PREVENTING SOCIAL MEDIA PHISHING IN SOCIAL NETWORKS

Project Report

Submitted in partial fulfillment for the award of degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By

Sripathi Dinesh Naga Tej (Regd.No.20L31F0058)

Under the esteemed guidance of

Mrs. P. Pavithra M.Tech (Ph.D)

Asst.Professor

VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE - New Delhi & Affiliated to JNTUK, Kakinada)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE - New Delhi & Affiliated to JNTUK, Kakinada)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

Department of Master of Computer Applications

CERTIFICATE

This is to certify that the Project report entitled **"PREVENTING SOCIAL MEDIA PHISHING IN SOCIAL NETWORKS "** is a bonafide record of project work carried out under my supervision by **Sripathi Dinesh Naga Tej** bearing Regd. No : 20L31F0058 in partial fulfillment of the degree of Master of computer Applications **of** Vignan's Institute of Information Technology(A) affiliated to Jawaharlal Nehru Technology University Kakinada(JNTUK), during the academic year 2020-2022.

Signature

Mrs. P. Pavithra

Signature

Dr.G.Rajendra Kumar (HOD)

EXTERNAL EXAMINER

DECLARATION

I here by declare that this project report entitled **" PREVENTING SOCIAL MEDIA PHISHING IN SOCIAL NETWORKS"** has undertaken by me for the fulfillment of Degree in Master of computer Applications. I declare that this project report has not been submitted anywhere in the part of fulfillment for any degree of any other University.

Place : Visakhapatnam **Student Name**

Date : **Sripathi Dinesh Naga Tej**

ACKNOWLEDGEMENT

An endeavor over a long period can be successfully with the advice and support of many well-wishers. I take this opportunity to express our gratitude and appreciation to all of them.

I express my sincere gratitude to my internal guide, **Mrs. P. Pavithra** for her encouragement and cooperation in completion of my project. I am very fortunate in getting the generous help and constant encouragement from her.

I would be very grateful to our project coordinator, Mrs.A.Sirisha for the continuous monitoring of my project work. I truly appreciate for her time and effort spent.

I would like to thank our Head of the Department, Dr. G. Rajendra Kumar and all other teaching and non-teaching staff of the department for their cooperation and guidance during my project.

I sincerely thank to **Dr. B. Arundhati,** Principal of VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A) for her inspiration to undergo this project.

I wanted to convey my sincere gratitude to **Dr. V. Madhusudhan Rao**, Rector of VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A) for allocating the required resources and for the knowledge sharing during my project work.

I extended my grateful thanks to our honorable Chairman **Dr. L. Rathaiah** for giving me an opportunity to study in his esteemed institution.

Sripathi Dinesh Naga Tej

(20L31F0058)

VIGNAN's INSTITUTE OF INFORMATION TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE - New Delhi & Affiliated to JNTUK, Kakinada)
Beside VSEZ, Duvvada, Vadlapudi Post, Gajuwaka, Visakhapatnam - 530 049.

MASTER OF COMPUTER APPLICATIONS

VISION :

- We aim to generate groomed, technical competent and skilled intellectual professionals.
- We serve as a valuable resource for modern industry and current society.

MISSION:

- Providing strong theoretical and practical knowledge in computer science discipline with an emphasis on software development.
- To provide need based quality training in the field of information technology.
- Impart quality education to meet global standards and achieve excellence in teaching-learning and research.
- To provide students with the tools to become productive, participating global citizens and life-long learners.

MASTER OF COMPUTER APPLICATIONS

PROGRAMME OUTCOMES

At the end of the programme the student shall be able to

- Application of Enginneering Knowledge
- problem analysis
- Design development of solultions
- conduct investigation of complex problems
- Modern tool usage
- The engineer and society
- Environment and sustainability
- Ethics
- Individual team work
- communication
- Project management and finance
- Lifelong learning

PREVENTING SOCIAL MEDIA PHISHING IN SOCIAL NETWORKS

Abstract

In this project, we employ machine learning, specifically a man-made neural network, to evaluate the probability that a social account is authentic or not. We also describe the relevant classes and libraries. We also examine the sigmoid function and the selection and application of the weights. Finally, we take into account the social network page's parameters, which are crucial to the offered solution.

INDEX

CHAPTER 1

INTRODUCTION

1. Introduction

Facebook became the most popular social media platform in 2017 after reaching a total user base of 2.46 billion. Users' information is used by social media networks to generate cash [1]. The usual user is unaware that when they use the service of a social media network, their rights are forfeited. Social media firms stand to earn greatly at the expense of the user. Facebook generates income from adverts and user data whenever a user posts new content, such as locations, images, likes, and dislikes. More specifically, the average American user produces around $26.76 every three months [2]. When several users are involved, that sum soon grows.

In the current digital era, the growing reliance on technology has made the average person more susceptible to crimes like data breaches and potential identity theft. These attacks can happen suddenly and frequently without warning to those who have experienced a data breach. Social networks currently have little reason to strengthen their data security. These hacks frequently target Twitter and Facebook, among other social media platforms. Banks and other financial organisations will also be on their radar.

Social media platforms getting hacked seem to be a newsworthy topic every day. About 50 million Facebook users were impacted by a recent knowledge breach [3]. Facebook outlines a number of clearly stated provisions that detail how they use user data [4]. The policy does very little, if anything, to stop the ongoing invasion of privacy and security. The built-in security measures on Facebook appear to be overcome by fake profiles.

Bots and phoney profiles represent the opposite threat of personal information being collected for illegal reasons. Bots are computer programmes that collect data about users without their knowledge. Web scraping is the term for this activity. Even worse, this action is legal. On social networks, bots are frequently concealed or seem as phoney friend requests to get private data.

This project proposed approach aims to highlight the risks posed by a bot that poses as a false social media presence. An algorithm would be available to implement this solution. Python is the language we decided to utilise. The algorithm would be prepared to determine whether a user is receiving a friend request from a legitimate person, a bot, or a phoney friend request that is gathering information. Since we might need a training dataset from the social media companies to build our model and then determine if the profiles are phoney or real, our algorithm would function with their help. The method might potentially be used as a standard layer on the user's web browser as a browser plug-in.

CHAPTER 2

LITERATURE SURVEY

2. Literature Survey (list of papers referred)

Sybil rank, a ranking graph-based method, was created in late 2012 to effectively identify fraudulent profiles [5]. To spread trust, the algorithm combines an early terminated random walk technique with seed selection [5]. The computational expense is expressed in O (nlogn). According to the quantity of interactions, tags, wall postings, and friends throughout time, profiles are ranked. High-ranking profiles are taken into account to be genuine, whereas low-ranking ones are deemed to be false. Unfortunately, it was discovered that this system was largely unreliable because it overlooked the possibility that actual accounts may be ranked poorly and false profiles could be ranked well.

A unique method for identifying bogus profiles was proposed by Sarode and Mishra [6]. They created a script to retrieve the seen data and used the Facebook graph API tool to gain access to many profiles. This gathered data is later used to create the features that the classifier will incorporate into their algorithm. The information is first in JSON format, which is then further converted to a structured format (CSV) that is simpler for machine learning techniques to understand. Later on, the classifier will function more effectively thanks to these comma separated values. The authors experimented with both supervised and unsupervised machine learning methods. Supervised machine learning algorithms performed better in this example, with an accuracy rate of approximately 98%. The dataset for supervised machine learning is divided into training and testing sets. They used 80% of the samples to coach the classifier and the rest to test it [6].

Sybil Frame classifies at multiple levels. There are both content-based and structure-based methodologies. The dataset is analysed using a content-based technique, which extracts data necessary to compute historical information about nodes and edges. Using a Markov random field and loopy belief propagation, which uses prior knowledge, the structure-based technique correlates nodes. The first step of the Sybil Frame technique uses a content-based approach, and the second stage uses a structure-based approach.

CHAPTER 3

SYSTEM ANALYSIS

3. System Analysis

3.1 Existing System :

Malicious users create fake profiles to phish login information from unsuspecting users. A fake profile will send friend requests to many users with public profiles. These counterfeit profiles bait unsuspecting users with pictures of people that are considered attractive. Once the user accepts the request, the owner of the phony profile will spam friend requests to anyone this user is a friend.

In Existing system they using Convolutional Neural Network(CNN) and bi-SN-LSTM to extract lower and higher features of the input data, e the combination of SELU and alpha dropout for preserving the self-normalizing property and relieving the overfitting problem along the training process.

3.2 Proposed System :

In this project using Artificial Neural Networks(ANN) we are identifying whether given account details are from genuine or fake users. ANN algorithm will be trained with all previous users fake and genuine account data and then whenever we gave new test data then that ANN train model will be applied on new test data to identify whether given new account details are from genuine or fake users.

3.3 Feasibility study :

We are using sigmoid activation function for binary classification and it provides more accurate results by classifying on online social networks such as Facebook or Twitter contains users details and some malicious users will hack social network database to steal or breach users information. To protect users data we are using ANN Algorithm and sigmoid activation function.

To train ANN algorithm we are using below details from social networks

Account_Age, Gender, User_Age, Link_Desc, Status_Count, Friend_Count, Location, Location_IP, Status

For the training set, the features that we use to determine a fake profile are Account age, Gender, User age, Link in the description, Number of messages sent out, Number of friend requests sent out, Entered location, Location by IP, Fake or Not. Each of these parameters is tested and assigned a value. For example, for the gender parameter if the profile can be determined to be a female or male a value of (1) is assigned to the training set for Gender. The same process is applied to other parameters. We also use the country of origin as a factor.

CHAPTER 4

SYSTEM SPECIFICATIONS

4. Software Specifications

4.1 Functional Requirements :

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.

