Detection of Phishing Website Using ML Technique

DOMAIN: MACHINE LEARNING

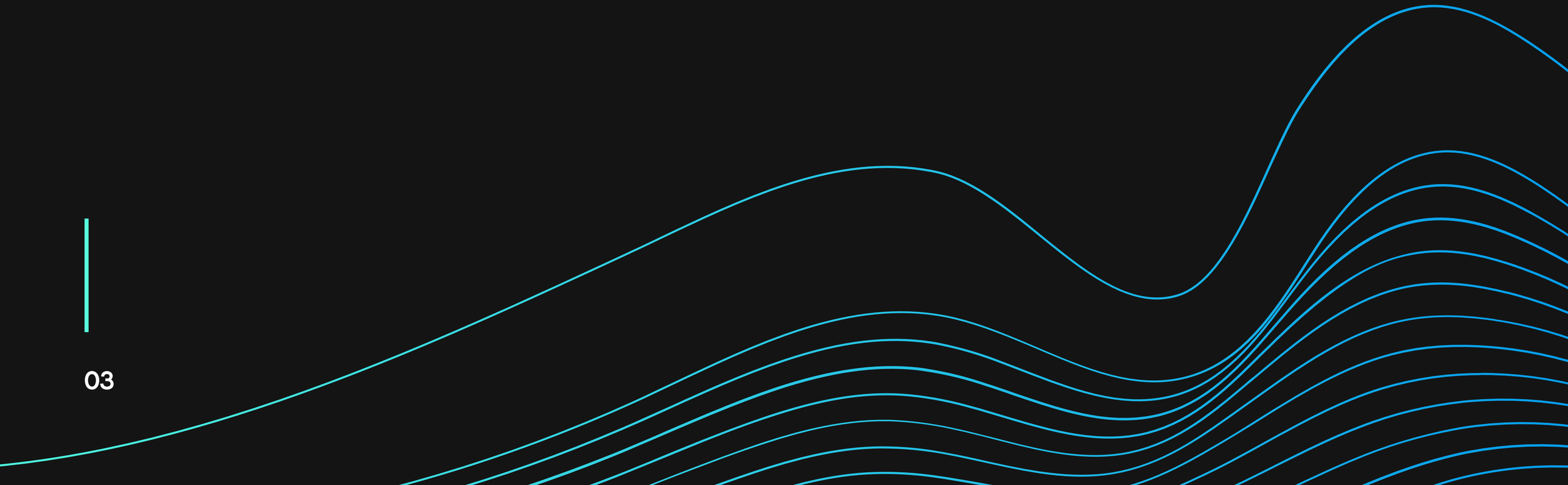
By: Jash Shah
Roll no: 31457
Div: TE-4

Guided By: Prof. R.A Kulkarni

Index

Problem Definition

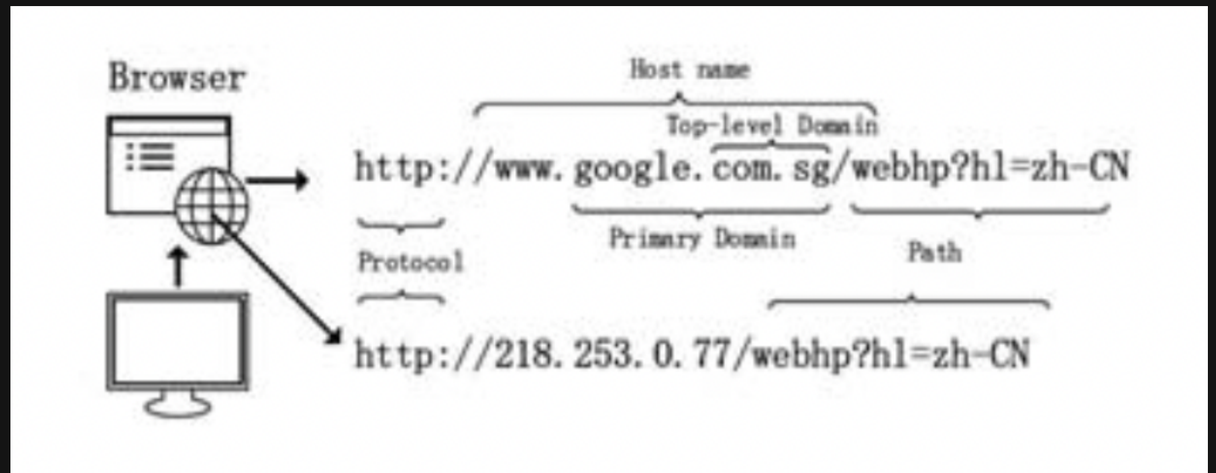
Identification of Phishing websites (URLs) using Machine Learning Techniques based on various factors such as IP address, URL length, Shortening Service, HTTPS token etc



