**Project report on**

**Phishing attack detection using machine learning techniques**

**CIS 660**

**Professor: Dr. Sunnie Chung**

**Student :Chaitanya Mungi (CSUID:2783234)**

**Date: 12/3/2021**

**Abstract**

**This is the detailed project report for the project “Phishing attack detection using machine learning techniques”.**

**This report describes how a prediction model can be made to predict whether a URL is malicious or legitimate. This document describes how features of training data set can be created/selected to train a model , preprocessing of dataset and also How we can create different classification models in python. Datasets used and other useful resources used to create this project are mentioned in the “resource” section. Improvements and future work is also mentioned in the “future work” section.**

**Table of Contents**

1. **Title page**
2. **Table of Contents.**
3. **Abstract.**
4. **Introduction.**
   1. **Cyberattacks since last few days**
   2. **Use of machine learning to avoid cyber attacks**
   3. **What is phishing attack??**
   4. **Project specific.**
5. **Steps performed for the project.**
6. **Data collection.**
7. **Feature selection/creation.**
8. **Machine learning Algorithms used** 
   1. **Decision tree classifier**
   2. **KNN algorithm**
   3. **Support vector machine algorithm.**
9. **Code and outputscreenshots:**
   1. **feature selection and preprocessing.**
   2. **Machine learning algorithms used**
   3. **Web applicaiton created.**
10. **Problems and errors encountered.**
11. **Resources**

**Introduction:**

**1.Cyber-attacks since last few days**

Chart

Description automatically generated

**As we can see from the above figure the cost of cyber security is increasing globally as the cyberattacks and the ways to perform cyberattacks by hackers are increasing day by day.**

**Different cyber attacks like DDos attack, phishing attack, website fingerprinting attack are being performed to get money, personal details like username/password, credit card details. Also new cyber attacks like meltdown and specter got introduce in last few years(meltdown-specter 2018).**

**No of cyber-attacks in USA are also increased significantly in last few days. The figure below shows the increase in the cost of overall cybersecurity industry over last few years.**

Chart

Description automatically generated

**2. Use of machine learning to avoid cyber attacks**

what is machine learning? A subset of artificial intelligence (AI), machine learning is a set of techniques and technologies using algorithms to examine large volumes of information or training data to discover unique patterns, which it can then analyse, group and use to make predictions. Machine learning can learn from data and make decisions without the aid of human interaction.

With their ability to sort through millions of datasets, recognise patterns and identify anomalies, machine learning systems are increasingly being used to proactively uncover data breaches or intrusions.

[Cybersecurity solutions powered by machine learning](https://www.dualog.com/protect) use data from past cyberattacks and threats, learn from the data, and identify and respond to similar threats.

An adaptive monitoring solution that leverages machine learning eliminates any threat in its infancy and sends off alerts to the right people before it can compromise your systems and operations.

1. Using machine learning to detect malicious activity and stop attacks

Machine learning algorithms will help businesses to detect malicious activity faster and stop attacks before they get started. David Palmer should know. As director of technology at UK-based start-up Darktrace – a firm that has seen a lot of success around its machine learning-based Enterprise Immune Solution since the firm’s foundation in 2013 – he has seen the impact on such technologies.

Palmer says that Darktrace recently helped one casino in North America when its algorithms detected a data exfiltration attack that used a “connected fish tank as the entryway into the network.” The firm also claims to have prevented a similar attack during the [Wannacry ransomware](https://www.csoonline.com/article/3227906/ransomware/what-is-wannacry-ransomware-how-does-it-infect-and-who-was-responsible.html) crisis last summer.

2. Using machine learning to analyze mobile endpoints

Google is using machine learning to analyze threats against mobile endpoints running on Android , as well as identifying and removing malware from infected handsets, while cloud infrastructure giant Amazon has [acquired start-up harvest.AI](https://techcrunch.com/2017/01/09/amazon-aws-harvest-ai/) and launched [Macie](https://www.csoonline.com/article/3217029/cloud-security/amazon-macie-automates-cloud-data-protection-with-machine-learning.html), a service that uses machine learning to uncover, sort and classify data stored on the S3 cloud storage service.

3. Using machine learning to enhance human analysis

In 2016, MIT’s Computer Science and Artificial Intelligence Lab (CSAIL) developed[a system called AI2](http://news.mit.edu/2016/ai-system-predicts-85-percent-cyber-attacks-using-input-human-experts-0418), an adaptive machine learning security platform that helped analysts find those ‘needles in the haystack’. Reviewing millions of logins each day, the system was able to filter data and pass it onto the human analyst, reducing alerts down to around 100 per day.

