MALICIOUS URL CLASSIFICATION

By

Gopalkrishna Ratilal Waja

MOTIVATION

- Malicious URLs are a common vector for cyberattacks, including phishing, malware distribution, and fraud.
- According to Cisco attacks conducted through malicious URLs account for 36% of all US data breaches.
- Major Safe Browsing tool providers like Google and Norton heavily rely on signature-based detection using blacklist lookups.
- Blacklist lookups cannot avoid zero-day attacks.
- Machine learning based detection can be used as a good alternative to blacklist lookups

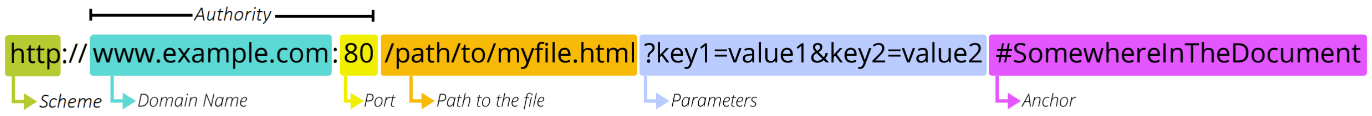
PROBLEM STATEMENT

- Given a URL classify it into one of the 3 categories : Safe, Phishing or Malware
- Data collected from different open sources active URL lists for each class:-

Class	Source	Size
Safe	DMOZ	15000
Phishing	PhishTank	22000
Malware	URLhaus	4078

- All the URL taken here are live URLs

METHODOLOGY



Authority

http:// www.example.com : 80 /path/to/myfile.html ?key1=value1&key2=value2 #SomewhereInTheDocument

Scheme

Domain Name

Port

Path to the file

Parameters

Anchor

1. ## Manual Feature Extraction

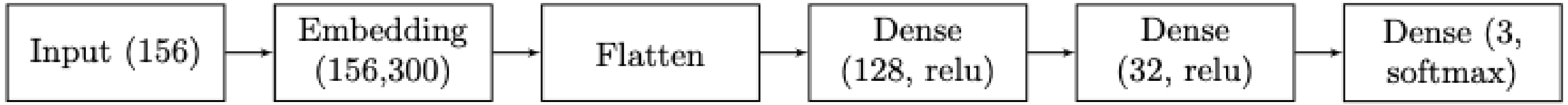
Extracted 28 features manually like:-

- **Lexical Features**: 23 features from the literal URL string. For example, length of the URL string, number of digits, number of parameters in its query part, if the URL is encoded, etc.
- **Host-Based Features**: These provide information about the host of the webpage, for example country and domain age.
- **Content Features**: These features capture the structure of the webpage and the content embedded in it. These will include information on iframes, image count and mouse over effect.

Experiment with different models like Logistic Regression, Decision Trees, Random Forest, SVM, XgBoost, KNN etc.

2. ## Automated Feature Extraction

- Design deep learning models for extracting high level features and perform classification.
- Study the effect of pretrained and random word embeddings GloVe.
- Compare the results with those obtained from manual feature extraction using metrics like accuracy, recall, precision, F1 score.



Input (156)

Embedding (156,300)

Flatten

Dense (128, relu)

Dense (32, relu)

Dense (3, softmax)

RESULTS & DISCUSSION

The results for best parameters for each of the models as the result of grid search using 10-fold cross validation are as follows :-

