Spam URL Detection System

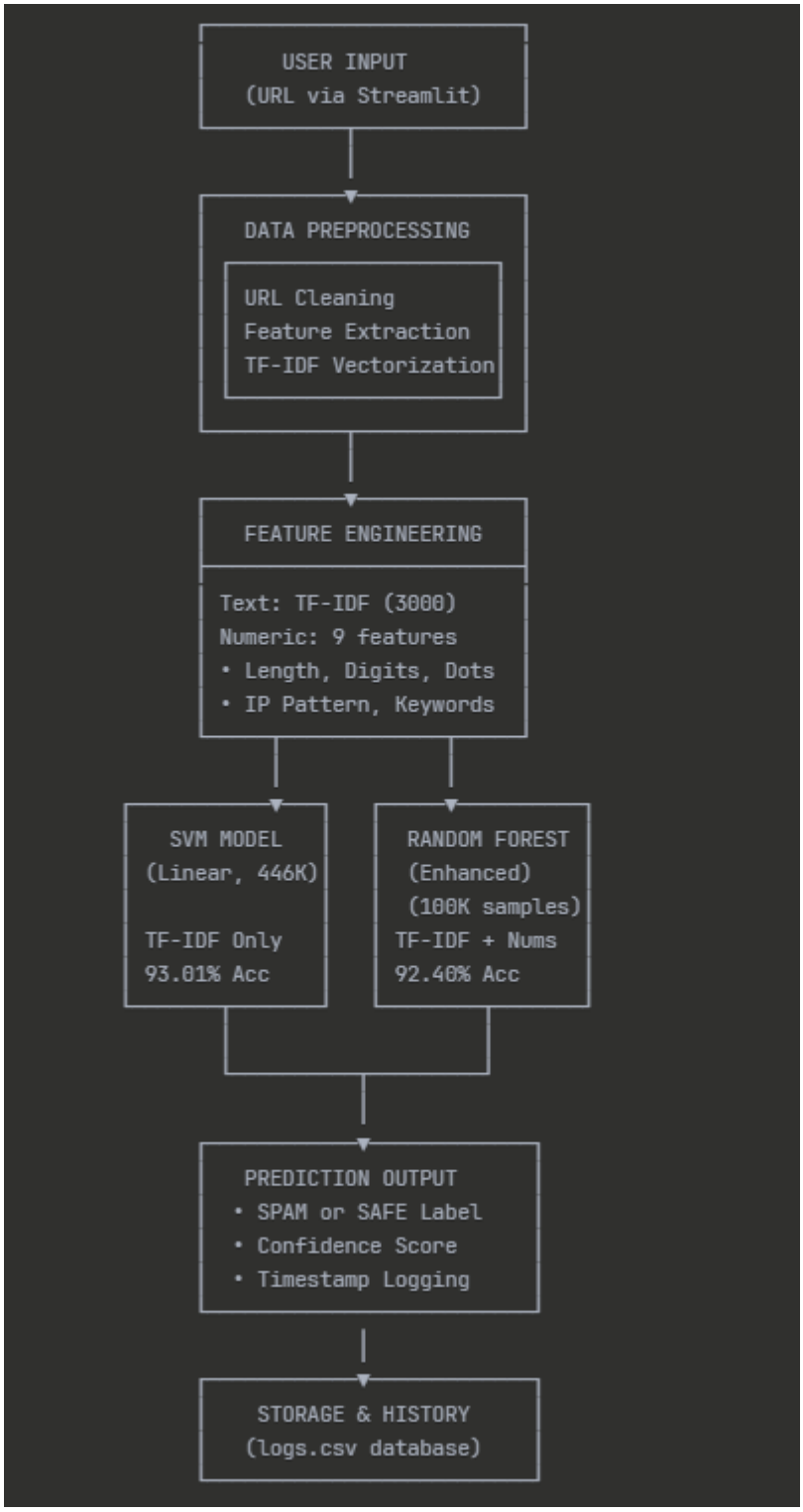
Project Title

Spam URL Detection using Machine Learning

Problem Statement

Spam websites consume valuable search engine crawl resources and degrade the quality of search results. Traditional spam detection methods often rely on analyzing page content, but crawling every page is resource-intensive and not scalable. The challenge is to develop a machine learning model capable of classifying URLs as spam or not spam using only the URL string and associated metadata—such as domain registration details, first-seen date, and traffic— without crawling the page. By implementing Support Vector Machine and Random Forest models, this approach can reduce crawl waste, improve search result quality, and provide a scalable solution to spam detection.

High level Architecture



Key Components:

- **Dataset:** 651K URLs (benign, phishing, malware, defacement)
- **Features:** TF-IDF text vectors + 9 engineered numeric features (length, digits, dots, suspicious words, etc.)
- **Models:** SVM (from ml_project_largedataset.ipynb) + RF (from enhanced_model.ipynb)

Results

Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
SVM	93.01%	0.96	0.90	0.93
Random Forest	92.40%	0.92	0.85	0.88

Confusion Matrices

SVM: FP: 1,670 | FN: 4,571 | Better precision, fewer false alarms
Random Forest: FP: 491 | FN: 1,030 | Enhanced with feature engineering

Sample Predictions

- `http://login-verify-secure-paypal.com` → **SPAM**
- `https://google.com` → **SAFE**
- `https://pes.edu` → **SAFE**

Deployment

Platform: Streamlit web application
Model Files:

- SVM: `svm_model.pkl` + `vectorizer.pkl`
- RF: `spam_url_model.pkl` + `tfidf_vectorizer.pkl`

Features: Real-time classification, dual-model selection, prediction history logging

---

Conclusion

Successfully developed a dual-model spam URL detection system achieving 93%+ accuracy. The system combines SVM's high precision with Random Forest's feature engineering for robust malicious URL identification. Production-ready with intuitive UI and comprehensive logging.

Future Work: Deep learning integration, ensemble voting, browser extension deployment

---

Tech Stack: Python | Scikit-learn | Streamlit | TF-IDF | SVM | Random Forest