4. Using machine learning to automate repetitive security tasks

The real benefit of machine learning is that it could automate repetitive tasks, enabling staff to focus on more important work. Palmer says that machine learning ultimately should aim to “remove the need for humans to do repetitive, low-value decision-making activity, like triaging threat intelligence. “Let the machines handle the repetitive work and the tactical firefighting like interrupting ransomware so that the humans can free up time to deal with strategic issues — like modernizing off Windows XP — instead.”

5. Using machine learning to close zero-day vulnerabilities

A team at Arizona State University used machine learning to monitor traffic on the dark web to [identify data relating to zero-day exploits](http://www.forbes.com/sites/kevinmurnane/2016/08/08/machine-learning-goes-dark-and-deep-to-find-zero-day-exploits-before-day-zero/), according to Forbes. Armed with this type of insight, organizations could potentially close vulnerabilities and stop patch exploits before they result in a data breach.

Companies using machine learning against cyberattacks:

**Microsoft** - Microsoft uses its own cybersecurity platform,  [Windows Defender Advanced Threat Protection (ATP)](https://www.microsoft.com/en-us/windowsforbusiness/windows-atp?ocid=cx-blog-mmpc), for preventative protection, breach detection, automated investigation and response.

**Chronicle** - Its first product, Backstory, analyzes large amounts of security data (such as internal network activity, known bad domains and suspected malware) and uses machine learning to condense it into more easily digestible insights.

**Splunk** -  Products like [Splunk Enterprise Security](https://www.splunk.com/en_us/software/enterprise-security.html) and [Splunk User Behavior Analytics](https://www.splunk.com/en_us/software/user-behavior-analytics.html) use machine learning to detect threats so they can be quickly eliminated.

**Sqrrl** - [Sqrrl](https://www.builtinboston.com/company/sqrrl) has designed a cyber-threat hunting platform that searches through networks to find code that can evade security measures in place. The product uses machine learning to turn data points into a behavior map, which acts as a visual representation of a computer network and shows where threats could be coming in.

**3.What is phishing attack??**

Phishing is a type of [social engineering attack](https://www.imperva.com/learn/application-security/social-engineering-attack/) often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a [malicious](https://www.imperva.com/learn/application-security/malware-detection-and-removal/) link, which can lead to the installation of malware, the freezing of the system as part of a [ransomware attack](https://www.imperva.com/learn/application-security/ransomware/) or the revealing of [sensitive information](https://www.imperva.com/learn/data-security/sensitive-data/).

An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

Moreover, phishing is often used to gain a foothold in corporate or governmental networks as a part of a larger attack, such as an [advanced persistent threat](https://www.imperva.com/learn/application-security/apt-advanced-persistent-threat/) (APT) event. In this latter scenario, employees are [compromised](https://www.imperva.com/learn/application-security/cyber-security/) in order to bypass security perimeters, distribute malware inside a closed environment, or gain privileged access to secured data.

An organization succumbing to such an attack typically sustains severe financial losses in addition to declining market share, reputation, and consumer trust. Depending on scope, a phishing attempt might escalate into a security incident from which a business will have a difficult time recovering.

**Notable Phishing Attacks of the Last Decade**

**2013**

Over 110 million Target customers had their credit card records stolen in a phishing attack. The scheme involved a malware-laden email to the company’s HVAC subcontractor, allowing the cybercriminals to gain access credentials to the data.

**2015**

A successful spear-phishing attack against high-value Defense Department targets with customized emails led to a data breach, which compromised information for 4,000 military and civilian personnel who worked for the Joint Chiefs of Staff. The attack forced the Pentagon to shut down its email system. Only unclassified information was said to be leaked.

**2016**

The Austrian Aerospace firm, FACC AG, was [defrauded of 50 million Euros](https://www.securityweek.com/cybercriminals-steal-54-million-aircraft-parts-maker) in a spear-phishing scheme that tricked a finance employee to transfer the money into back accounts controlled by the cybercriminals. As a result, the company’s CEO was fired.

**2017**

A Lithuanian cybercriminal posed as an Asian manufacturer to deceive Google and Facebook employees into wiring over $100 million to untraceable offshore bank accounts. The swindle occurred over the course of two years before his capture. For their part, Google claimed to have recouped the funds it had lost.