Introduction

- Nowadays people do the majority of their work on digital platforms in their daily lives. In many ways, having a computer and access to the internet makes our work and personal lives easier.
- Its effortless to connect to internet from any where at any time.
- Despite the fact that this arrangement is extremely convenient
- It has revealed significant information gaps. As a result, people in cyberspace must take precautions against potential cyber-attacks.
- The most common and dangerous of these cyber attacks is phishing

- Phishing is the most commonly used social engineering and cyber attack.
- Through such attacks, the phisher targets naïve online users by tricking them into revealing confidential information, with the purpose of using it fraudulently.
- In order to avoid getting phished,
  - users should have awareness of phishing websites.
  - have a blacklist of phishing website which requires the knowledge of website being detected as phishing.
  - detect them in their early appearance, using machine learning and deep neural network algorithms.
- Of the above three, the machine learning based method is proven to be most effective then the other methods
- Even then, online users are still being trapped into revealing sensitive information in phishing websites.



URL Structure

Literature Survey

Sr no.	Paper Title	Summary	Limitations
1	Detection of Phishing Websites by Using Machine Learning-Based URL Analysis	• trained only with the features obtained from the URL. • is expected to classify in a shorter time than other models	• Accuracy and F-measure obtained is less.
2	Phishing Web Page Detection Methods: URL and HTML Features Detection	• rules-based method with the aim of making the application more effective in terms of accuracy and faster detection ability	• Features not selected optimally • Lesser accuracy
3	Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection	• Fuzzy Rough Set (FRS) to select the most effective features. • Best accuracy achieved was 91.46%	• Hard to implement. • Lesser accuracy
4	Detecting Phishing Websites through Deep Reinforcement Learning	• The proposed model can adapt to the dynamic behaviour of the phishing websites and thus learn the features associated with phishing website detection.	• Not optimised for real world implementation. • Only lexical features were considered • Parameters not tuned.
5	OFS- NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network	• FS-NN, an effective phishing website detection model based on the optimal feature and neural network	• Hard to implement. • Takes time to select features.
6	Proactive Phishing Sites Detection	• authors proposed suspicious domain names generation and to predicts likely phishing web sites from the given legitimate brand domain name and scores and judges suspects by calculating various indexes to detect phishing websites	• Method based on heuristic therefore not automatic.

Motivation

- The sectors such as e-commerce and online banking have grown multi-folds in recent years. And a major boom in this sector was seen during the pandemic in the past two years .
- People are now much inclined toward online purchasing and banking rather than the traditional methods.
- And with this rise in online business there is significant rise in the cyber attacks such as phishing in which phishers try to exploit users and try to gain sensitive information from them through phishing websites.
- Such attacks have cost internet users more than billions of dollars per year.
- The reasons why people are getting duped are they are not well educated about these frauds and attacks
- It sometimes becomes hard for a human to identify which websites are legitimate and which are phishing
- Thus, a machine learning approach which would extract important features and would be trained by advanced ML boosting algorithms such as xgboost will help identify users which URLs are legitimate and which aren't , is a good approach to solve this problem.

Software Requirements

- Anaconda Navigator 3.0.1
- scikit-learn 1.0.1
- Numpy 1.21.4
- Pandas 1.3.4
- Matploitlib 3.4.2
- xgboost 1.6.0

Hardware Requirements

- Core i5 machine

Execution/Contribution

- In this seminar topic, I have analysed and have tried to predict if a URL is legitimate or phishing .For this I have used data set from PhishTank website

- As mentioned above, one needs to analyse different algorithms in order to predict the best fit. Here, I have used a few types of algorithms that were tried in various papers and have analysed the accuracy, F1 Score, precision etc. values for the same. These values essentially help predict which algorithm is better suited for the particular dataset.

- The algorithms that I have tried to implement are KNN, SVM, Decision tree,Random Forest, XGBoost classifiers.