4.2 Non-functional requirements :

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

4.3 Hardware requirements :

- Operating System supported by
- Windows 7

- Windows XP
- 3 . Windows 8
- Processor – Pentium IV or higher
- RAM -- 256 MB
- Space on Hard Disk -- Minimum 512 MB

4.4 Software requirements (SRS) :

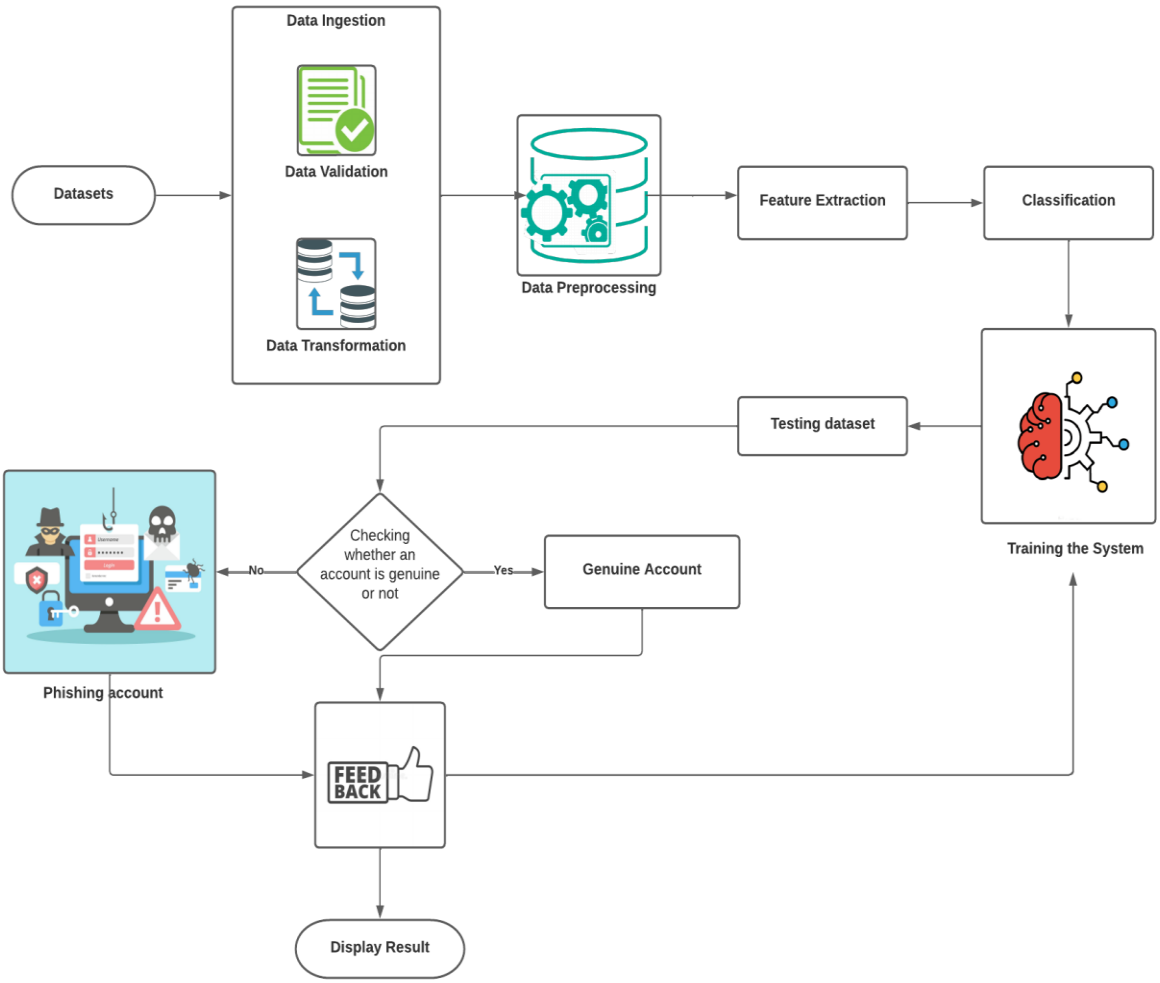
- For developing the Application
  - Python
  - Django
- Technologies and Languages used to Develop -- Python

CHAPTER 5

SYSTEM DESIGN

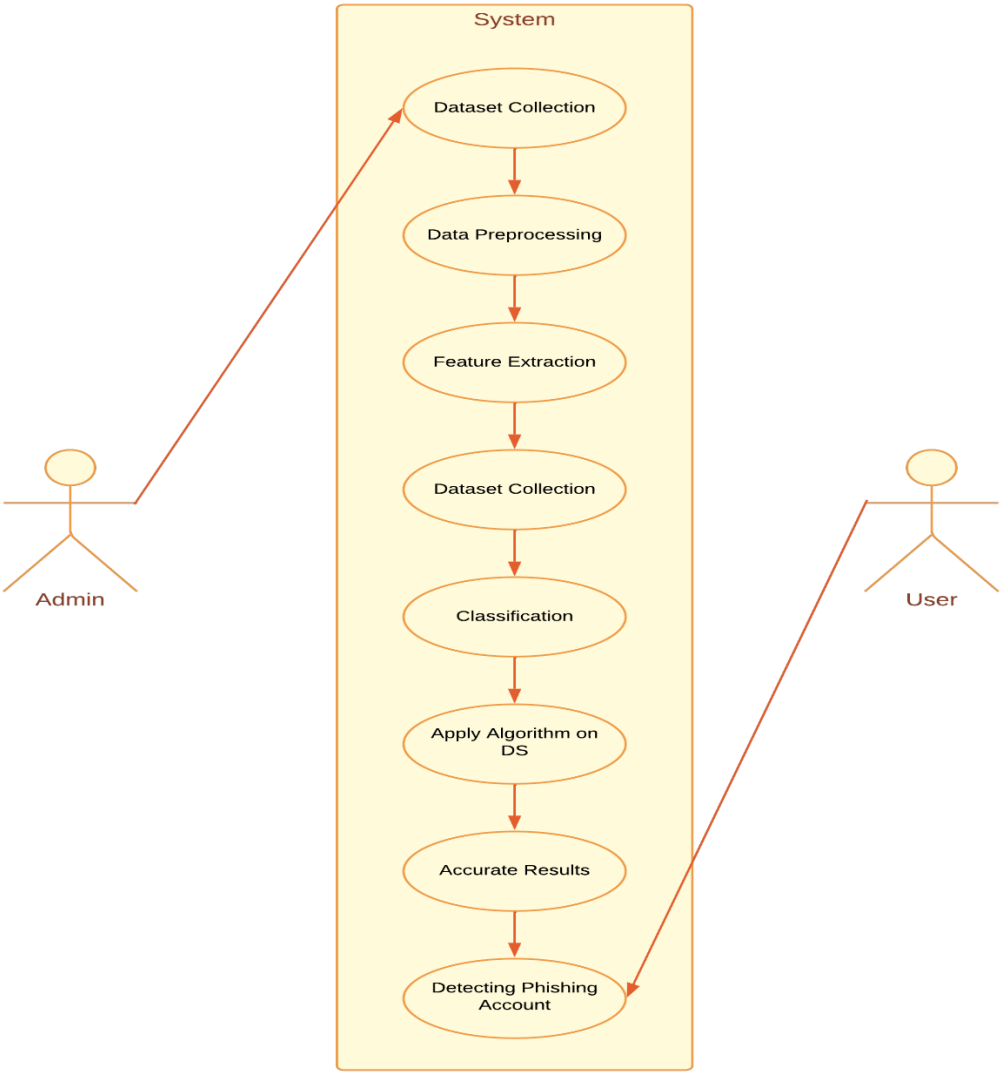
5. System Design

5.1 System Architecture :



5.2 UML Diagrams :

5.2.1 Usecase Diagram :

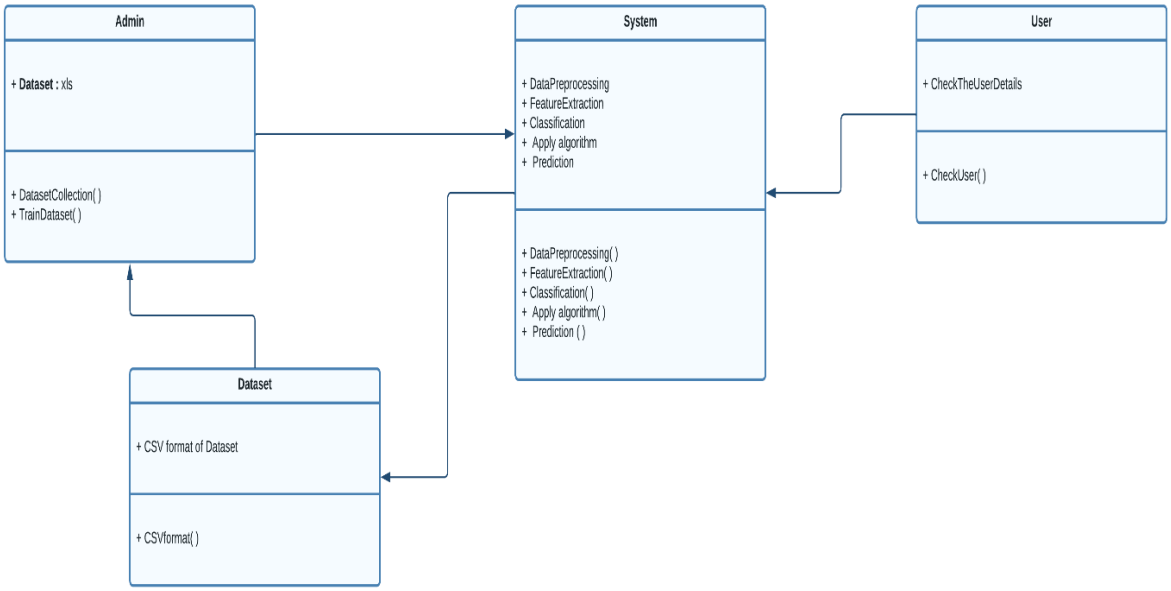


The use case diagram is used to represent all the functional use cases that are involved in the project.