**2018**

Cryptocurrency company EOS.IO was attacked. Cybercriminals posed as company representatives and contacted potential investors. The scam succeeded in luring many to provide their private key in order to claim unsold tokens.

**4.Project specific**

**Phishing attack is performed to get the personal details like username , password , credit/debit card details of a user using phishing URL. Phishing URL is a URL which mimics the original URL of any legitimate website with few minor changes in it. Hacker generally create such URLs and make users open these URLs by using any social engineering method.**

**In our project we are going to train our model by using different features of both legitimate and phishing URLs. The source of URLs is mentioned in the resource section. After training our classification models with nearly 80% of data from the dataset we can accept a new URL from user and predict whether this URL is legitimate or phishing.**

**We are using different classifiers like decision tree, K nearest neighbor, Support vector machine to create the classification model.**

**We have also made a small web application to accept input URLs from user. The web application is made from stream lit library in python.**

**5.Steps performed for the project**

* **Collect dataset containing phishing and legitimate websites from the open-source platforms.**
* **Code to extract the required features from the URL database.**
* **Analyze and preprocess the dataset by using EDA techniques.**
* **Divide the dataset into training and testing sets.**
* **Run selected machine learning and deep neural network algorithms like Decision tree, KNN algorithm, SVM, Xgb.**
* **Write a code for displaying the evaluation result considering accuracy metrics.**
* **Compare the obtained results for trained models and specify which is better.**
* **Build a simple webpage using stremlit library to accept input urls from user and display whether the url is legitimate or not.**

**6.Data Collection**

Data which we are going to use for this project is nothing but the URLs(both legitimate and phishing). The different resources of URLs are as follows:

I went through some data sets available on Kaggle and some open source platforms:

* <https://www.kaggle.com/akashkr/phishing-website-dataset>
* https://www.kaggle.com/shashwatwork/web-page-phishing-detection-dataset --- dataset -- currently using
* <https://www.kaggle.com/shashwatwork/phishing-dataset-for-machine-learning>
* Legitimate URLs are collected from the dataset provided by University of New Brunswick, https://www.unb.ca/cic/datasets/url-2016.html.
* Phishing URLs are collected from opensource service called Phish Tank . This service provide a set of phishing URLs in multiple formats like csv, json etc. that gets updated hourly.
* https://www.kaggle.com/shashwatwork/web-page-phishing-detection-dataset --- dataset
* https://www.kaggle.com/shashwatwork/phishing-dataset-for-machine-learning --- dataset
* <https://www.kaggle.com/akashkr/phishing-website-dataset>.

**7.Feature selection/creation**

I was planning to use PCA for feature selection. But PCA is used when there are huge number of features (~500). And it is difficult to interpret the model and explain it to other people (for example clients) with features created using PCA.

The plan was to make it like a real-life project in which input URL from the user will be accepted and the output will be shown to user. The URL given by user can’t be given directly to the “predict” function. We must convert into the data format which we used to train our model. We can’t expect from user to convert the URL into that data format. We have to do it at our end.

We have written code for that.

Following are the features created to train the model:

* Domain – domain of the url is generally next to the “www.” field.

we will the find the domain of the url using gedomain function.

* Have\_ip - ip address of a url. We are using getip function to get the ip of the url.
* Have\_At- presence of “@” in a website or url
* Length of url – if a length of url is greater than 54 it may be a malicious url.
* Getdepth – depth of url is number of “/” in the url. url is legitimate or

Or not can be decided from the number of “/”.

* Redirections – redirections in a url can be calculated from “//”.

If number of redirections are more than a certain number then there are high chances of url being fake.

* Prefixsuffix – we must carefully check the refix and suffix of a url separated by “-”.
* Httpdomain – protocol used with the website must be checked. Protocol used should be “https” of “shttp” where s stands for secured.

* No of Login pages pop ups : If website is redirecting us to a unnecessary login page then it can be a fake website.
* There are many other features of a url can be considered while training the model like Domain end, Domain feature, right Click response, webtraffic, iframe. On mouseover , tinyurl.

**8. Machine learning algorithms used:**

**1.Decision tree classification algorithm:**

Decision Trees are some of the most used machine learning algorithms. They are used for both classification and Regression. They can be used for both linear and non-linear data, but they are mostly used for non-linear data. Decision Trees as the name suggests works on a set of decisions derived from the data and its behavior. It does not use a linear classifier or regressor, so its performance is independent of the linear nature of the data. Boosting and Bagging algorithms have been developed as ensemble models using the basic principle of decision trees compiled with some modifications to overcome some important drawbacks of decision trees and provide better results. One of the other most important reasons to use tree models is that they are very easy to interpret.

