A Project report on

FAKE ACCOUNT DETECTION ON SOCIAL MEDIA USING ARTIFICIAL NEURAL NETWORK

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

M. MOUNIKA
19H51A05N8

MOIN ASHIQ
19H51A05P0

N. HARIKA RATNA
19H51A05P4

Under the esteemed guidance of

Mrs. M. KAMALA
Assistant Professor

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase-II report entitled **"Fake Account Detection on Social Media using ANN"** being submitted by Madham Mounika (19H51A05N8), Moin Ashiq (19H51A05P0), N. Harika Ratna (19H51A05P4) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out her under my guidance and supervision.

The results embody in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mrs.M.Kamla	**Dr. Siva Skandha Sanagala**
Asst. Professor	**Associate Professor and HOD**
Dept. of CSE	**Dept. of CSE**

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs. M. Kamala** Assistant Professor, Department of Computer Science and Engineering for her valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

M.MOUNIKA	**(19H51A05N8)**
MOIN ASHIQ	**(19H51A05P0)**
N. HARIKARATNA	**(19H51A05P4)**

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	iii
	LIST OF TABLES	v
	ABSTRACT	vi
1	**INTRODUCTION**	1
	1.1 Problem Statement	2
	1.2 Research Objective	3
	1.3 Project Scope and Limitations	5
2	**BACKGROUND WORK**	6
	2.1. Fake Account Detection Using SVM	7
	2.1.1. Introduction	7
	2.1.2. Merits, Demerits and Challenges	6
	2.1.3. Implementation of Fake Account Detection Using SVM	8
	2.2. Fake Account Detection Using DNN	11
	2.2.1. Introduction	11
	2.2.2. Merits, Demerits and Challenges	11
	2.2.3. Implementation of Fake Account Detection Using DNN	12
	2.3. Fake Account Detection Using Random Forest	15
	2.3.1. Introduction	15
	2.3.2. Merits, Demerits and Challenges	15
	2.3.3. Implementation of Fake Account Detection Using Random Forest	16
3	**PROPOSED SYSTEM**	**18**
	3.1. Objective of Proposed Model	19
	3.2. Algorithms Used for Proposed Model	21
	3.3. Designing	25
	3.3.1.UML Diagram	25

	3.3. Stepwise Implementation and Code	29
4	**RESULTS AND DISCUSSION**	**41**
	4.1. Performance metrics	42
5	**CONCLUSION**	**45**
	5.1 Conclusion and Future Enhancement	46
6	**REFERENCES**	**47**
	GitHub Link	**51**

List of Figures

FIGURE NO.	TITLE	PAGE NO.
1.1	Fake Accounts	02
1.2	Graph Showing Increase fake accounts	03
1.3	Node Links	05
2.1	System Architecture	07
2.2	Workflow of proposed methodology	08
2.3	Use Case Diagram	09
2.4	Activity Diagram	08
2.5	Fake Account Detection Using SVM	08
2.6	Neural Network Layers	09
2.7	UML Diagram	10
2.8	Fake account detection using DNN	10
2.9	Framework of fake profiles	11
2.10	Flow Diagrams	12
2.11	Fake account detection using Random Forest	12
3.1	ANN algorithms neuron	21
3.2	ANN Layers	21
3.3	Synapse	21
3.4	Neuron	22
3.5	System Design	24
3.6	Use Case Diagrams	26
3.7	Activity Diagram	27
3.8	Class Diagram	27

3.9	Collaboration Diagram	28
3.10	Deployment Diagram	28
3.11	Sequence Diagram	28
3.12	Dashboard	30
3.13	Datasets	31
3.14	Uploaded train dataset	31
3.15	Preprocess Dataset	32
3.16	Epoch Iterations	32
3.17	Accuracy of test dataset	33
3.18	Accuracy vs Loss Graph	33
3.19	Test Dataset	34
3.20	Results of Test dataset	34
4.1	Graph of Comparison	42
4.2	Metrics of Algorithms	42
4.3	ROC Graphs	43
4.4	Metrics of ANN	43

List of Tables

TABLE NO.	**TITLE**	**PAGE NO.**
2.1	Attributes used in proposed work	09
2.2	Attributes used in proposed work	16
3.1	Train Data Set	19
3.2	Test Data Set	19
4.1	Train Data Set	44
4.2	Test Dataset	44

ABSTRACT

Nowadays, online social media is dominating the world in several ways. Day by day the number of users using social media is increasing drastically. The main advantage of online social media is that we can connect to people easily and communicate with them in a better way. This provided a new way of a potential attack, such as fake identity, false information, etc. A recent survey suggests that the number of accounts present in the social media is much greater than the users using it. This suggests that fake accounts have been increased in the recent years. Online social media providers face difficulty in identifying these fake accounts. The need for identifying these fake accounts is that social media is flooded with false information, advertisements, etc.

Traditional methods cannot distinguish between real and fake accounts efficiently. Improvement in fake account creation made the previous works outdated. The new models created used different approaches such as automatic posts or comments, spreading false information or spam with advertisements to identify fake accounts. Due to the increase in the creation of the fake accounts different algorithms with different attributes are use. Previously use algorithms like naïve bayes, support vector machine, random forest has become inefficient in finding the fake accounts. In this research, we came up with an innovative method to identify fake accounts. We used Artificial Neural Network algorithm; bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. We combined Machine learning and Data Science to accurately predict fake accounts

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

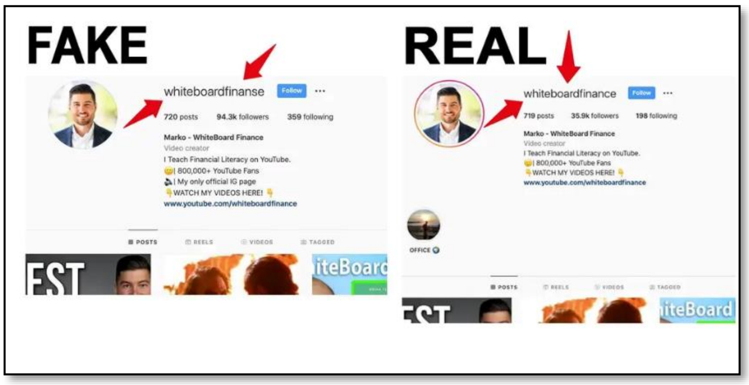
1.1 Problem Statement:

To detect and identify fake accounts from social media websites using machine learning algorithms.

Social media is an essential part of everyone's life in today's modern world. 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. The fake profile's contents typically have links that lead to an external website where the damage happens. An unaware curious user clicking the bad link will damage their computer. The cost can be as simple as catching a virus to as bad as installing a rootkit turning the computer into a zombie. While Facebook has a rigorous screening to keep these fake accounts out, it only takes one fake profile to damage the computers of many.

Why Fake account detection is important?

- ➢ Gives more security for our profiles.
- ➢ Reduces creation of false news feeds & accounts.
- ➢ Decreasing of abuse, payment fraud and identity of theft.



Marko - WhiteBoard Finance
Video creator
I Teach Financial Literacy on YouTube.
| 800,000+ YouTube Fans
| My only official IG page
WATCH MY VIDEOS HERE!
www.youtube.com/whiteboardfinance

Marko - WhiteBoard Finance
Video creator
I Teach Financial Literacy on YouTube.
| 800,000+ YouTube Fans
WATCH MY VIDEOS HERE!
www.youtube.com/whiteboardfinance

Fig 1.1 Fake Accounts

1.2 Research Objective

In today's Modern society, social media plays a vital role in everyone's life. The general purpose of social media is to keep in touch with friends, sharing news, etc. The number of users in social media is increasing exponentially. With more than 1 billion active users, Instagram has become one of the most used social media sites. A recent survey suggests that the number of accounts present in the social media is much greater than the users using it. This provided a new way of a potential attack, such as fake identity, false information, etc. These social media influencers have now become a go-to place for the business organization to advertise their products and services.

In 2017 Facebook reached a total population of 2.46 billion users making it the most popular choice of social media. Social media networks make revenues from the data provided by users. The average user does not know that their rights are given up the moment they use the social media network's service. Social media companies have a lot to gain at the expense of the user. Every time a user shares a new location, new photos, likes, dislikes, and tag other users in content posted, Facebook makes revenue via advertisements and data. More specifically, the average American user generates about $26.76 per quarter. That number adds up quickly when millions of users are involved. In today's digital age, the ever-increasing dependency on computer technology has left the average citizen vulnerable to crimes such as data breaches and possible identity theft. These attacks can occur without notice and often without notification to the victims of a data breach. At this time, there is little incentive for social networks to improve their data security. These breaches often target social media networks such as Facebook and Twitter. They can also target banks and other financial institutions.

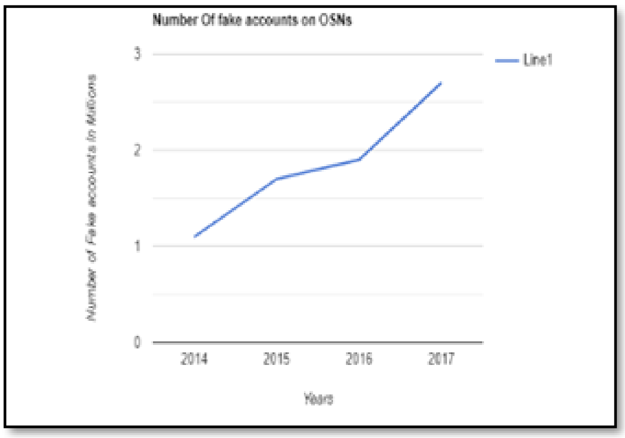


Fig 1.2 Graph showing increase number of fake accounts

There seems to be a newsworthy issue involving social media networks getting hacked every day. Recently, Facebook had a data breach which affected about 50 million users [3]. Facebook provides a set of clearly defined provisions that explain what they do with the user's data [4]. The policy does very little to prevent the constant exploitation of security and privacy. Fake profiles seem to slip through Facebook's built-in security features. The other dangers of personal data being obtained for fraudulent purposes is the presence of bots and fake profiles. Bots are programs that can gather information about the user without the user even knowing. This process is known as web scraping. What is worse, is that this action is legal. Bots can be hidden or come in the form of a fake friend request on a social network site to gain access to private information. The solution presented in this paper intends to focus on the dangers of a bot in the form of a fake profile on your social media. This solution would come in the form of an algorithm. The language that we chose to use is Python. The algorithm would be able to determine if a current friend request that a user gets online is an actual person or if it is a bot or it is a fake friend request fishing for information. Our algorithm would work with the help of the social media companies, as we would need a training dataset from them to train our model and later verify if the profiles are fake or not. The algorithm could even work as a traditional layer on the user's web browser as a browser plug-in.

The increasing popularity of Facebook or Twitter or Twitter from the year 2006 to 2016 allows the users to add friends and share various kinds of information such as personal, social, economic, educational, political, business, etc. Moreover, they can also share photos, videos, and other day-to-day interaction. However, some people don't use these sites with a good objective. Therefore, they create fake accounts on social sites. Fake accounts do not have any real identity so we can call them Attacker. This attacker uses incorrect information or statistics about some real-world person to create a fake account. Using these fake accounts, attackers spread fake information which affects other users. To protect such sensitive data of users is one of the major challenges of social sites. There are several techniques in the field of machine learning that have been developed to detect fake accounts in social networking sites such as Neural Network (NN), Naive Bayes, Markov Model, and Bayesian Network. In recent researches, it has been found that these techniques make available enhanced results to detect fake accounts.

1.3 Project Scope and Objectives:

Project Scope:

In the present generation, the social life of everyone has become associated with the online social networks. Adding new friends and keeping in contact with them and their updates has become easier. Online social networks have impact on the science, education, grassroots organizing, employment, business, etc. Researchers have been studying these online social networks to see the impact they make on people. Teachers can reach the students easily through this making a friendly environment for the students to study, teachers nowadays are getting themselves familiar to these sites bringing online classroom pages, giving homework, making discussions, etc. which improves education a lot. The employers can use these social networking sites to employ the people who are talented and interested in the work, their background check can be done easily.

Objectives:

1. To reduce creation of fake accounts.

2. Save money, time, & details.

3.Give more security for their profiles

4. No abuse content is created

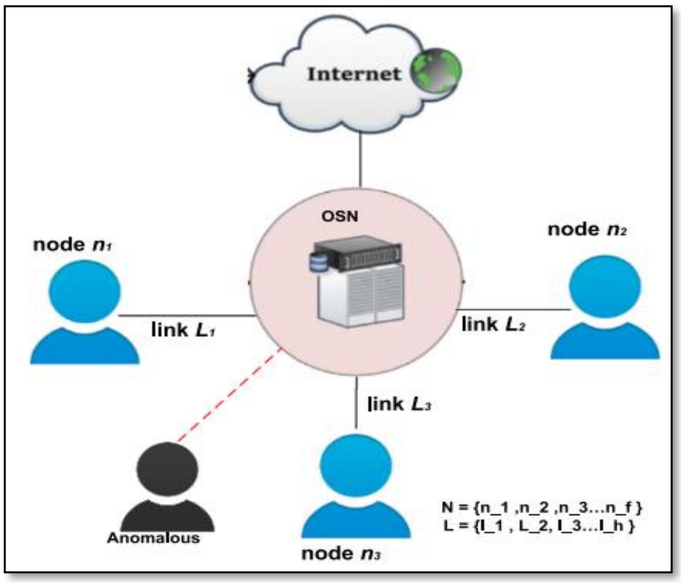


Fig 1.3 Node links

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. FAKE ACCOUNT DETECTION USING SVM:

2.1.1. Introduction

SVM (Support Vector Machine) is a binary classification algorithm that finds the maximum separation hyper plane between two classes. It is a supervised learning algorithm that gives enough training examples, divides two classes fairly well and classifies new examples. It offers a principle approach to machine learning problems because of their mathematical foundation in statistical learning theory. SVM construct their solution as a weighted sum of SVs, which are only a subset of the training input.

2.1.2. Merits, Demerits and Challenges

Merits:

i. New features and parameters are added.

ii. Perform better than existing ones.

Demerits:

i. Not accurate.

ii. Limited data to train.

iii. High variance problems on increasing dataset.

iv. Only few parameters are used.

Challenges:

i. Use more parameters.

ii. Make all problems satisfied.

