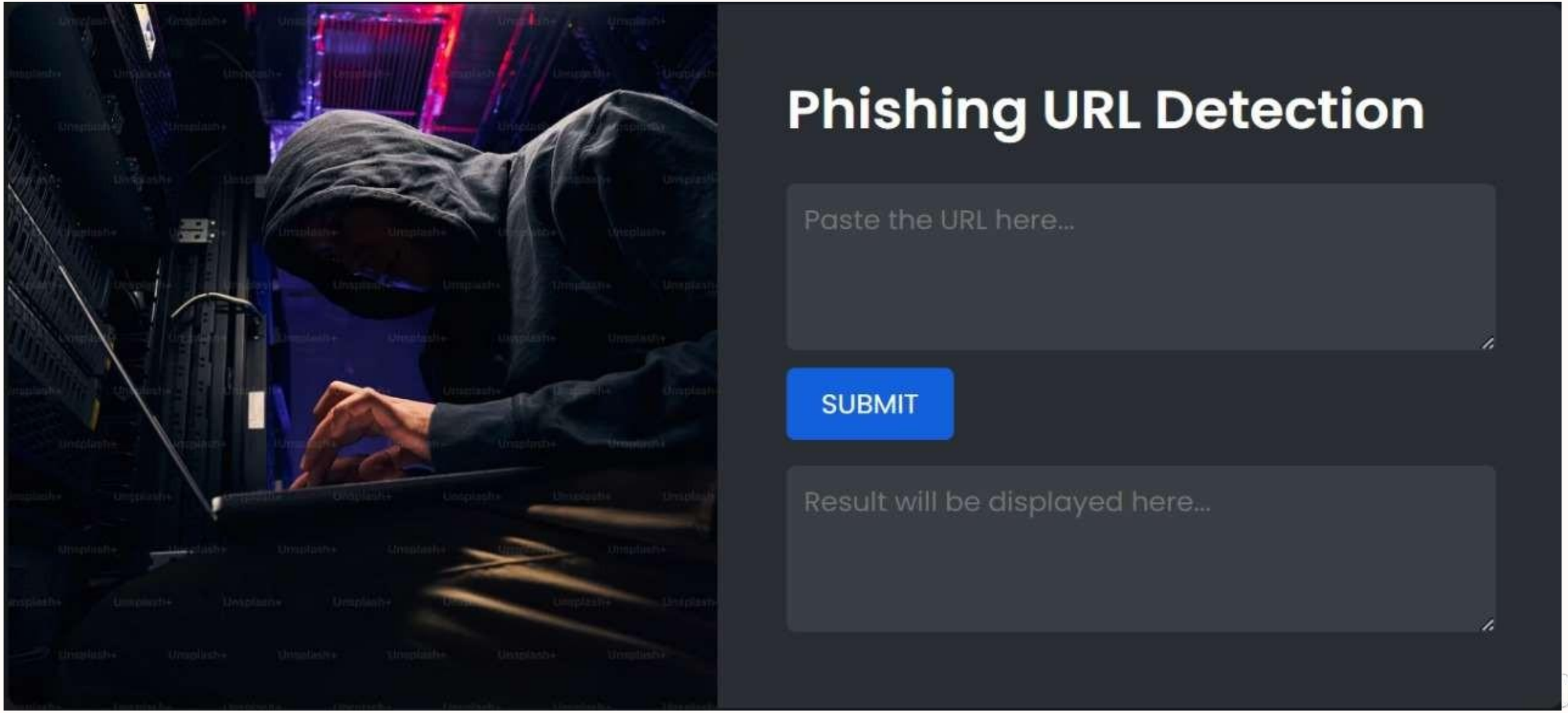
- Iterative approach optimizing precision, recall, and F1-score.
- Used pipelines to make preprocessing and modeling more seamless.
- Incorporated hyperparameter tuning and PCA keeping 95% of the variance which reduced noise and boosted model performance
- Best performing models included SVM followed by random forest and boosted models.

Most Important Features



- URL was the most important feature across boosted models and random forests.
- Models seemed to have picked up more patterns from the URLs themselves and our extracted features
- Further feature extraction and engineering would likely improve the models' generalizability and balance precision and recall.

Deploying the Detector



Key Insights and Recommendations

1

Better URL description

Enhance feature extraction to help the model learn more patterns from urls

2

Data augmentation and continuous training

Increasing the model training sample could better capture a wider array of url patterns and ongoing training would improve models over time

3

User Feedback

Incorporate user feedback to check model performance on unseen data

Challenges and Next Steps

1

Compute Constraints

Scaling to large datasets.

