Project Report on

"Malicious URL Classification"

MACHINE LEARNING (UML501)

Lab Evaluation

Submitted by:

102103750 Sanya Mahajan

BE 3rd Year

Submitted to-

DR. ASHUTOSH AGGARWAL

Computer Science and Engineering Department

TIET, Patiala

Aug-Dec 2023

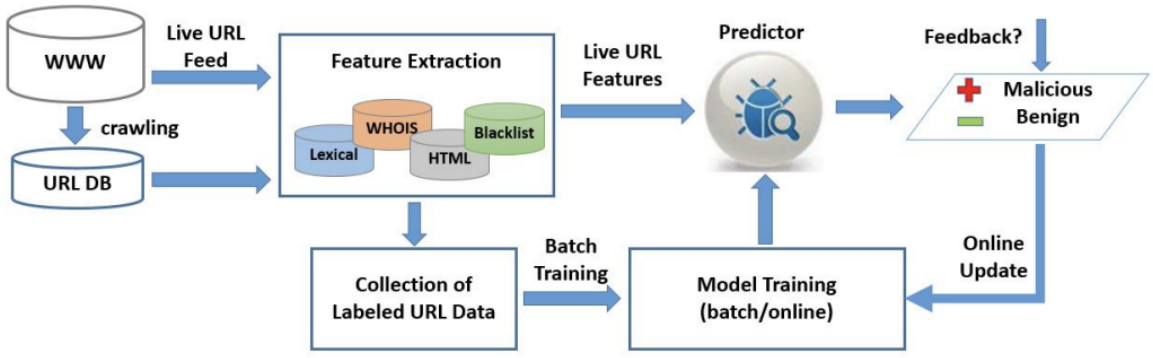
Introduction

Modified or compromised URLs employed for cyber attacks are known as malicious URLs.

A malicious URL or website generally contains different types of trojans, malware, unsolicited content in the form of phishing, drive-by-download, spams.

The main objective of the malicious website is to fraud or steal the personal or financial details of unsuspecting users.

Detection of malicious URLs is a multi-class classification problem by classifying the raw URLs into different class types such as benign or safe URLs, phishing URLs, malware URLs, or defacement URLs.



Dataset description

A Malicious URLs dataset of 6,51,191 URLs, out of which 4,28,103 benign or safe URLs, 96,457 defacement URLs, 94,111 phishing URLs, and 32,520 malware URLs.

- Benign URLs: These are safe to browse URLs.

- Malware URLs: These type of URLs inject malware into the victim's system once he/she visit such URLs.

- Defacement URLs: Defacement URLs are generally created by hackers with the intention of breaking into a web server and replacing the hosted website with one of their own, using techniques such as code injection, cross-site scripting, etc. Common targets of defacement URLs are religious websites, government websites, bank websites, and corporate websites.

- Phishing URLs: By creating phishing URLs, hackers try to steal sensitive personal or financial information such as login credentials, credit card numbers, internet banking details, etc.

Feature Construction

- **having_ip_address**
- **abnormal_url:** This feature can be extracted from the WHOIS database
- **google_index:** check whether the URL is indexed in google search console or not.
- **Count.** : The phishing or malware websites generally use more than two sub-domains in the URL. Each domain is separated by dot (.).
- **Count-www**: This feature helps in detecting malicious websites if the URL has no or more than one www in its URL.
- **count@**: The presence of the "@" symbol in the URL ignores everything previous to it.
- **Count_dir**: The presence of multiple directories in the URL generally indicates suspicious websites.
- **Count_embed_domain**: The number of the embedded domains can be helpful in detecting malicious URLs. It can be done by checking the occurrence of "//" in the URL.
- **Suspicious words in URL**: Malicious URLs generally contain suspicious words in the URL such as login, sign in, bank, account, update, bonus, service, ebayisapi, token, etc.
- **Short_url**: This feature is created to identify whether the URL uses URL shortening services like bit. \ly, goo.gl, go2l.ink, etc.

- **Count_https**: Generally malicious URLs do not use HTTPS protocols as it generally requires user credentials and ensures that the website is safe for transactions..
- **Count_http**: Most of the time, phishing or malicious websites have more than one HTTP in their URL.
- **Count%**: As we know URLs cannot contain spaces. URL encoding normally replaces spaces with symbols (%). Malicious websites generally contain more spaces in their URL hence more number of %.
- **Count?:** The presence of symbol (?) in URL denotes a query string that contains the data to be passed to the server. More number of ? in URL definitely indicates suspicious URL.
- **Count-**: Phishers or cybercriminals generally add dashes(-) in prefix or suffix of the brand name so that it looks genuine URL. For example. www.flipkart-india.com.
- **Count**=: Presence of equal to (=) in URL indicates passing of variable values from one form page to another.
- **url_length**: average length of a safe URL is 74.
- **hostname_length**
- **Count_digits**: The presence of digits in URL generally indicate suspicious URLs.
- **Count_letters:** as attackers try to increase the length of the URL to hide the domain name and this is generally done by increasing the number of letters .