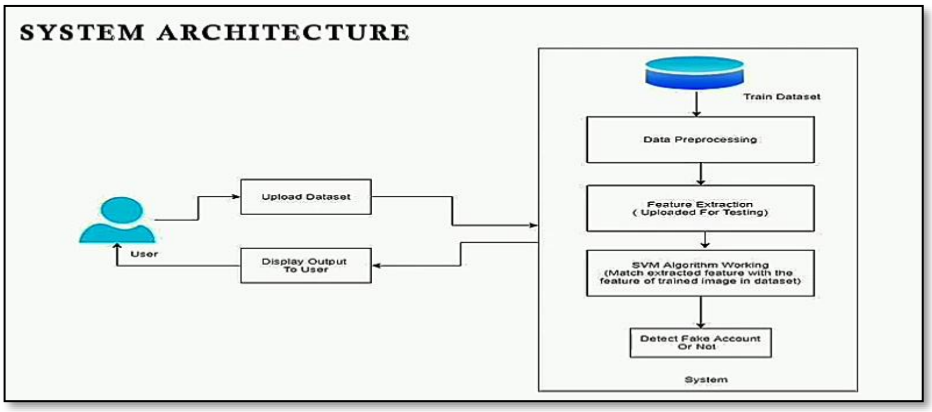

SYSTEM ARCHITECTURE
Train Dataset
Data Preprocessing
Upload Dataset
Feature Extraction (Uploaded For Testing)
User
Display Output To User
SVM Algorithm Working (Match extracted feature with the feature of trained image in dataset)
Detect Fake Account Or Not
System

Fig 2.1 System Architecture

2.1.3. Implementation of FAKE ACCOUNT DETECTION USING SVM

Fig 2.1 System Architecture

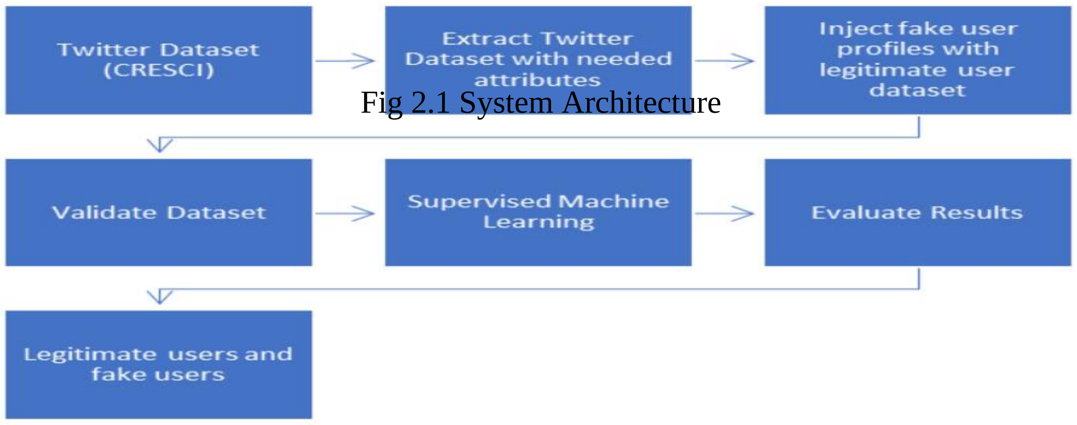

Twitter Dataset (CRESCI)
Extract Twitter Dataset with needed attributes
Inject fake user profiles with legitimate user dataset
Validate Dataset
Supervised Machine Learning
Evaluate Results
Legitimate users and fake users

Fig 2.2 Workflow of proposed methodology

Steps involved in Methodology to find the fake users.

Step 1: The twitter dataset used here was manually generated by ***Cresci et al***. A new dataset cannot be manually generated from any online social networks for identifying fake user on twitter.

Step 2: Form the dataset, the necessary attributes such as id_name, screen_name, date_of_joining, tweet frequency, followers_count (followers), friends_count (following), favourites_count (liked posts), language, geo_location(location), verified and tweetswere extracted.

Step 3: In the legitimate user dataset, the fake user dataset was injected in which the attributes are similar to the legitimate users but the values on the attributes are different from the legitimate users.

Step 4: The next step was validating the dataset, the data set with the same attributes from the real user and fake user dataset was extracted, since the R studio did not support characters or string it only accepts the numerical values, so the attributes which are having only the numerical values are extracted from the above dataset such as followers_count, friends_count, favorites, tweet, tweet_frequency, geo_enabled and verified.

Step 5: After validating the dataset, the supervised machine learning technique was used to classify the users based on SVM (Support Vector Machine) classification algorithm.

Step 6: Evaluating the results based on the given attributes and the condition, the fake users and legitimate users are classified separately.

Attribute	Description
FOLLOWERS_COUNT	No of people follows the account holder
FRIENDS_COUNT	Mutual users (Friends who follows back the account holder)
FAVURITES_COUNT	The number tweets liked/favorited by the user
TWEET	Total number of tweets made by a user
TWEET_FREQUENCY	The no of tweets by a user from the date of creation to till date
GEO_ENABLED	Whether the account holder provided their location
VERIFIED	Whether the account is verified or not

Table 2.1: Attributes used in proposed work

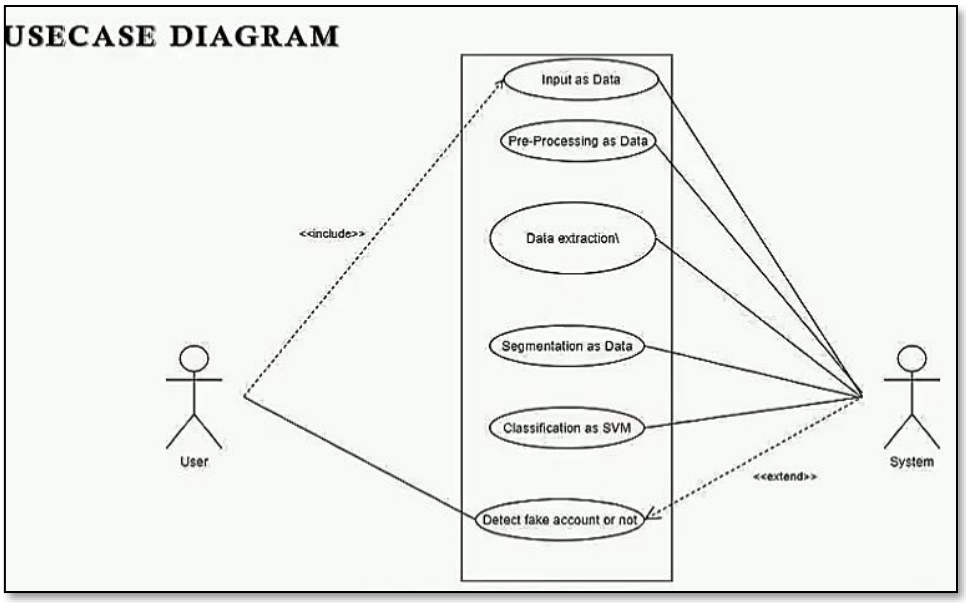


Fig 2.3 UML Diagrams

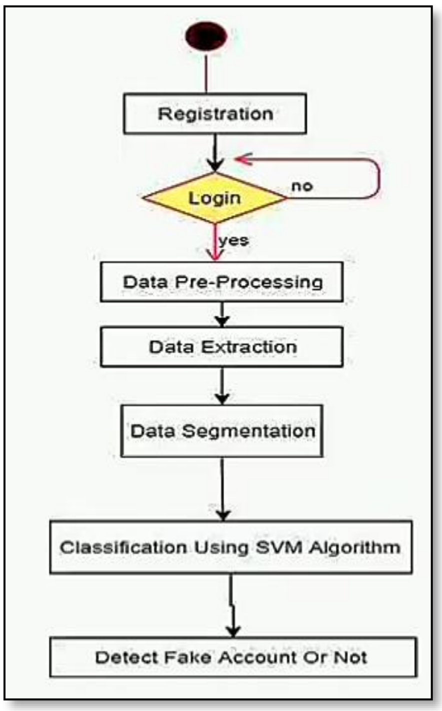
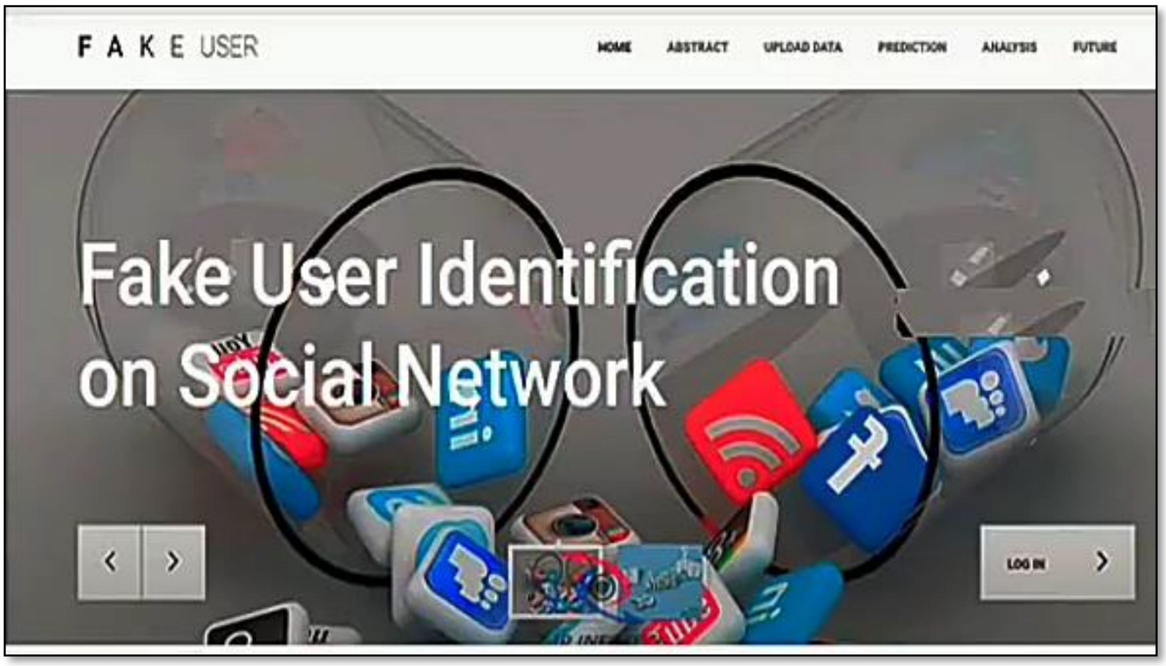


Fig 2.4 Activity Diagrams

CONCLUSION

In artificial intelligence, machine learning methods are applied in order to naturally improve and learn from the experience without being obviously programmed. This work provides the view about the fake user identification on twitter using classification algorithm. It helps to identify the fake users on twitter by using machine learning techniques. Fake users are increasing day by day, huge amount of private and personal data is shared using online social networks which are dangerous if disclosed to unknown people may cause potential risk. The collision of machine learning techniques had been used as effective mechanism to counter fake profiles.

Results:



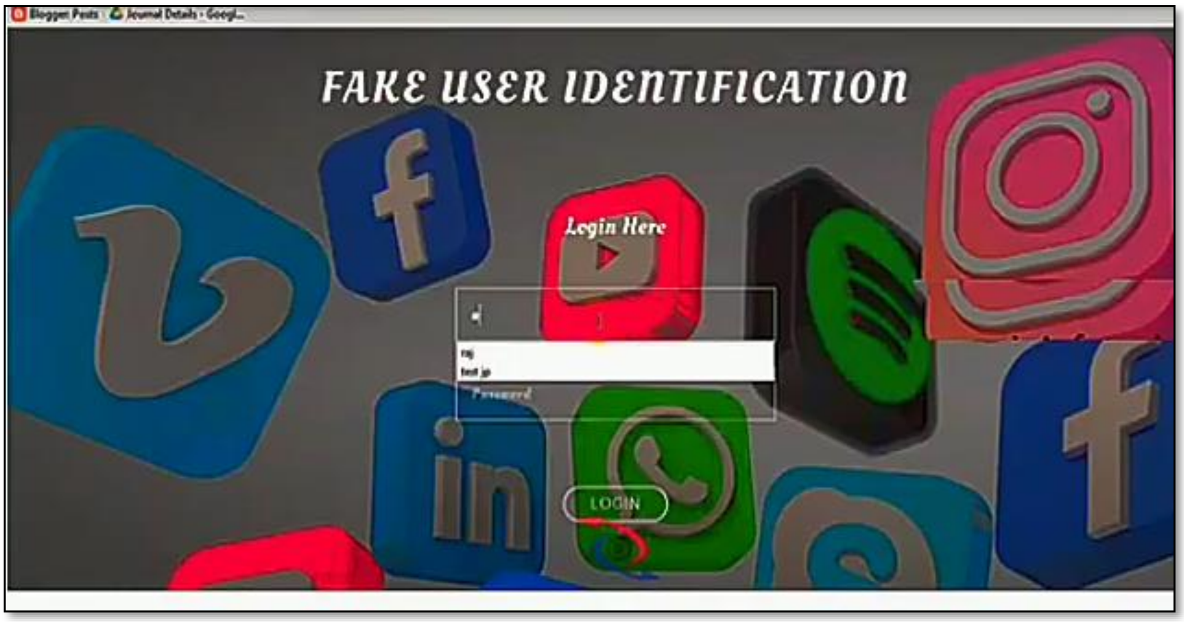


Fig 2.5 Fake Account Detection Using SVM

2.2. FAKE ACCOUNT DETECTION USING DNN:

2.2.1. Introduction

The research study by Simranjit Kaur et al is based on implementing a k-mean clustering algorithm on vector set to increase efficiency. To detect spam emails using neural networks the two phases namely training and testing are needed to be done. The process of detecting spam and phishing emails using feed forward neural network. The paper has 11 features have been implemented as a binary value 0 or 1 with value 1 indicating this feature appeared in the tested email and value 0 indicating non-appearance case.

2.2.2. Merits, Demerits and Challenges

Merits:

i. Making grouping increase accuracy.

ii. Detects fast.

iii. Implementation done by using binary values.

Demerits:

i. No spam detection method is used.

ii. Not robust.

iii. It only maps to small number of character classes.

iv. Not a language-insensitive pattern matching features.

Challenges:

i. Need more fastest programming.

ii. All the character classes should be involved.