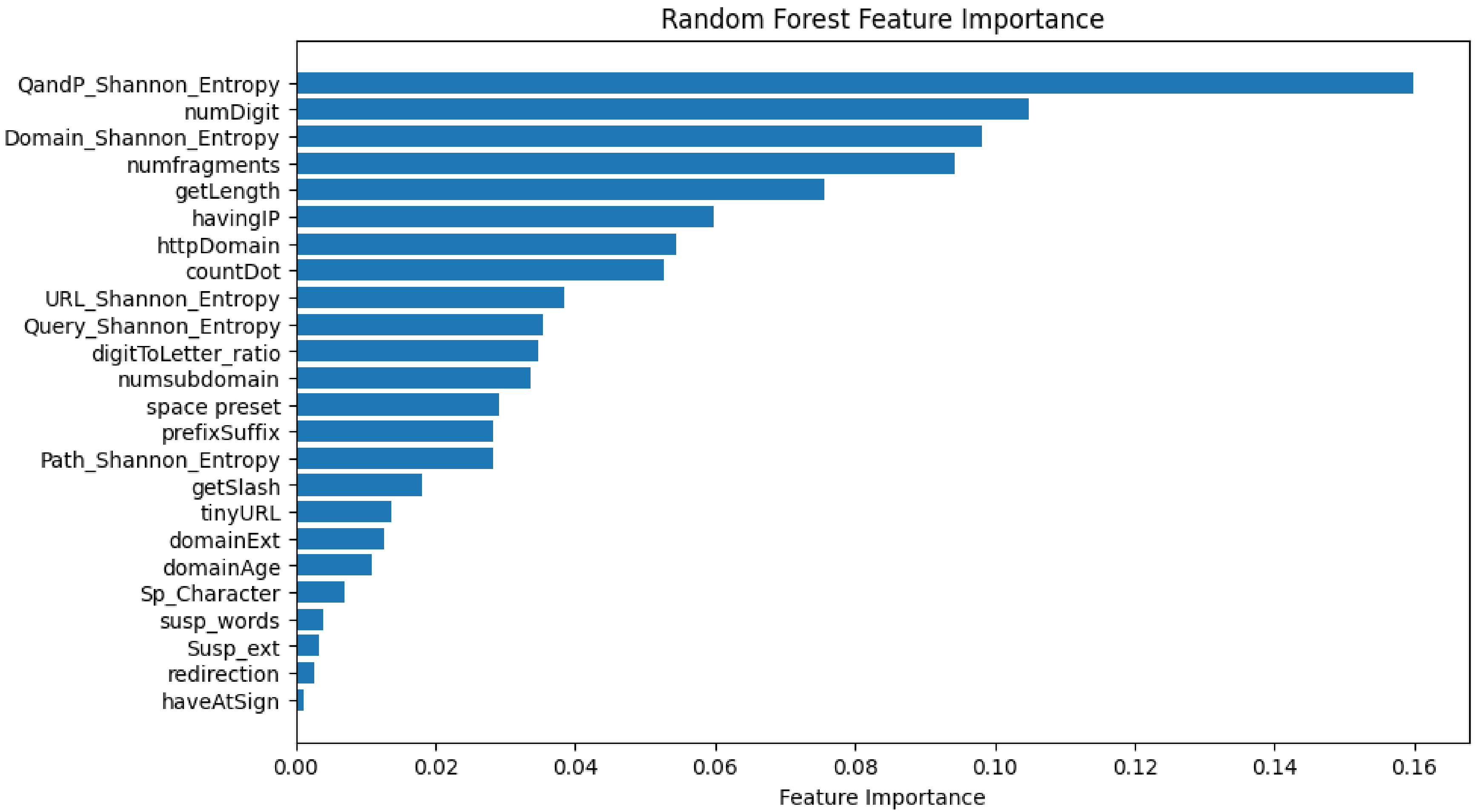
Model	Best Parameters
Logistic Regression	C: 100
Decision Tree	max depth: 20
Random Forest	max depth: 20, n estimators: 200
SVM	C: 10, kernel: RBF
XG Boost	max depth: 5, n estimators: 100
KNN	n neighbors: 3

RESULTS & DISCUSSION

Index	Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
0	Logistic Regression	0.8823	0.8754	0.8760	0.8754	0.8756
1	Decision Tree	0.9857	0.9371	0.9373	0.9371	0.9371
2	Random Forest	0.9878	0.9550	0.9552	0.9550	0.9550
3	SVM	0.9116	0.9046	0.9081	0.9046	0.9051
4	XG Boost	0.9701	0.9546	0.9548	0.9546	0.9547
5	KNN	0.9464	0.9074	0.9084	0.9074	0.9076
6	Pretrained Embeddings + NN	0.9922	0.9778	0.9779	0.9778	0.9778
7	Normal Embeddings + NN	0.5561	0.5511	0.3832	0.5511	0.4061

RESULTS & DISCUSSION

Feature Importance



CONCLUSION & FUTURE WORK

- Decision Tree, Random Forest, and XG Boost showcase exceptional performance in URL classification, leveraging manual feature extraction for high precision and recall.
- Models using pretrained embeddings, especially in combination with neural networks, achieve an impressive 98% accuracy, highlighting the power of automated feature extraction.
- Tree-based models excel in capturing explicit patterns, while pretrained embeddings provide automated feature extraction, showcasing the synergy of these distinct strategies.
- Future work includes exploring imbalance reduction techniques for enhanced model robustness, reflecting a commitment to continuous improvement in URL classification.

THANK YOU