

**Malicious URL Detection using Machine Learning**

**By**

|  |  |
| --- | --- |
| **Student Name** | **Academic ID** |
| Hailah Alharthi | 445306944 |
|  |  |
|  |  |
|  |  |
|  |  |

**Supervisor: Dr.Turki Alalyani**

Contents

[1. Abstract 4](#_Toc167129212)

[2. Introduction 4](#_Toc167129213)

[3. Background 4](#_Toc167129214)

[4. Literature Review 5](#_Toc167129215)

[4. Objective 14](#_Toc167129216)

[5. Methodology 14](#_Toc167129217)

[**Research Design** 14](#_Toc167129218)

[**Dataset** 15](#_Toc167129219)

[6. Data Analysis 15](#_Toc167129220)

[**Experiment 1:** 16](#_Toc167129221)

[**1-** **Collecting datasets** 16](#_Toc167129222)

[**2-** **Exploring the Dataset** 16](#_Toc167129223)

[**3-** **Cleaning the collected dataset** 17](#_Toc167129224)

[**4-** **Analyzing the dataset and preparing it for modeling** 18](#_Toc167129225)

[**5-** **Splitting the collected dataset** 21](#_Toc167129226)

[**6-** **Apply ML algorithms** 22](#_Toc167129227)

[**7-** **Evaluate the used algorithms** 22](#_Toc167129228)

[**Experience 2:** 23](#_Toc167129229)

[**Experience 3:** 25](#_Toc167129230)

[**Experience 4:** 25](#_Toc167129231)

[**1-** **Collecting datasets** 25](#_Toc167129232)

[**2-** **Exploring the Dataset** 26](#_Toc167129233)

[**3-** **Cleaning the collected dataset** 26](#_Toc167129234)

[**4-** **Analyzing the dataset and preparing it for modeling** 26](#_Toc167129235)

[**5-** **Splitting the collected dataset** 27](#_Toc167129236)

[**6-** **Apply ML algorithms** 28](#_Toc167129237)

[**7-** **Evaluate the used algorithms** 28](#_Toc167129238)

[**Experience 5:** 29](#_Toc167129239)

[**1-** **Apply ML algorithms** 29](#_Toc167129240)

[**2-** **Evaluate the used algorithms** 29](#_Toc167129241)

[6. Conclusion 33](#_Toc167129242)

[7. References 34](#_Toc167129243)

[8. Appendix 36](#_Toc167129244)

1. **Abstract**

Organizations are at significant risk from cybersecurity threats, especially phishing attacks, in today's rapidly evolving technological environment. Phishing attack has several types and techniques that aim to have initial access to the victim by using social engineering techniques to fool the victim. To reduce this risk we will examine how modern machine learning algorithms may be used to detect and minimize the impact of phishing URLs. With a focus on the ongoing risk of phishing URLs in the field of cybersecurity, this research aims to investigate innovative methods for analyzing URLs to spot potential risks. We seek to create creative approaches that leverage machine learning to improve phishing detection capabilities by utilizing publically accessible data. Our goal in conducting this research is to improve security measures and protect enterprises against phishing attacks.

**Keywords** Cyber security,  Phishing,  Machine learning,  Classifier  Algorithm, Phishing Detection

1. **Introduction**

The Internet and digital transactions are growing at an exponential rate, which has made the cybersecurity threat landscape more dynamic and complicated. Malicious Uniform Resource Locators (URLs) are powerful cyberattack vectors that may breach sensitive data, compromise systems, and cause disaster on networks, making them stand out among a variety of other cyber threats. Malicious URLs cover a wide range of criminal actions, such as malware distribution, phishing attacks, and taking advantage of holes in web applications.

Conventional techniques for identifying fraud URLs, which rely on static rules and signature-based procedures, frequently find it difficult to keep up with the constantly changing strategies used by online fraudsters. Since attackers are constantly coming up with new techniques that avoid detection, there is an urgent need for more adaptable and accurate detection systems. “The previous work done to detect malicious URLs using blacklisting or heuristics is found to lack reliability as they lack capabilities to detect new malicious URLs”[30]. In recent years, machine learning (ML) has emerged as a crucial instrument in the cybersecurity toolkit, able to both identify and reduce the threat posed by malicious URLs. Machine learning algorithms may identify small variations between Malicious and normal URLs by utilizing patterns and features present in URLs, which improves detection speed and reduces the impact. Modern approaches in machine learning for malicious URL identification are thoroughly reviewed in this study. We investigate which machine learning methods—supervised, unsupervised, and semi-supervised learning—are most useful for detecting malicious URLs. We also look at feature extraction methods specifically designed for URL analysis, model performance evaluation criteria, and difficulties in implementing machine learning models in real-world situations.

The idea of this research was inspired by research[1] where the author illustrates the effective use of machine learning algorithms to detect phishing URLs.

1. **Background**

“Phishing is a cyberattack that can be carried out using various approaches and techniques”[6], including email, website, and social media-based techniques, all aimed at gaining initial access to the victim through social engineering where the phishers(attacker) use sophisticated social engineering tactics to trick victims into revealing sensitive information or performing actions that compromise security.

As cybersecurity measures improve, threat actors continuously adapt their techniques, making it crucial to stay vigilant and explore new detection methods.

1. **Literature Review**

We reviewed 31 research papers; 25 included experiments on detection techniques, 5 were literature reviews and one was an experiment to evaluate social awareness about phishing. Summarization of those experiments shown in the tables