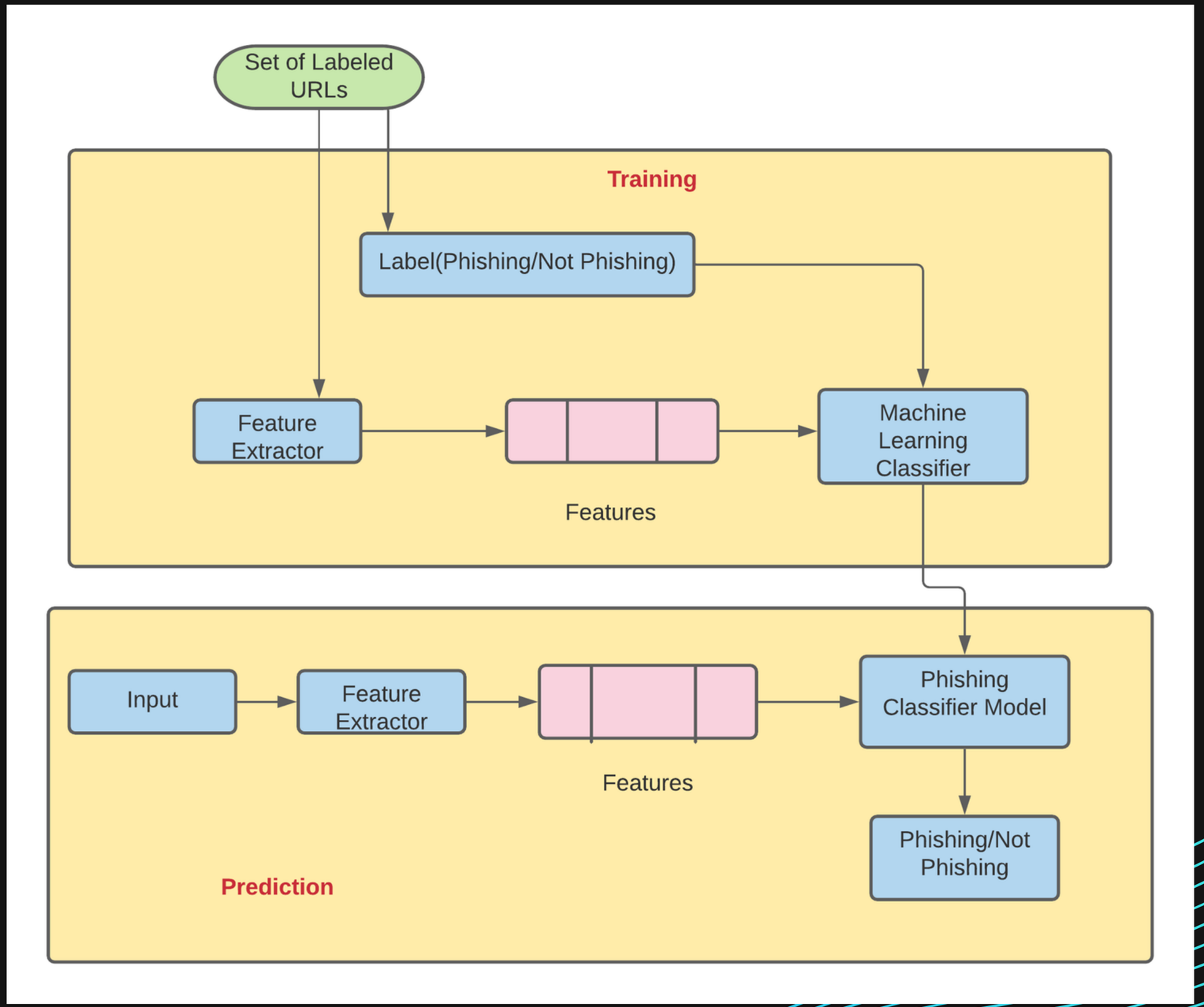
- Conclusively,I have compared all of them in order to choose which algorithm gives more accurate results