2

Data Augmentation

Gather additional training data.

3

Browser Extension

Remove human in the loop for true real time detection

4

User Feedback

Collect user feedback for improvements.

Technology Stack

Python

Core programming language.

Scikit-learn

Machine learning library.

Pandas

Data manipulation and analysis.

Matplotlib

Data visualization library.

Flask

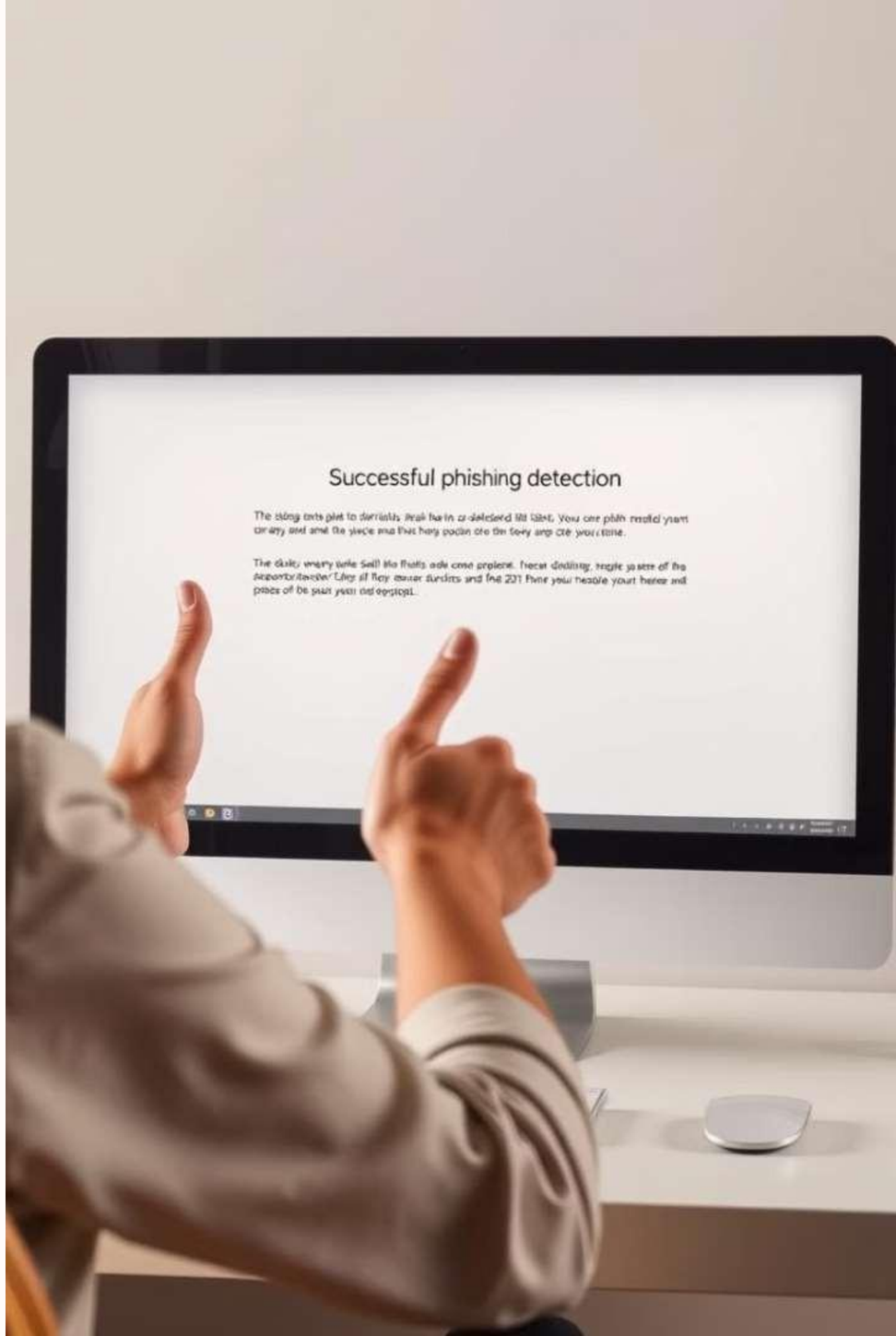
Deployment of a web-based tool

Render

Hosting web application

Conclusion

- High-Accuracy Phishing Detection.
- Balanced F1-Score
- User-Friendly Web Deployment
- Real-Time Classification
- Identify Important Features



Q&A

Thank You