Decision Trees

Decision Trees can be used for both classification and regression. The methodologies are a bit different, though principles are the same. The decision trees use the CART algorithm (Classification and Regression Trees). In both cases, decisions are based on conditions on any of the features. The internal nodes represent the conditions and the leaf nodes represent the decision based on the conditions.

Diagram

Description automatically generated

Classification

*A decision tree is a graphical representation of all possible solutions to a decision based on certain conditions.*

On each step or node of a decision tree, used for classification, we try to form a condition on the features to separate all the labels or classes contained in the dataset to the fullest purity. Let’s see how the idea works.

Table

Description automatically generated

Say, the table given above is our dataset. If we try to analyze the dataset and create a decision tree based on the dataset, we will obtain something like the tree given below

Diagram

Description automatically generated

In the root node, we are using “Feat 2 ≥3” as the condition to segregate our dataset for the first time. We can see if the answer is false, we reach a leaf, that has only the entries with label 3. So, label 3 is completely separated. But, if the answer is True, we reach an intermediate node. In this case, there are 3 entries, 2 of which belong to label 2 and one belongs to label 1. So, this result has impurity as there are two labels mixed. We apply another condition on the intermediate state and obtain the purely separated labels. On Leaf 2, we have obtained only the Label 1 entry and on leaf 3, we have only the Label 2 entries.

At the beginning, the whole training set is considered as the root.

Feature values need to be categorical. If the values are continuous then they are discretized prior to building the model.

The primary challenge in the Decision Tree implementation is to identify the attributes which we consider as the root node and each level. This process is known as the attributes selection. There are different attributes selection measure to identify the attribute which can be considered as the root node at each level.

There are 2 popular attribute selection measures. They are as follows:-

* Information gain
* Gini index

While using Information gain as a criterion, we assume attributes to be categorical and for Gini index attributes are assumed to be continuous.

By using information gain as a criterion, we try to estimate the information contained by each attribute. To understand the concept of Information Gain, we need to know another concept called Entropy.

Entropy

Entropy measures the impurity in the given dataset. In Physics and Mathematics, entropy is referred to as the randomness or uncertainty of a random variable X. In information theory, it refers to the impurity in a group of examples. Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.

Entropy is represented by the following formula:-

Entropy:

Text

Description automatically generated

**2. K nearest neighbor algorithm :**

K-nearest neighbors (kNN) is a supervised machine learning algorithm that can be used to solve both classification and regression tasks. I see kNN as an algorithm that comes from *real life*. People tend to be effected by the people around them. Our behaviour is guided by the friends we grew up with. Our parents also shape our personality in some ways. If you grow up with people who love sports, it is higly likely that you will end up loving sports. There are ofcourse exceptions. kNN works similarly.

*The value of a data point is determined by the data points around it.*

* If you have one very close friend and spend most of your time with him/her, you will end up sharing similarinterestsand enjoying same things. That is kNN with k=1.
* If you always hang out with a group of *5,*each one in the group has an effect on your behavior and you will end up being the average of 5. That is kNN with *k=5*.

kNN classifier determines the class of a data point by majority voting principle. If k is set to 5, the classes of 5 closest points are checked. Prediction is done according to the majority class. Similarly, kNN regression takes the mean value of 5 closest points.

We observe people who are close but how data points are determined to be close? The distance between data points is measured. There are many methods to measure the distance. [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance) (minkowski distance with p=2) is one of most commonly used distance measurement. The figure below shows how to calculate euclidean distance between two points in a 2-dimensional space. It is calculated using the square of the difference between x and y coordinates of the points.

Graphical user interface

Description automatically generated with medium confidence

In the case above, euclidean distance is the square root of (16 + 9) which is 5. Euclidean distance in two dimensions remind us the famous [pythagorean theorem](https://en.wikipedia.org/wiki/Pythagorean_theorem" \t "_blank).

It seems very simple for two points in 2-dimensional space. Each dimension represents a feauture in the dataset. We typically have many samples with many features.

**So now how do we choose K?**

* Generally we use the Square root of the number of samples in the dataset as value for K. An optimal value has to be found out since lower value may lead to overfitting and higher value may require high computational complication in distance. So using an error plot may help.
* Another method is the elbow method. You can prefer to take root else can also follow the elbow method.