TABLE 1. Summary of background literature that includs experience.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Study** | **Proposed System** | **Dataset** | **Classifier** | **Accuracy** |
| **1** | Robust Ensemble Machine Learning Model for Filtering Phishing URLs: Expandable Random Gradient Stacked Voting Classifier (ERG-SVC)[1] | Expandable Random Gradient Stacked Voting Classifier (ERG-SVC) | 73,575 URLs and 100,000 URLs | Random Forest (RF), Decision Tree (DT), XgBoost, Gradient Boosting (GB), AdaBoost, K-Nearest Neighbors (KNN), and Logistic Regression (LR) | 98.25% |
| **2** | An intelligent cyber security phishing detection system using deep learning techniques[4] | Detection model using machine learning techniques for combating phishing attacks | First dataset: 5,25,754 instances with 8351 phishing emails and 5,17,402 legitimate emails.  The second dataset with 50 features: 5000 phishing emails and 5000 legitimate emails .  Third dataset with only text feature: 2500 legitimate emails and 500 phishing emails . | 1.Locally-deep support vector machine  2. Support vector machine  3.Boosted decision tree  4.Logistic regression  5.Averaged perceptron  6.Neural network  7.Decision forest | The best ML algorithm accuracy were 0.88, 1.00, and 0.97 consecutively for boosted decision tree |
| **3** | Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification[5] | Cuckoo Search Optimization Algorithm with a Deep Learning-based Phishing Email Detection and Classification (ICSOA-DLPEC) model. | 7,781 legitimate emails and 999 phishing emails | ICSOA-DLPEC | 99.72%. |
| **4** | Assessment of End-User Susceptibility to Cybersecurity Threats in Saudi Arabia by Simulating Phishing Attacks[7] | Develop an efficient solution for detecting phishing attacks in URLs by utilizing Artificial Neural Network (ANN) and Deep Neural Network (DNN) based systems | 37,175 phishing web pages and 36,400 legitimate web pages, | Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs | (ANN) approach achieved an accuracy rate of 92% . (DNN) approach achieved an accuracy rate of 96% |
| **5** | Detection of Phishing Websites by Using Machine Learning-Based URL Analysis[8] | The proposed system in the research focuses on implementing a phishing detection system using machine learning algorithms to analyze URLs | Dataset-1: Total of 83,857 URLs (40,668 phishing URLs + 43,189 legitimate URLs)  Dataset-2: Total of 82,888 URLs (40,668 phishing URLs + 42,220 legitimate URLs)  Dataset-3: Total of 126,077 URLs (40,668 phishing URLs + 85,409 legitimate URLs) | 1. Logistic Regression (LR)  2. K-Nearest Neighborhood (KNN)  3. Support Vector Machine (SVM)  4. Decision Tree (DT)  5. Naive Bayes (NB)  6. XGBoost  7. Random Forest (RF)  8. Artificial Neural Network (ANN) | Best result for RF with 94.59 |
| **6** | Phishing web site detection using diverse machine learning algorithm[10] | proposed system aims to detect phishing websites using a combination of diverse machine learning algorithms and feature selection techniques. | 11,055 websites with 32 attributes related to phishing sites | 1.Random Forest (RF)  2.Support Vector Machine (SVM)  3.Bagging  4.k-Nearest Neighbor (kNN)  5.Neural Network (NN)  6.J48  7.Gradient Boosted Decision Tree  8.Light Gradient Boosting Machine (LightGBM)  9.XGradientBoost  10Decision Tree  11-K-star  12.AdaBoost  13.SMO  14.Naïve Bayes | Stacking model (RF + NN + Bagging): Achieved the highest classification accuracy of 97.4% in detecting phishing websites . |
| **7** | Mutual Information based Logistic Regression for phishing URL detection[11] | method utilizing mutual information and logistic regression techniques for the detection of phishing URLs | 134,850 legitimate URLs and 100,945 phishing URLs | Logistic Regression | 99.97% |
| **8** | 1URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection[12] | URLNet, an end-to-end deep learning framework designed for malicious URL detection | 15 million URLs | URLNet, Convolutional Neural Networks (CNNs) | evaluated based on the Area under the ROC Curve (AUC) and True Positive Rates at different False Positive Rates (FPR). |
| **9** | Malicious URL Detection based on Machine Learning[13] | The proposed system uses machine learning algorithms SVM and RF to detect malicious URLs by training on labeled datasets, extracting features, and classifying URLs as safe or malicious. | 70,000 malicious URLs and 400,000 safe URLs | Support Vector Machine (SVM) and Random Forest (RF) | Best result RF with 99.77% |
| **10** | Malicious URL Detection Based On A Parallel Neural Joint Model[14] | proposed system in the research is a novel approach for detecting malicious URLs by combining visual and semantic features using a parallel neural joint model. | 66,017 URLs, including 32,519 benign URLs and 33,498 malicious URLs | Capsule Networks, Independent Recurrent Neural Networks (IndRNN), attention mechanisms | 99.78 |
| **11** | Malicious URL Detection Based on Associative Classification Objectives[15] | The proposed system utilizes the Classification Based on Association (CBA) algorithm for detecting malicious URLs by extracting features from URLs and webpage content. | Not specified | CBA | 95.83%, |
| **12** | A malicious URLs detection system using optimization and machine learning classifiers[17] | The proposed system applied feature optimization by using a bio-inspired algorithm , specifically particle swarm optimization (PSO) to enhanced detection of malicious URLs. | Not specified | RF,  KNN,  AdaBoost,  Naïve Bayes and Support Vector Machine (SVM) | 97%,  97%,  97%,  99%,  99% |
| **13** | Phishing Detection System Through Hybrid Machine Learning Based on URL[18] | The proposed phishing detection system combines a hybrid model of machine learning algorithms, advanced feature selection techniques, and optimization methods to effectively identify and classify phishing URLs | over 11,000 phishing URL | 1. Decision Tree (DT)  2. Linear Regression (LR)  3. Support Vector Classifier (SVC) | 98.12% |
| **14** | An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework[19] | Analyze the performance of four machine learning techniques | 3,980,870 URL | Random Forest, Gradient Boost, XGBoost and AdaBoost | 99.63% |
| **15** | Malicious URL Prediction Using Machine Learning Techniques[20] | The proposed system in the paper aimed to enhance user security by predicting malicious URLs using machine learning algorithms, decision tree, and logistic regression. | 420,000 webpages | decision tree and logistic regression | - Decision Tree: Achieved an accuracy rate of 85%.  - Logistic Regression: Achieved a higher accuracy rate of 97.5%. |
| **16** | Detecting Malicious URLs Using Machine Learning[21] | The proposed system in the study involved applying a Variational Quantum Classifier (VQC) to cybersecurity datasets to detect fraudulent URLs | volume of the dataset and the required processing times, a first approximation was made with a reduced (200 observations), balanced (100 malicious URLs and 100  non-malicious ones) dataset | VQC | 97% |
| **17** | Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques[22] | The proposed system is a hybrid feature-based phishing detection approach that utilizes machine learning classifiers to identify phishing websites effectively. | with 6000 URLs, containing 3000 legitimate URLs and 3000 phishing URLs | Decision Tree, XGBoost, Random Forest, Support Vector Machine, and Naive Bayes. | high accuracy rate of 99.17% |
| **18** | Malicious URL Detection: A Comparative Study Objectives[24] | proposed system in the study involved collecting a dataset of 450,000 URLs, training various machine learning classifiers, and selecting the Random Forest model as the best performer for detecting malicious URLs. | of 450,000 URLs | Logistic Regression, Stochastic Gradient Descent, Random Forest, Support Vector Machine, Naïve Bayes, K-Nearest Neighbors, and Decision Tree | Not clearly define |
| **19** | Machine learning based phishing detection from URLs[25] | The proposed system in the research involved the development of a phishing detection system using machine learning algorithms and various features | 73,575 URLs, comprising 36,400 legitimate URLs and 37,175 phishing | Naive Bayes, Random Forest, kNN (n = 3), Adaboost, K-star, SMO, and Decision Tree . | The highest accuracy score achieved in the research was by the Decision Tree algorithm using NLP Features, with a precision of 96.4%, sensitivity of 97.7%, F-measure of 97.1%, and an overall accuracy of 97.02% |
| **20** | Classification of Malicious URLs Using Naive Bayes and Genetic Algorithm[26] | aims to classify safe and dangerous websites to mitigate financial losses for vulnerable websites. The system utilizes factor analysis of website categories and accurate identification of unknown information to distinguish between safe and malicious URLs. | Not specified | the Naive Bayes algorithm for website classification, combined with Genetic Algorithm optimization to enhance the classification process | Naive Bayes algorithm achieved a high accuracy probability of 96% |
| **21** | An intelligent identification and classification system for malicious uniform resource locators (URLs)[27] | The proposed system utilizes ensemble learning models in phases of preparation, learning, assessment, and deployment to detect and classify malicious URLs accurately. | 57,000 URL | ensemble of bagging trees(En\_Bag), ensemble of k-nearest neighbor (En\_kNN), ensemble of boosted decision trees(En\_Bos) , and ensemble of subspace discriminator (En\_Dsc) | Best Result was En\_Bag : 99.30 |
| **22** | A Comparative Study of Malicious URL Detection: Regular Expression Analysis, Machine Learning, and VirusTotal API[28] | The proposed system for detecting malicious URLs using machine learning involves data collection, preprocessing, and feature engineering to extract relevant characteristics. Machine learning models like Gradient Boosting Classifier are trained and evaluated for accurate detection, integrated with the VirusTotal API for verification, enabling real-time monitoring and continuous updates to enhance cybersecurity measures. | 11,054 samples with 32 features | 1. Naïve Bayes  2. Decision Tree  3. Support Vector Machine (SVM)  4. Gradient Boosting Classifier  5. Random Forest  6. Multi-layer Perceptron  7. K-Nearest Neighbors  8. Logistic Regression  9. CatBoost Classifier | Height score Gradient Boosting Classifier:97.4 |
| **23** | A Machine Learning Based Three-Step Framework for Malicious URL Detection[29] | propose a three-step framework consisting of segmentation, embedding, and machine learning for the detection of malicious URLs | 160,000 URLs | the Alphabet method and Token method, as well as embedding algorithms like Word2Vec, FastText, GloVe, and TF-IDF | Not specified |
| **24** | Machine Learning for Malicious URL Detection[30] | develop a model, the MuD (Malicious URL Detection) model, for effectively detecting malicious URLs using machine learning techniques | 11,055 URLs | 1.Support Vector Machine  2.Logistic Regression  3.Naïve Bayes | Naïve Bayes:100% |
| **25** | Phishing URL detection: A real-case scenario through login URLs[31] | Sánchez-Paniagua, M., Fernández, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. IEEE Access, 10, 42949-42960. | Not mentioned | 1.lightGBM  2.XGBoost  3.AdaBoost  4.RF  5.Knn  6.SVM  7.LR  8.NB  9.TF-IDF+N-gram | Best result was .TF-IDF+N-gram : 96.93 |

