Phishing Detection and Influence Analysis

Submitted By: **Anik Chatterjee**
Instructor: **Mr. Bibekananda Kundu and Mr. Asok Bandyopadhyay**

DECLARATION

I hereby declare that the project work entitled Phishing Detection and influence analysis submitted to the Centre for Advance Computing(C-DAC), Kolkata, is a record of an original work done by me under the guidance of **Mr. Bibekananda Kundu, Joint Director, C-DAC, Kolkata** and under the supervision of **Mr. Asok Bandyopadhyay, Associate Director, C-DAC, Kolkata.**

ACKNOWLEDGEMENT

It is a sense of great satisfaction that I am able to present a field of Computer Science, that has vast applications in a variety of fields, other than Computer Science itself, in the form of this Project Work.
I have great pleasure to express my most sincere regards and deep of gratitude to **Mr. Bibekananda Kundu, Joint Director, C-DAC, Kolkata** my Guide at CDAC, for not only his guidance in developing the Project but also for helping me develop a problem-solving mindset which will help me throughout my future career endeavours.
I'm very much thankful to **Mr. Asok Bandyopadhyay, Associate Director, C-DAC,Kolkata** for his valuable support and helping attitude.
I'm also highly obliged to **Dr. Nabarun Bhattacharyya, Center Head, C-DAC**, Kolkata for his unconditional help and inspiration and for giving us a wonderful opportunity to enhanceour skills.
Finally, I would like to extend our thank to **Mr.Chanchal Patra** and to all those who have contributed, directly or indirectly to make this project successful

Anik Chatterjee
B.Tech 3rd Year
St. Thomas' College of Engineering and Technology

Abstract

Phishing is a common tactic used by cybercriminals to steal personal and financial information from people. Phishing scams often take the form of an email or a bunch of text that influences people and pretends to be someone they are not, such as your bank or anything you will want to check-in. Day by day Cybercriminals have become more sophisticated and use many tricks to manipulate people. So, the problem is to detect all of their phishing activities and make the world a safer place. It's an important topic because, to be honest, the world is becoming more dangerous in the digital market every single day and millions of people are getting manipulated by these phishing activities and losses their everything money to personal details. So, the basic approach is to make a web app where everything related to phishing will be there, mainly in research shows there are top 3 ways we can detect phishing activities so in this app firstly a user can easily detect any mail-id and got a result as that particular mail-id is spam or not (trustworthy or not), then here we focused on detecting a link is phishing or not eve it's a google form what are the question is asking in that form we also detect that form is asking any confidential and any forbidden information or not. then developed tweeter scraper where for a particular Twitter id a user will get the links that have been used in the recent 100 tweets, and of course, it'll show each link is phishing or not or even a google form it'll do the same. Now for the text data or the paragraph, this model will predict that what types of influence is that text and even in percentage it'll show how much percentage of each influence belongs to that sentence based on **Cialdini's 6 Principles of Persuasion.**

Contents

1 Introduction **7**

2 Contribution **9**

3 Related Work **12**

4 Problem Definition and Algorithm **15**
4.1 Task Definition 15
4.2 Algorithm Definition 17

5 Experimental Evaluation **25**
5.1 Methodology 25
5.2 Results 28
5.3 Discussion 33

6 Conclusion **36**

List of Tables

1 Standard Dataset vs Compound Dataset 13
2 Model on each class 13
3 Smaller Dataset on Different Accuracy vs Learning_rate 31
4 Main Dataset on Different Accuracy vs Learning_rate 31
5 Model Precision, Recall, and f1-score for each class 35

List of Figures

1 Graph of Phishing activites in 2020 from in present 7
2 Dataset On Site Detection 9
3 Dataset On Cialdini's 6 principle 11
4 Flow Chart of Email Id Checking 15
5 Flow Chart of Site Checking 16
6 Flow Chart of Twint 16
7 Flow Chart of Influence analysis 17
8 Logistic Regression 18
9 Logistic Regression Formula 18
10 Bitly 19
11 Url Expand 20
12 Transformer Architecture 21
13 MultiHead Attention 22
14 Attention Formula 22
15 Summarized 22
16 Bert 23
17 Adam Optimizer 24
18 Categorical Cross Entropy 24
19 Validation Checking 26
20 Frequency of each class 26
21 Model for Influence analysis 27
22 Real id Checking 28
23 Fake Id checking 28
24 Site Checking 29
25 Confusion metrics for site Checking 29
26 Twitter Scrapping 30
27 Evaluate the first model(last 100 epochs) 30
28 Evaluate the 2nd mode(last 50 epochsl 31
29 Result 32
30 Confusion Metrics for Influence analysis 33
31 Formula for Recall and Precision 34
32 F1 - Score 34

1 Introduction

Phishing can mean many things hacker, spam, fraud so basically it's a Security related issue where fraud or a hacker sends a message or send a link and influence people to click it and share information in many ways the information can be the bank details or the personal details or any other thing the hacker want from that user, with that information that fraud can hack the bank account and take all the money or any personal harassment can be done from those, So it's relevant to stop that, here the main question and the topic is how to detect what is spam and what is safe ? how to trust an URL? how to check the information they are asking is forbidden?
In this 20th century we can see almost everything is moving to the digital world, most elaborately the last one year with this covid situation where we don't have any other option, we have to do almost everything online and that was more beneficial for the phishing people, so for that according to the FBI, phishing was the most common type of cybercrime in 2020—and phishing incidents nearly doubled in frequency, from 114,702 incidents in 2019 to 241,324 incidents in 2020. The FBI said there were more than 11 times as many phishing complaints in 2020 compared to 2016. According to Verizon's 2021 Data Breach Investigations Report (DBIR), phishing is the top "action variety" seen in breaches in the last year, and 43% of breaches involved phishing and/or pretexting. The frequency of attacks varies from industry to industry. But 75% of organizations around the world experienced some kind of phishing attack in 2020. Another 35% experienced spear phishing, and 65% faced BEC attacks. Below the diagram, you can see the frequency of Phishing activities going on currently[1].

Figure 1: Graph of Phishing activites in 2020 from in present

Statistics shows that 96% of phishing attacks are received by email. Another 3% are done by malicious websites and only 1% by phone.According to Sonic Wall's 2020 Cyber Threat Report, PDF and Microsoft Office Files (sent by email) were the vehicles of choice for today's cybercriminals. why? Because these files are internationally trusted in the modern workplace[2].

Now, The importance of the research is to protect online users from becoming victims of online scams, the disclosure of confidential information to the attacker, among other effective

[1]https://www.tessian.com/blog/phishing-statistics-2020/
[2]https://www.tessian.com/blog/phishing-statistics-2020/

phishing scams, this phishing detection tool play an important role in ensuring a safe online experience for users. So, the main research goal here to detect Phishing activities with deep learning, In the research we analyzed mainly there are 3 ways to check., a particular mail is coming from a good or trustable organization or not, for that developed a model that can check the particular email id is safe or not. Then the particular site they are giving it's secure or not? for that build a machine learning model that can detect the URL type easily we know in these types of classification cases machine learning is becoming more important day by day. Now if it's a form then what type of information they are asking? for these the model will scrap the question and show to the user with that also shows any forbidden information is asking or not and lastly the text is sent by them is it somehow influencing people in a forbidden way or not? here we used the sentiment analysis technique identify the influence level for a particular text and got some good accuracy so the user can trust the app to prevent fraud.

2 Contribution

During the project we have got to learn many things and got bold experience in this field, we can start with the case where we have to detect the particular site is phishing or not. this is one of the tasks what usually requires a lot of data to make the model to understand what type of domains are actually safe and what is not, and there are much domain exists on the internet and every day it's increasing so without a lot of data it's almost not possible to make your model to understand about understanding the domain even maximum humans are not perfectly good at this. So, we tried to find as much data as we can to make the model stronger, then start collecting the dataset and find 3 relevant readymade datasets on Kaggle, merge them together and totally got 5,49,347 data together for both classes, you can see some of the data below to have an understanding how's the dataset. For access the full data[3].

	URL	Label
2	nobell.it/70ffb52d079109dca5664cce6f317373782/login.SkyPe.com/en/cgi-bin/verification/login/70ffb52d079109dca5664cce6f317373/index.php?cmd=_profile-ach&outdated_page_tmpl=p/gen/fail	Phishing
3	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrcmd=_home-customer&nav=1/loading.php	Phishing
4	serviciosbys.com/paypal.cgi.bin.get-into.herf.secure.dispatch35463256rzr321654641dsf654321874/href/href/href/secure/center/update/limit/seccure/4d7a1ff5c55825a2e632a679c2fd5353/	Phishing
5	mail.printakid.com/www.online.americanexpress.com/index.html	Phishing
6	thewhiskeydregs.com/wp-content/themes/widescreen/includes/temp/promocoessmiles/?84784787824HDJNDJDSJSHD//2724782784/	Phishing
7	smilesvoegol.servebbs.org/voegol.php	Phishing
8	premierpaymentprocessing.com/includes/boleto-2via-07-2012.php	Phishing
9	myxxxcollection.com/v1/js/jih321/bpd.com.do/do/l.popular.php	Phishing
10	super1000.info/docs	Phishing
11	members.tripod.com/~CRBA/main.html	Safe
12	homepage.ntlworld.com/peter.snape/PrestonBackgammonClub.htm	Safe
13	members.tripod.com/leaderdragon/bgammon.html	Safe
14	www.hardyhuebener.de/engl/geschichte.html	Safe
15	bckg.pagesperso-orange.fr/english/accueil.htm	Safe
16	greekteam2000.tripod.com/BackGammon	Safe
17	home.online.no/~warnckew/gammon/	Safe
18	www.edcollins.com/backgammon/index.html	Safe
19	www.centralconnector.com/GAMES/BACKGAMM.htm	Safe
20	www.bginga.com/chouette.html	Safe