The above diagram represents the main two **actors** in the project, they are

- User
- Admin

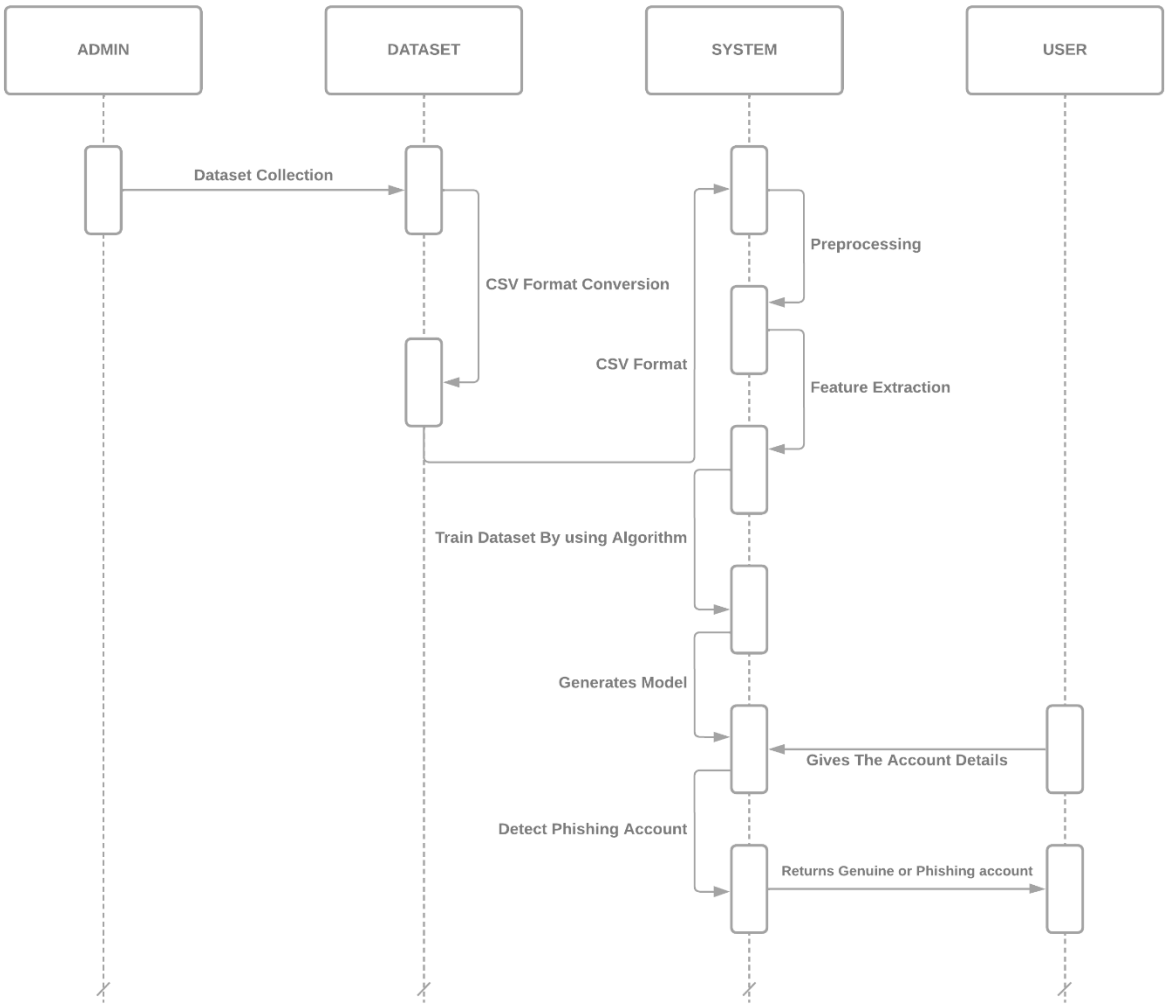
5.2.2 Class Diagram :



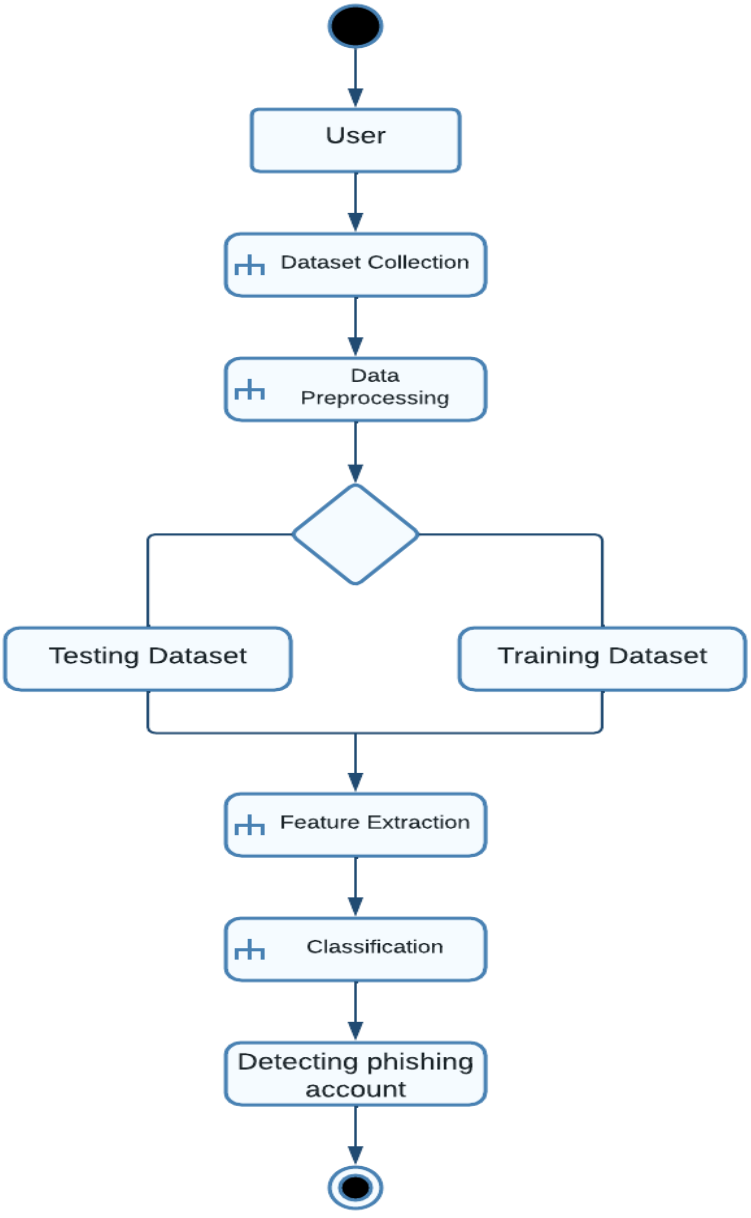
The above mentioned class diagram represents the Chatbot system workflow model. This diagram has class models with class names as