Table2: Summary of background literature that doesn’t encloud experience

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **paper** | **Year** | **Summary** | **Challenge & Recommendation** | **Finding/ Outcomes** |
| **1** | Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions[2] | 2022 | Provides a taxonomy of deep learning algorithms for phishing detection, discusses challenges and future directions, and evaluates the performance of deep learning techniques in practical contexts | Challenges: Limited availability of high-quality phishing datasets, computational requirements, manual feature engineering in traditional ML. Recommendations: Address dataset quality and diversity issues, explore efficient computational solutions, automate feature extraction in ML. | Identifies limitations, gaps, and areas for improvement in using deep learning for phishing detection. |
| **2** | A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Computer Science Review, 29, 44-55[3] | 2018 | Discusses the development of anti-phishing tools, including database-driven approaches like blacklists, aiming to reduce false positives and increase true positives in identifying phishing emails. | Challenges: Short-lived phishing websites, limitations of legal actions. Recommendation: User education to raise awareness about phishing risks. | Emphasizes the importance of reducing false positives in anti-phishing tools to enhance user confidence in email filtering systems. |
| **3** | Machine Learning Techniques for Detection of Website Phishing: A Review for Promises and Challenges[9] | 2021 | Comprehensive review of ML techniques for phishing detection, challenges faced, and proposals for improvement | Challenges include inefficiency on large datasets, need for automated frameworks. Recommendation: propose automated framework based on ensemble learning and deep learning | ML methods effective in eradicating phishing, ongoing research for new approaches, challenges include data size, overfitting, and low accuracy, need for automated framework. |
| **4** | Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions[16] | 2022 | Reviewed 91 studies from 2012 to 2021 on ML/DL in classifying malicious URLs. Classified websites as Arabic or English. Provided taxonomies on detection aspects like language, URL features, ML techniques, and datasets. Discussed challenges affecting ML detection quality. | - Data sample size limitation in reviewed papers. - Recommended evaluating ML models with sufficient samples. - Balancing techniques suggested for improving detection rate. - Big data collection challenges due to processing requirements. | - Lexical features most used for detecting malicious URLs in both Arabic and non-Arabic content.  - Arabic website studies did not utilize network-based features.  - SVM model achieved 86.72% accuracy in detecting suspicious Arabic tweets. |
| **5** | An Identification and Analysis of Harmful URLs through the Application of Machine Learning Techniques[23] | 2024 | The research paper delves into the significant cybersecurity threat posed by malicious URLs, which can compromise user security and result in substantial financial losses. Traditional detection methods relying on blacklists are insufficient in addressing the dynamic nature of these threats, leading to the adoption of machine learning approaches for more effective detection. The paper offers a structured insight into various aspects of identifying and analyzing harmful URLs through machine learning techniques, focusing on feature representation, algorithm design, and formally defining the machine learning task of identifying malicious URLs. | One of the key challenges highlighted in the research is the need for ample samples with a balanced ratio of normal to malicious URLs to enhance detection accuracy. Balancing strategies are recommended to address this issue while maintaining an adequate sample size. Additionally, the challenge of adapting machine learning models to swiftly adjust to evolving trends, especially in the face of polymorphic attacks where URL structures are regularly modified, is emphasized. The recommendation includes the development of adaptable models capable of withstanding such dynamic threats while ensuring data confidentiality during the training process. | The research paper showcases the effectiveness of machine learning techniques in detecting malicious URLs. Various machine learning algorithms such as SVM, RF, NB, LSTM, LR, GB, DT, and deep learning methods have been explored for this purpose. |
| **6** | Assessment of End-User Susceptibility to Cybersecurity Threats in Saudi Arabia by Simulating Phishing Attacks[6] | 2020 | The study assessed cybersecurity knowledge and awareness in Saudi Arabia through phishing attack simulations, including clone phishing, spear-phishing, and social networking phishing. The research aimed to evaluate end-user susceptibility, analyze attack effectiveness, and understand user behaviors in response to cyber threats. | The challenge identified was the high susceptibility of participants, particularly students, to phishing attacks. The recommendation was to implement cybersecurity awareness programs and educational initiatives to enhance user knowledge and behavior in cyberspace. | The findings revealed a significant vulnerability among end-users in Saudi Arabia, with a high percentage falling victim to various phishing attacks. The study emphasized the importance of improving cybersecurity knowledge and fostering a culture of security awareness to effectively mitigate cyber threats. |

As you can see from the above tables the result of using machine learning seems promising where it registers accuracy scores from 0.88 till 1.00 which means that we can eliminate the risk of phishing URLs by employing machine learning . however the used dataset was limited to specific area which may lead to unreliable result in some studies. In our research we will examine machine learning capabilities to detect malicious URLs on recent dataset contain malicious and legit URLs then we will evaluate the result.

1. **Objective**

This research aims to evaluate Machine learning algorithms' capabilities in detecting malicious URLs, this will give us more insight into the effectiveness of using machine learning algorithm to detect malicious URLs.

**Research question**

We will do a set of experiments to answer the following question :

* Can machine learning algorithms detect malicious URLs with high accuracy and low error rates?
* What algorithm has the highest accuracy among ML algorithms?

1. **Methodology**

## **Research Design**

In this research we will go through the following stages:

1. Collecting datasets from public resources such as Kagle, URLhaus, and PhishTank
2. Exploring the collected dataset by visualizing the content of the dataset and checking the validity of the assigned classification using third-party services such as VirusTotal [33] which is a well-known website for identifying malicious URLs
3. Cleaning the collected dataset, if needed by removing unnecessary data
4. Analysing the dataset and preparing it for modelling by creating new column for URLs related feature that would help to detect malicious URLs and visualize them to have clear insight
5. Splitting the collected dataset to testing and training dataset to feed the model
6. Apply ML algorithms
7. Evaluate the used algorithms

These steps will be used during the explements, and we will skip som of them if it was already done in previous explements .

## **Dataset**

In the first phase of this research, we gathered datasets from different websites such as Kagle, URLhaus, and PhishTank. Details about the collected data in the bellow table:

|  |  |  |
| --- | --- | --- |
| **Dataset Name** | **Size** | **Source** |
| Malicious and Benign Website dataset | 69699 | kagle: <https://www.kaggle.com/datasets/jackcavar/malicious-and-benign-website-dataset> |
| Phishtank | 60733 | Phishtank: https://phishtank.org |
| Alexa | 999999 | https://www.kaggle.com/datasets/cheedcheed/top1m |

We used the first dataset as our starting point for this research and noticed some issues with the results. Subsequently, we used two additional datasets.

1. **Data Analysis**

In this phase, we conducted several experiments, so we will devid this section by experiments

## **Experiment 1:**

### **Collecting datasets**

The dataset we will use in this experiment had a predefine description by the publisher, where he mentions that :

the dataset construct of :

* 20,175 phishing websites sourced from PhishTank and PhishStats.
* 49,524 legitimate websites extracted from the top 1 million websites listed by Alexa.

It's divided into two parts: an Excel spreadsheet and a corresponding TXT file containing the initial page scraped from each URL. Additional information about the websites, such as the number of redirects per request and WHOIS information, was also collected.

The dataset was collected over a period of 38 days. Among various methods tested, LightGBM was found to perform best with this dataset. This suggests that models built on this framework can achieve high accuracy, especially when dealing with larger datasets, *Kaggle[32]*.

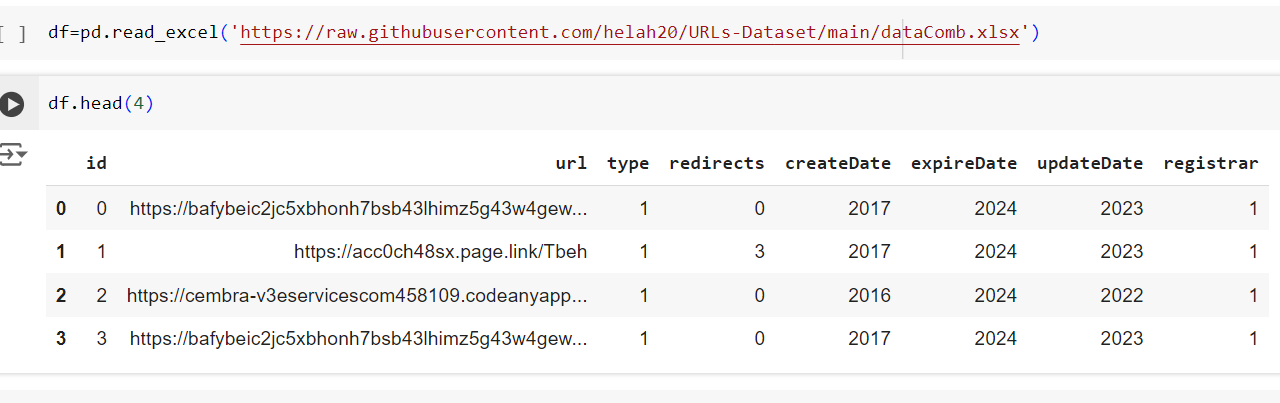


Figure Loading Dataset

### **Exploring the Dataset**

Now we have a simple overview of the dataset let's explore it by ourselves. So, we did a manual check to ensure the reliability of the dataset by selecting random URLs and then testing them on VirusTotal[33] to check the accuracy of classification. We found that it was valid and accurate, and below is a sample of the result.