Figure 2: Dataset On Site Detection

After that we used the classical ML approach check each word of the domain individually then used ML model to classify first we have start with **Multinomial Binomial** which is a probabilistic model and got 88% of accuracy with the validation data after that used **Naïve Based** model **Xgboost** model got 74% and 82% for those particular model lastly used the Logistic Regression for classify and got **96%** accuracy for the validation data and that's the optimal one.
Within that same part made a google form detector that will check any particular site is a google form or not and if it's a google form what are the question that form is asking, for this have to take some help of **beautifulsoup** library, in the algorithm definition (4.2) part you can see how we're detecting the full form Then comes the last topic, It's the most unique one and here we have faced the most difficulty in almost everywhere because the task is to classify different influence, where we don't particularly have a readymade dataset, we have researched a lot for finding dataset, maximum of data we was getting was either of an attack detection where they are classifying a message is spam or ham and though we got a influenced based data but it's on a case study and for a particular region and generally the

[3]https://github.com/starboi2000/Phishing-detection-and-influence-analysis/blob/main/phishing_site_urls.zip

message was on the offline seller conversion effect on each gender, gender topic aside now you can probably think what is the difference between offline conversion and online conversion, let me tell you there is a big difference there is a particular limitation on offline business on business type also on place in research we get to know that there's a big difference in between Canada USA on influencing people in offline market but in online you will get a bigger difference than offline market, as an example you can think the amazon site we're here from India is getting the same influence on each product as the other country people are getting and for each product It's almost same types of influence. The variety comes into play when we check the business type in online because in offline there's a fix business type in online there's a huge verity as an example you can check of a Dating Sites there is almost no business of this in offline but it creates a huge business in online. For this reasons we didn't think this data will be appropriate for this task, so we planned to make a dataset, in research we have found out for representing an influence Cialdini's 6 Principles of Persuasion is the most valuable weapon, we started learning about this get to know the different classes those are

1. Reciprocity - This means we don't like to feel that we owe other people. Generally speaking, when people have these social obligations they try to settle them. For example, if someone sends you a birthday card, you'll almost certainly want to send them one in return. You'll do this when their birthday next rolls around so that you settle your sense of social obligation.

2. Scarcity - The less of something there is, the more people tend to want it. This holds true for experiences as well as for material products. From a persuasion and influence perspective this means that to increase interest in your product or service, you may benefit from reducing its availability (or at least creating a sense of scarcity)

3. Authority - Individuals who are authoritative, credible and knowledgeable experts in their fields are more influential and persuasive than those who are not. Part of the reason for this is that authority and credibility are some of the core building blocks of trust. When we trust people we are more likely to follow them

4. Commitment and consistency - People like to be consistent with their identity or sense of self-image. In other words, if I'm a person who thinks of myself as a "healthy" person, then I'm more likely to undertake actions that I consider to be "healthy".

5. Liking - It might seem totally obvious, but people are much more likely to be influenced and persuaded by those that they like, than those that they don't. Given human nature, people are much more likely to like people who pay them compliments and who cooperate with them, than those who don't. And, unfortunately, given positive evidence in relation to certain benefits of diversity, people are also much more likely to like people who are similar to them, than those who are not.

6. Consensus (social proof) - Humans are social by nature and generally feel that it's important to conform to the norms of a social group. This means that when it comes to decision-making, we often look around us to see what others are doing, before making our minds up.

Now you can see the examples of the dataset that we have collected, one thing to focus is that here 1 – Reciprocity, 2 – Scarcity, 3 – Authority, 4 – Commitment, 5 – Liking, 6 – Social proof, 7 - Normal[17]

	A	B
1	Phrase	Sentiment
2	Thanks for your help. We really appreciate that.	1
3	Merry Christmas. We are giving away free deliveries in this big occasion. Hope you enjoy the day.	1
4	Hurry up, the special offer is going to end tonight.	2
5	Order before it gets cold. We are giving free food deliveries in your area and 30% off in your every food.	2
6	We are sponsored by Nike, the best fancy and comfy shoe seller.	3
7	We are supported by Government and our company follows by govt policies and rules.	3
8	We are here to take care of your children while you are at work and we promise to make them happy.	4
9	Sign up now for a 7days&6nights Rajasthan tour. We promise to give you the best travelling experience.	4
10	Today actress Jacqueline Fernandez is going to use our all new beauty product and make a review on it. Please come by at our shop or visit this link.	5
11	We prefer your liking and want to improve based on that. Please make a review here.	5
12	This anime movie is the most anticipated of this month.	6
13	We are so proud having the largest fan base in the world.	6
14	That's cool dude	7
15	wassup? People	7

Figure 3: Dataset On Cialdini's 6 principle

Now you have a quick understanding about this data for access the full data[4]
And for the implementation part, we have 2 columns Phrase is the Value and Sentiment is the key fine-tuned the transformer Bert model in that whole data first normalize the class from (1 – 7) to (0 - 6) for a bit simpler in the transformer, that's why you can see the result section it's (0 – 6) then used our model (you will get more information about it in the methodology part) and got good accuracy. As a result, the model will show each influence percentage for every particular example, we have analyzed this is the best way to show the influence level because Nowadays online market is evaluating a lot, and we observed that different companies are merging more than one influence into one sentence as an example: **Happy birthday to you Click the link to check the offer for you.** It's a Reciprocity example we know, **The offer will last till midnight.** this is a Scarcity example. If we merge 2 examples it'll sound like **'Happy birthday to you Click the link to check the offer it'll last till midnight.'** it's even impossible for a human to understand which particular type of influence it's because by the example we can see it's both Reciprocity Scarcity. For these reasons we build a model that won't say which influence it's, it'll predict what percent of which influence is in that sentence, if it's a straightforward single influence example model will predict that influence percentage with a higher number like 90% or above, but if it's a mixture of 2 it'll show the percentage low for both the influence though the percentage of scarcity is max in the last example model will predict scarcity with a bit max percentage and the sentance is a mixture of both influence so there will be some percentage of reciprocity also like this, and lastly model will take the higher value position and show that particular sentence is most likely to be that influence. With that result, we have made a **forbidden word list (example – account, address, password, records, confidential, credentials, data, database)** there is a total of 44 words in it[5]. These are the words used in a forbidden text, so if any of these words have been used in the text our model will show that sentence is containing forbidden words to make the user more careful. For more understanding check out the result part there we have shared the screenshot of the result part, you can check this also for the video explanation[6].

[4]https://github.com/starboi2000/Phishing-detection-and-influence-analysis/blob/main/main_data.xlsx
[5]https://drive.google.com/drive/u/1/folders/1djyMa2-V-7HJfeHj17fdNeLvmmi0dE9K
[6]https://drive.google.com/file/d/1iqKwNJJeyNwY4iKqKBhoBIu04w2lcNyK/view

3 Related Work

In persuasive technology research, fewer studies have been investigated.
Cialdini's 6 Principles of Persuasion effects on different gender[2] - Here it's focusing on how culture and gender influence the effectiveness of Cialdini's principles of persuasion they conducted a study to investigate the culture, gender, and age differences concerning individuals' susceptibility to Cialdini's persuasive strategies. They found that, in general, people are more susceptible to Commitment and Reciprocity. However, the main focus of their studies influence on African people with each gender like which gender of people are influencing in which type of influence They observed that there is a main effect of gender and the main effect of strategy Between-Group Comparison. The between-subject effect Kruskal-Wallis rank-sum test based on the rating measure further shows a gender difference concerning Authority (p ¡ 0.01) and Commitment (p ¡ 0.05), with males being more responsive to both strategies. However, there is no gender difference concerning the other four strategies: Reciprocity, Liking, Scarcity, and Consensus. Moreover, there is no gender difference they have analyzed concerning all of the six strategies based on the ranking measure.

Cialdini's 6 Principles of Persuasion effects on E-Commerce Shopping Persuasive Strategies[3] - Here they propose the use of shoppers' online shopping motivation in tailoring six commonly used influence strategies: scarcity, authority, consensus, liking, reciprocity, and commitment. They identify how these influence strategies can be tailored or personalized to e-commerce shoppers based on the online consumers' motivation when shopping. To achieve this, a research model was developed by the team using Partial Least Squares-Structural Equation Modeling (PLS-SEM) and tested by conducting a study of 226 online shoppers. The result of their structural model suggests that persuasive strategies can influence e-commerce shoppers in various ways depending on the shopping motivation of the shopper In this case study, they observed the behavior and classify in different classes as a result **Balanced buyers**—the shoppers who typically plan their shopping ahead and are influenced by the desire to search for information online have the strongest influence on commitment strategy and have insignificant effects on the other strategies. **Convenience shoppers**—those motivated to shop online because of convenience have the strongest influence on scarcity, while **store-oriented shoppers**—those who are motivated by the need for social interaction and immediate possession of goods have the strongest influence on consensus. **Variety seekers**—consumers who are motivated to shop online because of the opportunity to search through a variety of products and brands, on the other hand, have the strongest influence on authority.