- User
- Admin
- System

5.2.3 Sequence Diagram :

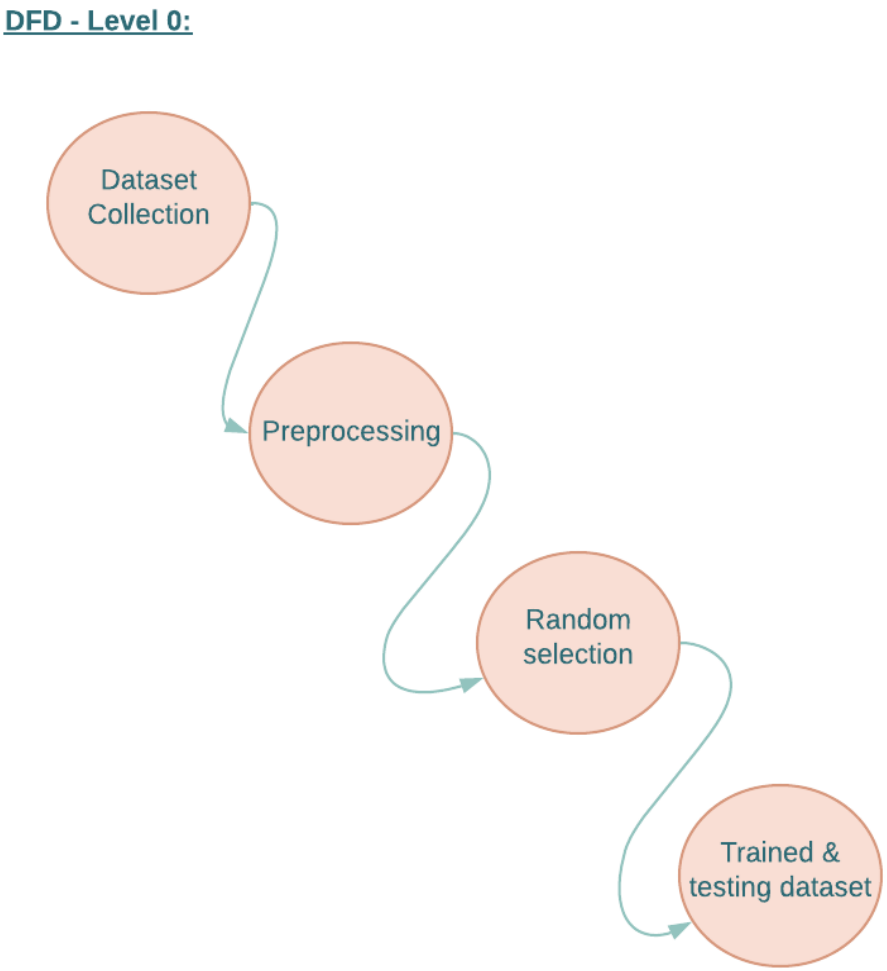


5.2.4 Activity Diagram :

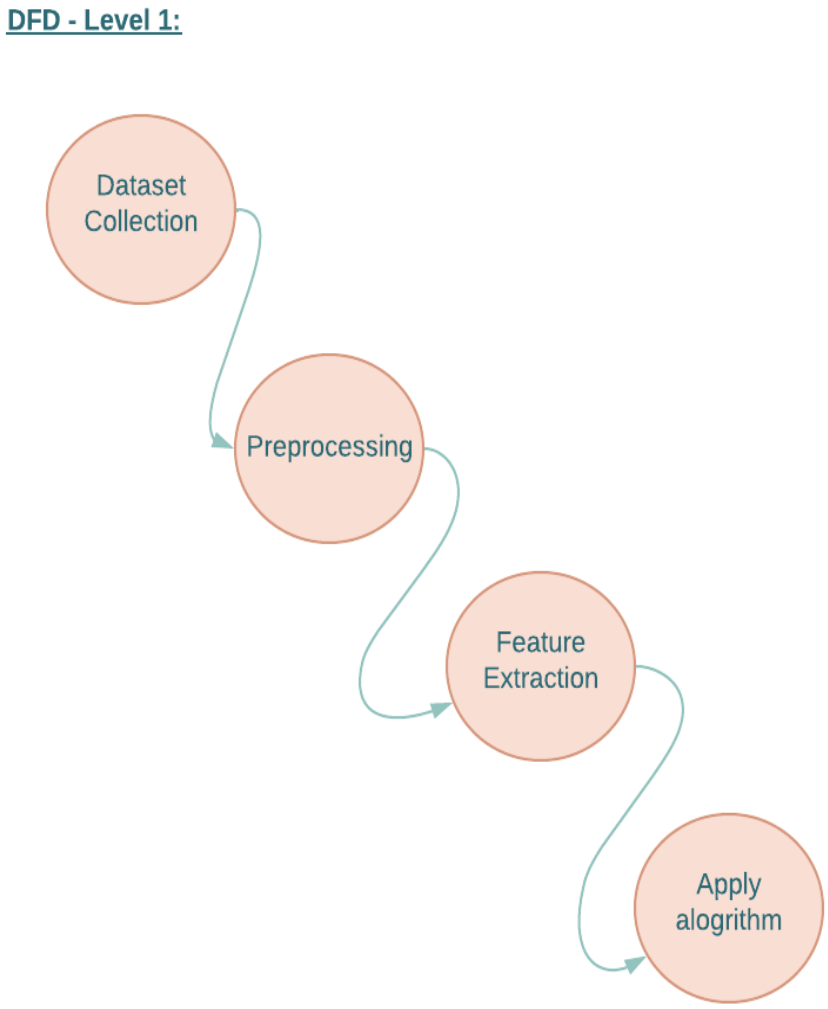


5.3 Data Flow Diagram :

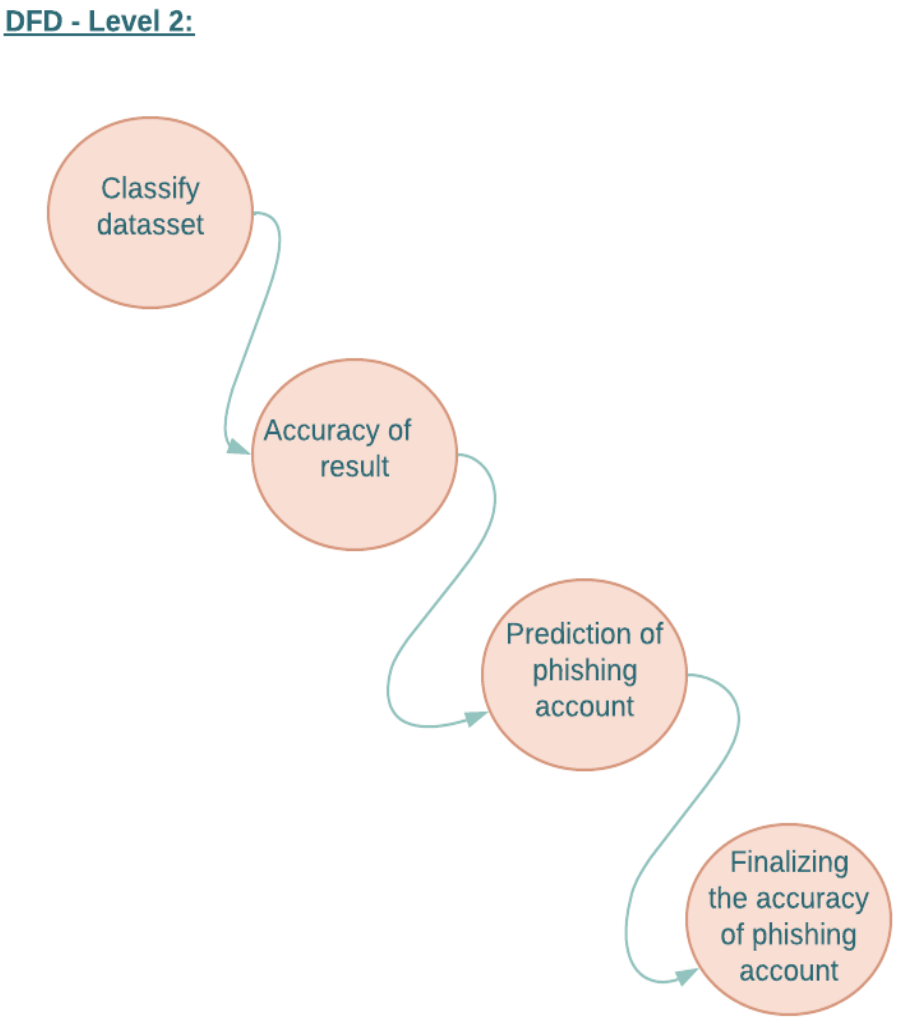
5.3.0 Level 0 :



5.3.1 Level 1 :


DFD - Level 1:
Dataset Collection
Preprocessing
Feature Extraction
Apply alogrithm

5.3.2 Level 2 :


DFD - Level 2:
Classify datasset
Accuracy of result
Prediction of phishing account
Finalizing the accuracy of phishing account

CHAPTER – 6

SYSTEM IMPLEMENTATION

6. System Implementation

6.1 Project Modules :

6.1.1 Admin Module:

The administrator will enter the application with the credentials "admin" and "admin" and then carry out the following tasks.

- **Generate ANN Train Model:** The administrator will upload the profile dataset. By using test data from new accounts, this train model is frequently used to determine whether an account is real or false.
- **View ANN Train Dataset:** The admin can access all the datasets used to train the ANN model by using this module.

6.1.2 User Module:

Anyone can use this application, input test data from the most recent account, and invoke the ANN algorithm. A new set of test data will be used, and an ANN algorithm will be employed to forecast if the test data comprises real accounts or phishing accounts.

6.2 Methodology (Algorithms) :

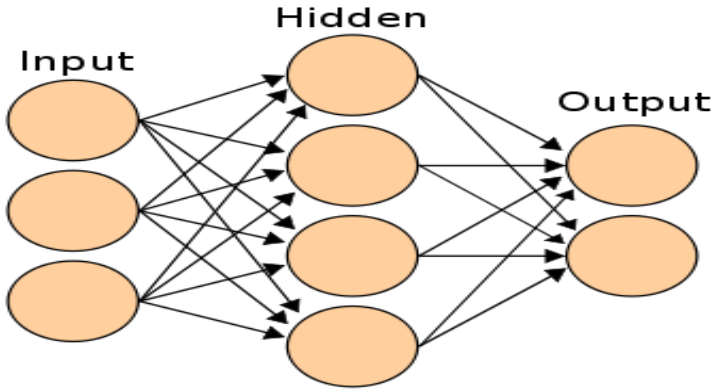
6.2.1 Artifical Neural Networks Algorithm :

I'm applying artificial neural networks created through machine learning to evaluate the likelihood that a friend request is genuine or not. Every

neuron (node) processes each equation through a Sigmoid function to keep the answers between the range of 0.0 and 1.0. This could easily be multiplied by 100 at the output end to provide us with the likelihood that the request is malicious. Our approach would consist of a single deep neural network with a single hidden layer.

Formulation of Neural network :

Let's start by understanding formulation of a simple hidden layer neural network. A simple neural network can be represented as shown in the figure below:



The most amazing fact in an ANN is the linkage between nodes. The ANN algorithm only uses inputs as known values. Weights represent the nodes' connectivity during this method.

Following is the framework in which artificial neural networks work :

1. To begin the algorithm, assign random weights to all or some of the links.
2. Determine the activation rate of Hidden nodes by using the inputs and the ensuing (input -> Hidden node) linkage.

3. Calculate the activation rate of the output nodes using the connections and hidden nodes' activation rates.
4. Determine the output node's error rate and recalibrate every link between hidden nodes and output nodes.
5. Cascade the error to Hidden nodes using the weights of and error discovered at output node.
6. Adjust the weights between the input nodes and the hidden nodes.