1. URL with 0 classification (which indicates it is not malicious)

A screenshot of a computer

Description automatically generated

Figure URL classification validation

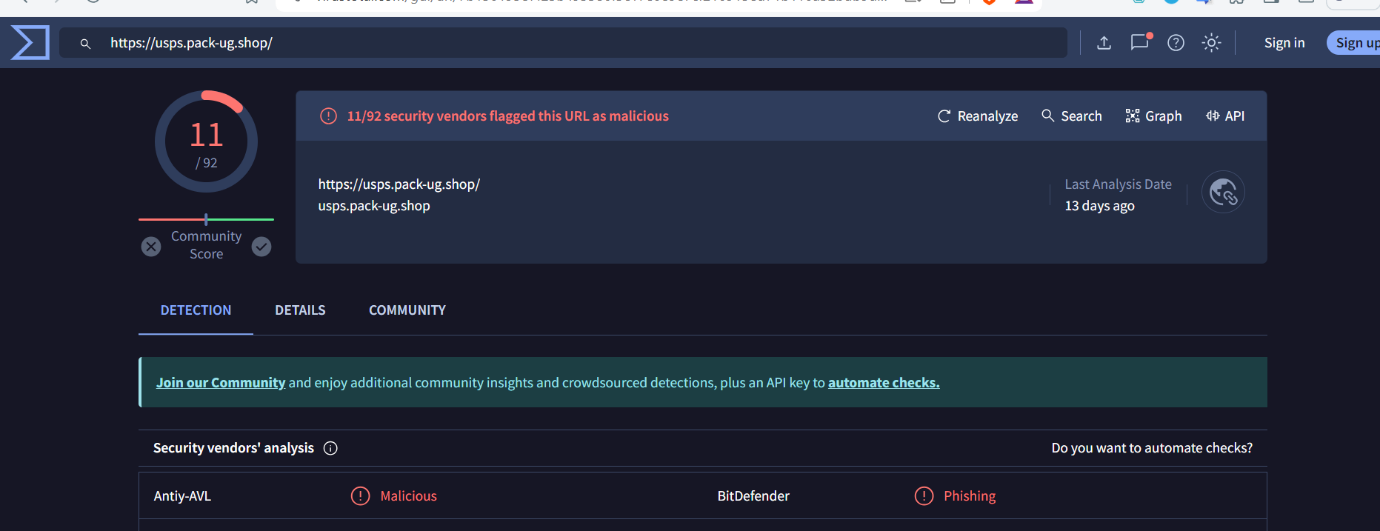
1. URL with 1 classification(which indicates it is malicious)

Figure URL classification validation

We can see that these URLs are still valid for analysis, as the sample shows that the classifications remain accurate and the sites have been active recently.

### **Cleaning the collected dataset**

We use Google Colab for analysis since it supports Python and we subscribe for a month to speed up the CPU processing.

After the exploratory phase, we found that the dataset was clean and there was no duplicated or unnecessary data.

A sample of the applied result is shown in the bellow screenshots

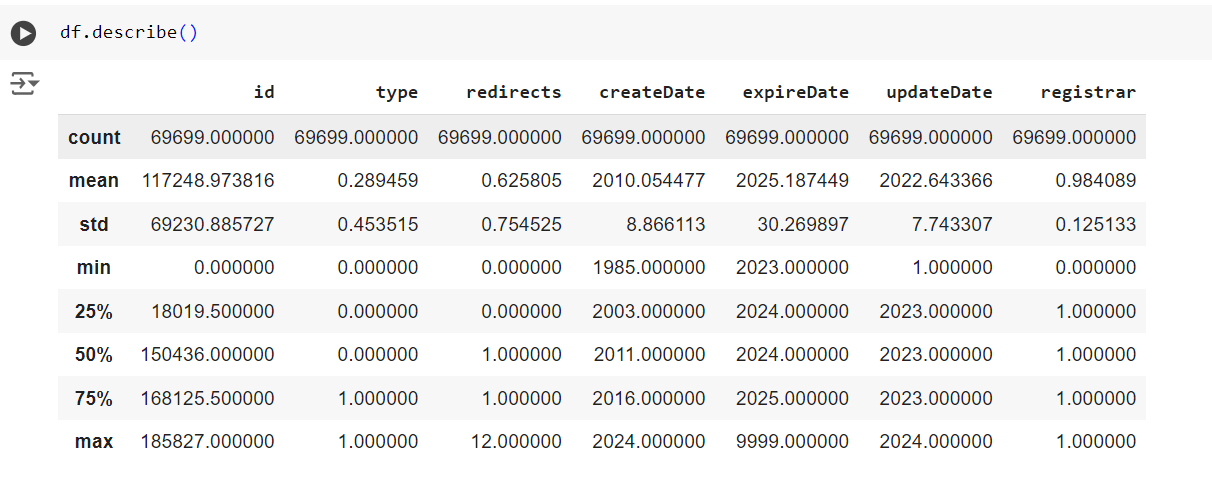


Figure : Description of the dataset

### **Analyzing the dataset and preparing it for modeling**

In this phase, we make some plots as shown in the bellow screenshots

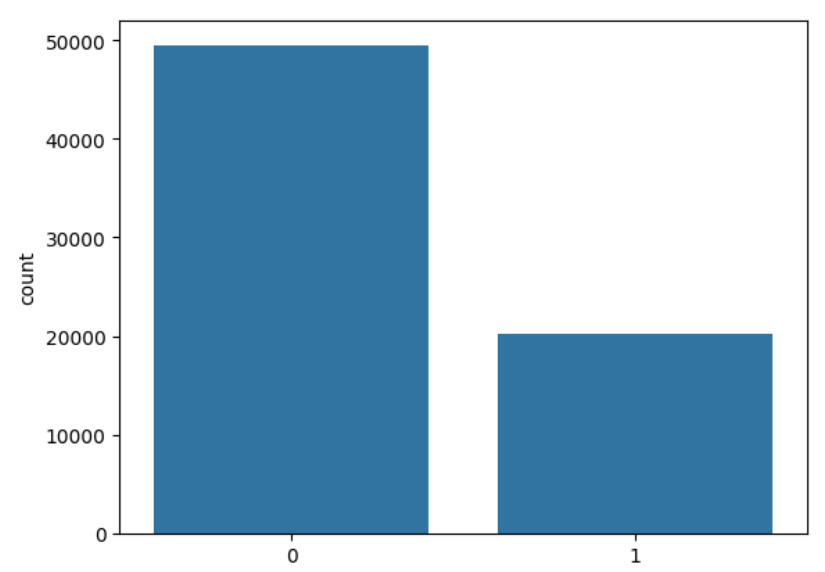


Figure Distribution of URL types: malicious(1) URLs via legit(0)

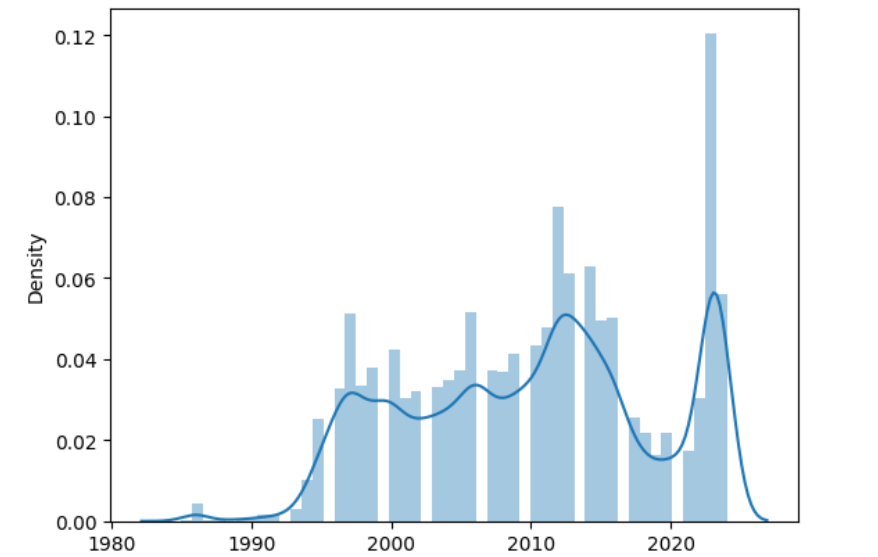


Figure Distribution of creation dates:

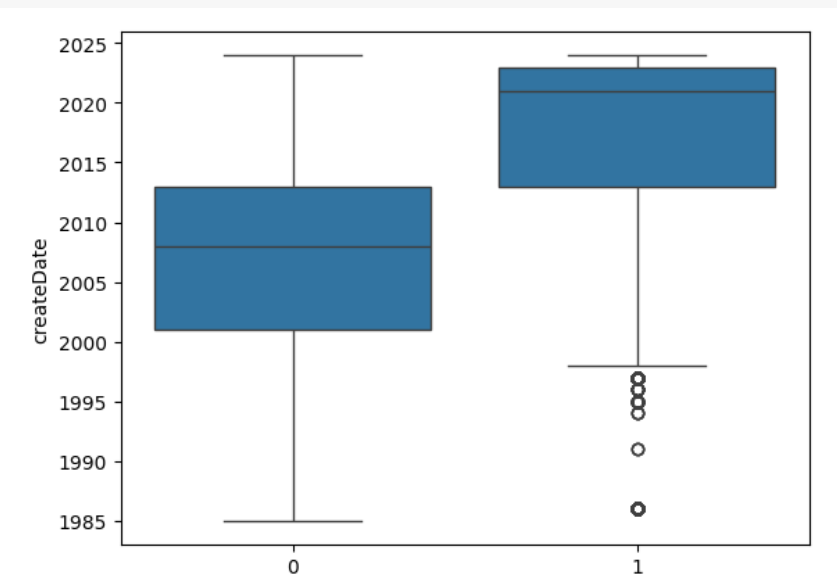


Figure Relations between creation date and URLs type: more recent more suspicious

We did add a feature for URL length to maximize the accuracy and enhance the dataset as per the following cod and we delete URL column .

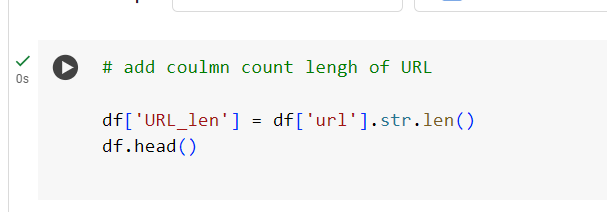


Figure : add a feature for URL length



Figure : Delete URL column

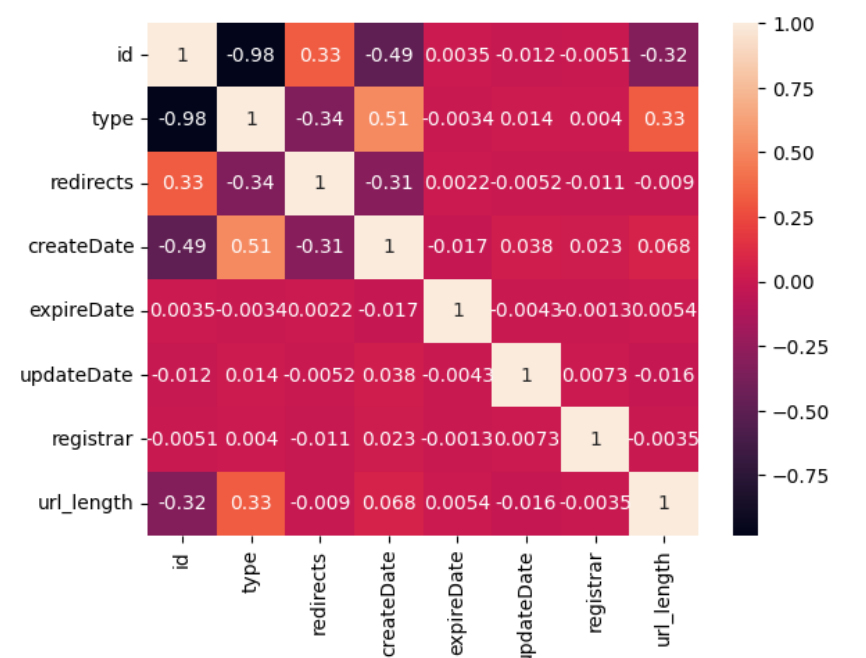


Figure Relation between variables

We can extract some information regarding the dataset from the previous plots, which can be summarized in the following points:

* The number of non-malicious URLs is greater than the number of malicious ones.
* The creation dates of the URLs start from the late 1980s and have increased in recent years.
* The more recent the creation date, the more likely the URL is to be malicious.

### **Splitting the collected dataset**

A screenshot of a computer code

Description automatically generated

Figure : splitting the dataset

### **Apply ML algorithms**

A screen shot of a computer code

Description automatically generatedWe start creation classifier ; we used stacking techniques that allow us to run multiple algorithms in the same time . we used Random Forest ,KNeibors and Decision Tree algorithm initially .

Figure stacking

### **Evaluate the used algorithms**

In this Phase we define cross validation function to evaluate our models

A screenshot of a computer code

Description automatically generated

Figure cross validation

"After that, we calculate the baseline score, which basically predicts the most common value and evaluates its accuracy and we got 71% of accuracy

A screenshot of a computer code

Description automatically generated

Figure baseline

Then we apply the model and find out the high accuracy; this stage , where the model show 100% accuracy across all algorithm , this means that ether the models so good or the models is overfitting to the training data. (Time , more than 45 minutes)

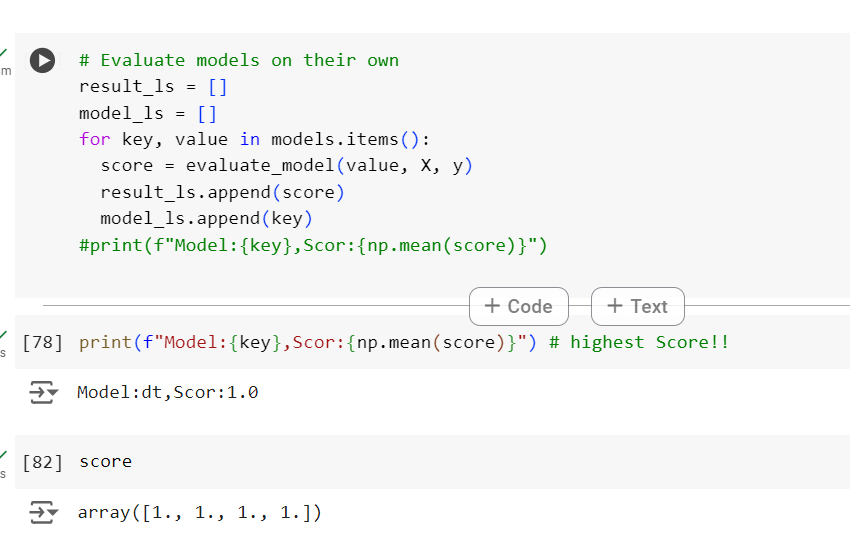


Figure accuracy

## **Experience 2:**

We used the same dataset as per in the previous experiment , but we tried on single classifier and use new techniques for splitting the dataset