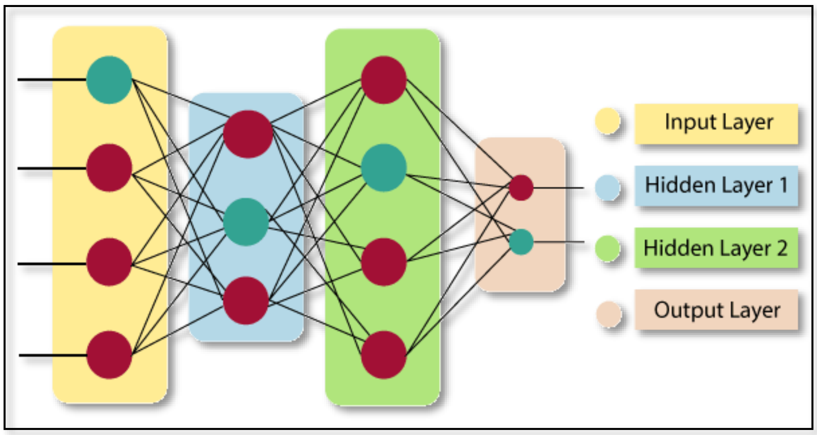


Fig 2.6 Neural Network Layers

2.2.3. Implementation of Fake account detection using DNN:

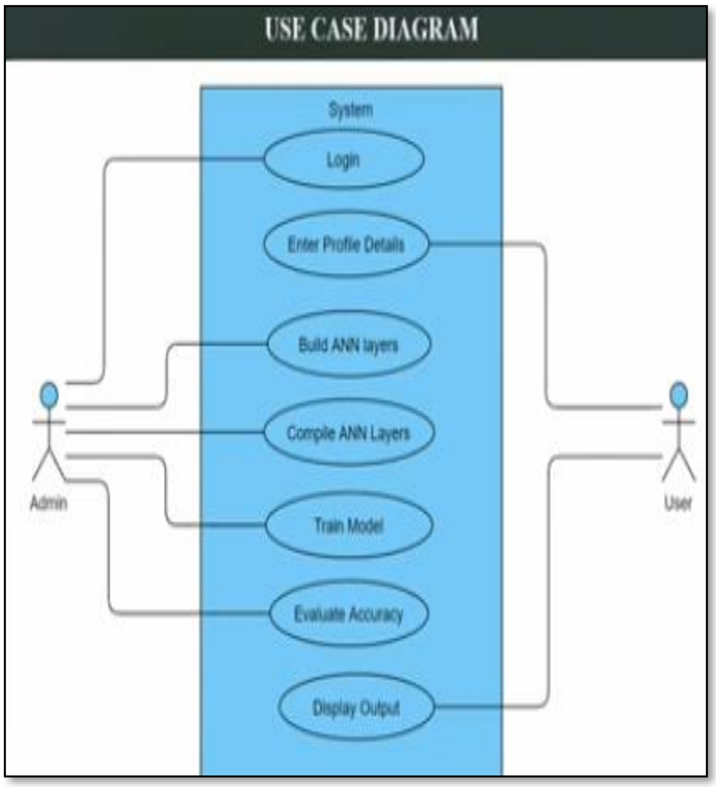
- **Input:** Social media account profile data
- **Output:** Binary label indicating whether the account is genuine or fake

- **Data Collection**: Gather a large labeled dataset of social media account profiles, consisting of both genuine and fake accounts. The dataset should be diverse, covering various types of fake accounts, and include relevant features such as profile picture, account age, posting frequency, follower-to-following ratio, and engagement rate.
- **Data Preprocessing**: Clean and normalize the dataset by removing irrelevant features, handling missing values, converting categorical variables to numerical representations, and normalizing numerical features.
- **Feature Extraction:** Extract relevant features from the social media account profiles that can serve as input to the DNN. This may involve converting text and image data into numerical representations using techniques such as word embeddings and image embeddings.
- **Model Training:** Split the preprocessed dataset into training, validation, and testing sets. Use the training set to train the DNN model. The DNN architecture can consist of multiple layers with various activation functions, such as ReLU, sigmoid, or tanh. Experiment with different architectures and hyperparameters to find the optimal model. Train the model using backpropagation and an appropriate optimization algorithm such as stochastic gradient descent (SGD).
- **Model Evaluation:** Evaluate the trained DNN model on the validation set to tune the hyperparameters and assess its performance. Use common evaluation metrics for binary classification tasks, such as accuracy, precision, recall, and F1 score, to measure the performance of the model.
- **Model Testing:** Test the trained DNN model on the testing set to evaluate its generalization performance. Calculate the evaluation metrics on the testing set to estimate the model's ability to detect fake accounts on unseen data.
- **Model Deployment:** Once the DNN model has been trained and tested, deploy it in a production environment to automatically detect fake accounts on social media platforms. This may involve integrating the model into the platform's backend system and regularly updating it with new data to ensure its effectiveness over time.

- **Monitoring and Maintenance:** Continuously monitor the performance of the deployed DNN model and update it as needed to maintain its accuracy and effectiveness in detecting fake accounts, as fake account creators may change their strategies over time.

It's important to note that the above algorithm is a high-level outline, and the specific implementation details may vary depending on the dataset, model architecture, and platform requirements. Fine-tuning and experimentation may be necessary to achieve the best performance for fake account detection using DNNs.

UML Diagrams



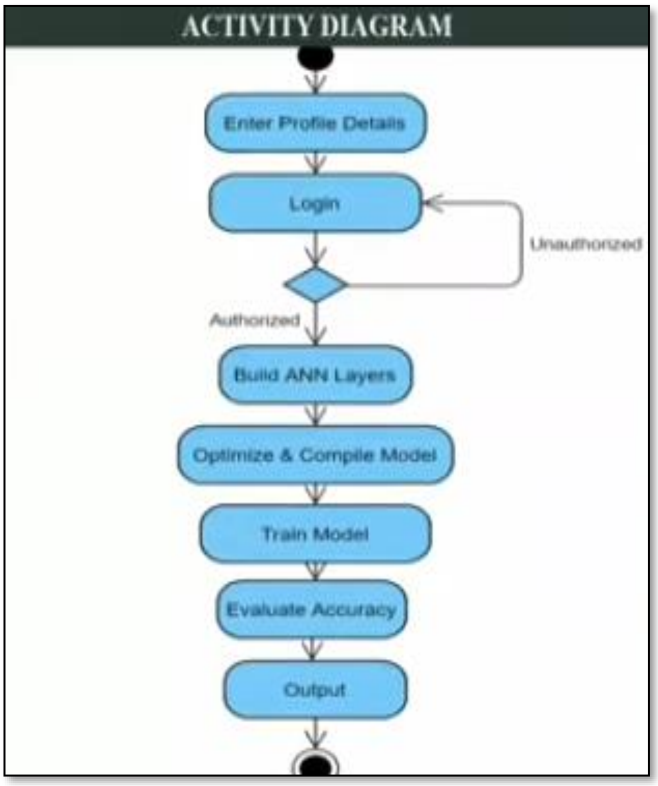
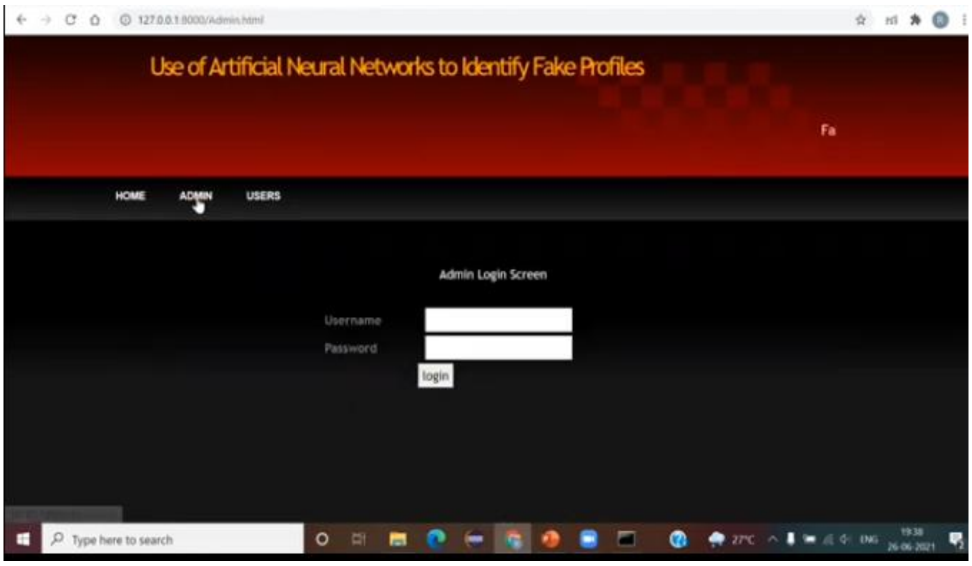
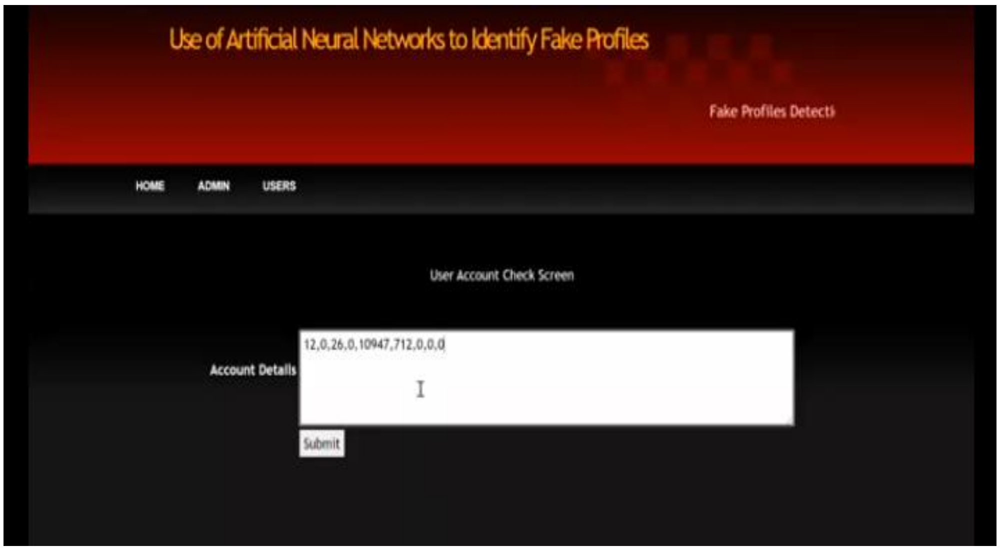


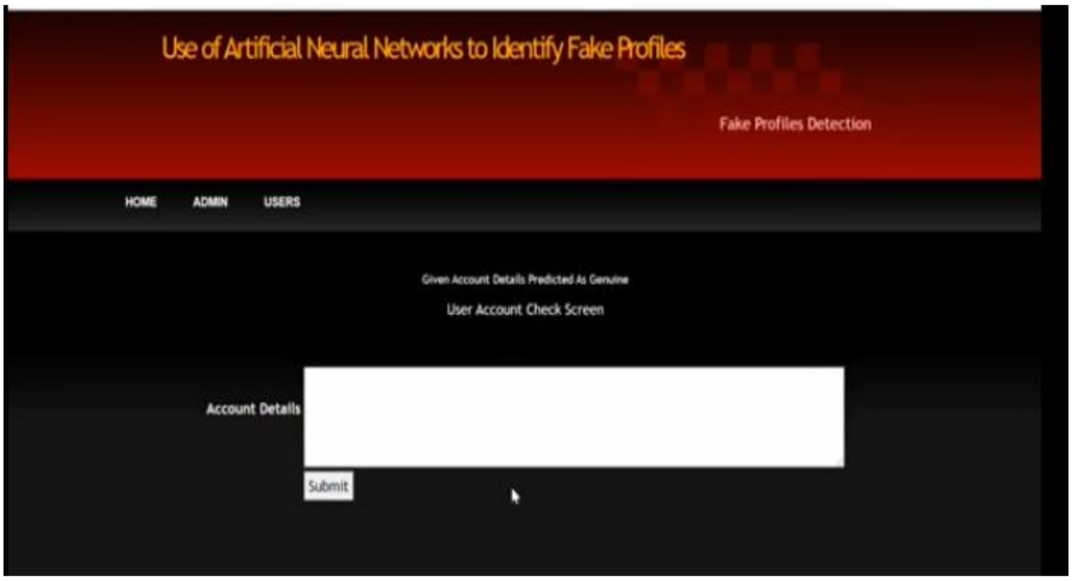
Fig. 2.7 UML Diagrams

Results:


127.0.0.1:8000/Admin.html
Use of Artificial Neural Networks to Identify Fake Profiles
HOME
ADMIN
USERS
Admin Login Screen
Username
Password
login


Use of Artificial Neural Networks to Identify Fake Profiles
Fake Profiles Detecti
HOME
ADMIN
USERS
User Account Check Screen
Account Details
Submit

12,0,26,0,10947,712,0,0,0


Use of Artificial Neural Networks to Identify Fake Profiles
Fake Profiles Detection
HOME
ADMIN
USERS
User Account Check Screen
Account Details
Submit

Given Account Details Predicted As Genuine

Fig 2.8 Fake account detection using DNN

2.3. FAKE ACCOUNT DETECTION USING RANDOM FOREST

2.3.1. Introduction

Random forest is a supervised learning algorithm that is used for both classifications as well as regression. Similarly, the random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method that is better than a single decision tree because it reducesthe over-fitting by averaging the result.

2.3.2. Merits, Demerits and Challenges

Merits:

i. More accurate.

ii. Creates many variations of trees.

iii. Give best outcomes than others.

Demerits:

iv. Not accurate than ours.

v. Uses only few features like comments & comportments.

Challenges:

vi. Use more features like complaints, etc.

vii. Can automatically remove the fake one.

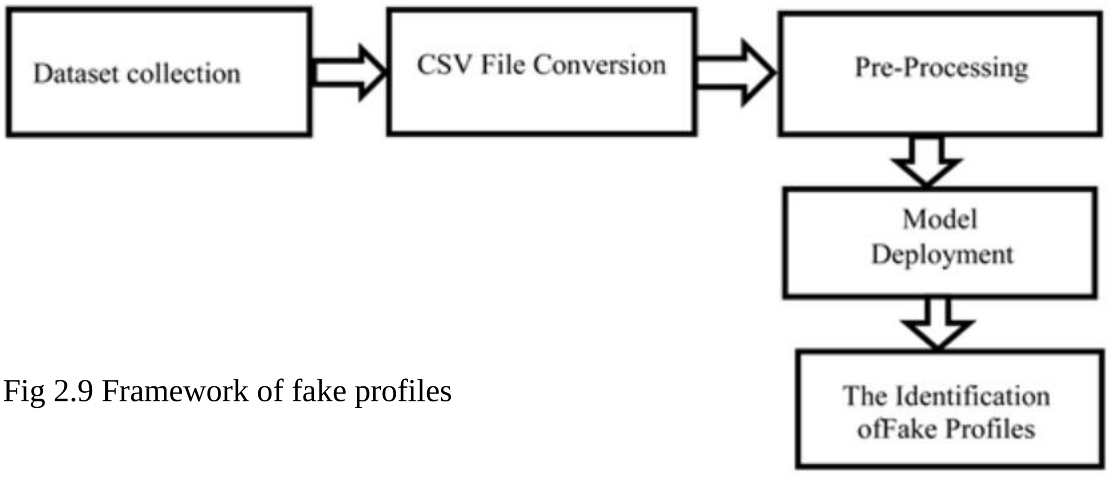


Fig 2.9 Framework of fake profiles

2.3.3. Implementation of Fake account detection using Random Forest

We can understand the working of the Random Forest algorithm with the help of following steps:

1. First, start with the selection of random samples from a given dataset.
2. Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
3. In this step, voting will be performed for every predicted result.
4. At last, select the most voted prediction result as the final prediction result.