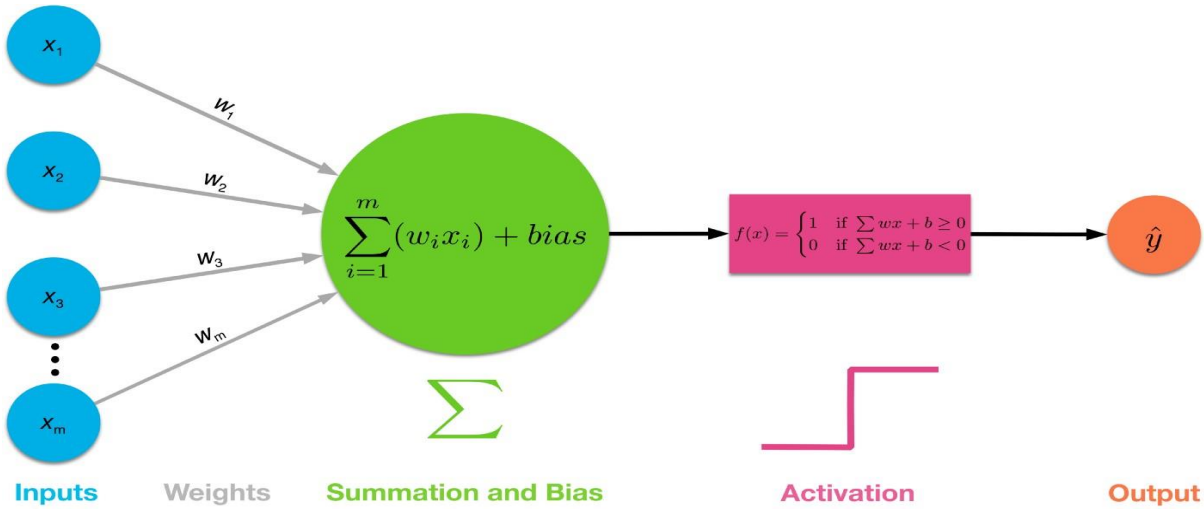
   Error **@ H1 is equal to W(H1O1)*Error@O1 plus W(H1O2)*Error@O2.**
7. Continue using this approach until the convergence requirement is reached.
8. Score the activation using the ultimate linkage weights. rate of the output nodes.

6.2.2 Sigmoid Activation Function :

Something that is curved in two directions is described by the term "sigmoid." We are only interested in one of the several sigmoid functions. Given that it is known as the logistic function, the mathematical formulation is simple:

f(x) = 1/1+e-x

The maximum value of the curve is determined by the constant L, hence the constant k affects how steep the transition is. The plot below displays examples of the logistic function for various values of L; thus, the plot below displays curves for various values of k.



The threshold value is where classification occurs, and the graph of the sigmoid activation function will have an S-shape.

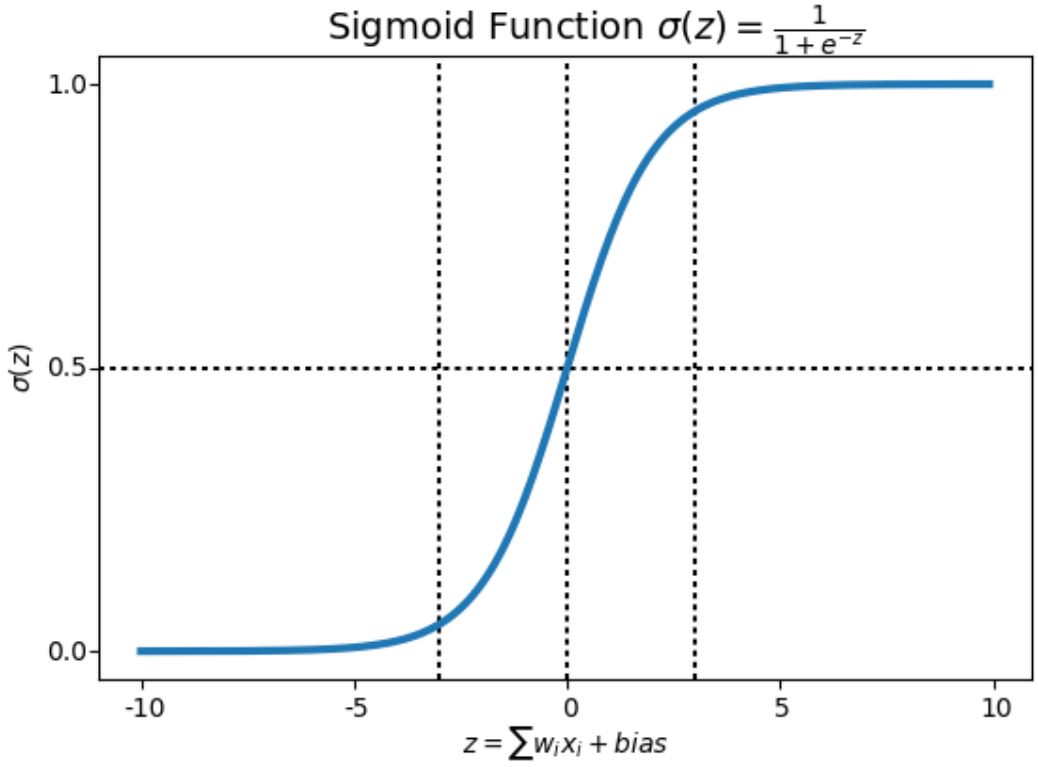


fig : Sigmoid Graph

6.2.3 Implementation :

Our deep learning algorithm is written in the Python language. The used libraries are:

• Pandas

• NumPy

• MATLAB

• theano

• scikit-learn

• Keras

• TensorFlow

We utilize Microsoft Excel to store old and new fake data profiles. The algorithm then stores the data in a data frame. This collection of data will be divided into a training set and a testing set. We would need a data set from the social media sites to train our model.

For the training set, the features that we use to determine a fake profile are Account age, Gender, User age, Link in the description, Number of messages sent out, Number of friend requests sent out, Entered location, Location by IP, Fake or Not. Each of these parameters is tested and assigned a value. For example, for the gender parameter if the profile can be determined to be a female or male a value of (1) is assigned to the training set for Gender. The same process is applied to other parameters. We also use the country of origin as a factor. The top country with highest bot activity is China with the United States coming in as a strong second [8].

As stated before, there are different layers to the algorithm. For example, there are 128 nodes in the hidden layer. There is also an input layer and an output layer. With a single hidden layer is used, it is called a one deep machine learning algorithm. These layers are intended to mimic a neural network. In our case, it is named an artificial neural network or ANN [9]. This system can be used in AI programs to solve problems. It is often used in face recognition, pattern recognition and even in training virtual assistants (Siri, Alexa, etc.). This type of model is used to behave like the human brain. Different nodes would represent specific parameters; for example, there may be a node for the Age parameter and another one for the Gender parameter and so forth. Based on the inputs provided an output (decision) is produced. The inputs are directed to the hidden layer.

The data is to read using the readCSV() method. This is read into the training set. We use the dropna() method to clean up the data set. Using the correct parameters and formatting the data correctly is one of the most important things when building an ANN. We then convert the parameters such as Account Age, Gender, etc. to a numerical form somewhat like an enumerated data type. We convert the Account Age into weeks. We then compare the IP address parameter with the actual IP address of the profile. If it is a match, then a value of (1) is assigned to locationMatch. If it does not match, then the value of (0) is assigned to locationMatch. The same process is repeated for Gender. We used the match() method to compare the link in the description with the training set's link in the description. A Boolean value of true or false is assigned to the url_found variable. If it is true, the value (1) is assigned to the Link in the description parameter. If it is false, the value (0) is assigned.

We then determine the Number of messages sent out parameter by dividing the number of messages sent by the age of the account. We then determine the Number of

friend requests sent out parameter by dividing the Number of friend requests sent out by the age of the account. We use the Iloc() method to adjust the columns in the data set. We use this method for our input set and our outputs set.

Next, we split the input set and the output set in half. This will leave us with four different sets. The first input set is assigned to the input train. This variable contains the second half of the input set. The second input set is assigned to the input test. This variable contains the first half of the input set. The first output set is assigned to the output train. This variable contains the second half of the output set. The second output set is assigned to the output test. This variable contains the first half of the output set.

We use the standardscaler() class [10]. This allows us to convert the data into a uniform distribution form so that it has a mean value of (0) and a standard deviation of (1). We use fitTransform() method on the input train parameter and the transform() method on the input_set parameter. This converts the data to the intended distribution.

The next step is to use Sequential from the Keras library to create the model. We use the add() method of the Sequential class to activate the Sigmoid function and again to generate the output layer. Afterward, we compile the model using Stochastic Gradient Descent as an optimizer. The neural network is then fitted to our training set using the fit() method.

We have to test the accuracy of the input_test variable which contains one half of the input set. We do this using the predict() method. The result is then converted to a percentage. This result is stored in the output_pred variable, which is in turn stored in our database under a new column. The input_test and the output_test variables are then passed as parameters to the evaluate() method. The result of the evaluate() method is assigned to score. The score is used to determine whether or not the profile is real or fake.

Libraries :

Through the use of different libraries, we can easily make machine learning and data mining possible. The most common language in use for artificial intelligence today is Python, due to its popularity, compactness and the various ready-to-use libraries that are perfect for mathematical models.

One such library is [11] Pandas, which is an excellent tool for data analysis. It can be used in a wide range of fields including academic and commercial domains such as finance, economics, statistics, analytics, etc [12]. It can easily access and modify different datasets and optimize them for later use. Correct formatting of the datasets and finding the optimal vital features that are used later are crucial. Another library worth mentioning is [13] NumPy. NumPy can be used for scientific computing and used primarily for multi-dimensional matrix multiplication as we are dealing with a large amount of numbers that are very dependent on each other.

A similar tool,[14] MATLAB is often used to plot and visualize mathematical models for analysis or as Numpy, for matrix multiplications. The Theano library [15] is also ideal for evaluating mathematical expressions involving multidimensional arrays. For plotting and visualizing, data analysis and data mining, you can use the Scikit-learn (sklearn) machine learning library [16]. It is built on top of the NumPy, SciPy, and matplotlib libraries. Scikit-learn features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, kmeans and DBSCAN. It was initially developed as a Google summer code project in 2007.

The two most essential libraries are Keras and TensorFlow . Keras [17] itself is capable of running on top of TensforFlow [18], CNTK, or Theanoc. It enables fast

experimentation and prototyping. Keras core structure is a model, a way to organize layers. It was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). The name Keras means horn is Greek.

Another viral library is TensorFlow[18] , which was developed by Google Brain with Google's AI organization for internal use initially. It is now an open-source software library that is ideal for machine learning, mainly using neural networks.