1. Split the data as per the following code

A screenshot of a computer code

Description automatically generated

Figure Split the data

1. Applying logistic regression Model

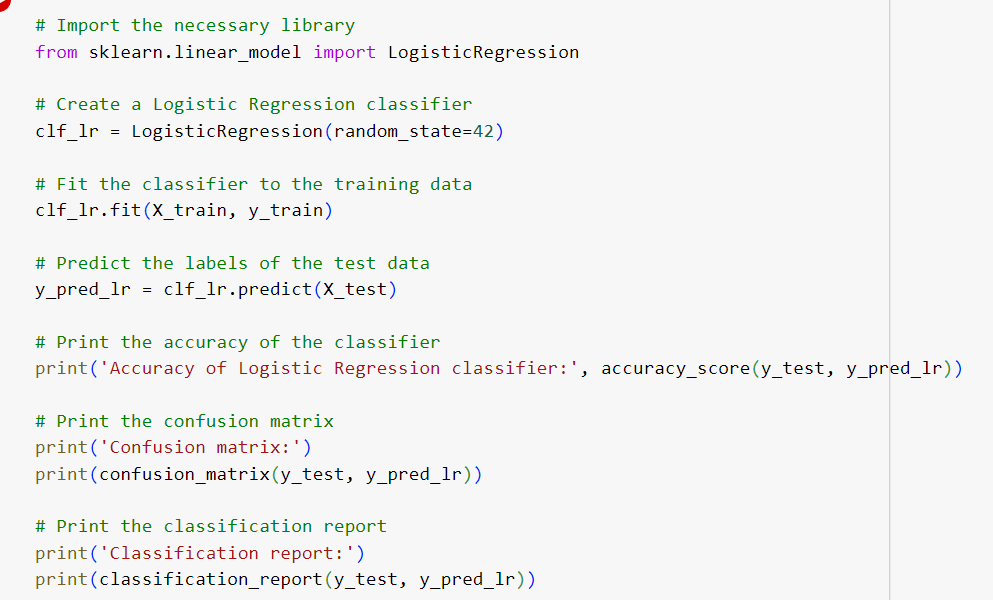


Figure Applying logistic regression

And got the same result of precise as experience 1

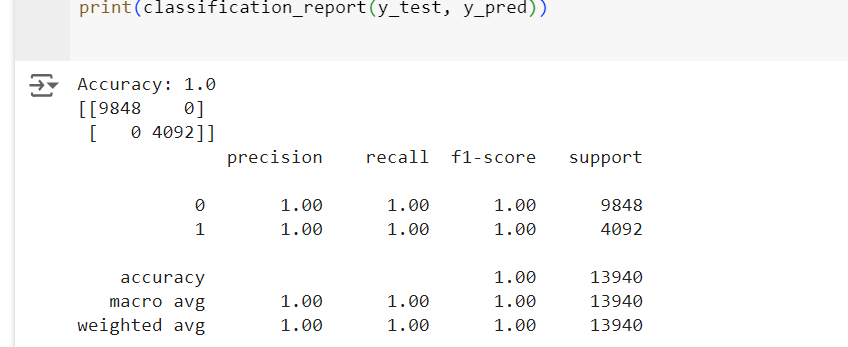


Figure Acuracy

## **Experience 3:**

We try to reduce the amount of non-malicious type of data and fix the overfitting problem but with no lock, same problem appeared, and we got 100%.this means that the data seems to be useless so we will move to another approach.

## **Experience 4:**

In this approach we will use new dataset , extracted from the following sites :

* Alexa dataset – 100 URLs
* Phishtank- 100 URLs

### **Collecting datasets**

**Loading Alexa dataset**

A screenshot of a computer

Description automatically generated

Figure Loading dataset1

We notice that the dataset does not include header ,type nor protocol .

**Loading Phishtank dataset**



Figure Phishtank dataset

We can notice that there is an unneeded column that will not serve the experiments

### **Exploring the Dataset**

To save time, we will directly explore the final dataset. Initially, we merged a new dataset from our sources containing only two columns: URL and its corresponding type (either good or bad). Then, we merged the two datasets to create the following comprehensive dataset:

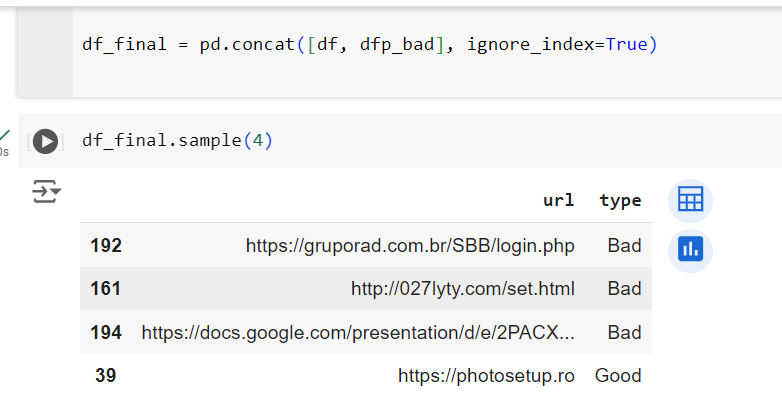


Figure : final dataset

### **Cleaning the collected dataset**

We conducted the cleaning phase before merging the data. In the Alexa dataset, we added a header row and a type column indicating "Good", after which we dropped the ID column and added the URL protocol(http, https ). On the other hand, the Phishtank dataset included numerous columns that were unnecessary, so we dropped them and retained only the URL column. Additionally, we added the type column to the dataset, indicating "Bad"."

### **Analyzing the dataset and preparing it for modeling**

We used only 200 rows to test our approach initially to reduce processing time. If successful, we plan to expand the dataset later. To create a robust dataset, we merged a sample of 100 rows from each source, resulting in the following dataset.

We prepared the dataset and extracted the following features from the URL:

* Sensitive Word Count
* Special Character Count
* URL Length
* Dot Count
* Parameter Count
* IP Presence

We then conducted data analysis to examine the relationship between these features and the label.

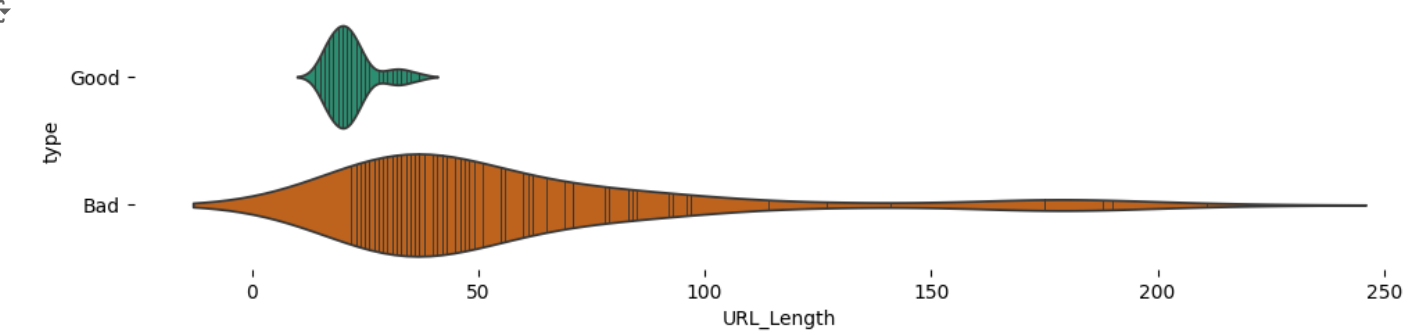


Figure type vs URL\_Length

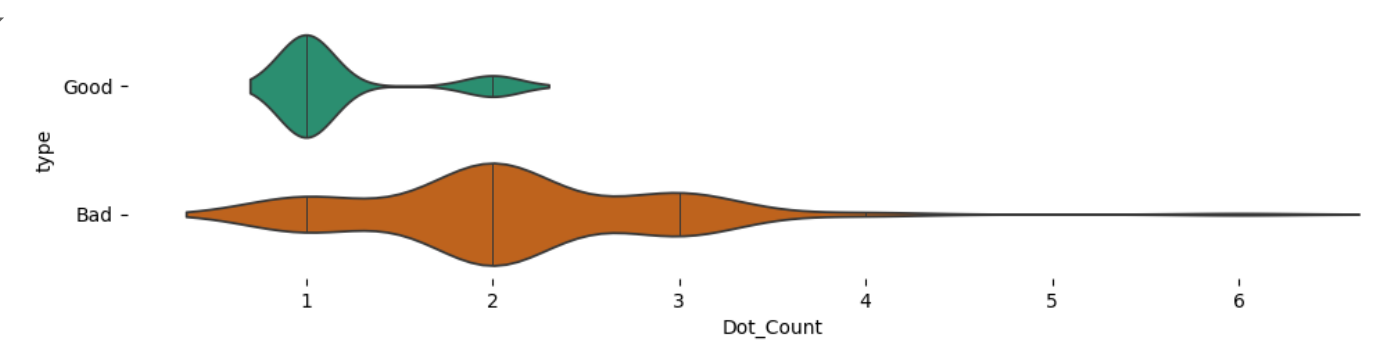


Figure type vs Dot Count

We can conclude from the plot the following findings:

* The longer the URL length, the more likely it is to be bad.
* The more dots in the URL, the more likely it is to be bad.