S. No	Attribute	Description
1.	Profile ID	The Profile ID of account holder
2.	Profile Name	The name of the account holder
3.	Status Count	The number of tweets made by the account
4.	Followers Count	The number of followers for the account
5.	Friends Count	The number of friends for the account
6.	Location	The location of the account holder
7.	Created Date	The date the account was created
8.	Share count	The number of shares done by account holder
9.	Gender	The Gender of the account holder
10.	Language Code	The language of account holder

Table 2.2 Attributes used in proposed work

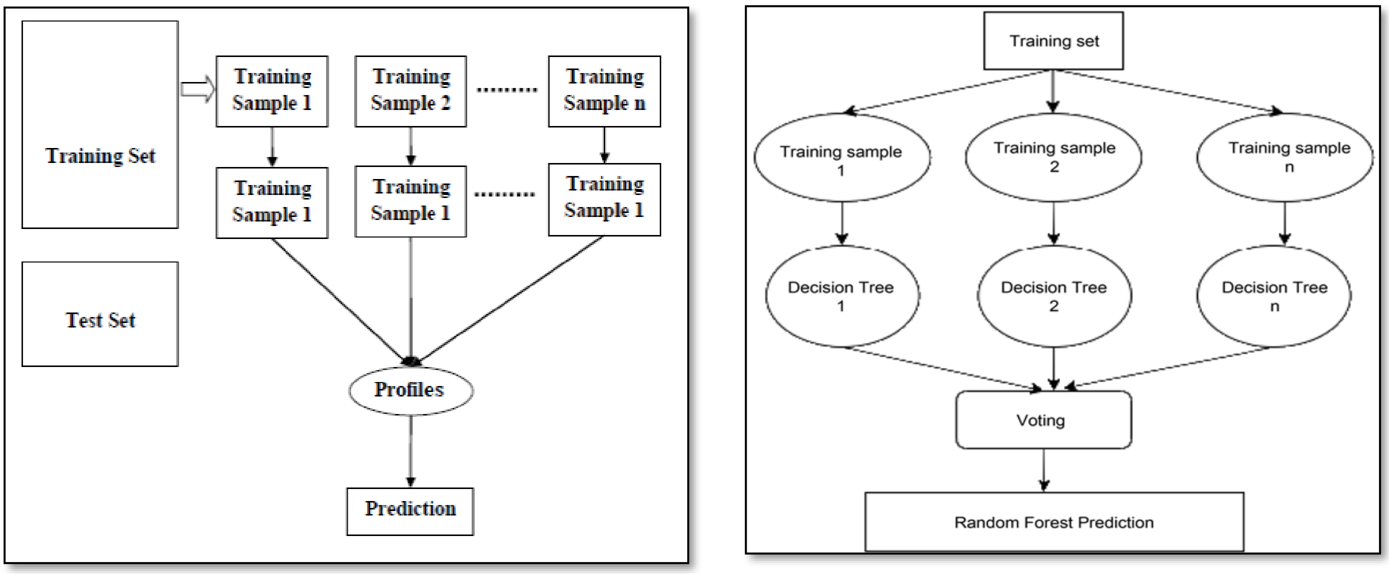
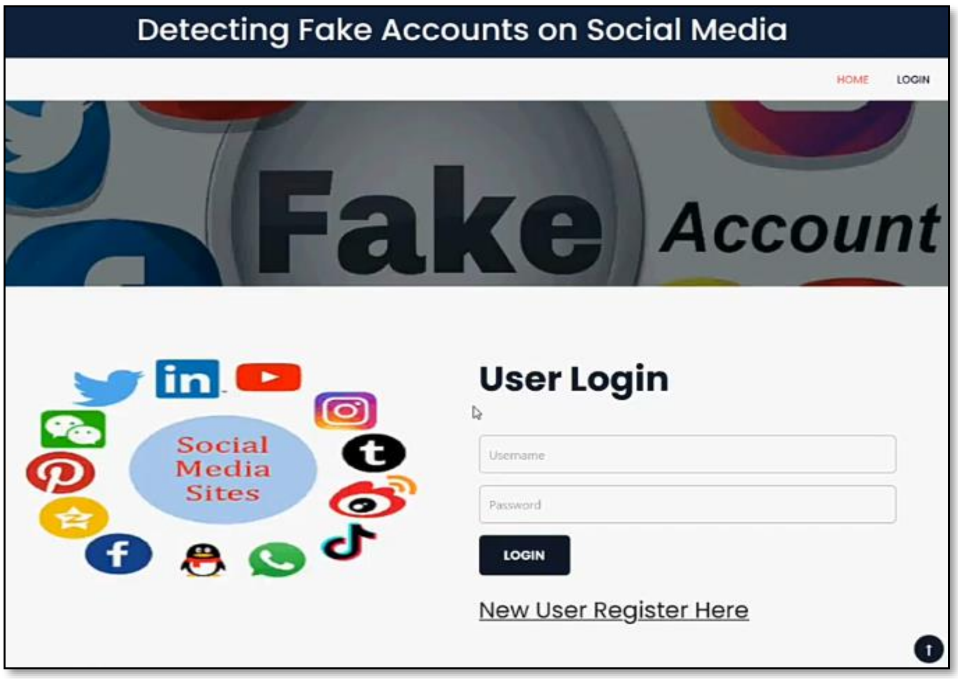


Fig 2.10 Flow Diagrams

CONCLUSION:

In a framework using which we can identify fake profiles in any online social network by using Random Forest Classifier with a very high efficiency as high as around 95%. Fake profile Identification can be improved by applying NLP techniques and Neural Networks to process the posts and the profiles. In the future we wish to classify profiles by taking profile pictures as one of the features.

Results



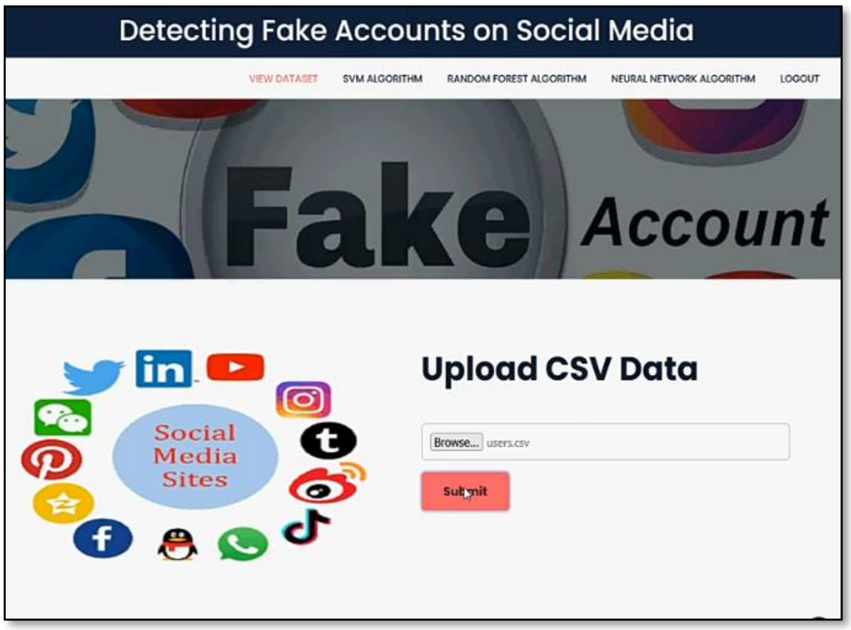


Fig 2.11 Fake account detection using Random Forest

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3
PROPOSED SYSTEM

3.1. Objective of Proposed Model

In our solution, we use machine learning, namely an Artificial Neural Network to determine what are the chances that a friend request is authentic or not.

• We utilize excel sheet 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.

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

All fake users main intention is to send friend request to normal users to hack their machine or to steal their data and never they will have many number of posts or have many following friends and their account age also will have less number of years. By analyzing these features, Facebook account will mark whether user profile is fake or genuine. This Facebook profile data we downloaded from Facebook website and using this data to train ANN model. Below are some values from profile dataset.

Account_Age, Gender, User_Age, Link_Desc, Status_Count, Friend_Count, Location, Location_IP, Status
10,1,22,0,1073,237,0,0,0
10,0,33,0,127,152,0,0,0
10,1,46,0,1601,405,0,0,0
7,1,34,1,64,721,1,1,1
7,1,30,1,69,587,1,1,1
7,1,36,1,61,782,1,1,1

Table 3.1 Train Data set

In above dataset all bold names are the dataset column names and all integer values are the dataset values. As ANN will not take string value so we convert gender values to 0 or 1, if male value is 1 and if female value is 0. In above dataset last column give us information of fake or genuine account if last column contains value 0 then account is genuine otherwise fake. All fake account will have less number of posts as their main intention is to send friend requests not posts, so by analyzing this features Facebook mark that record with value 1 which means it's a fake account. We are using above dataset to train ANN model and this dataset saved inside code 'dataset' folder. After building train model we input test data with account details and ANN will give result as fake or genuine. Below are some values from test data.

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

10,1,44,0,280,1273,0,0
10,0,54,0,5237,241,0, 0
7,0,42,1,57,631,1, 1
7,1,56,1,66,623,1, 1

Table 3.2 Test Data set

In above test data STATUS column and its value is not there and ANN will predict status and give us result whether above test data is fake or genuine. In output we can see result of above test data as genuine or fake.

<u>SOFTWARE REQUIREMENTS:</u>

❖ **Operating system**	**:**	Windows 7 Ultimate.
❖ **Coding Language**	**:**	Python.
❖ **Front-End**	**:**	Python.
❖ **Designing**	**:**	Html, CSS, JavaScript.
❖ **Data Base**	**:**	Excel Sheet

3.2. Algorithm Used for Proposed Model:

ANN algorithms Details:

What is an Artificial Neural Network?

Artificial Neural Network is much similar to the human brain. The human brain consists of **neurons**. These neurons are connected with each other. In the human brain, neuron looks something like this…

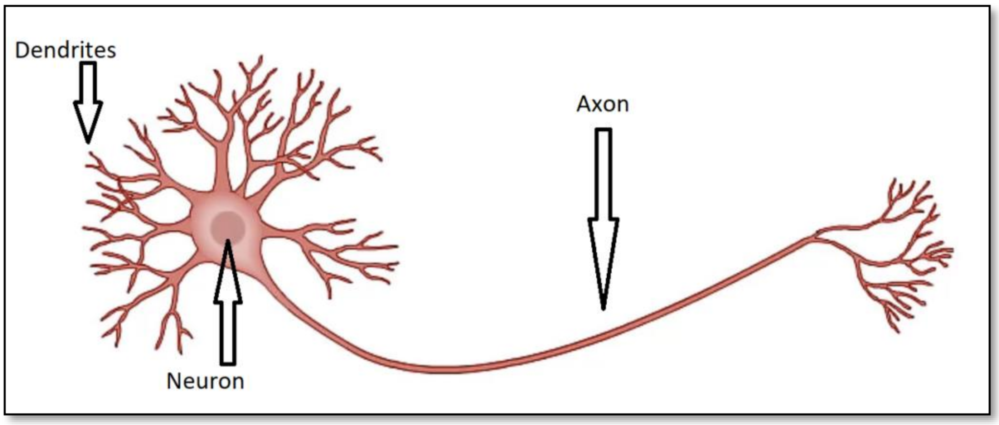


Fig 3.1 ANN algorithms neuron

Artificial Neural Network has three layers-

1. Input Layer.
2. Hidden Layer.
3. Output Layer.

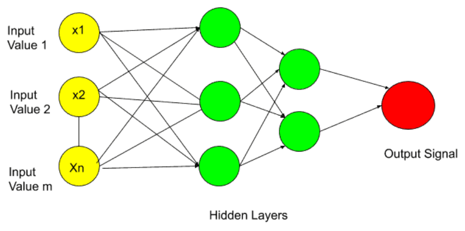


Fig 3.2 ANN Layers

Data is passed to the **input layer**. And then the input layer passed this data to the next layer, which is a **hidden layer**. The hidden layer performs certain operations. And pass the result to the **output layer**.

Synapses-

Synapses are nothing but the connecting lines between two layers.

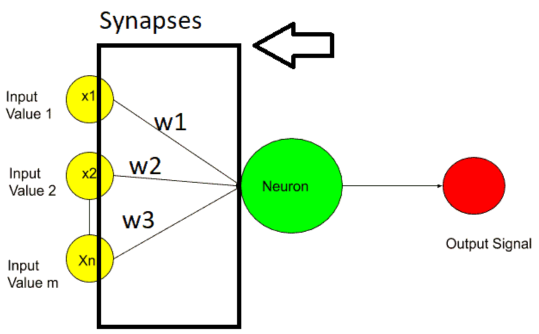


Fig 3.3 Synapses

In synapses, weights are assigned to each synapse. These weights are crucial for artificial neural network work. Weights are how neural networks learn. By adjusting the weights neural network decides what signal is important and what signal is not important.

Neuron-

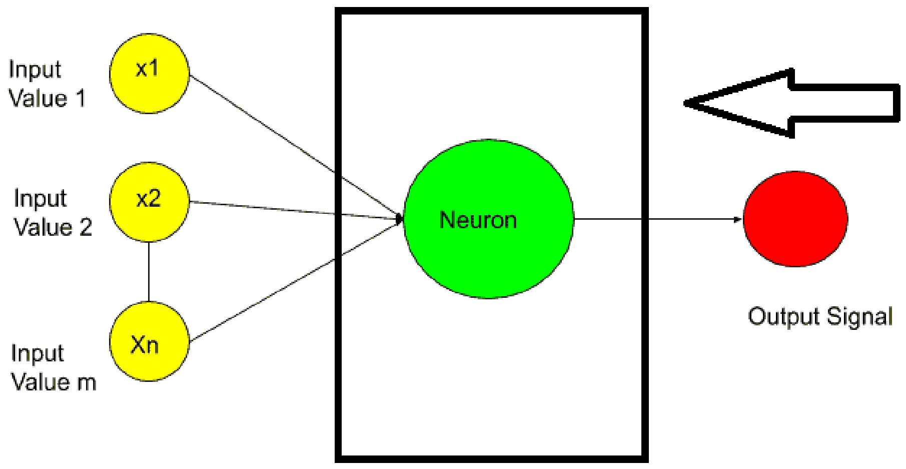


Fig 3.4 Neuron

What Happens inside the neurons?