6.3 Source Code :

6.3.1 AdminLogin page.html :

```
{% load static %}
<html>
<head>
<title>Preventing Social Media Phishing in social networks </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'default.css' %}" rel="stylesheet" type="text/css" media="screen" />
 <script LANGUAGE="Javascript" >
function validate(){
        var x=document.forms["f1"]["username"].value;
        var y=document.forms["f1"]["password"].value;
        if(x == null || x==""){
```

```
                window.alert("Username must be enter");
                document.f1.t1.focus();
                return false;
        }
        if(y == null || y==""){
                window.alert("Password must be enter");
                document.f1.t2.focus();
                return false;
        }
        return true;
}



        </script>
</head>
<body>
<div id="wrapper">
  <div id="header">
    <div id="logo">
      <h1><center><font color="orange" size=6>Preventing Social Media Phishing in social networks </font></center></h1>
          <marquee><font color="pink" size=4>Social Media Phishing Profiles Detection</font></marquee>
    </div>
```

```
 </div>
</div>
<div id="menu">
  <ul>
   <li><a href="{% url 'index' %}">Home</a></li>
   <li><a href="{% url 'Admin' %}">Admin</a></li>
   <li><a href="{% url 'User' %}">Users</a></li>
</ul>
</div>
<div class="entry">
 <br/><br/><br/>
      <font size="" color="white"><center>{{ data }}</center></font>
       <br>
       <center><font size="5" color="white">Admin Login Screen</font></center>
      <form name="f1" method="post" action={% url 'AdminLogin' %} OnSubmit="return validate()">
      {% csrf_token %}
      <br><br>
      <TABLE align=center width="35%" class="notepad">
            <TR><TH align="left">Username
            <TD>  <Input type=text name="username" value=" class="form-control">
            <div id='nameid'></div>
            </TD>
```

```
                </TR>
                <TR></TR>
                <TR></TR>
                <TR></TR>
                <TR><TH align="left">Password
                <TD>  <Input type='password' name="password"
value=" class="form-control">
                </TR>
                 <TR>
                <TD></TD>
                        <TD>
                        <br><br>
                 <input type="submit" value="login">
                 </TABLE>
</form>
</div>
</body>
</html>
```

6.3.2 UserPannel.html :

```
{% load static %}
```

```
<html>
<head>
<title>Preventing Social Media Phishing in social networks</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'default.css' %}" rel="stylesheet" type="text/css"
media="screen" />
 <script LANGUAGE="Javascript" >
function validate(){
        var x=document.forms["f1"]["t1"].value;
        if(x == null || x==""){
                window.alert("Please enter account details");
                document.f1.t1.focus();
                return false;
        }

        return true;
}
        </script>
</head>
<body>
<div id="wrapper">
  <div id="header">
    <div id="logo">
```

```
    <h1><font color="orange" size=6>Preventing Social Media Phishing in social networks</font></h1>
        <marquee><font color="pink" size=4>Social Media Phishing Profiles Detection</font></marquee>
   </div>
  </div>
 </div>
 <div id="menu">
   <ul>
    <li><a href="{% url 'index' %}">Home</a></li>
    <li><a href="{% url 'Admin' %}">Admin</a></li>
    <li><a href="{% url 'User' %}">Users</a></li>
     </ul>
 </div>
 <div class="entry">
  <br/><br/><br/>
                <br>
       <center><font style="Times New Roman" size="9" color="white">User Account Check Screen</font></center><br><br><br>
                <center><p><font size="4" color="white"><b>Enter data in form of numbers.</b><br>
                <font size="3" color="white">Account_Age, Gender<b>(Male - 1/ Female -0)</b>, User_Age, Link_Desc, Status_Count, Friend_Count, Location, Location_IP </b> </p></center>
```

```
<font size="4" color="white"><center>{{ data }}</center></font>
<form name="f1" method="post" action={% url 'UserCheck' %} OnSubmit="return validate()">
{% csrf_token %}
<br><br>
<TABLE align=center width="35%" class="notepad">
<TR><TH align="left"><font size="3" color="white">Account Details
<TD>  <textarea name="t1" rows="5" cols="80"></textarea>
<div id='nameid'></div>
</TD>
</TR>
<TR>
<TD></TD>
<TD>
<input type="submit" value="Submit">
</TABLE>
</form>
</div>
</body>
</html>
```

6.3.3 AdminLogin.html :

```
{% load static %}
<html>
<head>
<title>Preventing Social Media Phishing in social networks </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'default.css' %}" rel="stylesheet" type="text/css"
media="screen" />
 <script LANGUAGE="Javascript" >
function validate(){
        var x=document.forms["f1"]["username"].value;
        var y=document.forms["f1"]["password"].value;
        if(x == null || x==""){
                window.alert("Username must be enter");
                document.f1.t1.focus();
                return false;
        }
        if(y == null || y==""){
                window.alert("Password must be enter");
                document.f1.t2.focus();
                return false;
        }
        return true;
```

```
}
</script>
</head>
<body>
<div id="wrapper">
  <div id="header">
    <div id="logo">
      <h1><center><font color="orange" size=6>Preventing Social Media Phishing
in social networks </font></center></h1>
          <marquee><font color="pink" size=4>Social Media Phishing Profiles
Detection</font></marquee>
    </div>
      </div>
  </div>
  <div id="menu">
     <ul>
      <li><a href="{% url 'index' %}">Home</a></li>
      <li><a href="{% url 'Admin' %}">Admin</a></li>
      <li><a href="{% url 'User' %}">Users</a></li>
        </ul>
  </div>
  <div class="entry">
   <br/><br/><br/>
        <font size="" color="white"><center>{{ data }}</center></font>
```

```
<br>
<center><font size="5" color="white">Admin Login Screen</font></center>
<form name="f1" method="post" action={% url 'AdminLogin' %}
OnSubmit="return validate()">
{% csrf_token %}
<br><br>
<TABLE align=center width="35%" class="notepad">
<TR><TH align="left">Username
<TD>  <Input type=text name="username" value="
class="form-control">
<div id='nameid'></div>
</TD>
</TR>
<TR></TR>
<TR></TR>
<TR></TR>
<TR><TH align="left">Password
<TD>  <Input type='password' name="password"
value=" class="form-control">
</TR>
<TR>
<TD></TD>
<TD>
<br><br>
```

```
                    <input type="submit" value="login">
                    </TABLE>
</form>
</div>
</body>
</html>
```

6.3.4 Home.html :

```
{% load static %}
<html>
<head>
<title>Preventing Social Media Phishing in social networks</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'default.css' %}" rel="stylesheet" type="text/css"
media="screen" />
</head>
<body>
<div id="wrapper">
  <div id="header">
    <div id="logo">
      <h1><font color="orange" size=6>Preventing Social Media Phishing in social
networks</font></h1>
```

```
        <marquee><font color="pink" size=4>Social Media Phishing Profiles
Detection</font></marquee>
    </div>
    </div>
  </div>
  <div id="menu">
    <ul>
      <li><a href="{% url 'index' %}">Home</a></li>
      <li><a href="{% url 'Admin' %}">Admin</a></li>
      <li><a href="{% url 'User' %}">Users</a></li>
   </ul>
  </div>
  <div id="page">
   <div id="content">
     <div class="post">
       <div class="title">
         <h2>Preventing Social Media Phishing in social networks</h2>
        </div>
       <div class="entry">
                  <br/><br/><br/>
{{ data }}
         <br>
         <p><img src="{% static 'images\images.jpg' %}" alt="" width="890"
height="200" class="left" /><p></p><font size="4" color="white">
```

```
<center><br/>Social Media Phishing Profiles Detection</center></font></p>
<p><font size="" color="white"><b>About this project</b></p>

<p><font size="" color="white">This project will help the user to find whether a social media account is phishing account or not.</p>
</div>
</div>
</div>
</div>
</body>
</html>
```

6.3.5 Python Code :

6.3.5.1 Importing Libraries :

```
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponse
import pandas as pd
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers.core import Dense,Activation,Dropout
```

```
from keras.callbacks import EarlyStopping
from sklearn.preprocessing import OneHotEncoder
from keras.optimizers import Adam
```

6.3.5.2 Connection Modules :

```
def index(request):
    if request.method == 'GET':
        return render(request, 'Home.html', {})
def User(request):
    if request.method == 'GET':
        return render(request, 'UserPannel.html', {})
def Admin(request):
    if request.method == 'GET':
        return render(request, 'AdminLogin.html', {})
def AdminLogin(request):
    if request.method == 'POST':
        username = request.POST.get('username', False)
        password = request.POST.get('password', False)
        if username == 'admin' and password == 'admin':
            context= {'data':'welcome '+username}
            return render(request, 'AdminPannel.html', context)
        else:
            context= {'data':'login failed'}
```

return render(request, 'AdminLogin.html', context)

6.3.5.3 Predicting :

```
def UserCheck(request):
 if request.method == 'POST':
   data = request.POST.get('t1', False)
   input =
'Account_Age,Gender,User_Age,Link_Desc,Status_Count,Friend_Count,Location,Loc
ation_IP\n';
   input+=data+"\n"
   f = open("C:/FakeProfile/Profile/dataset/test.txt", "w")
   f.write(input)
   f.close()
   test = pd.read_csv('C:/FakeProfile/Profile/dataset/test.txt')
   test = test.values[:, 0:8]
   predict = model.predict_classes(test)
   print(predict[0])
   msg = ''
   if str(predict[0]) == '1':
     msg = "Given Account Details Predicted As Genuine"
   if str(predict[0]) == '0':
     msg = "Given Account Details Predicted As Phishing Account"
   context= {'data':msg}
```

```
    return render(request, 'UserPannel.html', context)


def GenerateModel(request):
    global model
    data = importdata()
    train_x, test_x, train_y, test_y = splitdataset(data)
    model = Sequential()
    model.add(Dense(200, input_shape=(8,), activation='relu', name='fc1'))
    model.add(Dense(200, activation='relu', name='fc2'))
    model.add(Dense(2, activation='softmax', name='output'))
    optimizer = Adam(lr=0.001)
    model.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
    print('CNN Neural Network Model Summary: ')
    print(model.summary())
    model.fit(train_x, train_y, verbose=2, batch_size=5, epochs=200)
    results = model.evaluate(test_x, test_y)
    ann_acc = results[1] * 100
    context= {'data':'ANN Accuracy : '+str(ann_acc)}
    return render(request, 'AdminPannel.html', context)
```

CHAPTER 7

SYSTEM TESTING

7. System Testing

7.1 Testing Methods :

7.1.1 Software Testing :

Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncover any defects/bugs that exist in the application. It assures the quality of the software. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc. Among these performance testing and load testing are the most important one for an android application and next sections deal with some of these types.