### **Splitting the collected dataset**

In this phase, we start preparing the dataset for applying ML algorithm be deleting url column and splitting the data

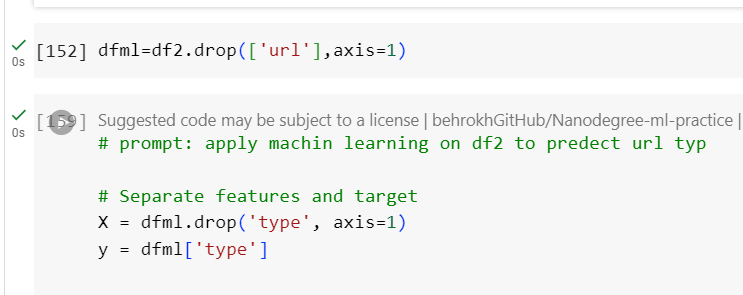


Figure splitting the data

### **Apply ML algorithms**

In this phase, we apply Random forest algorithm as per the pic

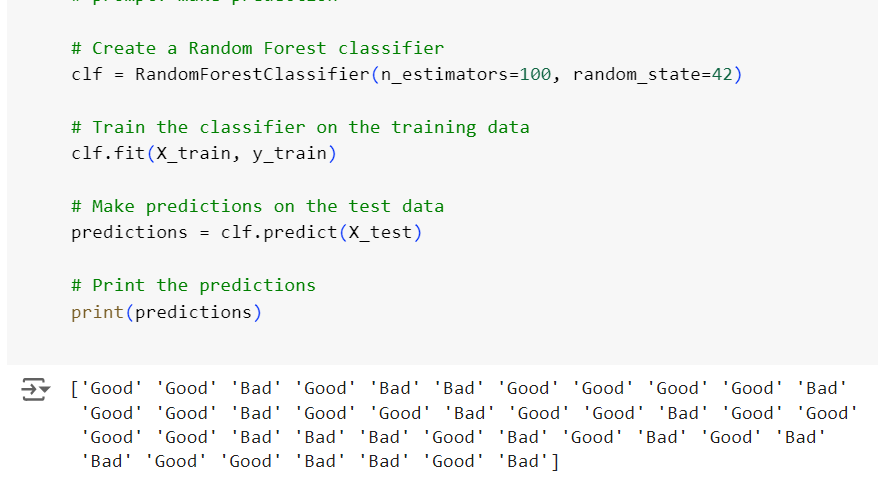


Figure Random forest

### **Evaluate the used algorithms**

In this phase, we evaluate the accuracy score as per the following code and we got 82%

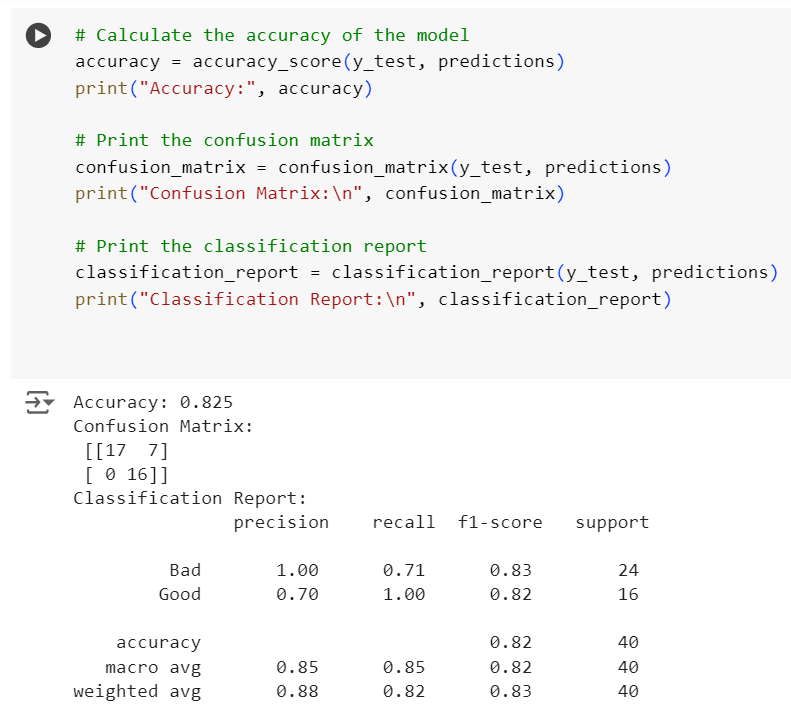


Figure accuracy

## **Experience 5:**

### **Apply ML algorithms**

On the same dataset we apply stacked algorithms as per the bellow

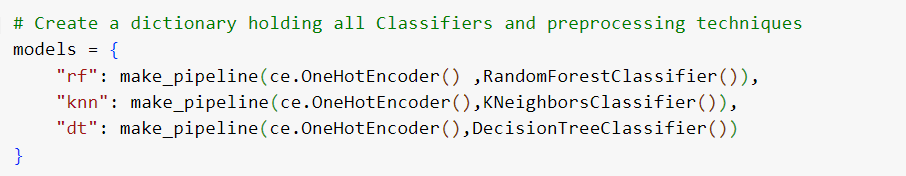


Figure apply stacking

### **Evaluate the used algorithms**

Then we define function to evaluate the scores

A computer code with green text

Description automatically generated with medium confidence

Figure evaluate the scores

Then we calculate the baseline and the result was 50%

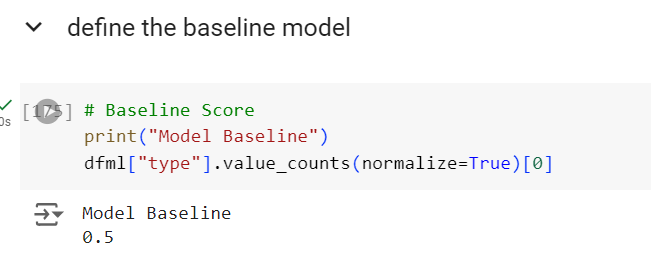


Figure baseline

Then we apply the stack an go 90% for decision tree algorithm

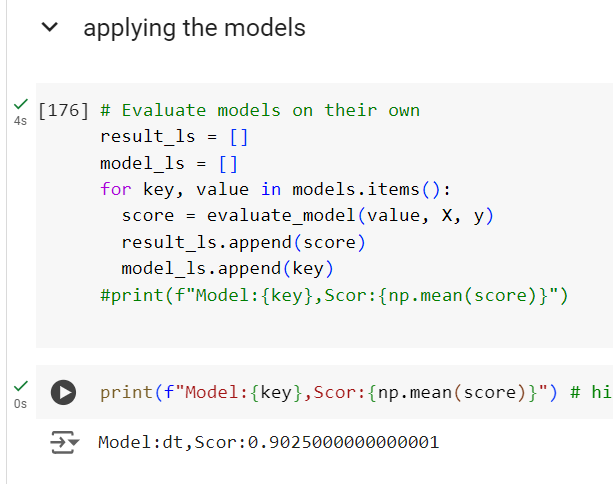


Figure Accuracy

Below is a chart defining a comparison between the algorithm scores and baseline



Figure comparison

Then we apply grid search to find out the best parameter to use and after 2 hour of execution the result was as pre the following pics