Social Engineering Attack Detection method based on NLP[4]- They have developed a method that detects social engineering attacks that are based on natural language processing and artificial neural networks. This method can be applied in offline texts or online environments and flag a conversation as a social engineering attack or not. Initially, the conversation text is parsed and checked for grammatical errors using natural language processing techniques, and then an artificial neural network is used to classify possible attacks. The proposed method has been evaluated using a real dataset The method was. 1 – Data Preprocessing (Meta Data, Synthesise Data), 2 - Feature Extraction (Malicious Links,

ALGORITHM RESULT (Standard Dataset)	ALGORITHM RESULT (Compound Dataset)
Decision Tree 0.681	Decision Tree 0.918
Random Forest 0.683	Random Forest 0.917
Multi-Layer Perceptron 0.691	Multi-Layer Perceptron 0.925

Table 1: Standard Dataset vs Compound Dataset

Dataset	Models	Train and Dev	Test
Anger	Word2Vec, BERT	0.7241	0.6388
Joy	SBERT, BERT	0.7788	0.7115
Sadness	SBERT	0.7719	0.6967
Fear	Word2Vec, SBERT	0.6930	0.5705

Table 2: Model on each class

Principles of persuasion, attack history), 3 – Aggregation of Results (Calculates the degree of truth output, Used Decision tree or Neural Network) For the result, they used 2 datasets, one standard and one synthesized or Compound, and used 3 techniques 1 – Random Forest, 2 – Decision Tree, 3 - Neural Network: MLP classifier The result was1

Sentiment Analysis Using Deep Learning[5] – Here used deep learning techniques to classify the sentiments of an expression into positive or negative emotions, The positive emotions are further classified into enthusiasm, fun, happiness, love, neutral, relief, surprise, and negative emotions are classified into anger, boredom, emptiness, hate, sadness, worry. They observed and evaluated the method using Recurrent Neural Networks and Long short-term memory on three different datasets to show how to achieve high emotion classification accuracy. A thorough evaluation shows that the system gains emotion prediction on LSTM model with 88.47% accuracy for positive/negative classification and 89.13% and 91.3% accuracy for positive and negative subclass respectively.

Fuzzy-Rough Nearest Neighbor for Emotion Detection in Tweets[6] - Here they consider using classification methods based on fuzzy rough sets. Specifically, develop an approach for the SemEval-2018 emotion detection task, based on the fuzzy rough nearest neighbor (FRNN) classifier enhanced with ordered weighted average (OWA) operators. They used tuned ensembles of FRNN(OWA models based on different text embedding methods. They try to classify 4 classes Anger, Joy, Sadness, Fear, and got a result in (Table 2) 2

Customer Perception Analysis Using NLP[7]- Here they are interested in unstructured data available that would be available for survey voice recordings of customer interactions and chat transcripts. Analyzing such data correctly is critical, as it reveals everything from buying trends to product flaws and provides a significant business advantage. They explore different technologies of Deep Learning and Natural Language Processing (NLP) that would

help analyze better the contextual information to capture customer feedback. Here they mentioned some strategies using Deep Learning how we can achieve a good result 1 – Deep MLP, 2 – CNN, 3 – RNN / LSTM, 4 – RNN/GRU As a result, each model has its own benefit and disadvantages

Twitter Sentiment Analysis Using Deep Learning[8] - Here they are focusing more on improvement of accuracy of sentiment system 1 to 5 star depending on the text 1 being the negative and 5 is the most positive they build a custom model with embedding, CONV1d, LSTM and got a result as
Training Loss 0.1907
Validation Loss 0.5636
Training Accuracy 0.9968
Validation Accuracy 0.9407

The literature review can be summarised as the identification of the Cialdini's 6 Principles of Persuasion and how that is influencing people and with that we can also see the effectiveness of deep learning models like transformers, RNN, CNN in the sentiment analysis based on the twitter corpus in a general scenario.

4 Problem Definition and Algorithm

4.1 Task Definition

So, the top 3 ways to check an email or a message is Phishing or not are
• Check the Email id whether it's from a safe organization or not
• Check the link is a Phishing site or a safe site
• Check the text whether it is trying to influence you into something and if it containing any forbidden words or not
For all of those above reasons, The Email ID checker used validate-email of python dictionary to check that the particular email id is reliable or not. The algorithm of the process is Given below.

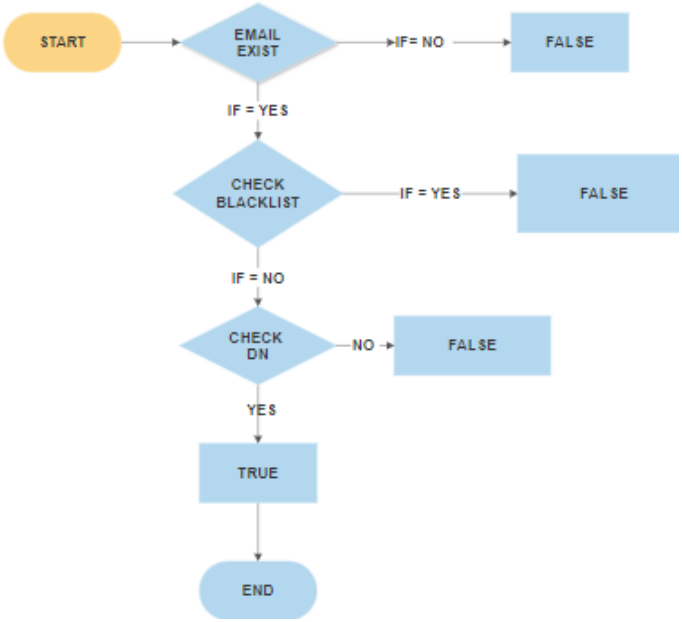


Figure 4: Flow Chart of Email Id Checking

Then we made a model that can detect whether a given site is a google form or an URL if it's a google form it'll scrap all those questions asked in that form and will show any of those questions is asking any forbidden things or not and if it's an URL I have trained a machine learning model LogisticRegression for that it'll check whether that site is a phishing site or not with other parameters

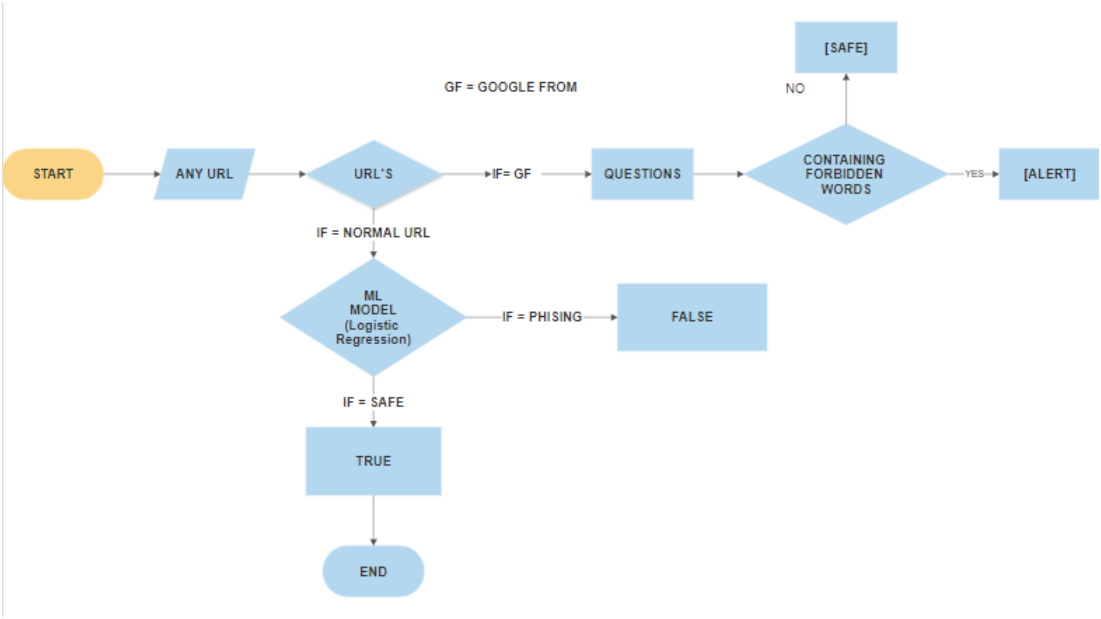


Figure 5: Flow Chart of Site Checking

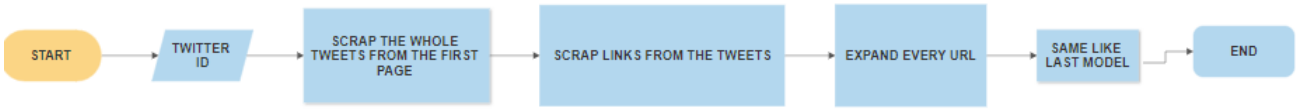


Figure 6: Flow Chart of Twint

After that made a Twitter scrapper that can scrap the Twitter id given by the user and will take the first 100 tweets and scrap their URL's and check each URL is Phishing or not or even a google form, in the same way, Used Twint for the Twitter scrapper and URLexpander for expanding each URL because many Phishing sites use bitly to shorten their URL and it'll be hard for our model to detect of a shorten URL

Lastly made a custom dataset by real mails or adds then for the text analysis I finetuned a BERTmodel which will analyze the text and as an output it'll say if the text is influencing or not if it's influencing in which way it's influencing and it'll show this text is containing any forbidden words or not