So Inside the neurons, the two main important steps happen-

1. Weighted Sum.
2. Activation Function.

The first step is the weighted sum, which means all of the weights assigned to the synapses are added with input values. Something like that-

[x1.w1+x2.w2+x3.w3+………………..Xn.Wn]

After calculating the weighted sum, the **activation function** is applied to this weighted sum. And then the neuron decides whether to send this signal to the next layer or not.

ANN Implementation in Python

1. **Data Preprocessing**
   - 1.1 Import the Libraries
   - 1.2 Load the Dataset
   - 1.3 Split Dataset into X and Y
   - 1.4 Encode Categorical Data(One Hot Encoding)
   - 1.5 Split the X and Y Dataset into the Training set and Test set
   - 1.6 Perform Feature Scaling

2. Build Artificial Neural Network

- 2.1 Import the Keras libraries and packages
- 2.2 Initialize the Artificial Neural Network
- 2.3 Add the input layer and the first hidden layer
- 2.4 Add the second hidden layer
- 2.5 Add the output layer

3. Train the ANN

- 3.1 Compile the ANN
- 3.2 Fit the ANN to the Training set

4. Predict the Test Set Results

5. Make the Confusion Matrix

Important Libraries.

Machine Learning:

- **TensorFlow** (deep learning with neural networks) *
- **scikit-learn** (machine learning algorithms)
- **keras** (high-level neural networks API)

Data Science:

- **pandas** (data analysis)
- **NumPy** (multidimensional arrays)
- **SciPy** (algorithms to use with numpy)
- **HDF5** (store & manipulate data)
- **matplotlib** (data visualization)

To demonstrate how to build a ANN neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is to show how to build a real-world convolutional neural network using TENSORFLOW.

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output(activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

If you stack neurons in a single line, it’s called a layer; which is the next building block of neural networks. To predict class label multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

SYSTM DESIGN

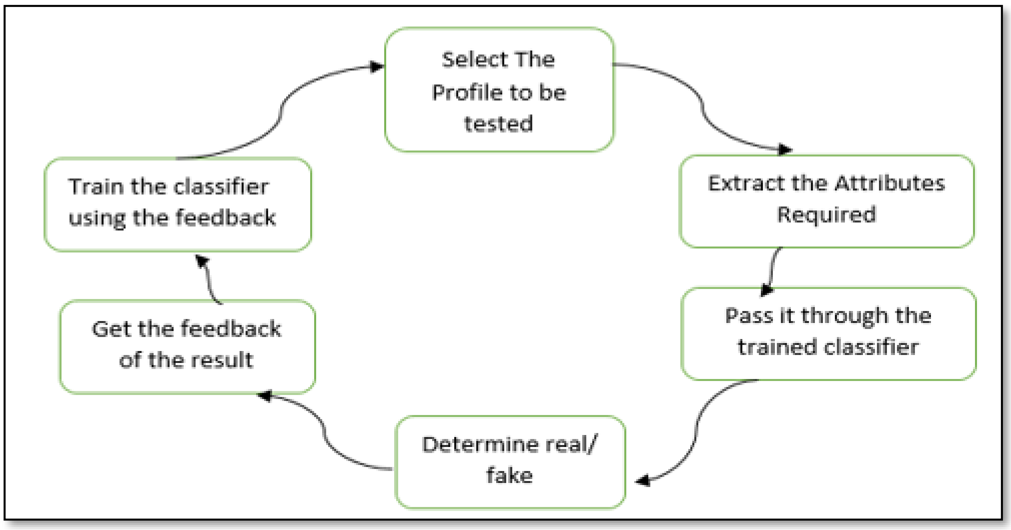

Select The Profile to be tested
Extract the Attributes Required
Pass it through the trained classifier
Determine real/ fake
Get the feedback of the result
Train the classifier using the feedback

Fig 3.5 System Design

3.3. Designing:

- **Upload Social Network Profiles Dataset**: Using this module we will upload dataset to application.
- **Preprocess Dataset**: Using this module we will apply processing technique such as removing missing values and then split dataset into train and test where application use 80% dataset to train ANN and 20% dataset to test ANN prediction accuracy.
- **Run ANN Algorithm**: Using this module we will train ANN algorithm with train and test data and then train model will be generated and we can use this train model to predict fake accounts from new dataset.
- **ANN Accuracy & Loss Graph**: To train ANN model we are taking 200 epoch/iterations and then in graph we will plot accuracy/loss performance of ANN at each epoch/iteration.
- **Predict Fake/Genuine Profile using ANN**: using this module we will upload new test data and then apply ANN train model to predict whether test data is genuine or fake.

3.3.1.UML Diagrams:

UML diagrams that could be relevant for fake account detection implementation:

- **Class Diagram:** A class diagram represents the static structure of a system, showing the classes, their attributes, methods, and relationships. In the context of fake account detection, a class diagram could represent classes such as "Social Media Account," "User," "Fake Account," and their relationships, such as inheritance, association, or aggregation.
- **Activity Diagram:** An activity diagram represents the flow of activities or processes in a system. It can be used to model the steps or actions involved in the fake account detection process, such as data preprocessing, feature extraction, model training, and model testing, along with decision points, parallel activities, and flow control.
- **Sequence Diagram:** A sequence diagram represents the interactions between objects or components in a system over time. It can be used to model the sequence of messages or method calls exchanged between different components or modules involved in the fake account detection process, such as data collection, model training, and model testing.

- **State Machine Diagram:** A state machine diagram represents the states and transitions of an object or system. It can be used to model the behavior of a system or object during the fake account detection process, such as the different states an account can be in (e.g., genuine, suspicious, fake) and the transitions between these states based on certain conditions or events.
- **Deployment Diagram:** A deployment diagram represents the physical deployment of software components and hardware infrastructure in a system. It can be used to model the deployment of the fake account detection system in a production environment, including the servers, databases, and communication channels involved.

These are just a few examples of UML diagrams that can be used for fake account detection implementation. Depending on the complexity and requirements of the system, other UML diagrams such as use case diagrams, component diagrams, and package diagrams may also be useful for modeling different aspects of the system. It's important to choose the appropriate UML diagrams based on the specific needs of the fake account detection system and use them to effectively communicate the system's structure and behavior to stakeholders.