**3.Support vector machine algorithm :**

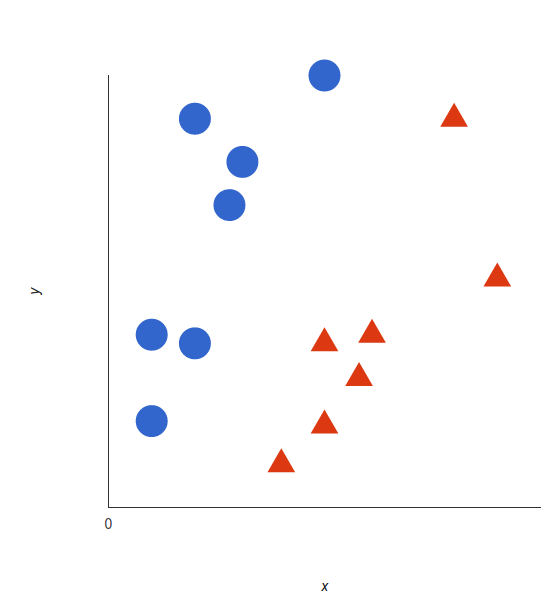
## What is Support Vector Machines?

A support vector machine (SVM) is a supervised [machine learning](https://monkeylearn.com/machine-learning/) model that uses [classification algorithms](https://monkeylearn.com/blog/machine-learning-algorithms/) for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they’re able to categorize new text.

Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it’s common to have access to a dataset of at most a couple of thousands of tagged samples.

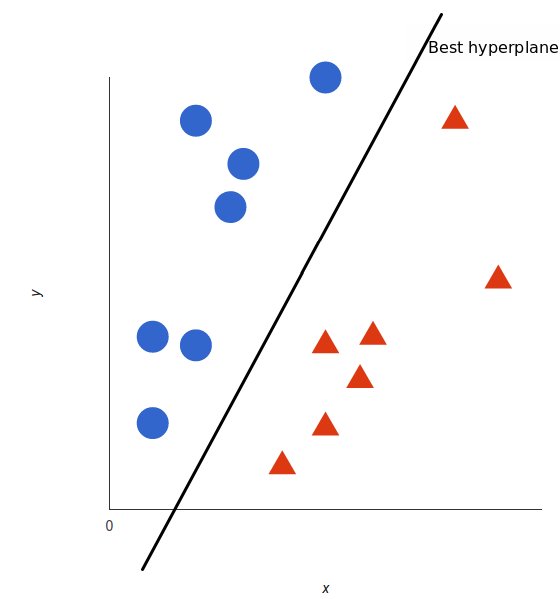
## How Does SVM Work?

The basics of Support Vector Machines and how it works are best understood with a simple example. Let’s imagine we have two tags: red and blue, and our data has two [features](https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/#feature-engineering): x and y. We want a classifier that, given a pair of (x,y) coordinates, outputs if it’s either red or blue. We plot our already labeled training data on a plane:

****

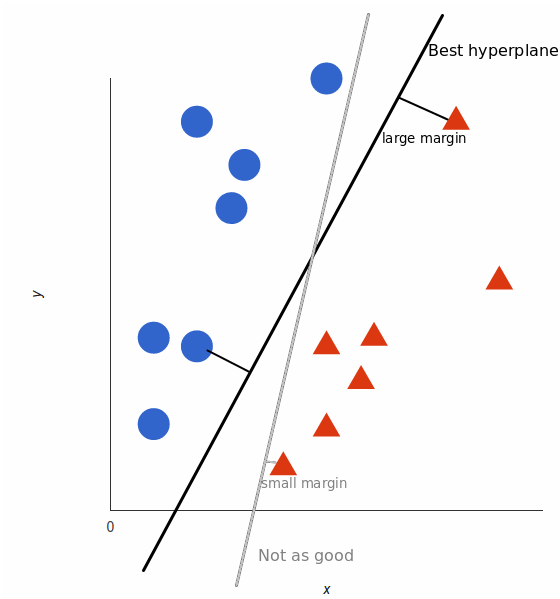
*Our labeled data*

A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it’s simply a line) that best separates the tags. This line is the **decision boundary**: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

****

*In 2D, the best hyperplane is simply a line*

But, what exactly is the best hyperplane? For SVM, it’s the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