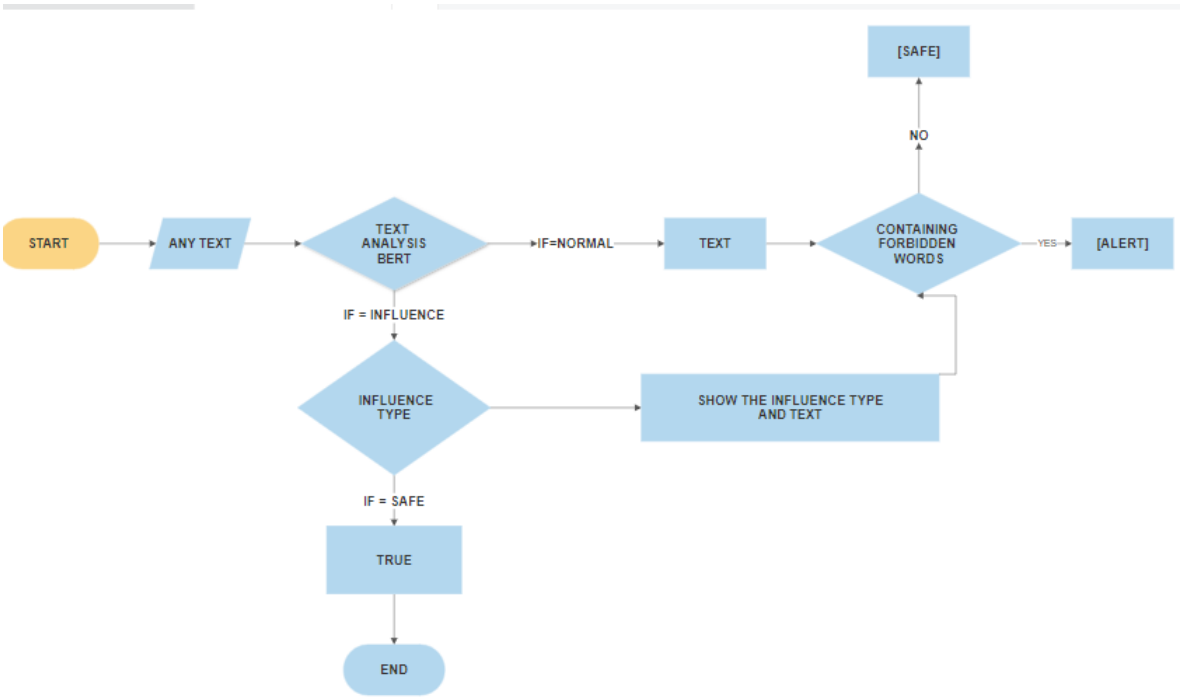


Figure 7: Flow Chart of Influence analysis

With all of this above process, we can almost detect Phishing.

4.2 Algorithm Definition

Validate-email is a package for Python that check if an email is valid, properly formatted, and really exists. It'll start checking the mail id is in the proper format or not then it'll look into the mail id is already on blacklist or notBlack List are containing all the mail id that are being blacklisted by the users or proven that those are spam. Then it'll look into the DN of the mail Distinguished Names is a (often referred to as a DN or FDN) is a string that uniquely identifies an entry in the DIT. A Distinguished Names is comprised of zero or more Relative Distinguished Name components that identify the location of the entry in the DITWith all of this above process, we can almost detect Phishing.
Now for checking a URL is Phishing or safe we trained a machine learning model Logistic Regression for this

Logistic Regression = There are many more complex extensions, logistical regression is a statistical model that uses logistics function to model binary dependency variables. By re-analysis, logistics delays (or lodge retreats) are estimating logistical model measurements (binary abuse form).

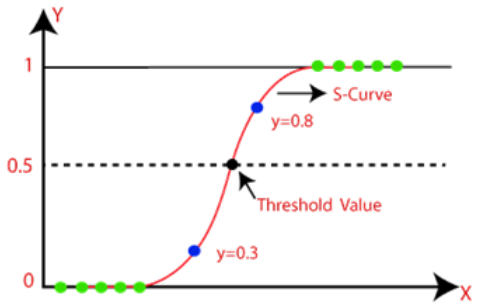


Figure 8: Logistic Regression

$$p = \frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1} = \frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}} = S_b(\beta_0+\beta_1 x_1+\beta_2 x_2).$$

Figure 9: Logistic Regression Formula

In this above picture, we can see the graph of a Logistic Regression where the threshold value is 0.5

Where S_b is the sigmoid function with base b. The above formula shows that once beta_i are fixed, we can easily compute either the log-odds that Y=1 for a given observation, or the probability that Y=1 for a given observation. The main use-case of a logistic model is to be given an observation (x_1,x_2), and estimate the probability p that Y=1. In most applications, the base b of the logarithm is usually taken to be e. However, in some cases, it can be easier to communicate results by working in base 2, or base 10.
RegexpTokenizer: With the help of the NLTK tokenize.regexp() module, we are able to extract the tokens from string by using the regular expression with RegexpTokenizer()
Snowball Stemmer: It is a stemming algorithm which is also known as the Porter2 stemming algorithm as it is a better version of the Porter Stemmer since some issues of it were fixed in this stemmer. Stemming It is a process of directing the word to the word and is applied to extensions and prefixes or to the roots of words known as lema. Simplified words are the same words that fall under a common trunk by subtracting a word from the root or stem. For example, the words caregiver, caregiver, and caregiver lie under the same trunk 'care'. Natural Language Processing (NLP)
CountVectorizer : Convert a set of text documents to a matrix of simulation counts. This application uses a few representations of numbers using scipy.sparse.csr_matrix. If you do not provide a "prerequisite" dictionary and do not use the same behavior selection analysis, then the number of attributes will be equal to the number of words obtained by analyzing the data. And if the site is a google form used Selenium to scrap that Selenium is a mobile framework for testing web applications. Selenium provides a playback tool for writing task tests without having to learn the test script language

Used **Twint** it's the Twitter API. It is an advanced Twitter-based Twitter tool that

Figure 10: Bitly

allows you to cut Twitter from Twitter without using it. It uses Twitter search operators to shave tweets from specific users, shave tweets related to certain topics, hashtags and trends, or release sensitive information from tweets with e-mail and phone numbers. We find this very useful, and you can really get creative with it. Twitter also requires special Twitter to allow Twitter users to copy the API, Selenium, or browser without verification of what the user likes and follows. Some of the benefits of using Twint vs Twitter API: 1. Can fetch almost all Tweets (Twitter API limits to last 3200 Tweets only)
2. Fast initial setup
3. Can be used anonymously and without Twitter sign up
4. No rate limitations.

Some Relevant Twint Functions which makes it eassier to use
• Twint.Search: This helps us to access the search function of Twitter app. We can search just anything with this function, not necessarily a username.
• Twint.Username: We can search for any specific Twitter user through this variable just using his/her username.
• Twint.Language: We can specify a language, in which we want to search the tweets. For example, if we want all tweets to be in English, then we have to make this variable 'en'.
• Twint.Pandas: This variable is made to 'True' as we want all the tweets to be stored.
• Twint.Since: Here we can specify from which date and time we want the tweets.
• Twint.Limit: If we had not set the limit, it would have run indefinitely and we had to interrupt it from keyboard. In that case the tweets had not stored in a list. The minimum limit can be 100 and maximum is the total no. of tweets a particular account has. We have used a Tweet limit of 100 in our code.

Many Phishing sites use bitly where Bitly is a URL shortening service and a link management platform, and the company Bitly, Inc., was established in 2008. It is privately held and based in New York City. Bitly shortens 600 million links per month, for use in social networking, SMS, and email.
It'll be hard for our model to detect a shorten URL so used, URL Expander is a service using which you can find out the destination of a shortened URL before you click and visit that link. By examining the link before clicking, you will have a better chance of avoiding phishing, malware, and viruses, from entering your network.

Here first we can see the input is a shorten URL the model first expand the URL the detect

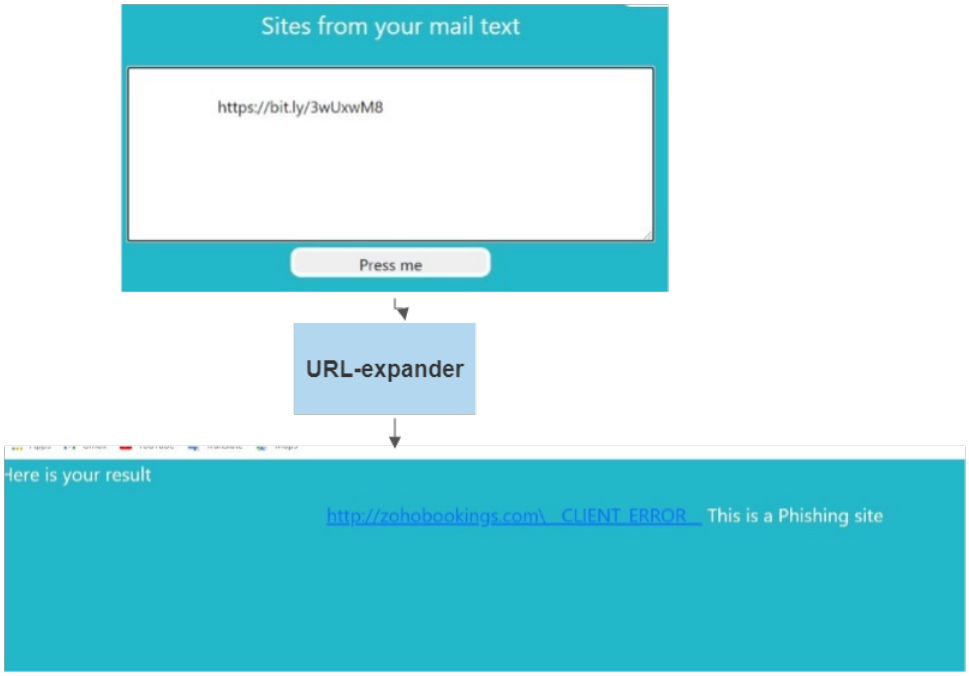


Figure 11: Url Expand

it's safe or Phishing That's not the end there are some google form even ask a forbidden question so for understanding that first analyzed some google from and found out there is a common class only a google form contains that "freebirdFormviewerViewFormContentWrapper" Used BeautifulSoup(Beautiful Soup is a Python package for filtering HTML and XML documents. Creates a tree for filtering pages that can be used to extract HTML information, which can be used for web browsing) to scrap these questions with that use regular function containing some forbidden words collected by research