7.1.2 Blackbox Testing :

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

7.1.3 Whitebox Testing :

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement this is called Whitebox testing.

7.1.4 Performance Testing :

Performance testing is executed to determine how fast a system or sub-system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system such as scalability, reliability and resource usage.

7.1.5 Load Testing :

Load testing is primarily concerned with testing that can continue to operate under specific load, whether that is large quantities of data or a large number of users.

7.1.6 Manual Testing :

Manual Testing is the process of manually testing software for defects. Functionality of this application is manually tested to ensure the correctness.

7.2 Test cases :

Test Case 1	
Test Case Name	Empty login fields testing
Description	In the login screen if the username and password fields are empty
Output	Login fails showing the same page, asking to enter username and password.

Table 7.2.1 Test Case for Empty Login Fields

Test Case 2	
Test Case Name	Wrong login fields testing
Description	A unique username and password are set by administrator. On entering wrong username or password gives.
Output	Login fails showing the same page, and displays an error message username or password incorrect.

Table 7.2.2 Test Case for Wrong Login Fields

Test Case 3	
Test Case Name	Model generation.
Description	Admin login to the page and will generate a model.
Output	A message will display on the screen that the **"**model was generated successfully**"**.

Table 7.2.3 Test Case for Generating model

Test Case 4	
Test Case Name	Table View.
Description	Admin login to the page and will requests to show the trained datasets.
Output	A table will display on the screen that contains the information of the data that is used to train the model.

Table 7.2.4 Test Case for Table View

Test Case 5	
Test Case Name	User Data.
Description	User will enter the data to check whether a account is phishing account or not.
Output	A message will display on the user panel whether an account is phishing account or genuine account.

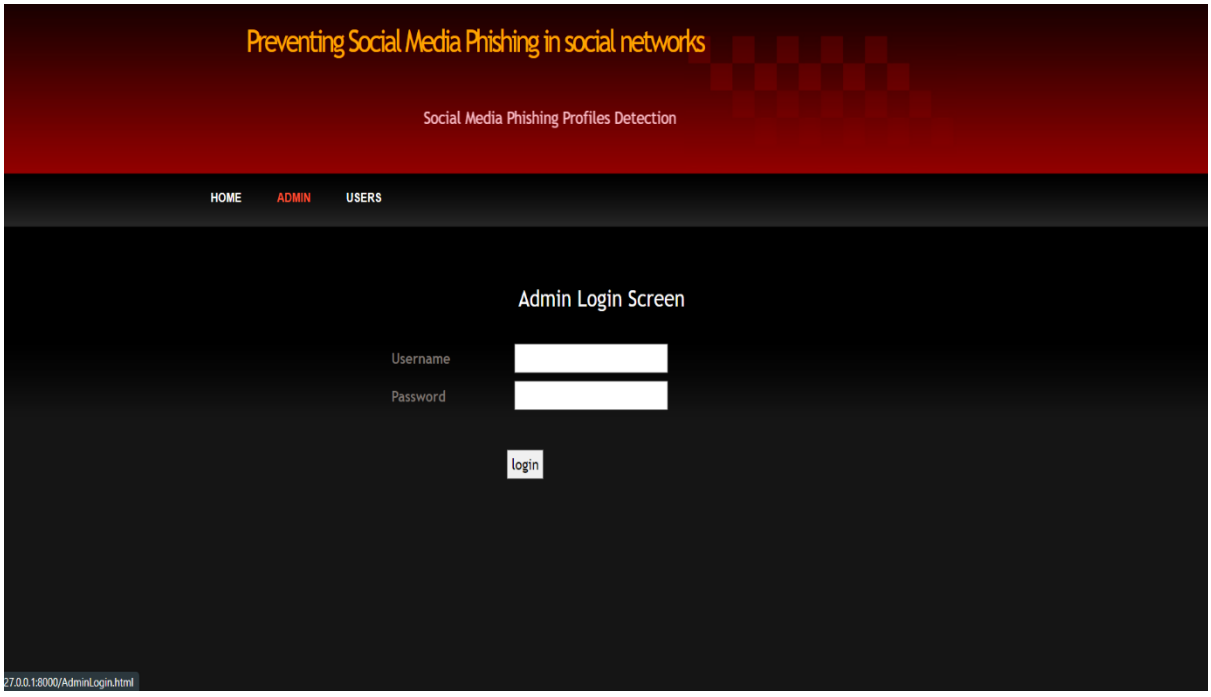
Table 7.2.5 Test Case for User Data

CHAPTER 8
EXPERIMENTAL RESULTS

8. Experimental Results

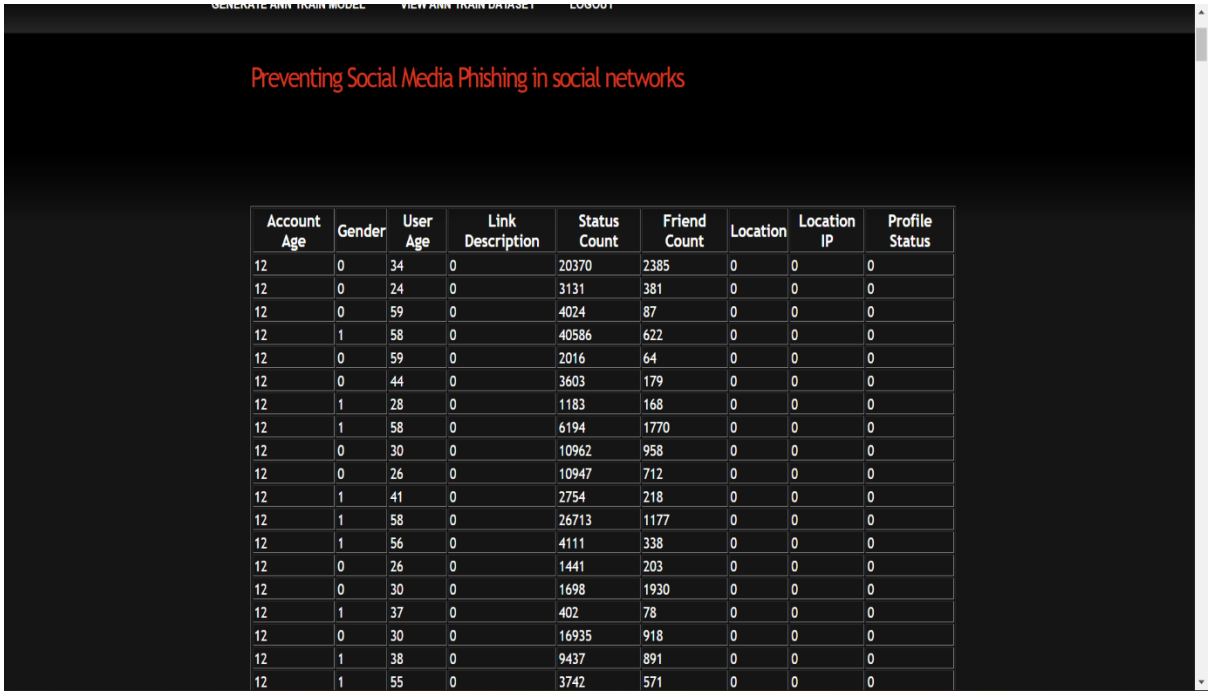
1.1 Home page :

1.2 Admin Login page :



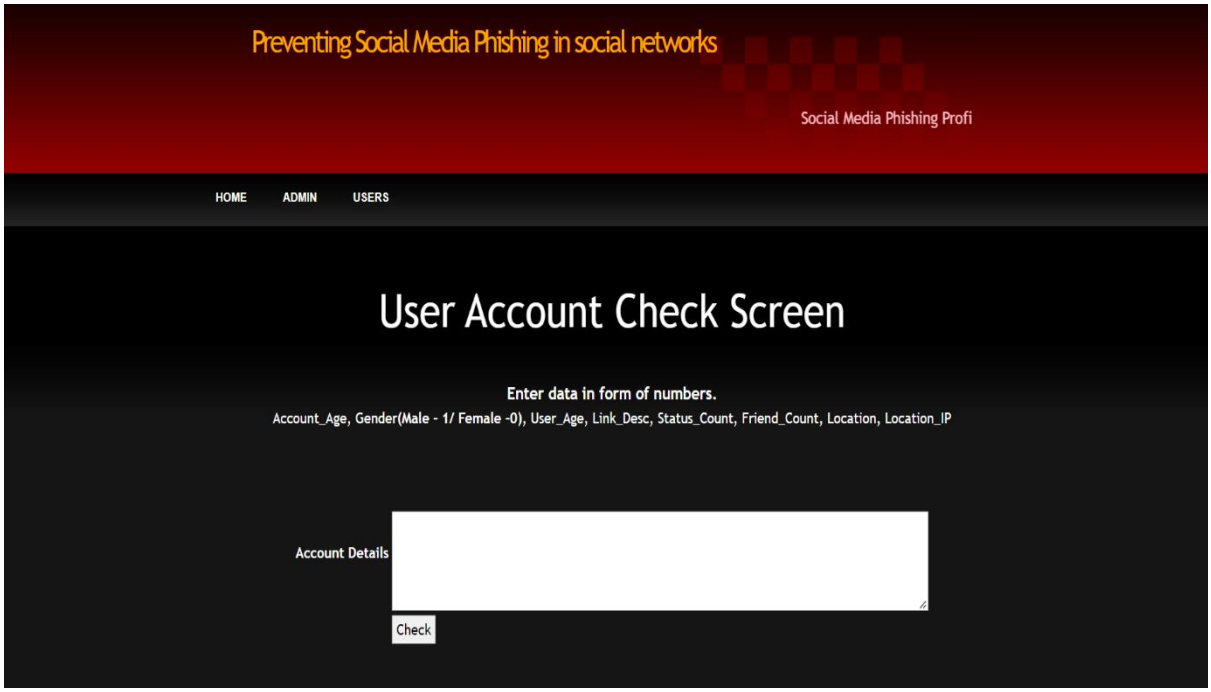
1.3 Model Generation page :