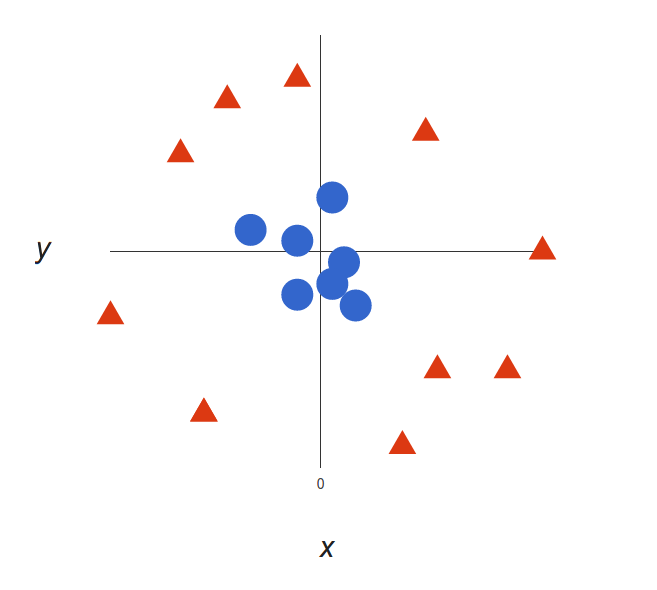
****

*Not all hyperplanes are created equal*

You can check out [this video tutorial](https://www.youtube.com/watch?v=1NxnPkZM9bc) to learn exactly how this optimal hyperplane is found.

### **Nonlinear data**

Now this example was easy, since clearly the data was linearly separable — we could draw a straight line to separate red and blue. Sadly, usually things aren’t that simple. Take a look at this case:

****

*A more complex dataset*

It’s pretty clear that there’s not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them.

So here’s what we’ll do: we will add a third dimension. Up until now we had two dimensions: x and y. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: z = x² + y² (you’ll notice that’s the equation for a circle).

I am going to discuss about some important parameters having higher impact on model performance, “kernel”, “gamma” and “C”.

* **kernel**: We have already discussed about it. Here, we have various options available with kernel like, “linear”, “rbf”,”poly” and others (default value is “rbf”).  Here “rbf” and “poly” are useful for non-linear hyper-plane.
* **gamma**: Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’. Higher the value of gamma, will try to exact fit the as per training data set i.e. generalization error and cause over-fitting problem.
* **C:**Penalty parameter C of the error term. It also controls the trade-off between smooth decision boundaries and classifying the training points correctly.

**9.Code and output screenshots:**

**1.feature selection and preprocessing:**

**Text

Description automatically generated**

**Above functions are written to get the different features of the URLs. To make csv file from these features two different methods are used in two files. You can try any one of them. It usually takes lot of time as python is a slow language as compared to C and C++. The data in csv created by above method is used further to train the model.**

**2.Data split for training and testing:**

**Text

Description automatically generated**

**2.Machine learning algorithms used:**

**Text

Description automatically generated**

**Output: to get the outputs of classification algorithm, run the “All\_class.py” file.**

**Text

Description automatically generated**

**Web applicaiton created.**

**To create a web application we have to use pickle library in python which creates a “.pkl” file for us.**

**Text

Description automatically generated**

**After creating the .pkl file we have to write the code which converts the input url into the data format given to predict function.**

**A picture containing text

Description automatically generated**

The above written function will be called from app.py file as follows:

Text

Description automatically generated

Above is the code which calls extract\_features\_of\_inp\_url function. Also it calls predict function using data created by extract\_features\_of\_inp\_url function. Out put of above code is as follows:

Graphical user interface, application, Teams

Description automatically generated

**Problems and errors encountered**

* **There are not many resources which will tell us about the parameters of a URL which will indicate whether the url is legitimate or fake.**
* **To create a dataset with features we have to pass all the urls through all the functions which will create the different features. As python is very slow feature creation for all the urls will take a lot of time. (tried to use 2 methods for this).**
* **We have to convert the input url into same data format before passing it to “predict” function which was used to train the model. Hence we have to pass input url also through all the feature creation functions.**

**Resources**

* <https://www.kaggle.com/akashkr/phishing-website-dataset>
* [https://www.analyticsvidhya.com/blog/2021/10/implementing-artificial-neural-networkclassification-in-python-from-scratch/#h2\_1](https://www.analyticsvidhya.com/blog/2021/10/implementing-artificial-neural-networkclassification-in-python-from-scratch/)
* <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>
* <https://www.analyticsvidhya.com/blog/2021/01/a-quick-introduction-to-k-nearest-neighbor-knn-classification-using-python>
* <https://www.kaggle.com/shashwatwork/web-page-phishing-detection-dataset>
* <https://www.kaggle.com/shashwatwork/phishing-dataset-for-machine-learning>