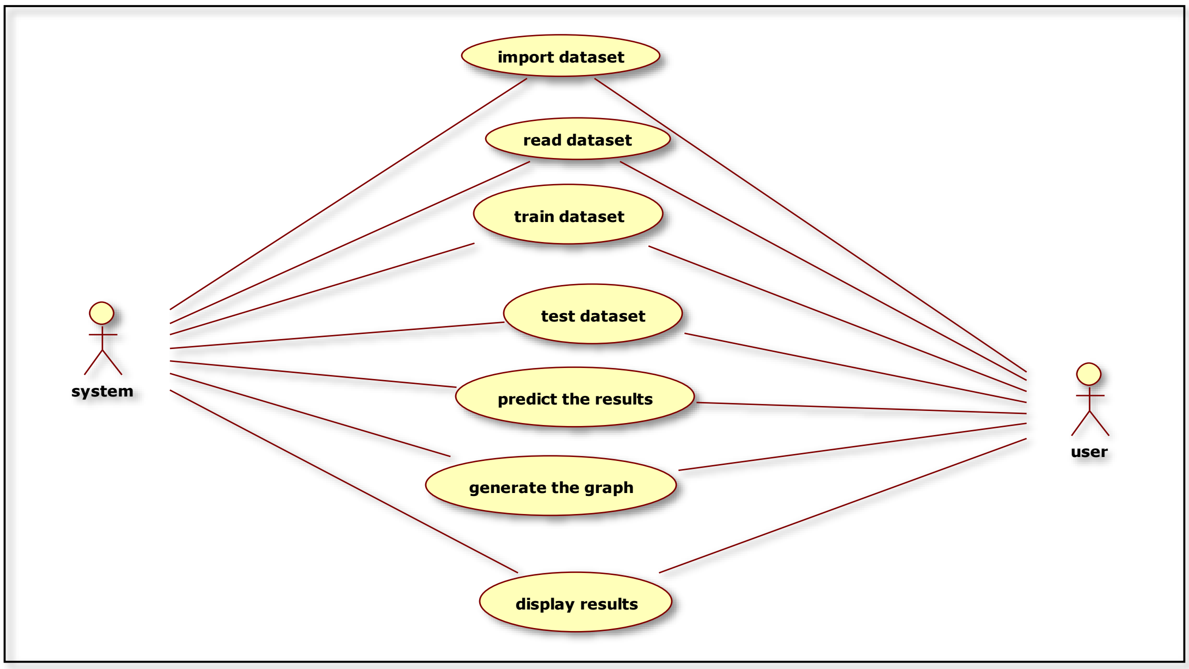


Fig 3.6 Use Case Diagram

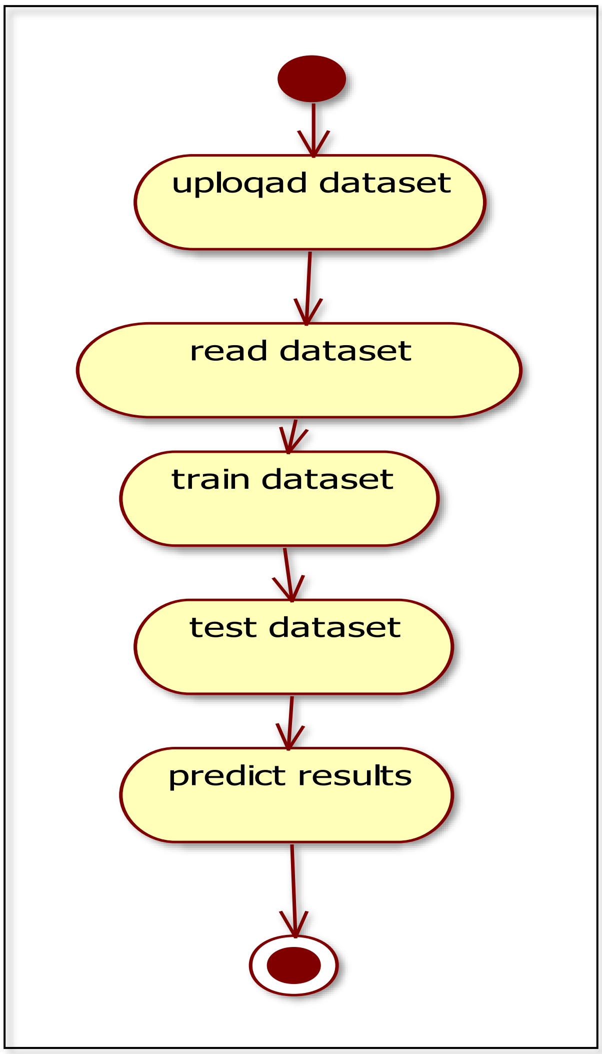


Fig 3.7 Activity Diagram

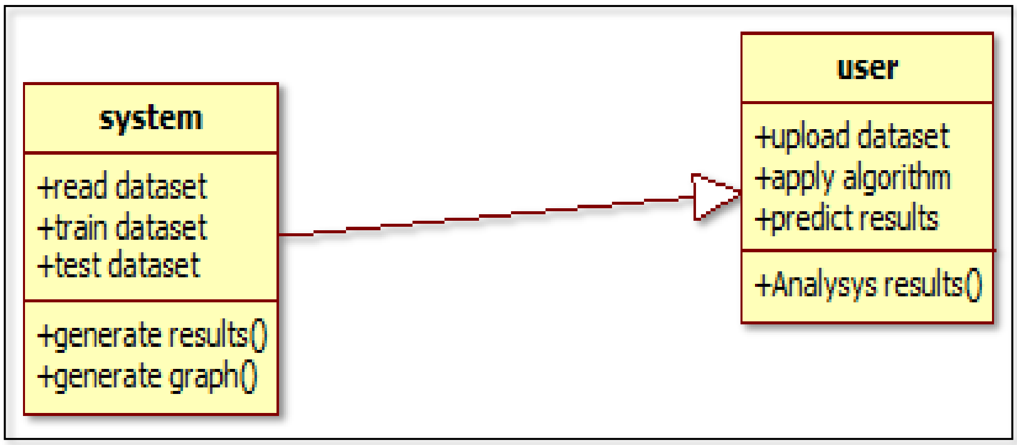


Fig 3.8 Class Diagram

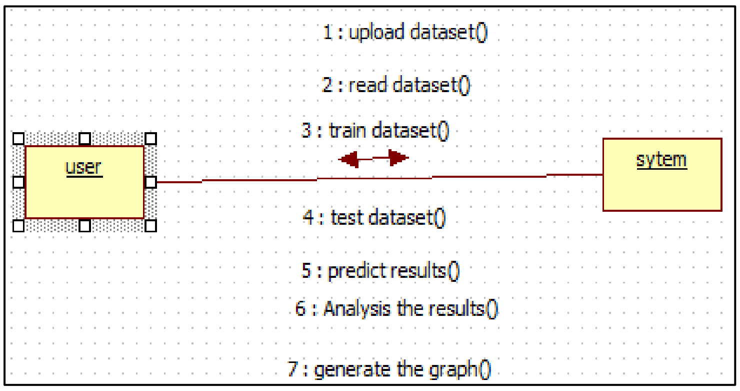


Fig 3.9 Collaboration Diagram

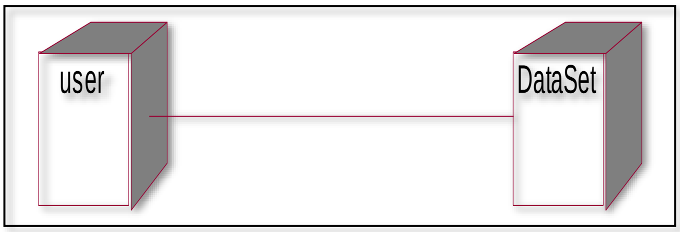


Fig 3.10 Deployment Diagram

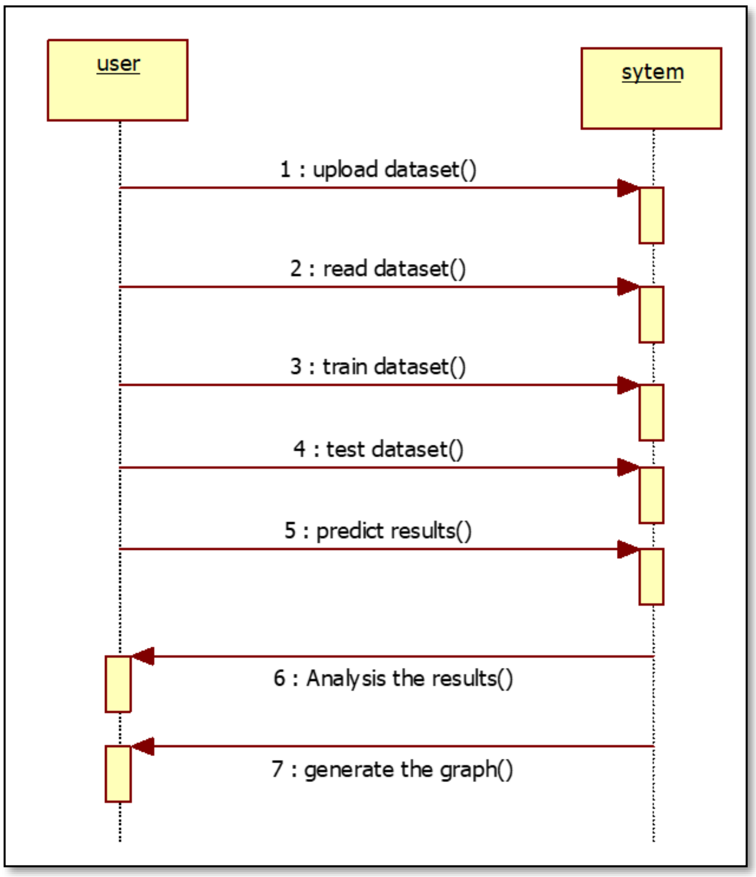


Fig 3.11 Sequence Diagram

3.4. Stepwise Implementation and Code:

A. Python

Python is our major programming language. It is used for our dataset to detect fake accpounts. It provides various tools and libraries that help in detecting fake accounts upto high accuracy levels.

B. Module description:

TensorFlow: TensorFlow may be a free and ASCII text file software package library for dataflow and differentiable programming across a spread of tasks. It's a symbolic scientific discipline library and is additionally used for machine learning applications like neural networks. It's used for each analysis and production at Google. TensorFlow was developed by the Google Brain team for internal Google use.

Pandas: Pandas is an associate degree ASCII text file Python Library providing superior knowledge manipulation and analysis tool victimization its powerful knowledge structures. Python was majorly used for knowledge munging and preparation. It had little contribution towards knowledge analysis. Pandas resolved this drawback. victimization Pandas, will accomplish 5 typical steps within the process and analysis of information, in spite of the origin of information load, prepare, manipulate, model, and analyze. Python with Pandas is employed in a very wide selection of fields as well as educational and industrial domains as well as finance, economics, Statistics, analytics, etc.

Matplotlib: Matplotlib is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels.

Scikit – learn Scikit – learn Scikit-learn provides a spread of supervised and unattended learning algorithms via an identical interface in Python. The library is made upon the SciPy (Scientific Python) that has to be put in before you'll use scikit-learn. This stack that includes:

- NumPy: Base n-dimensional array package
- SciPy: Fundamental library for scientific computing
- Matplotlib: Comprehensive 2D/3D plotting
- I Python: Enhanced interactive console
- Sympy: Symbolic mathematics
- Pandas: Data structures and analysis
- Extensions or modules for SciPy care conventionally named SciKits

Module Details:

- **Upload Social Network Profiles Dataset**: Using this module we will upload dataset to application.
- **Preprocess Dataset**: Using this module we will apply processing technique such as removing missing values and then split dataset into train and test where application use 80% dataset to train ANN and 20% dataset to test ANN prediction accuracy.
- **Run ANN Algorithm**: Using this module we will train ANN algorithm with train and test data and then train model will be generated and we can use this train model to predict fake accounts from new dataset.
- **ANN Accuracy & Loss Graph**: To train ANN model we are taking 200 epoch/iterations and then in graph we will plot accuracy/loss performance of ANN at each epoch/iteration.
- **Predict Fake/Genuine Profile using ANN**: using this module we will upload new test data and then apply ANN train model to predict whether test data is genuine or fake.

Application Screenshots:

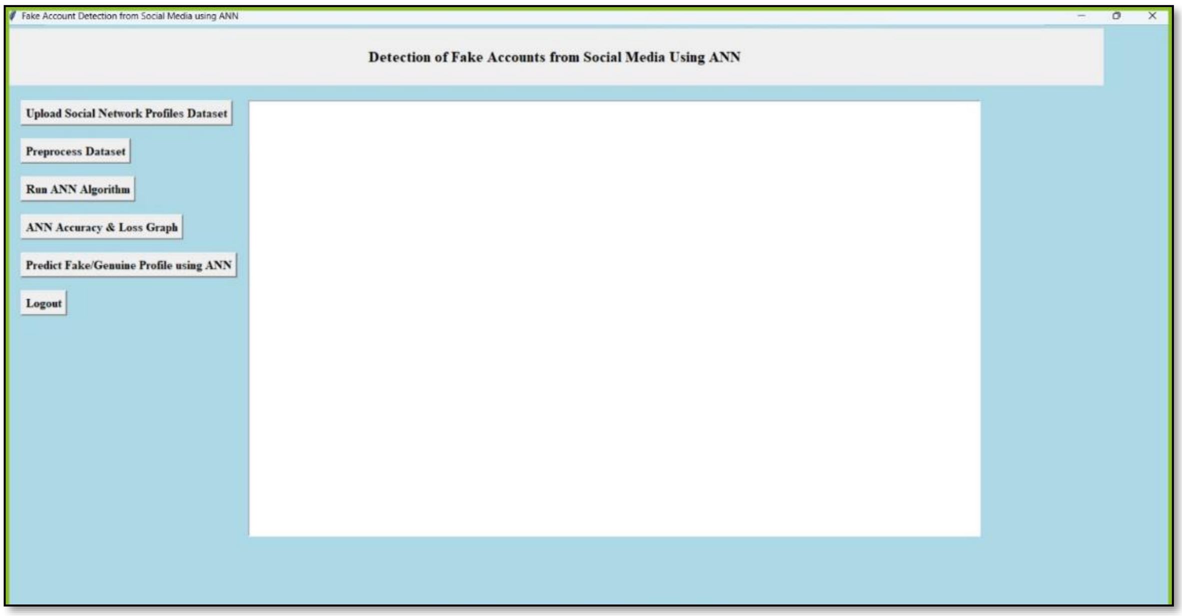

Fake Account Detection from Social Media using ANN
Detection of Fake Accounts from Social Media Using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

Fig 3.12 Dashboard

In above screen click on ‘Upload Social Network Profiles Dataset’ button and upload dataset

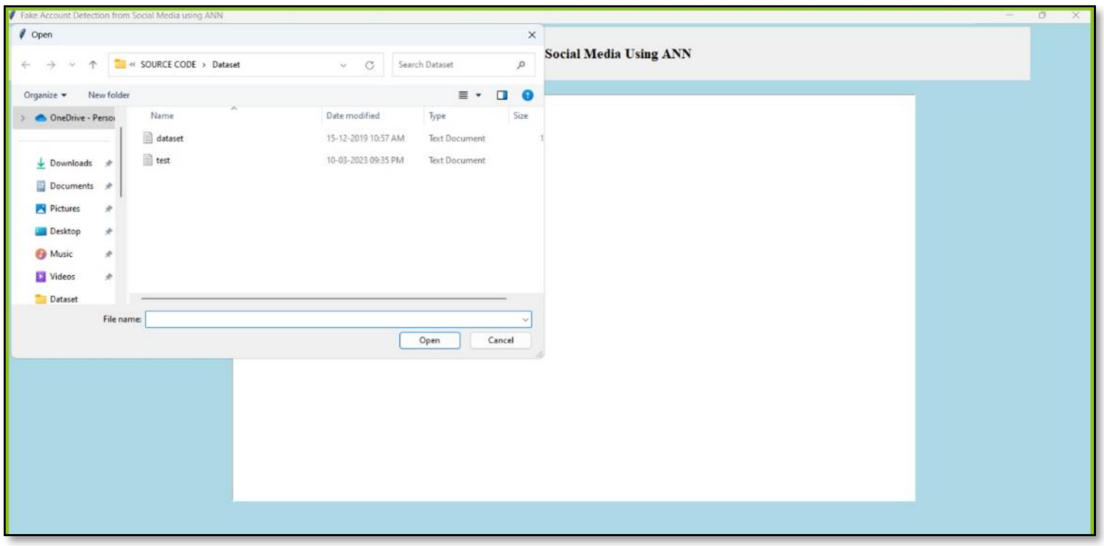
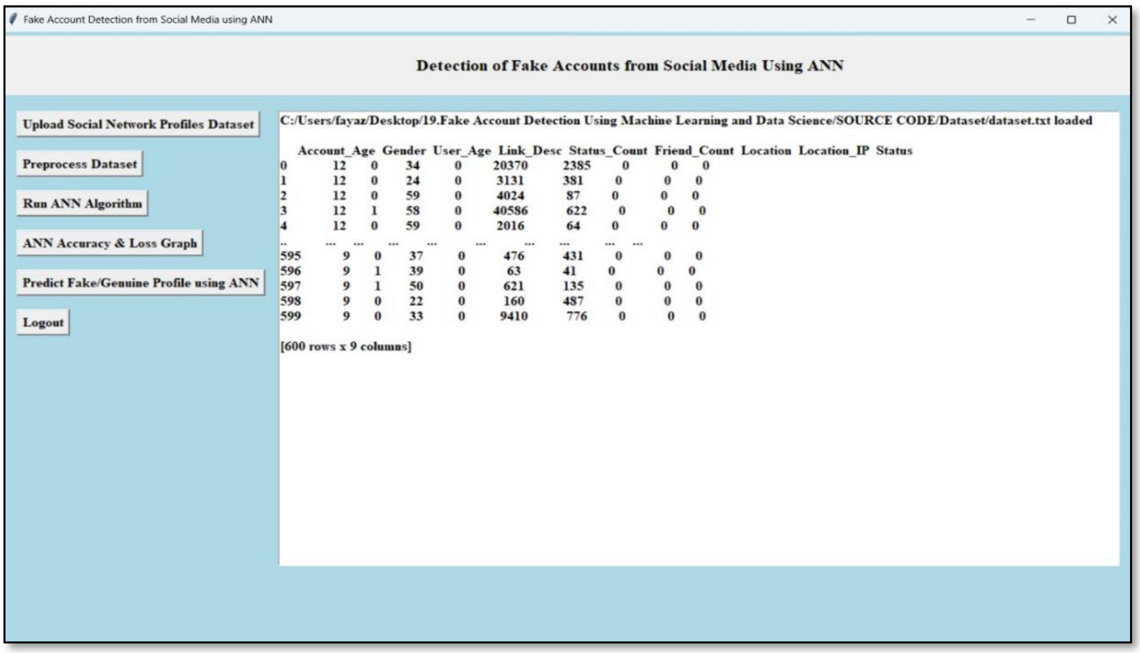


Fig 3.13 Datasets

In above screen selecting and uploading ‘dataset.txt’ file and then click on ‘Open’ button to load dataset and to get below screen


Fake Account Detection from Social Media using ANN
Detection of Fake Accounts from Social Media Using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

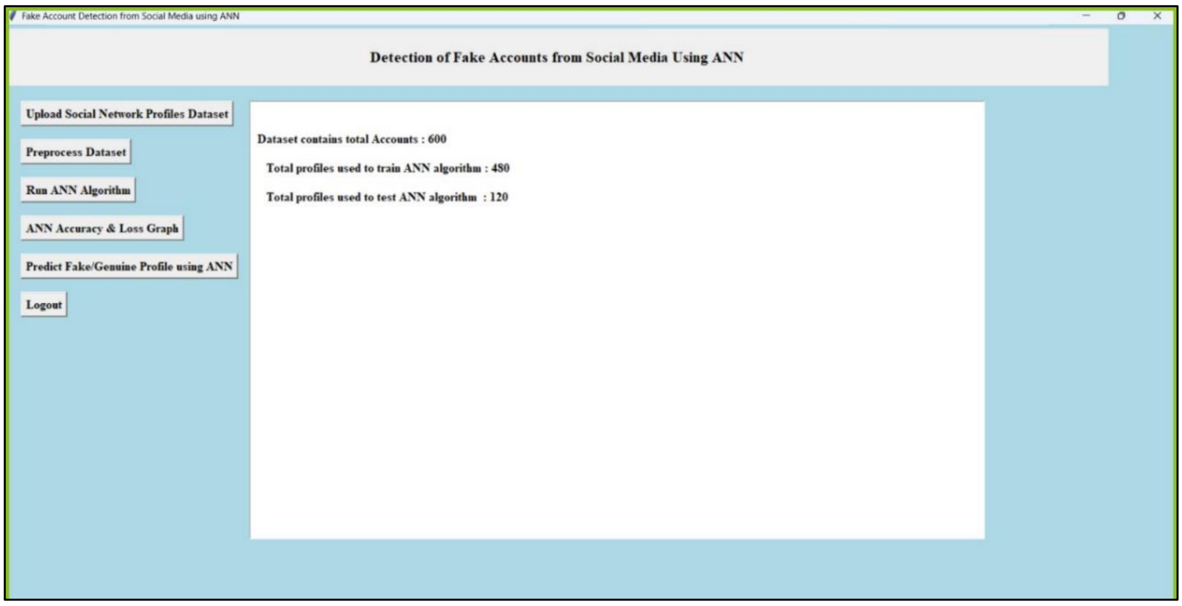
```
C:/Users/fayaz/Desktop/19.Fake Account Detection Using Machine Learning and Data Science/SOURCE CODE/Dataset/dataset.txt loaded

    Account_Age  Gender  User_Age  Link_Desc  Status_Count  Friend_Count  Location  Location_IP  Status
0          12       0      34         0      20370        2385       0          0      0
1          12       0      24         0       3131         381       0          0      0
2          12       0      59         0       4024          87       0          0      0
3          12       1      58         0      40586         622       0          0      0
4          12       0      59         0       2016          64       0          0      0
..        ...     ...     ...       ...        ...         ...     ...        ...    ...
595         9       0      37         0        476         431       0          0      0
596         9       1      39         0         63          41       0          0      0
597         9       1      50         0        621         135       0          0      0
598         9       0      22         0        160         487       0          0      0
599         9       0      33         0       9410         776       0          0      0

[600 rows x 9 columns]
```

Fig 3.14 Uploaded train dataset

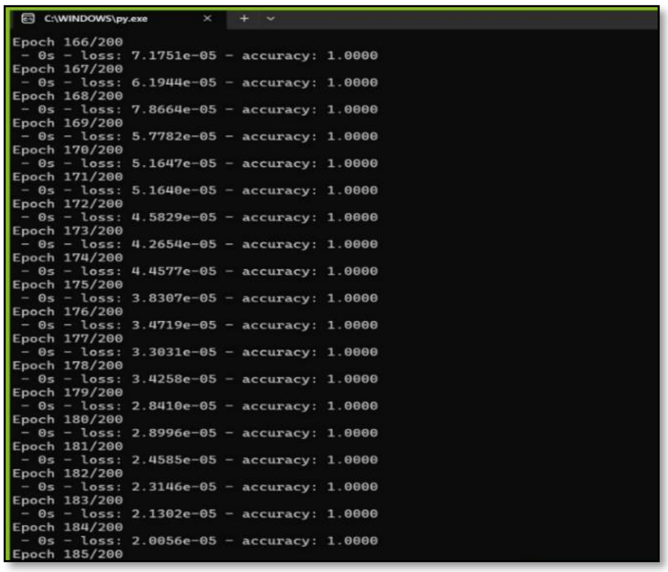
In above screen dataset loaded and displaying few records from dataset and now click on ‘Preprocess Dataset’ button to remove missing values and to split dataset into train and test part.