Methodology

The methodology employed for the classification task involves training three distinct machine learning models—Random Forest, XGBoost, and Light GBM—on the provided dataset, each used to predict the class labels associated with URLs, categorized into 'benign,' 'defacement,' 'phishing,' and 'malware.'

For each individual model, the dataset is split into training and testing sets, and the model is trained on the training set and evaluated on the testing set. Evaluation metrics, including precision, recall, and F1-score, are generated using the classification report, offering a detailed overview of the model's performance across different classes. Moreover, to enhance the predictive power, an ensemble model is created using a VotingClassifier, combining the predictions of the individual models.

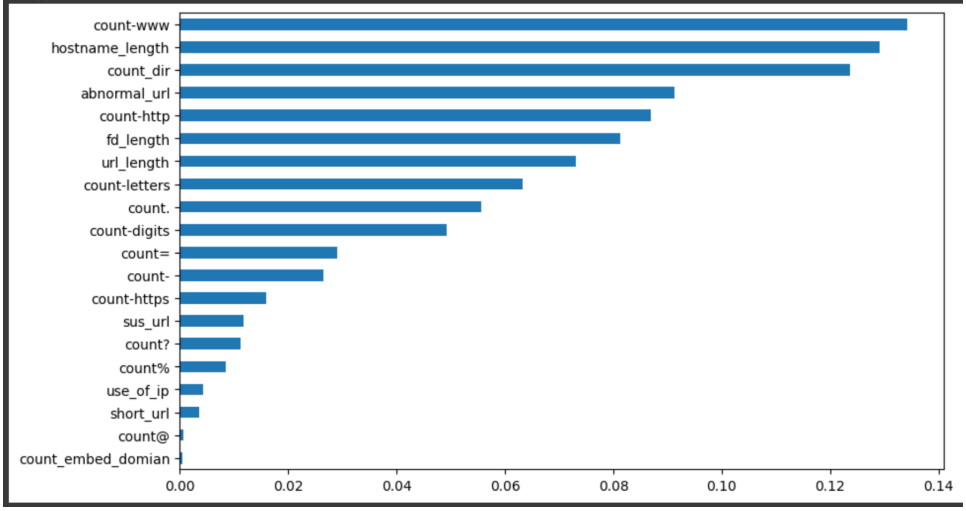
The performance of each individual model and the ensemble model is assessed through accuracy scores, providing insights into their effectiveness in accurately classifying URLs. The methodology aims to leverage the strengths of each model and enhance overall classification performance through ensemble learning.

Result Analysis

Feature Importance chart-

```
feat_importances = pd.Series(rf.feature_importances_, index=X_train.columns)
feat_importances.sort_values().plot(kind="barh",figsize=(10, 6))
```