Figure grid search

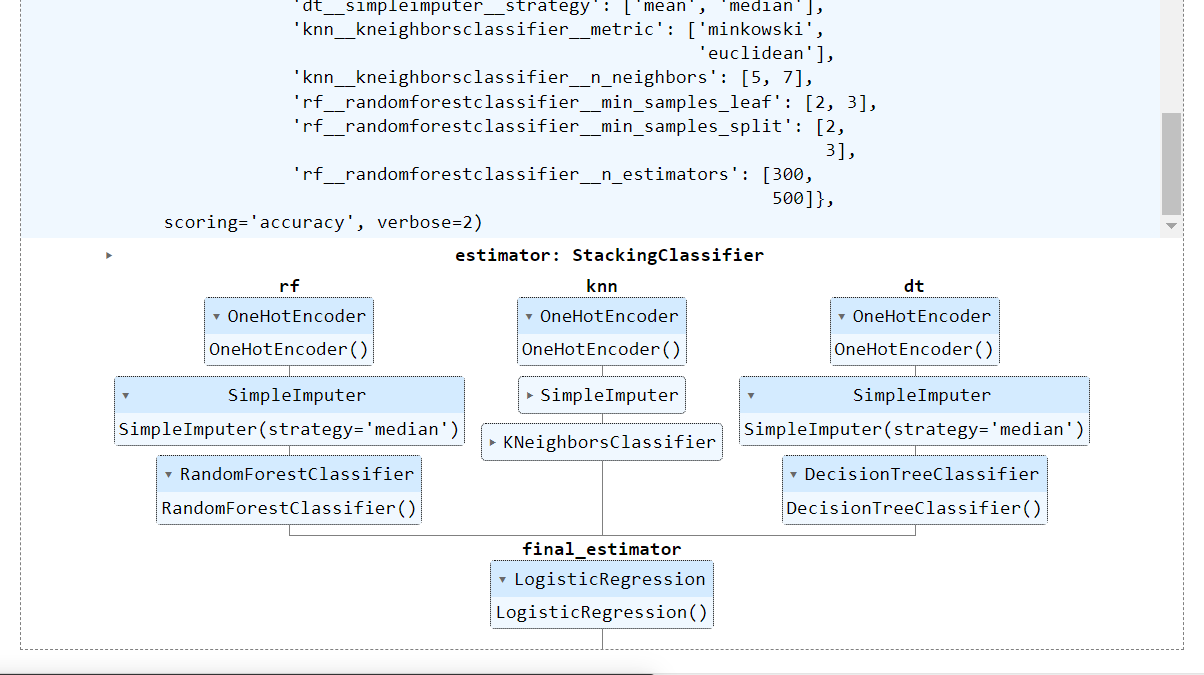


Figure grid search result

And the best score that we reach was 93%

A screenshot of a computer code

Description automatically generated

Figure After applying grid search bert parameter

1. **Conclusion**

In conclusion, this study focuses on utilizing machine learning algorithms to detect malicious URLs. We began by conducting a literature review of 31 papers to understand the state of the art. We collected datasets from various resources and proceeded to analyze them. During this process, we encountered challenges in finding the most suitable dataset, which ultimately enhanced the reliability of our research. We conducted three failed experiments before achieving success with two experiments, where we attained accuracy scores ranging from 80% to an optimized 93%. Moving forward, we intend to expand the dataset further and explore different scenarios to determine the potential accuracy scores.

1. **References**
2. Indrasiri, P. L., Halgamuge, M. N., & Mohammad, A. (2021). Robust ensemble machine learning model for filtering phishing URLs: Expandable random gradient stacked voting classifier (ERG-SVC). Ieee Access, 9, 150142-150161.
3. Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. Ieee Access, 10, 36429-36463.
4. Qabajeh, I., Thabtah, F., & Chiclana, F. (2018). A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Computer Science Review, 29, 44-55.
5. Mughaid, A., AlZu’bi, S., Hnaif, A., Taamneh, S., Alnajjar, A., & Elsoud, E. A. (2022). An intelligent cyber security phishing detection system using deep learning techniques. Cluster Computing, 25(6), 3819-3828.
6. Brindha, R., Nandagopal, S., Azath, H., Sathana, V., Joshi, G. P., & Kim, S. W. (2023). Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification. Computers, Materials & Continua, 74(3).
7. Aljeaid, D., Alzhrani, A., Alrougi, M., & Almalki, O. (2020). Assessment of end-user susceptibility to cybersecurity threats in Saudi Arabia by simulating phishing attacks. Information, 11(12), 547.
8. Sahingoz, O. K., Baykal, S. I., & Bulut, D. (2018). Phishing detection from urls by using neural networks. Computer Science & Information Technology (CS & IT), 41-54.
9. Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020, July). Detection of phishing websites by using machine learning-based URL analysis. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.
10. Odeh, A., Keshta, I., & Abdelfattah, E. (2021, January). Machine learningtechniquesfor detection of website phishing: A review for promises and challenges. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0813-0818). IEEE.
11. Zamir, A., Khan, H. U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A., & Hamdani, M. (2020). Phishing web site detection using diverse machine learning algorithms. The Electronic Library, 38(1), 65
12. Vajrobol, V., Gupta, B. B., & Gaurav, A. (2024). Mutual information based logistic regression for phishing URL detection. Cyber Security and Applications, 2, 100044.
13. Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. arXiv preprint arXiv:1802.03162.
14. Do Xuan, C., Nguyen, H. D., & Tisenko, V. N. (2020). Malicious URL detection based on machine learning. International Journal of Advanced Computer Science and Applications, 11(1).
15. Yuan, J., Chen, G., Tian, S., & Pei, X. (2021). Malicious URL detection based on a parallel neural joint model. IEEE Access, 9, 9464-9472.
16. Kumi, S., Lim, C., & Lee, S. G. (2021). Malicious URL detection based on associative classification. Entropy, 23(2), 182.
17. Aljabri, M., Altamimi, H. S., Albelali, S. A., Al-Harbi, M., Alhuraib, H. T., Alotaibi, N. K., ... & Salah, K. (2022). Detecting malicious URLs using machine learning techniques: review and research directions. IEEE Access, 10, 121395-121417.
18. Heryanto, A., Ab Razak, M. F., Raffei, A. F. M., Phon, D. N. E., Kasim, S., & Sutikno, T. (2020). A malicious URLs detection system using optimization and machine learning classifiers. Indonesian Journal of Electrical Engineering and Computer Science, 17(3), 1210-1214.
19. Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., & Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805-36822.
20. Nowroozi, E., Jadalla, N., Ghelichkhani, S., & Jolfaei, A. (2024). Mitigating Label Flipping Attacks in Malicious URL Detectors Using Ensemble Trees. arXiv preprint arXiv:2403.02995.
21. Aalla, H. V. S., Dumpala, N. R., & Eliazer, M. (2021). Malicious URL prediction using machine learning techniques. Annals of the Romanian Society for Cell Biology, 2170-2176.
22. Reyes-Dorta, N., Caballero-Gil, P., & Rosa-Remedios, C. (2024). Detection of malicious URLs using machine learning. Wireless Networks, 1-18.
23. Das Guptta, S., Shahriar, K. T., Alqahtani, H., Alsalman, D., & Sarker, I. H. (2024). Modeling hybrid feature-based phishing websites detection using machine learning techniques. Annals of Data Science, 11(1), 217-242.
24. Karve, S. M., Kakad, S., Amol, S., Gavali, A. B., Gavali, S. B., & Shirkande, S. T. (2024). An Identification and Analysis of Harmful URLs through the Application of Machine Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 12(17s), 456-468.
25. Janet, B., & Kumar, R. J. A. (2021, March). Malicious URL detection: a comparative study. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 1147-1151). IEEE.
26. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357.
27. Koca, M., Avcı, İ., & Al-hayani, M. A. S. (2023). Classification of malicious urls using naive bayes and genetic algorithm. Sakarya University Journal of Computer and Information Sciences, 6(2), 80-90.
28. Abu Al-Haija, Q., & Al-Fayoumi, M. (2023). An intelligent identification and classification system for malicious uniform resource locators (URLs). Neural Computing and Applications, 35(23), 16995-17011.
29. Misquitta, J., & Kannan, A. (2023, November). A Comparative Study of Malicious URL Detection: Regular Expression Analysis, Machine Learning, and VirusTotal API. In International Congress of Electrical and Computer Engineering (pp. 219-232). Cham: Springer Nature Switzerland.
30. Chen, Qisheng & Omote, Kazumasa. (2023). A Machine Learning Based Three-Step Framework for Malicious URL Detection. 10.21203/rs.3.rs-3150943/v1.
31. Wejinya, Gold & Bhatia, Sajal. (2021). Machine Learning for Malicious URL Detection. 10.1007/978-981-15-8289-9\_45.
32. Sánchez-Paniagua, M., Fernández, E. F., Alegre, E., Al-Nabki, W., & Gonzalez-Castro, V. (2022). Phishing URL detection: A real-case scenario through login URLs. IEEE Access, 10, 42949-42960.
33. <https://www.kaggle.com/datasets/jackcavar/malicious-and-benign-website-dataset>
34. [https://www.virustotal.com/](https://www.virustotal.com/gui/url/11c431ed416411490d3f7643a3883570557c6c7a53e644395578e5dd18a22c5d)
35. Phishtank opensourcd platform. <http://phishtank.org>
36. **Appendix**
    1. Research Code : TBA
    2. Dataset: TBA
    3. Presentation : TBA