Fake Account Detection from Social Media using ANN
Detection of Fake Accounts from Social Media Using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

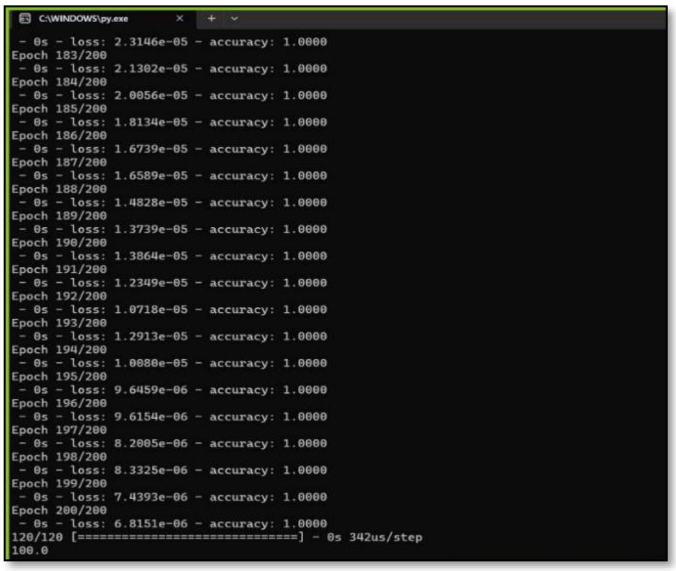
```
Dataset contains total Accounts : 600
  Total profiles used to train ANN algorithm : 480
  Total profiles used to test ANN algorithm  : 120
```

Fig 3.15 Preprocess Dataset

In above screen we can see dataset contains total 600 records and application using 480 records for training and 120 records to test ANN and now dataset is ready and now click on ‘Run ANN Algorithm’ button to ANN algorithm. In below screen we can see ANN start iterating model generation and at each increasing epoch we can see accuracy is getting increase and loss getting decrease.



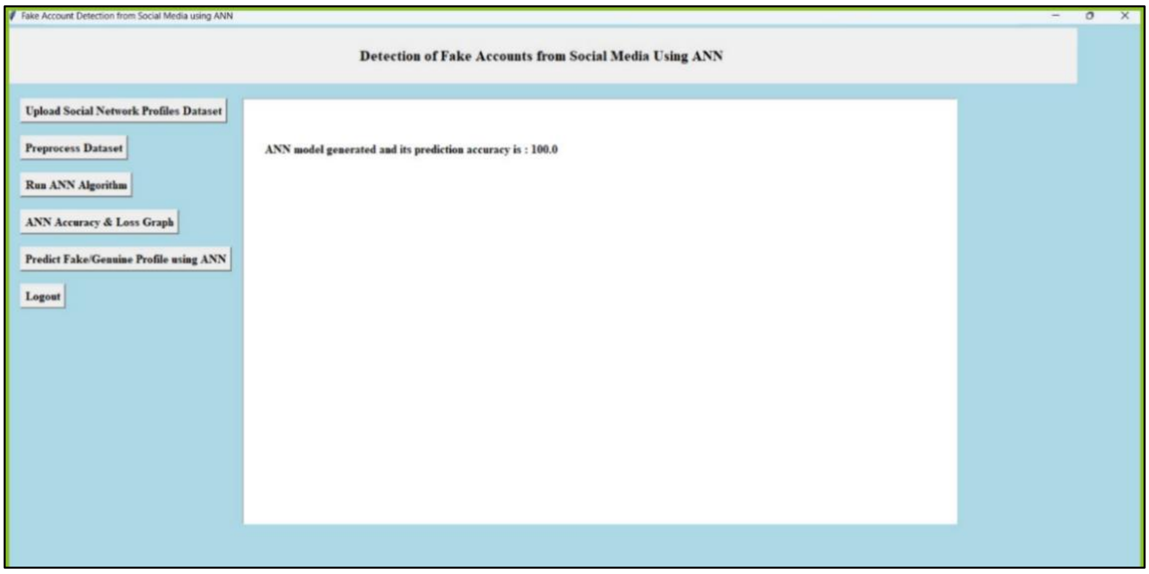
```
Epoch 166/200
 - 0s - loss: 7.1751e-05 - accuracy: 1.0000
Epoch 167/200
 - 0s - loss: 6.1944e-05 - accuracy: 1.0000
Epoch 168/200
 - 0s - loss: 7.8664e-05 - accuracy: 1.0000
Epoch 169/200
 - 0s - loss: 5.7782e-05 - accuracy: 1.0000
Epoch 170/200
 - 0s - loss: 5.1647e-05 - accuracy: 1.0000
Epoch 171/200
 - 0s - loss: 5.1640e-05 - accuracy: 1.0000
Epoch 172/200
 - 0s - loss: 4.5829e-05 - accuracy: 1.0000
Epoch 173/200
 - 0s - loss: 4.2654e-05 - accuracy: 1.0000
Epoch 174/200
 - 0s - loss: 4.4577e-05 - accuracy: 1.0000
Epoch 175/200
 - 0s - loss: 3.8307e-05 - accuracy: 1.0000
Epoch 176/200
 - 0s - loss: 3.4719e-05 - accuracy: 1.0000
Epoch 177/200
 - 0s - loss: 3.3031e-05 - accuracy: 1.0000
Epoch 178/200
 - 0s - loss: 3.4258e-05 - accuracy: 1.0000
Epoch 179/200
 - 0s - loss: 2.8410e-05 - accuracy: 1.0000
Epoch 180/200
 - 0s - loss: 2.8996e-05 - accuracy: 1.0000
Epoch 181/200
 - 0s - loss: 2.4585e-05 - accuracy: 1.0000
Epoch 182/200
 - 0s - loss: 2.3146e-05 - accuracy: 1.0000
Epoch 183/200
 - 0s - loss: 2.1302e-05 - accuracy: 1.0000
Epoch 184/200
 - 0s - loss: 2.0056e-05 - accuracy: 1.0000
Epoch 185/200
```



```
 - 0s - loss: 2.3146e-05 - accuracy: 1.0000
Epoch 183/200
 - 0s - loss: 2.1302e-05 - accuracy: 1.0000
Epoch 184/200
 - 0s - loss: 2.0056e-05 - accuracy: 1.0000
Epoch 185/200
 - 0s - loss: 1.8134e-05 - accuracy: 1.0000
Epoch 186/200
 - 0s - loss: 1.6739e-05 - accuracy: 1.0000
Epoch 187/200
 - 0s - loss: 1.6589e-05 - accuracy: 1.0000
Epoch 188/200
 - 0s - loss: 1.4828e-05 - accuracy: 1.0000
Epoch 189/200
 - 0s - loss: 1.3739e-05 - accuracy: 1.0000
Epoch 190/200
 - 0s - loss: 1.3864e-05 - accuracy: 1.0000
Epoch 191/200
 - 0s - loss: 1.2349e-05 - accuracy: 1.0000
Epoch 192/200
 - 0s - loss: 1.0718e-05 - accuracy: 1.0000
Epoch 193/200
 - 0s - loss: 1.2913e-05 - accuracy: 1.0000
Epoch 194/200
 - 0s - loss: 1.0080e-05 - accuracy: 1.0000
Epoch 195/200
 - 0s - loss: 9.6459e-06 - accuracy: 1.0000
Epoch 196/200
 - 0s - loss: 9.6154e-06 - accuracy: 1.0000
Epoch 197/200
 - 0s - loss: 8.2005e-06 - accuracy: 1.0000
Epoch 198/200
 - 0s - loss: 8.3325e-06 - accuracy: 1.0000
Epoch 199/200
 - 0s - loss: 7.4393e-06 - accuracy: 1.0000
Epoch 200/200
 - 0s - loss: 6.8151e-06 - accuracy: 1.0000
120/120 [==============================] - 0s 342us/step
100.0
```

Fig 3.16 Epoch Iterations

In above screen we can see after 200 epoch ANN got 100% accuracy and in below screen, we can see final ANN accuracy


Fake Account Detection from Social Media using ANN
Detection of Fake Accounts from Social Media Using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

ANN model generated and its prediction accuracy is : 100.0

Fig 3.17 Accuracy of test dataset

In above screen ANN model generated and now click on 'ANN Accuracy & Loss Graph' button to get below graph.

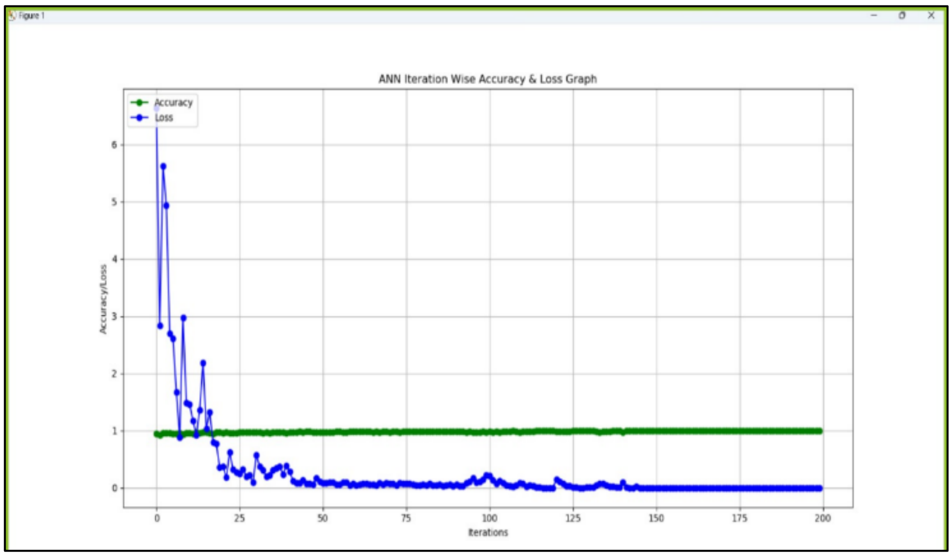


Fig 3.18 Accuracy vs Loss Graph

In above graph x-axis represents epoch and y-axis represents accuracy/loss value and in above graph green line represents accuracy and blue line represents loss value and we can see accuracy was increase from 0.90 to 1 and loss value decrease from 7 to 0.1. Now the model is ready and now click on 'Predict Fake/Genuine Profile using ANN' button to upload test data and then ANN will predict below result.