Lastly made a custom dataset by real mails or adds , and finetuned the 'BERT-base-uncased' model, developed a custom model with this using Tensorflow 2.0use that in our dataset to get the output

So what is BERT?
Bidirectional encoder representation is a machine learning method based on Transformers, a pre-training transformer for Google's natural language processing. BERT was created and published by Google in 2018 by Jacob Davlin and his colleagues
The key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-left training

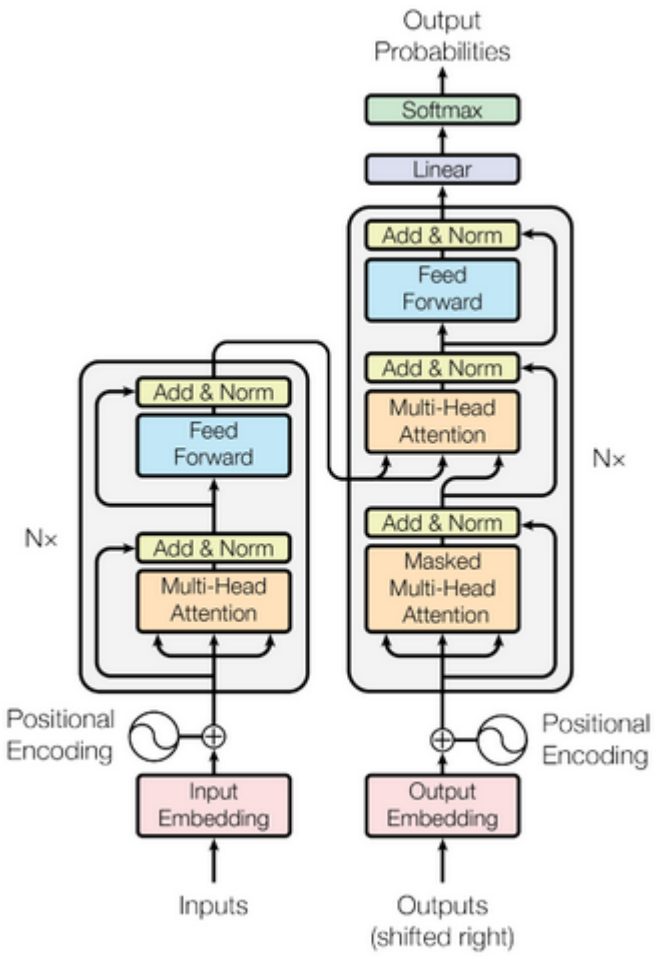


Figure 12: Transformer Architecture

Above we can see the whole transformer architecture[7] The encoder on the left and a decoder on the right. Both the encoder and the decoder are composed of modules that can be stacked on top of each other several times, as described in the diagram. We see that the modules are primarily composed of multiple heads and feed layers. Inputs and outputs (target sentences) are already in n-dimensional space because we cannot use strings directly. A small but important part of the model is to record the different words in the layout. Since we do not have a repetitive network of sequences that remember how to feed in the model, we must assign each word/part relative to the sequence because the sequence depends on the order of the elements. These layouts are added to the n-dimensional vector included in each word.

Multi-head Attention[8]

[7]https://www.kdnuggets.com/2019/07/pre-training-transformers-bi-directionality.html
[8]https://paperswithcode.com/method/multi-head-attentionl

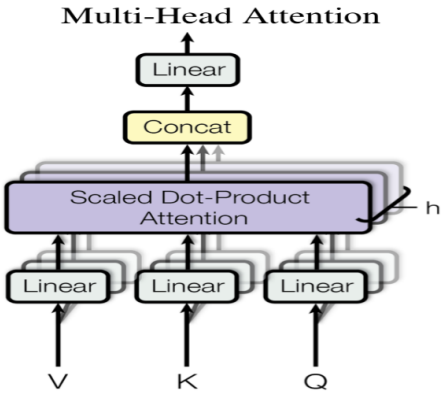


Figure 13: MultiHead Attention

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Figure 14: Attention Formula

And the main formula is working here is

Q is a matrix that contains the query (vector representation of one word in the sequence), K is all the keys (vector representations of all the words in the sequence) and V is the values, which are again the vector representations of all the words in the sequence. For the encoder and decoder, multi-head attention modules, V consists of the same word sequence as Q. However, for the attention module that is taking into account the encoder and the decoder sequences, V is different from the sequence represented by Q.
To alleviate this a little bit, the values in V are summed up and we can say that we are summed up by some of the points in which our weight is translated: a.
This means that the weights are defined by how each word of the sequence (represented by Q) is influenced by all the other words in the sequence (represented by K). Additionally, the SoftMax function is applied to the weights to have a distribution between 0 and 1. Those

$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Figure 15: Summarized

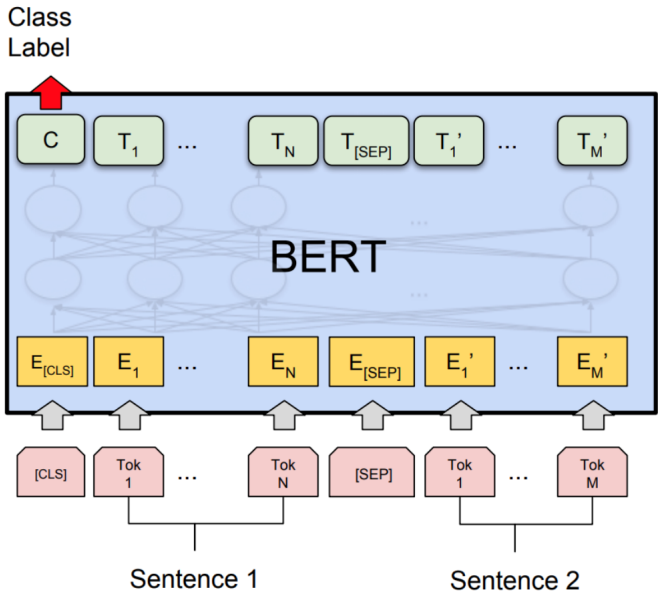


Figure 16: Bert

weights are then applied to all the words in the sequence that are introduced in V (same vectors than Q for encoder and decoder but different for the module that has encoder and decoder inputs).

How BERT works

BERT makes use of a Transformer, an attention mechanism that learns contextual relations between words (or sub-words) In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task.

Here we used the 'BERT-base-uncased model' to classify the sentiments it's a Prerequisite model in English using the purpose of hidden language modeling (MMM). This article was first published and released in 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding' this paper. This model is unchanged and does not differentiate between English and English.

Model description

The model randomly masks 15% of the words in the input then runs the entire masked sentence through the model and has to predict the masked word. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other. Models concatenate two masked sentences as inputs during pretraining. Sometimes they correspond to sentences that were next to each other in the original text. The model then has to predict if the two sentences were following each other or not. The model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks. If you have a dataset of labeled sentences, for instance, you can train a standard classifier we can see in Figure 16[9].

[9]https://pytorch.org/tutorials/intermediate/dynamic_quantization_bert_tutorial.html

Algorithm: Generalized Adam

S0. Initialize $m_0 = 0$ and x_1

For $t = 1, \ldots, T$, do

S1. $m_t = \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t$

S2. $\hat{v}_t = h_t(g_1, g_2, \ldots, g_t)$

S3. $x_{t+1} = x_t - \frac{\alpha_t m_t}{\sqrt{\hat{v}_t}}$

End

Figure 17: Adam Optimizer

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Figure 18: Categorical Cross Entropy

Used the Adam optimizer and loss as categorical-cross-entropy Adam Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient inherits the strengths or the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent
Mathematical Aspect of Adam Optimizer
Categorical-cross-entropyAlso called Softmax Loss. It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a sentiment classification to output a probability over the C classes for each sentence. It is used for multi-class classification.
If M¿2 (i.e. multiclass classification), we calculate a separate loss for each class label per observation and sum the result.
M - number of classes
log - the natural log
y - binary indicator (0 or 1) if class label c is the correct classification for observation o
p - predicted probability observation o is of class c

5 Experimental Evaluation

5.1 Methodology

In the first one as an email_id validation where the task is to check whether a particular email id even exists or the formation of the email id is correct or not? with this thought I analyzed some mail id and make a simple Regular expression that will check the correct formation of an email id, now the main question comes as if the email id exists what is the possibility to check that id is a spam id or not? with that, we did a bit of research and get to know that there's a way we should have a blacklist of different ids which have already been detected as spam and we should check the DNs of that id. if the particular id is from that blacklist then there's no point to check the DNs, but if the blacklist does not contain the id we have to go with checking the DNs to be secure, and if one of the results comes as false or negative then there's a high possibility that the mail id is a spam id, Luckily for research, we have found a python library that can do all of those things, so we applied that and got a satisfactory result.