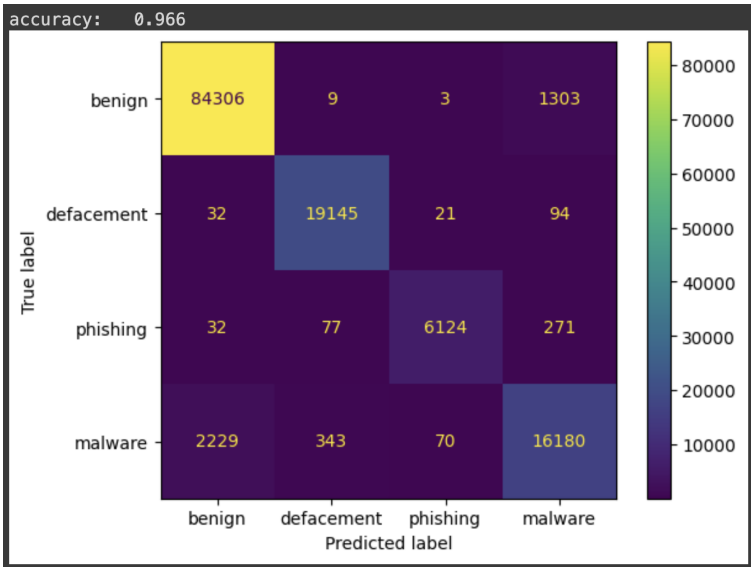
```
<Axes: >
```



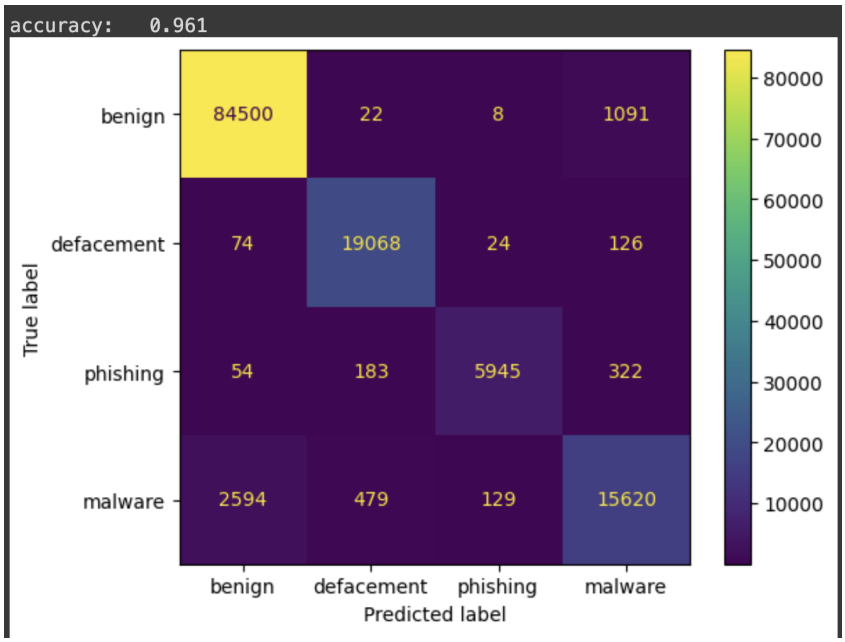
Accuracy Score Comparison

- Random Forest - 96.6%,

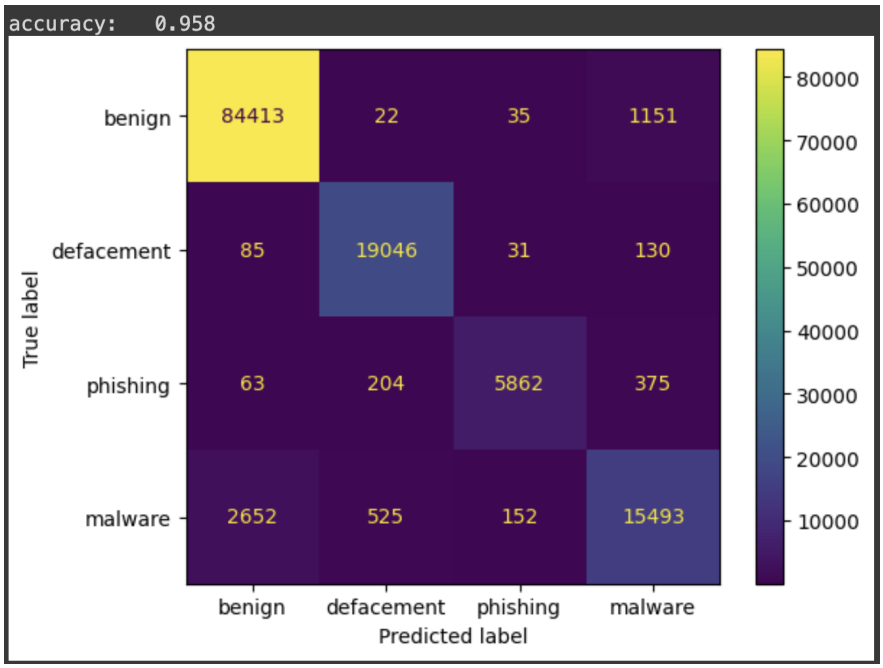
```
accuracy:  0.966
```



- XGBoost - 96.1%

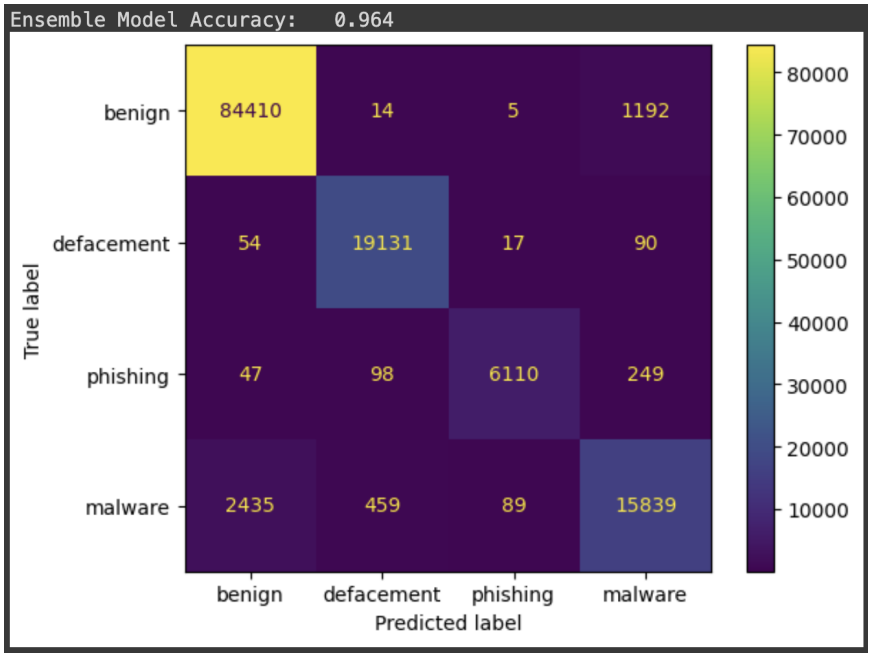


- LightGBM - 95.8%



- Ensemble Model

  RandomForest, **Logistic Regression**, **Decision Tree** - 96.4%



Observations-

The custom ensemble model made using majority voting classifier using simple base classifier outperforms solo performances of XGBoost and LightGBM.

Future scope-

Create a chrome based browser extension that uses the deployed ML classification model to prompt the user before accessing the malicious website.