Implementation

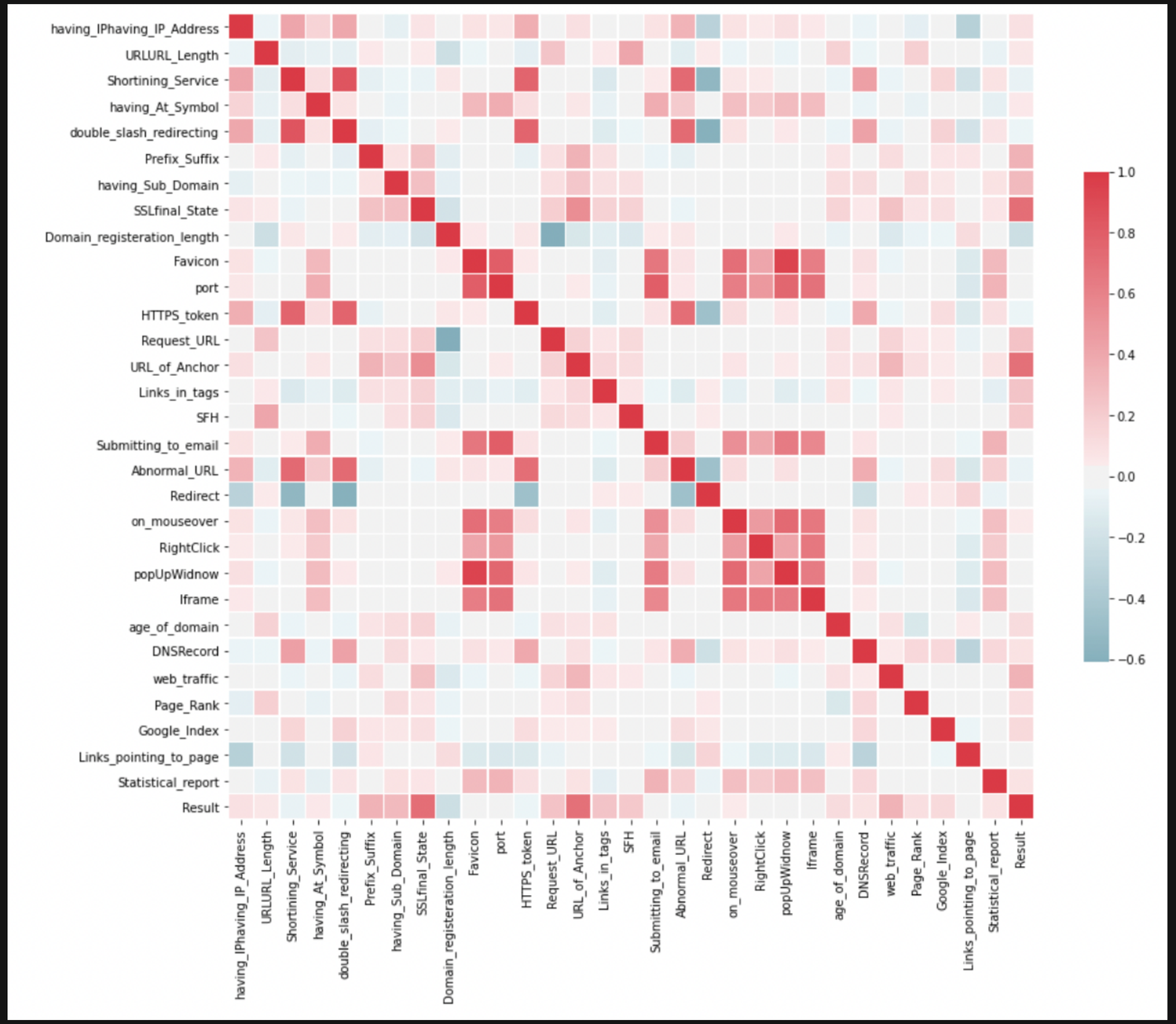
The URLs are taken from the PhishTank website.The dataset has around 11,000 sample websites URL. Each URL is marked either phishing or legitimate.

index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	S
2678	1	1	-1	1	1	1	-1	
5989	1	-1	1	1	1	-1	1	
616	-1	-1	-1	1	-1	-1	-1	
9731	1	-1	1	-1	1	-1	-1	
5329	-1	-1	1	-1	1	-1	-1	