In the 2nd case collected many URL's examples near about 5,50,000 data containing phishing and safe sites, Used RegexpTokenizer, SnowballStemmer for understanding each text of those URL's, used CountVectorizer for making input as an array, after that split the train/test data.Then developed a Pipeline model using logistic regression with stop_words and RegexpTokenizer to classify the URL use machine learning for that got some good result with the validation accuracy Still to be sure about the model performance plot the confusion metrics to check the actual scenario because there is a probability that all validation set got 90% of safe class and if you just print('Safe site') it'll give you 90% of accuracy like this you don't have to train a model, but in confusion metrics you can get the actual scenario about your model. In the result part you can see the metrics where we got a good result, but still those are the value of training and validation set to be more accurate about the model performance we took a help of virus total collect total of 50 new URLs from daily messages put all of this to the virus total and store the result used the same 50 samples to my model and got above 84% of accuracy in **Figure 19** you can see the result of the first 10 samples.

	A	B	C
1	URL	Model Prediction	Virus Total Result
2	submitaccountloginloadingcgibin.com/paypal/us/cgibin/login/cmdsubmit/sessionID953911/statusaspx/database/inde	This is a Phishing Site	PHISING SITE
3	saifenergy.com/adminsection/calendar/coxmail.htm	This is a Safe Site	PHISING SITE
4	seriestvtokio.ekiwi.es/index_files/avgls.php	This is a Phishing Site	PHISING SITE
5	primwood.se/broschyrer/Redirect.php	This is a Phishing Site	PHISING SITE
6	cephtex.com/Paypal/Processing.htm?cmd=_Processing&dispatch=5885d80a13c0db1fb6947b0aeae66fdbfb2119927117e3	This is a Phishing Site	PHISING SITE
7	paypal-confirmation.it.kizitowangalwa.co.ke/confirmation/6a9562a8bb2ec772cf2db78aea520ab1/Processing1.php?cmd	This is a Phishing Site	PHISING SITE
8	imetrica.net/css/	This is a Phishing Site	PHISING SITE
9	www.idxband.com/telin/images/e-online.php	This is a Phishing Site	PHISING SITE
10	www.vbacreation.fr/includes/Archive/Connect/SignOn.php	This is a Phishing Site	PHISING SITE
11	www.sec.gov/edgar.shtml	This is a Safe Site	SAFE SITE

Figure 19: Validation Checking

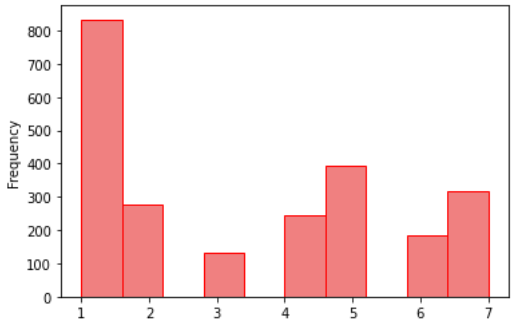


Figure 20: Frequency of each class

and if the URL is a google form will detect that and print all of the questions asking in that form and if any question contains any forbidden words it'll alert the user.

In the 3rd case Developed a twitter scrapper used twint and nest_asyncio to scrape any tweeter id, then developed a regular expression that will take URL's from a text, after that used urrlexpander to expand every URL then used the machine learning model developed in the last case and classify those URL's as a phishing / safe or a google form for google from it'll show the questions and is any question containing any forbidden words like password, number, etc or not.

In the 4th and final case build a sentiment classifier for influence or normal text we used the popular concept Cialdini's 6 Principles of Persuasion are reciprocity, scarcity, authority, commitment and consistency, liking, and consensus. By understanding these rules, people can use them to persuade and influence others. We made a manual dataset containing these 6 types of sentences with some examples of a normal sentence. Total there are near about 800 pieces of data we stored from the mail and many ads, we analyzed each data to check which sentence is on which type and which sentences are just normal and made the whole dataset. The frequency of the dataset is looks like in **Figure 20**

Then finetune the BERT-base-Uncased model and made a custom model with the help of the Tensorflowlibrary to classify those sentiments, in Figure 21 you can see the model

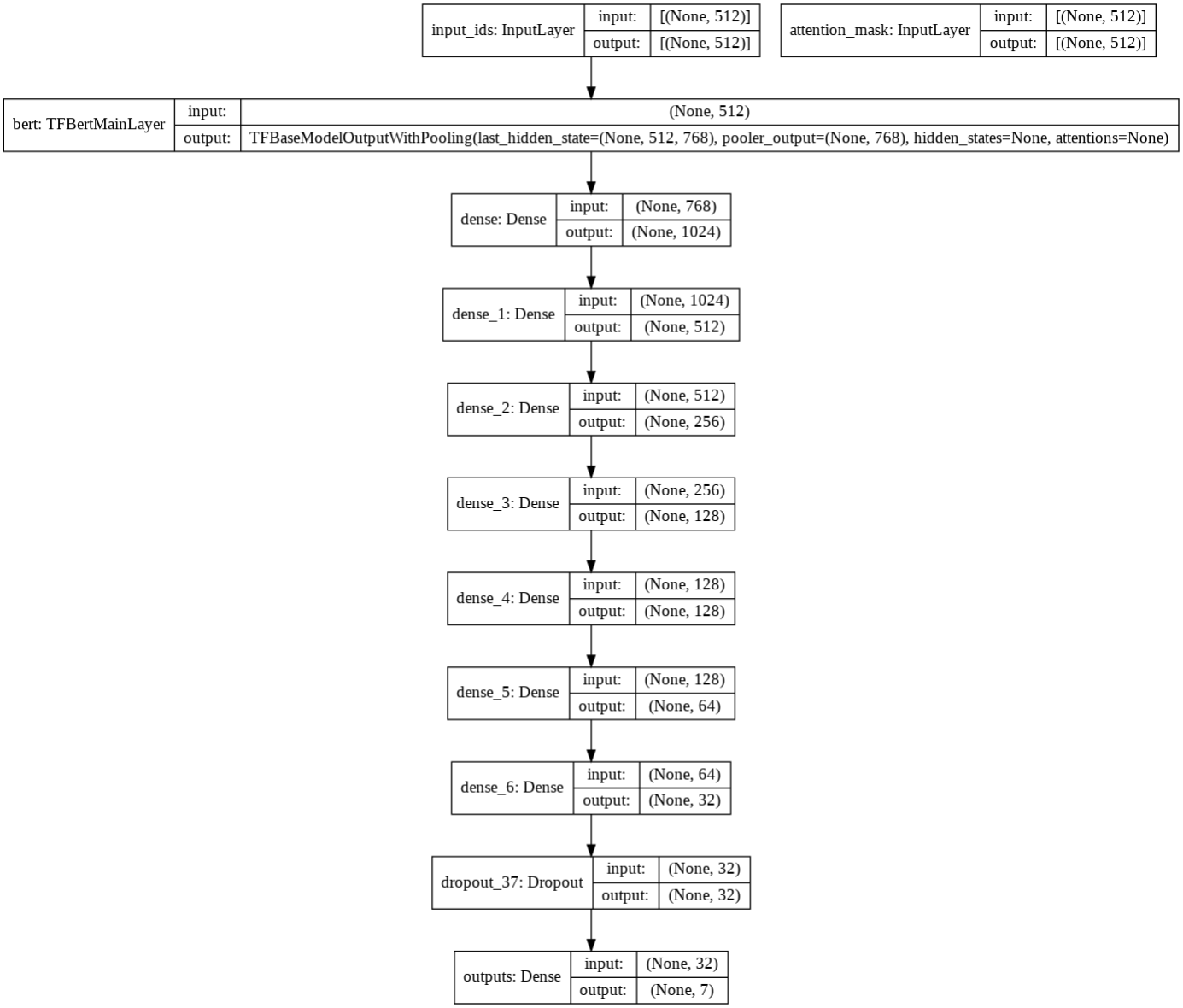


Figure 21: Model for Influence analysis

Figure 22: Real id Checking

Figure 23: Fake Id checking

We all know transformer works really well in these field, So we didn't use much time to make the model, main target was to check how each learning rate change effects the model, with the optimal one trained the data got some pretty good accuracybut for getting more variety we used the Twitter scraper build earlier used that to scrap some tweets from different types of companies (Flipkart, Amazon, Tinder, Tesla, Microsoft, SBI-Life, Rockstar-Games) used the previous model to predict all those tweets, then check the confidence of the model in each prediction and only took some prediction whose confidence is greater than 95%, in this way stored near about 2.5K data together. Then Checked all of those data manually to see if there is any wrong prediction or not and made the dataset again then trained the whole dataset with the previous model check the model performance with different learning rate then took the optimal one and trained it with more epoch to get the optimal accuracy.