1.4 View Table page :

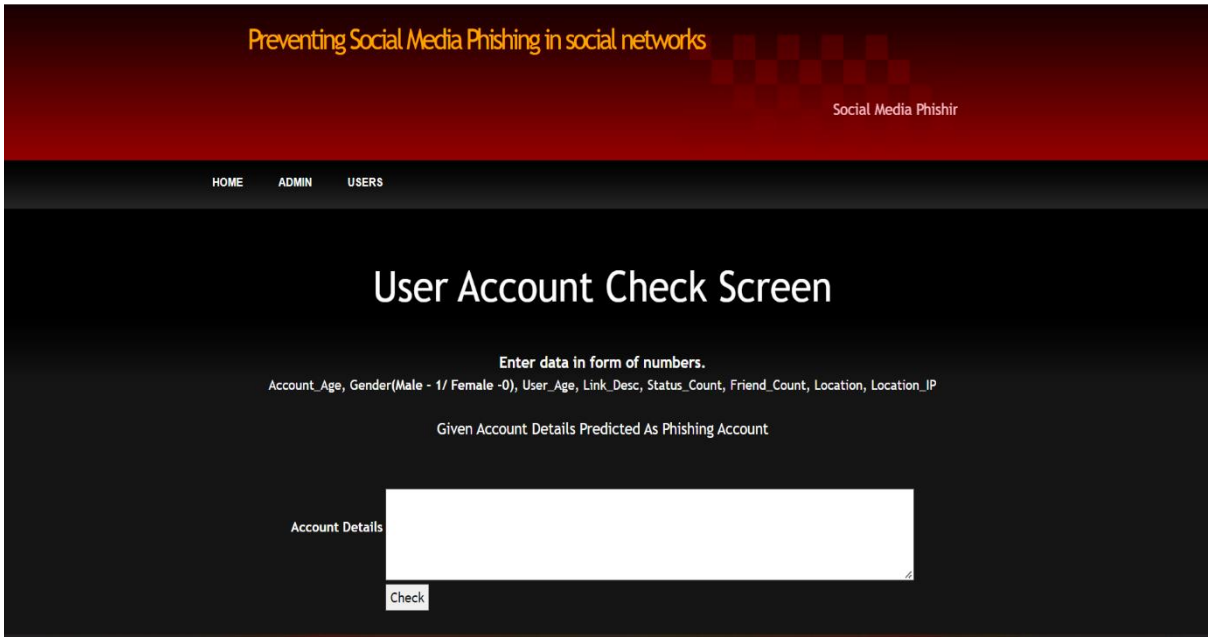


Account Age	Gender	User Age	Link Description	Status Count	Friend Count	Location	Location IP	Profile Status
12	0	34	0	20370	2385	0	0	0
12	0	24	0	3131	381	0	0	0
12	0	59	0	4024	87	0	0	0
12	1	58	0	40586	622	0	0	0
12	0	59	0	2016	64	0	0	0
12	0	44	0	3603	179	0	0	0
12	1	28	0	1183	168	0	0	0
12	1	58	0	6194	1770	0	0	0
12	0	30	0	10962	958	0	0	0
12	0	26	0	10947	712	0	0	0
12	1	41	0	2754	218	0	0	0
12	1	58	0	26713	1177	0	0	0
12	1	56	0	4111	338	0	0	0
12	0	26	0	1441	203	0	0	0
12	0	30	0	1698	1930	0	0	0
12	1	37	0	402	78	0	0	0
12	0	30	0	16935	918	0	0	0
12	1	38	0	9437	891	0	0	0
12	1	55	0	3742	571	0	0	0

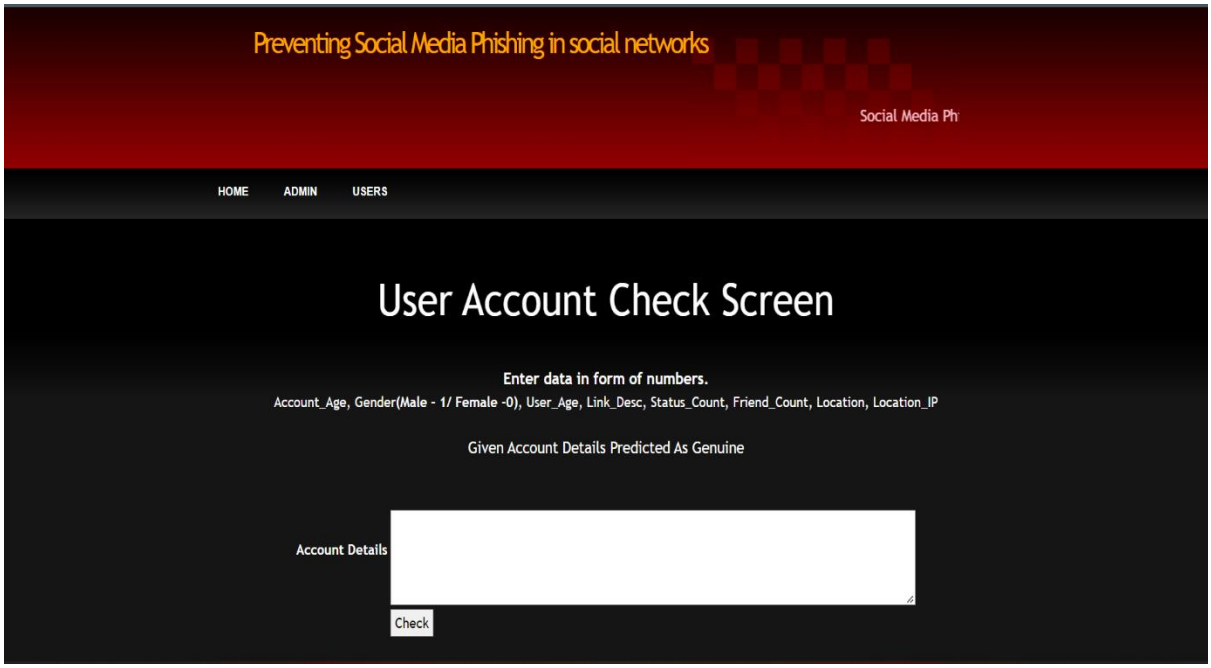
1.5 User Panel page :



1.5.1 User Output 1 :



1.5.2 User Output 2 :



CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

2. Conclusion & Future scope

2.1 Conculsion :

To evaluate the likelihood that a friend request is genuine or not, we employ machine learning, specifically an artificial neural network. At every neuron (node), a Sigmoid function is applied to every equation. We employ a training set of data from Facebook or other social networks. This makes it possible for the deep learning algorithm that is being used to learn patterns of bot behaviour to do so by using back propagation, minimising the overall cost function, and modifying the weight and bias of each neuron.

2.2 Future scope :

Each input neuron would be a unique, previously selected characteristic of each profile that was converted into a numerical value (for example, gender as a binary number, female 0 and male 1) and, if necessary, divided by an arbitrary number (for example, age is typically divided by 100) to reduce one characteristic having more influence on the result than the other. The neurons stand in for nodes. There would be one decision-making procedure assigned to each node.

CHAPTER 10
Bibliography

10.Bibliography

10.1 References :

[1] https://www.statista.com/topics/1164/social-networks/

[2] https://www.cnbc.com/2018/01/31/facebook-earnings-q4-2017-arpu.html

[3] https://www.cnet.com/news/facebook-breach-affected-50-millionpeople/

[4] https://www.facebook.com/policy.php

[5] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI'12). USENIX Association, Berkeley, CA, USA, 15-15.

[6] Akshay J. Sarode and Arun Mishra. 2015. Audit and Analysis of Impostors: An experimental approach to detect fake profile in an online social network. In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015 (ICCCT '15). ACM, New York, NY, USA, 1-8. DOI: https://doi.org/10.1145/2818567.2818568

[7] Devakunchari Ramalingam, Valliyammai Chinnaiah. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering, Volume 65, 2018, Pages 165-177, ISSN 0045-7906,

https://doi.org/10.1016/j.compeleceng.2017.05.020.

[8] https://www.enigmasoftware.com/top-20-countries-the-most-cybercrime

[9] pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

[10] **For code snipets :** https://stackoverflow.com/questions/40758562/can-anyone-explain-mestandardscaler

[11] https://pandas.pydata.org

[12] https://www.tutorialspoint.com/python_pandas/index.htm

[13] http://www.numpy.org

[14] https://www.mathworks.com/products/matlab.html

[15] http://www.deeplearning.net/software/theano/

[16] https://scikit-learn.org/stable/

[17] https://keras.io

[18] https://www.tensorflow.org

10.2 Book References :

[1] C#.NET2005 black book.

[2] Visual C#.NET book by Jason price and Mike Gunderloy

[3] Software Engineering Principles by Roger.S. Pressman

[4] Object oriented s/w engineering By Tata MCGRAW HILL

[5] Network Security by William Stallings.