Block Diagram

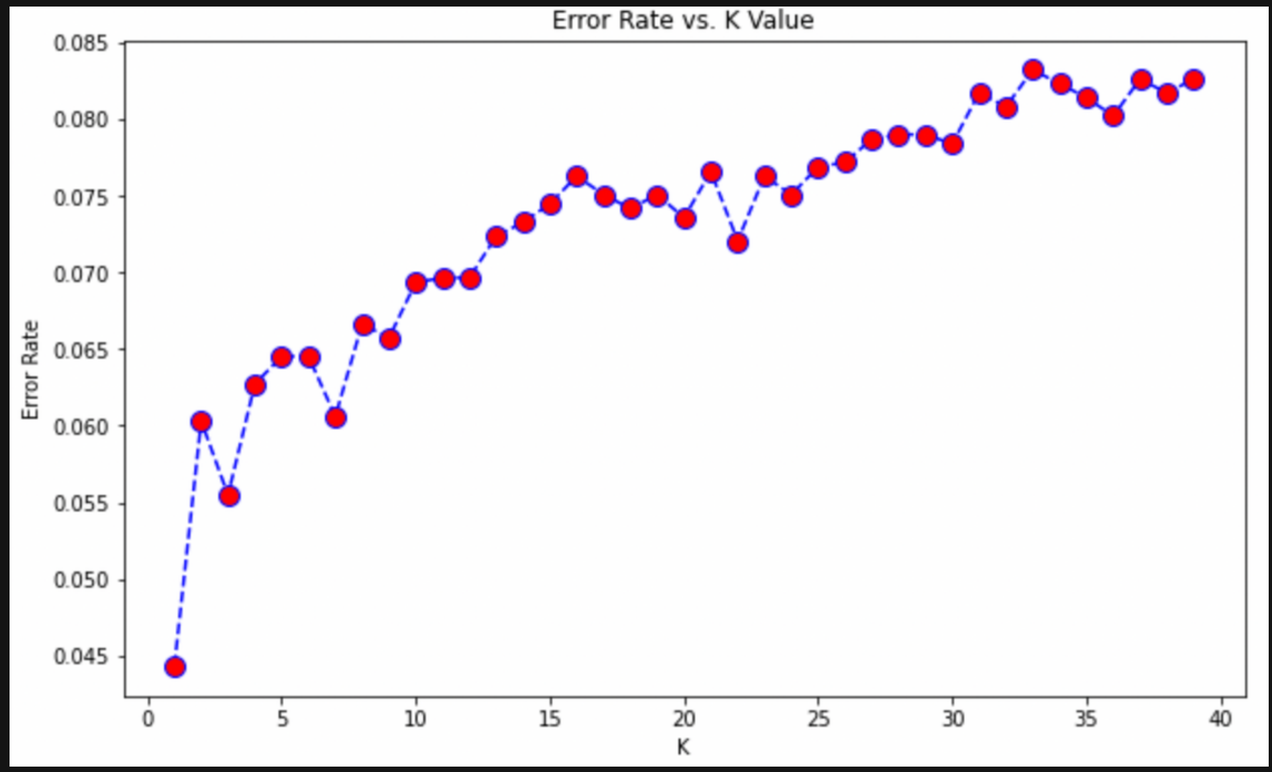


Corrolation Diagram



1.KNN

KNN is a ML technique for regression and classification problems. Feature similarity is used to predict the values for the new data points.The new datapoint is assigned its value depending on how closely it resemble the points in the training set.



KNN

In [39]:
```
KNeighbors_clf=KNeighborsClassifier(3)
cross_val_scores = cross_validate(KNeighbors_clf, X, y, cv=fold_count, scoring=scoring)
KNeighbors_clf_score = mean_score(cross_val_scores)
print(KNeighbors_clf_score)
```

```
{'fit_time': 0.11297237873077393, 'score_time': 0.3535628795623779, 'test_accuracy': 0.9527809643818579, 'test_recall': 0.9629
682715658324, 'test_precision': 0.9527832046787947, 'test_f1': 0.95782786394066}
```

2.SVM

SVM is a supervised machine learning (ML) technique that can be used to solve regression and classification issues. SVM's major goal is to establish a decision boundary that divides n-dimensional space into separate classes so that a new data point can be appropriately classified in the future.

```
###linear
linear_clf = svm.SVC(kernel='linear')
cross_val_scores = cross_validate(linear_clf, X, y, cv=fold_count, scoring=scoring)
linear_svc_clf_score = mean_score(cross_val_scores)
print(linear_svc_clf_score)
```

```
{'fit_time': 2.378216814994812, 'score_time': 0.05668981075286865, 'test_accuracy': 0.9272714850302342, 'test_recall': 0.9462377256889452, 'test_precision':
```

3.Decision Tree

A tree-like structure is used to perform classification in the Decision Tree algorithm.In this algorithm the dataset is separated into small subsets which are used to generate Tree Nodes . These tree nodes can be Leaf Nodes or Decision nodes depending upon their function.