5.2 Results

For evaluating the email id classifier I collected many email ids and it worked perfectly fine In figure 22, 23 you can see example with the real mail id and fake id of Flipkart

Then for the URL detection we used the test set containing 0.2% of the original data and got a result of 96% on the training data for the logistic regression used MultinomialNB and get 94% for the training data then used the pipeline of logistic regression with stop_words and RegexpTokenizer got 98% accuracy in a train data 96% in the test data.

Figure 24: Site Checking

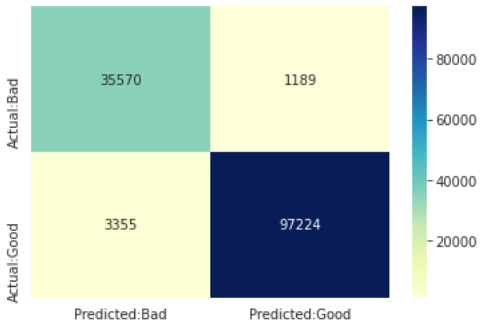


Figure 25: Confusion metrics for site Checking

And here's the confusion metrics.

Stored some random sites from daily emails and daily messages and evaluate each URL's from Virus Total and used my model for the same sites got near about 84% accuracy in this

Checked the Twitter model it's working fine for a given twitter id.
Example of this app is we can see in Figure 26, where I scraped my own twitter account and I have uploaded 2 google form in my account so we can see the results here how it can detect google from also And if you click any link it'll show the questions has been asked there and any question is asking forbidden information or not.

Fine-tune the Pre-Trained BERT model used the ADAM optimizer with the Categorical-Cross-entropy as a loss function made a custom model. Main focus was to check how a different learning rate changes the performance of a model for a particular dataset so keep the model same for both cases used different learning rate. First trained this model with the epochs number of 20 with differentLearning rate to check which one is more suitable for this datasetfirst used the learning rate as 0.01 , 0.001, 0.0001, 1e − 5 you can see the result in the next page at the table, Got the best accuracy at 0.0001 as a learning rate. Trained the model with 350 epochs got the best accuracy at 349thnumber of 97.46% with the loss of 0.2252, saved the best one with the callback.
After that in the next training time with that 2.5K data trained in the same optimizer and same loss first used the same type of learning rate 0.01, 0.001, 0.0001, 1e-5 and eventually got the best accuracy at the same 0.0001 within that 20 epochs trained it with 400 epochs got the best score as validation accuracy 92.76% with a loss of 0.2636(epoch - 365) so, here this is the optimal Learning rate we can assume, used this model on random real data from internet adds and got some good results.

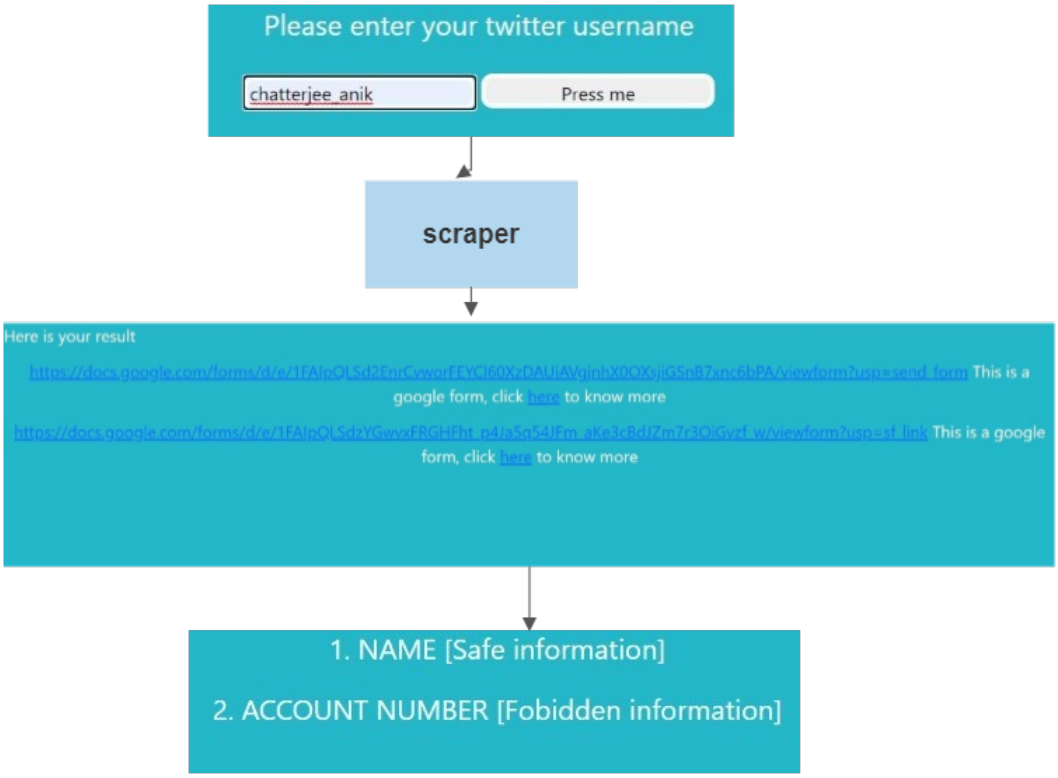
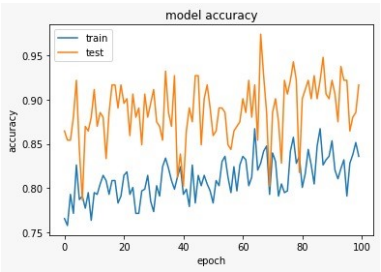


Figure 26: Twitter Scrapping

[Accuracy] 

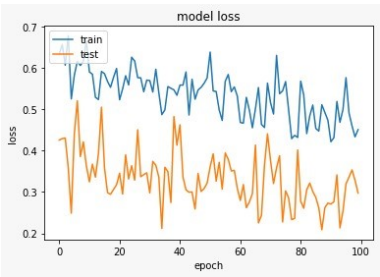
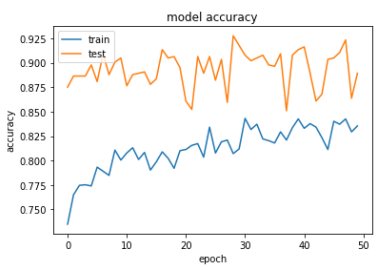
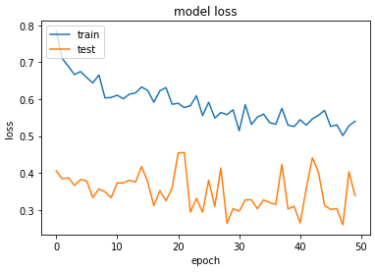
[Loss] 

Figure 27: Evaluate the first model(last 100 epochs)



[Accuracy]



[Loss]

Figure 28: Evaluate the 2nd mode(last 50 epochsl

Learning Rate	Accuracy	Epoch	Val$_A ccuracy$	Val$_L oss$
0.01	17.77%	20	17.61%	1.9349
0.01 (best)	17.97%	7	18.18%	1.9296
0.001 (best)	33.4%	20	36.65%	1.5483
0.0001	39.65%	20	39.2%	1.5046
0.0001 (best)	39.84%	19	49.15%	1.4150
1e-5	29.88%	20	31.68%	1.8800
1e-5 (best)	22.66%	15	32.67%	1.8964

Table 3: Smaller Dataset on Different Accuracy vs Learning_rate

So we can summarized the model performance in a table as For the first dataset with 735 data 3
For the final dataset with 2379 data 4
Note - In the above tables I checked the accuracy for the first 20 epochs (for those one write (best) beside means within 20 epochs In that particular epoch model gave the best accuracy)

The model reply with the probability of each influence and how much percentage of that sentence is which influence because in our research we observe maximum sentence are containing more than one type of influence so for those it's hard for even human to decide which one is the actual correct one so with the percentage it'll be easy for an user to understand

Learning Rate	Accuracy	Epoch	Val$_A ccuracy$	Val$_L oss$
0.01 (best)	35.22%	20	35.65%	1.7701
0.001	52.82%	20	51.56%	1.2727
0.001 (best)	48.98%	11	54.83%	1.2069
0.0001	56.25%	20	62.36%	1.0637
0.0001 (best)	53.61%	18	63.64%	1.1079
1e-5	44.17%	20	45.74%	1.5280
1e-5 (best)	41.65%	18	46.16%	1.5442

Table 4: Main Dataset on Different Accuracy vs Learning_rate

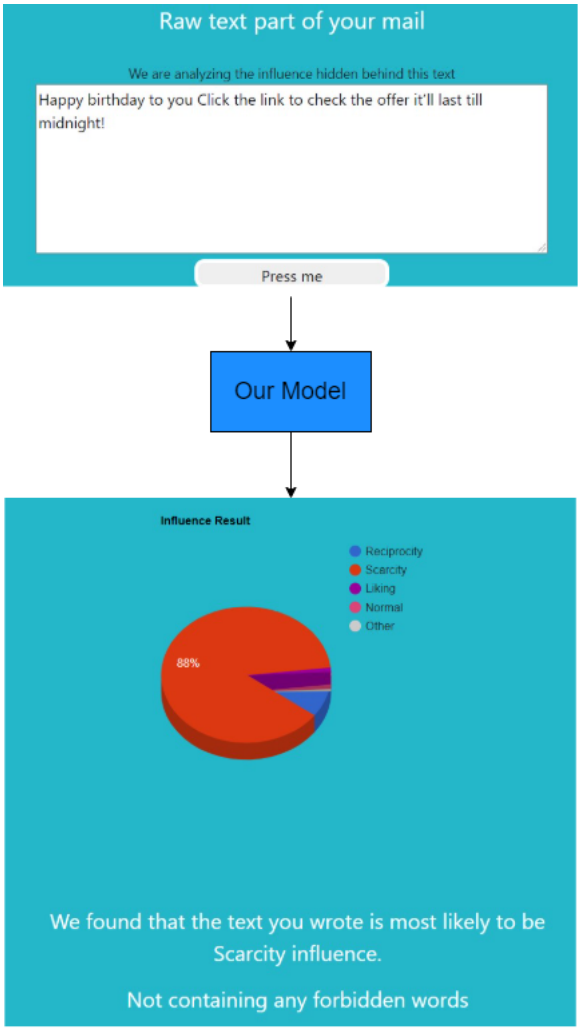


Figure 29: Result

the sentence and what its containing.for a user simplicity we have used pychart to make a user understand about the influence percentage. From this research we have made a list of words that are basically there if the sentence is forbidden so this model will inform the user about the influence and with that will notice it's containing any forbidden words or not
At **Figure 29** you can see one Real – Life example and how this model is reacting,for video explanation[10].