Descision Tree

```
dtree_clf=DecisionTreeClassifier()
cross_val_scores = cross_validate(dtree_clf, X, y, cv=fold_count, scoring=scoring)
dtree_score = mean_score(cross_val_scores)
print(dtree_score)
```

```
{'fit_time': 0.02145226001739502, 'score_time': 0.0037374973297119142, 'test_accuracy': 0.9659887246037655, 'test_recall':
714145813536058, 'test_precision': 0.9676815772041456, 'test_f1': 0.9695318553822136}
```

4.Random Forest

The random forest algorithm uses several different decision trees to anticipate the outcome. It's a collection of decision trees. To generate more precise answers, many decision trees are blended together. It aids in the elimination of decision tree flaws.

Random Forest

In [32]:
```
rforest_clf=RandomForestClassifier()
cross_val_scores = cross_validate(rforest_clf, X, y, cv=fold_count, scoring=scoring)
rforest_clf_score = mean_score(cross_val_scores)
print(rforest_clf_score)
```

```
{'fit_time': 0.4361262321472168, 'score_time': 0.021941876411437987, 'test_accuracy': 0.9726827751548527, 'test_recall': 0.981
4847956921128, 'test_precision': 0.9698521059240438, 'test_f1': 0.9756226033690053}
```

5.XGBoost

XGBoost the Algorithm operates on decision trees, models that construct a graph that examines the input under various "if" statements (vertices in the graph). Whether the "if" condition is satisfied influences the next "if" condition and eventual prediction. XGBoost the Algorithm progressively adds more and more "if" conditions to the decision tree to build a stronger model.

Gradient Boosting With XGBoost

In [34]:
```
XGB_clf=XGBClassifier()
cross_val_scores = cross_validate(XGB_clf, X, y, cv=fold_count, scoring=scoring)
XGB_clf_score= mean_score(cross_val_scores)
print(XGB_clf_score)
```

```
{'fit_time': 0.5060725927352905, 'score_time': 0.0062372684478759766, 'test_accuracy': 0.9712358750705734, 'test_recall': 0.97
90470911202618, 'test_precision': 0.9696243951193955, 'test_f1': 0.9742930356745975}
```

Observations

Table 2: Classification results for different methods

Classifier	train time(s)	test time(s)	accuracy	recall	precision	F1 score
KNN	0.1129	0.3535	0.9527	0.9629	0.9527	0.9578
SVM_linear	1.6475	0.0539	0.9277	0.9455	0.9262	0.9357
Decision Tree	0.0214	0.0037	0.9659	0.9714	0.9676	0.9695
Random Forest	0.4361	0.0219	0.9726	0.9814	0.9698	0.9756
XGBoost	0.5060	0.0062	0.9832	0.9810	0.9872	0.9768

Future Scope

The accuracy for the algorithms like decision tree and random forest can be increased by using various boosting algorithms and also by tuning the hyper parameters with proper analysis.

Conclusion

- ML classifiers were implemented and tested on the phishing website dataset, which consisted of 6157 authentic websites and 4898 phishing websites, in this study.
- The examined classifiers were KNN , SVM , Decision Tree , Random forest , XGBoost.
- The results are shown in table 2. From the table it can be said that excellent results were gained in ensembling classifiers like as Random forest and XGBoost in terms of both computation time and accuracy

References

- Korkmaz, Mehmet & Sahingoz, Ozgur & Diri, Banu. (2020). Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. 1-7. 10.1109/ICCCNT49239.2020.9225561.
- Chapla, Happy & Kotak, Riddhi & Joiser, Mittal. (2019). A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier. 383-388. 10.1109/ICCES45898.2019.9002145.
- H. Faris and S. Yazid, "Phishing Web Page Detection Methods: URL and HTML Features Detection," 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 167-171, doi: 10.1109/IoTaIS50849.2021.9359694.
- Chatterjee and A. -S. Namin, "Detecting Phishing Websites through Deep Reinforcement Learning," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 2019, pp. 227-232, doi: 10.1109/COMPSAC.2019.10211.
- Nakamura, Akihito & Dobashi, Fuma. (2019). Proactive Phishing Sites Detection. 443-448. 10.1145/3350546.3352565.
- Shantanu, Janet, B., & Joshua Arul Kumar, R. (2021). Malicious URL Detection: A Comparative Study. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). doi:10.1109/icais50930.2021.9396014
- Kumar, J., Santhanavijayan, A., Janet, B., Rajendran, B., & BS, B. (2020). Phishing Website Classification and Detection Using Machine Learning. 2020 International Conference on Computer Communication and Informatics (ICCCI). doi:10.1109/iccci48352.2020.91