One thing to notice that all influential text aren't Phishing maximum of them are trying to influence you into something they will get benefited, just some business polices, Obviously the user should be aware of that and check the forbidden words to analyze. For access the whole code check here[11].

[10]https://drive.google.com/file/d/1iqKwNJJeyNwY4iKqKBhoBIu04w2lcNyK/view
[11]https://github.com/starboi2000/Phishing-detection-and-influence-analysis

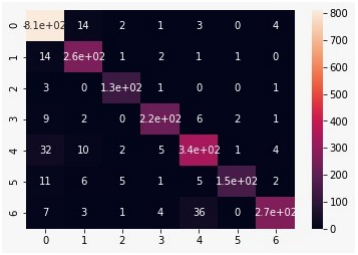


Figure 30: Confusion Metrics for Influence analysis

5.3 Discussion

This work is based on real-life data and real-life examples. There are many ways Phishing can fool anyone but these are the top points how we can stop phishing And be safe from the digital market. The benefit of using all realistic data is we are getting the results based on the reality so we can trust the model in this, and being another strong site is the models don't take a bunch of time to predict so the user will get their result quickly Now the main question comes how to be sure that this app is ready for the user to use. Though you can see my dataset which has a different variety, still there is a possibility that in the time of a split maximum of the same class is storing in the validation set. For example – if my validation set contains 90% of data that belongs to class 1(reciprocity), then by just printing (print('1')) I'll get 90% of accuracy. To make sure that is not happening in our case, to check this model is predicting well in the dataset, we use the help of confusion metrics

Confusion metrics -

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. Here,

Definition of the Terms:

Positive (P) : Observation is positive (for example: is an apple).

Negative (N) : Observation is not positive (for example: is not an apple).

True Positive (TP) : Observation is positive, and is predicted to be positive.

False Negative (FN) : Observation is positive, but is predicted negative.

True Negative (TN) : Observation is negative, and is predicted to be negative.

False Positive (FP) : Observation is negative, but is predicted positive.

In above picture we can see the confusion metrics of the prediction of my model. In this metrics classes are

$$\underline{Recall} = \frac{TP}{TP + FN} \qquad \underline{Precision} = \frac{TP}{TP + FP}$$

Figure 31: Formula for Recall and Precision

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

Figure 32: F1 - Score

0 – Reciprocity, 1 – Scarcity, 2 – Authority, 3 – Commitment, 4 – Liking, 5 – Social Proof, 6 – Normal

For understanding the performance better there are 3 things we need to check Recall, Precision and F1-Score
Recall: Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN). Recall is given by the relation:

Precision: To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP). Precision is given by the relation:
High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.
Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)
So, with that sense we can clear out that if my model is just printing 1 or something like that happening it'll give a low recall, high precision, let's see the model evaluation
Make it more-clear we can check the F1 – Score
F1 – Score: This combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision. In this case, the F1-scores for both the classifiers can be used to determine which one produces better results.
Now we can check these scores in our dataset by our model

Class and accuray	precision	recall	f1-score	support
0	0.91	0.97	0.94	834
1	0.88	0.93	0.91	277
2	0.92	0.96	0.94	132
3	0.94	0.92	0.93	244
4	0.87	0.86	0.87	392
5	0.97	0.84	0.90	183
6	0.96	0.84	0.89	317
accuracy			0.91	2379
macro avg	0.92	0.90	0.91	2379
weighted avg	0.92	0.91	0.91	2379

Table 5: Model Precision, Recall, and f1-score for each class

In this above result we can check the model precision, recall and f1-score on each class 0 to 6 and can understand how this model is fitting in this data even how is the accuracy for each class like for the 0th class it fitted 94% like this you can check the score, in this case for the overall model it fitted well 5.

6 Conclusion

We know there are many ways Phishing can make us fool within just one uncertain click or a mail or something like this. It'll best for all of us to be safe before clicking a link make sure you know about the site and you trust that URL, before trusting any mail make sure you know totally what is going on. So, the novelty of this application is here to help users to understand what they are dealing with, which link they are clicking is it safe or not? the text they are getting is it influential and is it containing any forbidden words? is the mail id is from a safe place or not? By those methods, you can decide that is Phishing or safe. And as we showed earlier those models got a pretty good accuracy of understanding this so you can trust this. Lastly, we are trying as much as we can to prevent phishing to make the world is a safer place but,lastly it's up to the user to click on something or response on something we can just suggest at the last check everything before diving into something

References

[1] MakuochiNkwo and Rita Orji. *Persuasive Technology in African Context: Deconstructing Persuasive Techniques in an African Online Marketplace. 2nd African Computer Human Interaction Conference (AfriCHI'18)*. Windhoek, NamibiaAt: Windhoek, Namibia, December 2018.

[2] Kiemute Oyibo, Ifeoma, Adaji, Rita Orji, and JulitaVassileva. *The Susceptibility of Africans to Persuasive Strategies: A Case Study of Nigeria. Persuasive Technology International Workshop 2018*. Waterloo, Canada, June 2018.

[3] Ifeoma Adaji ,Kiemute Oyibo, and JulitaVassileva. *E-Commerce Shopping Motivation and the Influence of Persuasive Strategies. ORIGINAL RESEARCH, Frontiers in Artificial Intelligence*. November 2020.

[4] Merton Lansley, Nikolaos Polatidis, and Stelios Kapetanakis. *SEADer: A Social Engineering Attack Detection Method Based on Natural Language Processing and Artificial Neural Networks*. In book: Computational Collective Intelligence, August 2019.

[5] Shilpa P C, Rissa Shereen, Susmi Jacob, Vinod P. *Sentiment Analysis Using Deep Learning,. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. February 2021.

[6] OlhaKaminska, Chris Cornelis, and Veronique Hoste. *Fuzzy-Rough Nearest Neighbour Approaches for Emotion Detection in Tweets, the IJCRS 2021 conference*. organized jointly with IFSA-EUSFLAT 2021, July 2021.

[7] Sridhar Ramaswamy, Natalie DeClerck . *Customer Perception Analysis Using Deep Learning and NLP Complex Adaptive Systems Conference with Theme: Cyber Physical Systems and Deep Learning*. CAS, November 2018.

[8] Neha Jadav, Sagar Pande, Aditya Khamparia . *Twitter Sentiment Analysis Using Deep Learning. IOP Conference Series: Materials Science and Engineering.* Rajpura, January 2021

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova . *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Annual Conference of the North American Chapter of the Association for Computational Linguistics.* May 2019 [Version 2].

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, IlliaPolosukhin. *Attention Is All You Need. Conference on Neural Information Processing Systems.* December 2017 [Version 5].

[11] Robert B. Cialdini 2004, *The Science of Persuasion.* Scientific American Mind 284, (2004), 76-84.

[12] Ambik Mitra, Lambodar Jena, Soumya Sahoo *Emoji Analysis Using Deep Learning, Advances in Intelligent Computing and Communication (pp.689-698).* May 2021

[13] Oza Pranali P., Deepak Upadhyay *Review on Phishing Sites Detection Techniques.International Journal of Engineering and Technical Research V9(04).* May 2020

[14] Shoma Tanaka, Takashi Matsunaka, Akira Yamada, Avumu Kubota *Phishing Site Detection Using Similarity of Website Structure.IEEE Conference on Dependable and Secure Computing (DSC).* Jan 2021

[15] Deb Prakash Chatterjee, Anirban Mukherjee, Sabyasachi Mukhopadhyay, Mrityunjoy Panday, *A Survey on Sentiment Analysis.Emerging Technologies in Data Mining and Information Security, Proceedings of IEMIS 2020, Volume 2 (pp.259-271)* May 2021

[16] Orestes Appel, Francisco Chiclana, Jennifer Carter, Hamido Fujita, *Consensus in Sentiment Analysis, Fuzzy Logic (pp.35-49)* May 2021

[17] Phishing Statistics: Updated 2021,
`https://worldofwork.io/2019/07/cialdinis-6-principles-of-persuasion/`

[18] Bert-base-uncased,
`https://huggingface.co/bert-base-uncased`

[19] Twint Project,
`https://github.com/twintproject/twint`

[20] validate_email 1.3
`https://pypi.org/project/validate`$_e$`mail/`.