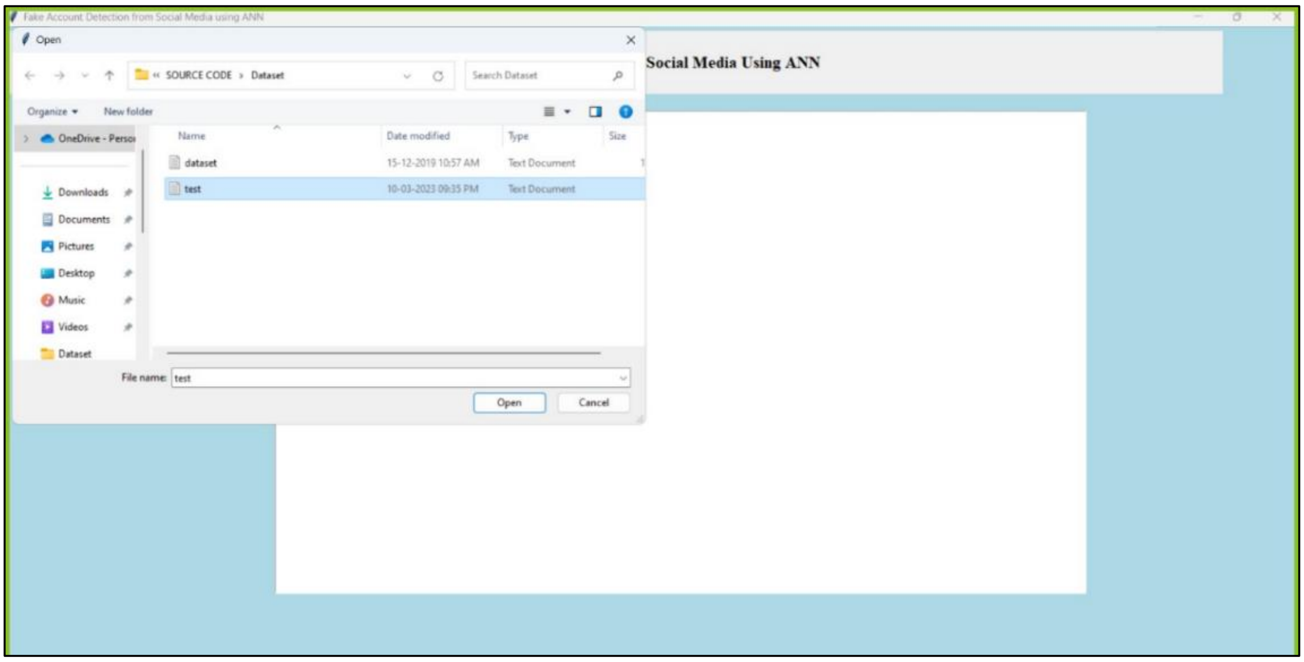
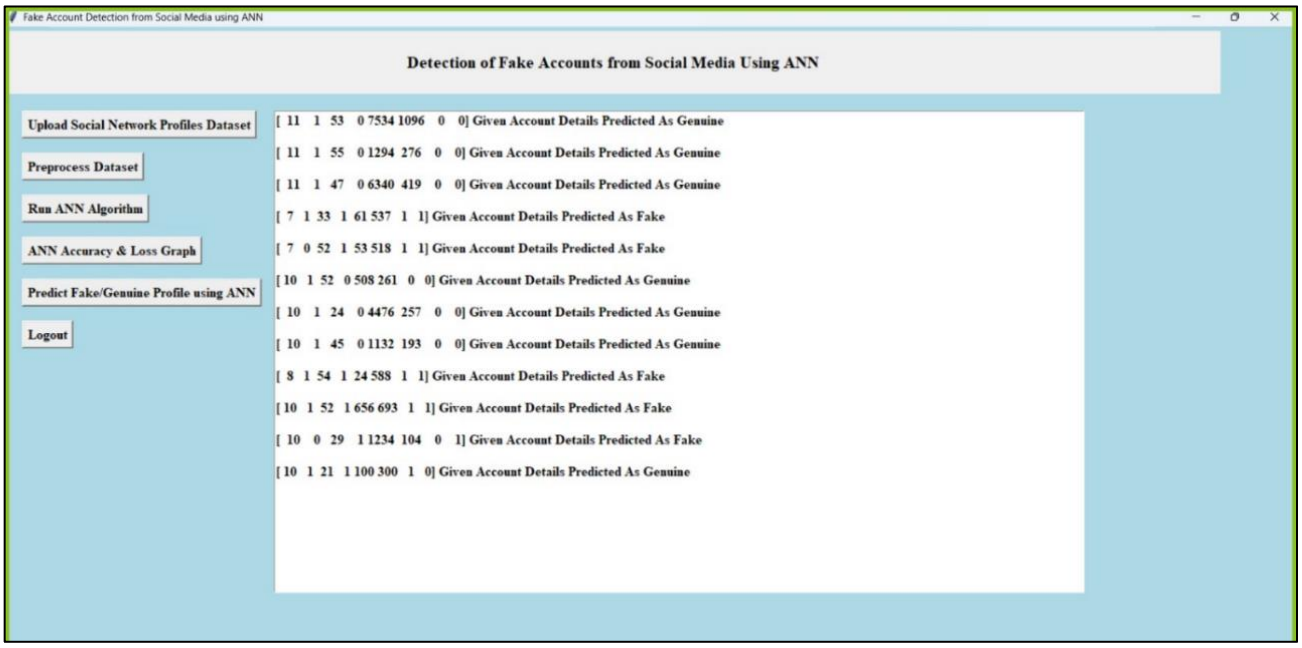


Fig 3.19 Test Dataset

In above screen we are selecting and uploading 'test.txt' file and then click on 'Open' button to load test data and to get below prediction result


Fake Account Detection from Social Media using ANN
Detection of Fake Accounts from Social Media Using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

```
[ 11   1  53   0 7534 1096   0   0] Given Account Details Predicted As Genuine
[ 11   1  55   0 1294  276   0   0] Given Account Details Predicted As Genuine
[ 11   1  47   0 6340  419   0   0] Given Account Details Predicted As Genuine
[  7   1  33   1   61  537   1   1] Given Account Details Predicted As Fake
[  7   0  52   1   53  518   1   1] Given Account Details Predicted As Fake
[ 10   1  52   0  508  261   0   0] Given Account Details Predicted As Genuine
[ 10   1  24   0 4476  257   0   0] Given Account Details Predicted As Genuine
[ 10   1  45   0 1132  193   0   0] Given Account Details Predicted As Genuine
[  8   1  54   1   24  588   1   1] Given Account Details Predicted As Fake
[ 10   1  52   1  656  693   1   1] Given Account Details Predicted As Fake
[ 10   0  29   1 1234  104   0   1] Given Account Details Predicted As Fake
[ 10   1  21   1  100  300   1   0] Given Account Details Predicted As Genuine
```

Fig 3.20 Results of Test dataset

In the above screen in square bracket, we can see uploaded test data and after square bracket we can see ANN prediction result as genuine or fake.

Code:

```
from tkinter import messagebox
 from tkinter import *
 from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from tkinter import filedialog
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 tensorflow.keras.optimizers import Adam
from keras.utils.np_utils import to_categorical

main = Tk()
main.title("Fake Account Detection On Social Media Using ANN")
main.geometry("1300x1200")
main.config(bg="lightgreen")

global filename
global X, Y
global X_train, X_test, y_train, y_test
global accuracy
global dataset
global model
 def loadProfileDataset():
        global filename
        global dataset
        outputarea.delete('1.0', END)
        filename = filedialog.askopenfilename(initialdir="Dataset")
        outputarea.insert(END,filename+" loaded\n\n")
        dataset = pd.read_csv(filename)
        outputarea.insert(END,str(dataset.head()))
```

```
def preprocessDataset():
            global X, Y
            global dataset
            global X_train, X_test, y_train, y_test
            outputarea.delete('1.0', END)
            X = dataset.values[:, 0:8]
            Y = dataset.values[:, 8]
            indices = np.arange(X.shape[0])
            np.random.shuffle(indices)
            X = X[indices]
            Y = Y[indices]
            Y = to_categorical(Y) X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
            outputarea.insert(END,"\n\n Dataset contains total Accounts : "+str(len(X))+"\n")
            outputarea.insert(END,"Total profiles used to train ANN algorithm:"+str(len(X_train))+"\n")
            outputarea.insert(END,"Total profiles used to test ANN algorithm : "+str(len(X_test))+"\n")

def executeANN():
             global model
            outputarea.delete('1.0', END)
            global X_train, X_test, y_train, y_test
            global accuracy
            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('ANN Neural Network Model Summary: ')
            print(model.summary())
            hist = model.fit(X_train, y_train, verbose=2, batch_size=5, epochs=200)
            results = model.evaluate(X_test, y_test)
            ann_acc = results[1] * 100
            print(ann_acc)
            accuracy = hist.history
            acc = accuracy['accuracy']
            acc = acc[199] * 100
            outputarea.insert(END,"ANN model generated and its prediction accuracy:"+str(acc)+"\n")
```

```
def graph():
	global accuracy
	acc = accuracy['accuracy']
	loss = accuracy['loss']
	plt.figure(figsize=(10,6))
	plt.grid(True)
	plt.xlabel('Iterations')
	plt.ylabel('Accuracy/Loss')
	plt.plot(acc, 'ro-', color = 'green')
	plt.plot(loss, 'ro-', color = 'blue')
	plt.legend(['Accuracy', 'Loss'], loc='upper left')
	#plt.xticks(wordloss.index)
	plt.title('ANN Iteration Wise Accuracy & Loss Graph')
	plt.show()

def predictProfile():
	outputarea.delete('1.0', END)
	global model
	filename = filedialog.askopenfilename(initialdir="Dataset")
	test = pd.read_csv(filename)
	test = test.values[:, 0:8]
	#predict = model.predict_classes(test)
	predict = (model.predict(test) > 0.5).astype("int32")
	print(predict)
	for i in range(len(test)):
		msg = ''
		if str(predict[i]) == '0':
			msg = "Given Account Details Predicted As Genuine"
		if str(predict[i]) == '1':
			msg = "Given Account Details Predicted As Fake"
		outputarea.insert(END,str(test[i])+" \t"+msg+"\n\n")
def close():
	main.destroy()
font = ('times', 15, 'bold')
title = Label(main, text='Fake Account Detection On Social Media Using ANN')
#title.config(bg='powder blue', fg='olive drab')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')
uploadButton = Button(main, text="Upload Social Network Profiles Dataset", command=loadProfileDataset)
uploadButton.place(x=20,y=100)
uploadButton.config(font=ff)
processButton = Button(main, text="Preprocess Dataset", command=preprocessDataset)
processButton.place(x=20,y=150)
processButton.config(font=ff)
annButton = Button(main, text="Run ANN Algorithm", command=executeANN)
annButton.place(x=20,y=200)
annButton.config(font=ff)
graphButton = Button(main, text="ANN Accuracy & Loss Graph", command=graph)
graphButton.place(x=20,y=250)
graphButton.config(font=ff)
predictButton = Button(main, text="Predict Fake/Genuine Profile using ANN", command=predictProfile)
predictButton.place(x=20,y=300)
predictButton.config(font=ff)
exitButton = Button(main, text="Logout", command=close)
exitButton.place(x=20,y=350)
exitButton.config(font=ff)
font1 = ('times', 12, 'bold')
outputarea = Text(main,height=30,width=85)
scroll = Scrollbar(outputarea)
outputarea.configure(yscrollcommand=scroll.set)
outputarea.place(x=400,y=100)
outputarea.config(font=font1)

main.config()
main.mainloop()
```

Description of Code:

- **Importing Libraries:**
    - The first few lines of the code are used to import the necessary libraries for the program. These include tkinter for creating the GUI, matplotlib and numpy for data visualization, pandas for data processing, scikit-learn for machine learning, and keras for building and training neural networks.
- **Creating the GUI:**
    - The Tk() method is used to create the main window of the GUI, which is assigned to the variable main. The title(), geometry(), and config() methods are then used to set the title, size, and background color of the window, respectively.
- **Loading and Preprocessing Data:**
    - The program allows the user to load a dataset of social network profiles using the loadProfileDataset() function. Once the dataset is loaded, the preprocessDataset() function is called to preprocess the data. This involves splitting the dataset into training and testing sets and performing one-hot encoding on the target variable.
- **Building and Training a Neural Network:**
    - The executeANN() function builds a neural network using the keras library. The neural network consists of three layers: two hidden layers with 200 nodes each and a SoftMax output layer with two nodes. The Adam optimizer is used to minimize the categorical cross-entropy loss. The model is then trained on the pre-processed dataset using the fit() method.

- **Visualizing the Model's Performance:**
  - The graph() function is used to generate a graph of the model's accuracy and loss over the training iterations. This is achieved using the matplotlib library.
- **Predicting on New Data:**
  - The predictProfile() function allows the user to select a new dataset of social network profiles and use the trained neural network to predict whether the profiles are genuine or fake.
- **Creating Buttons:**
  - Finally, the program creates several buttons using the Button() method from tkinter. These buttons are used to trigger the various functions described above.

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Performance metrics:

We also conducted experiments using SGD + Momentum weight updates and found out that it takes too long to cover the entire data set. We ran our model up to 20 epochs after which it began to over fit. Thus, identifying the profile is real or fake. We used sparse vector representation of tweets for training the classifier. We identify that the presence of bigrams features significantly improved the accuracy. The overall accuracy across all machine learning models was very high with the highest being 94.43% using Deep Neural Networks and 94% using Random Forest method and finally 90.01% using Support Vector Machine algorithm. These results are just below what one would expect from getting the prediction right by chance.

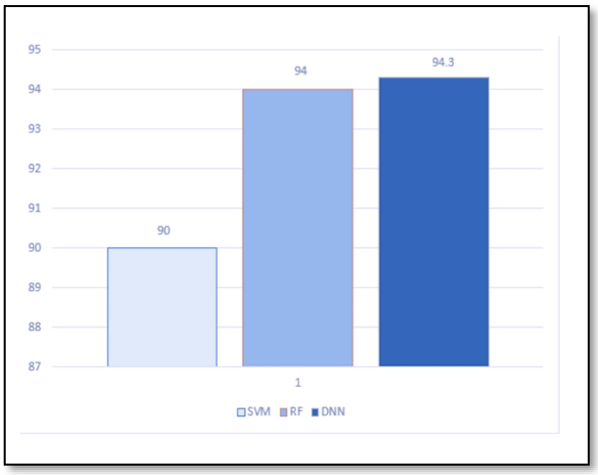
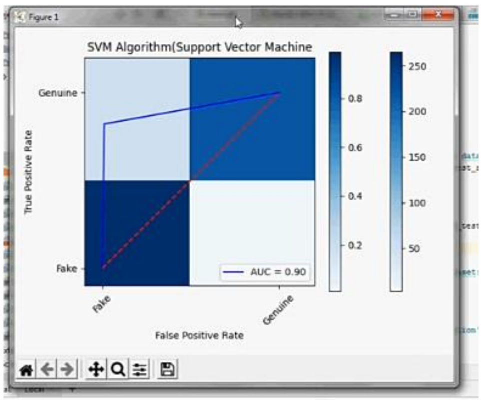
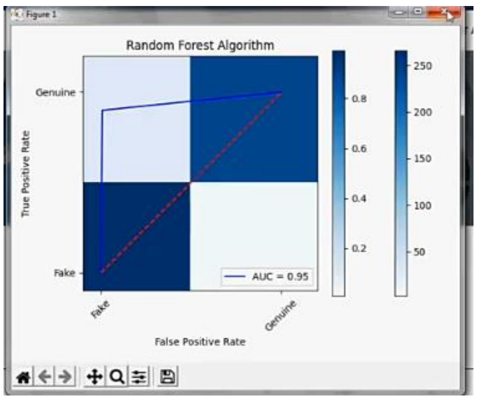
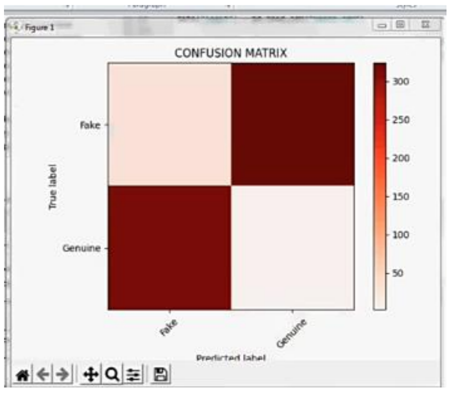


Fig 4.1 Graph of comparison





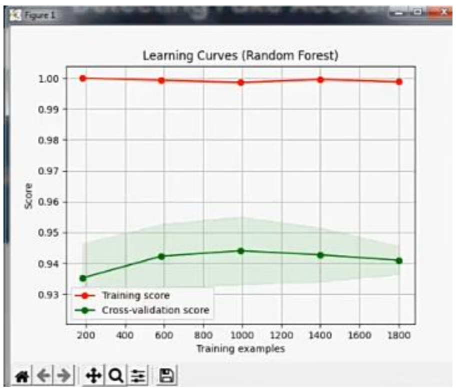


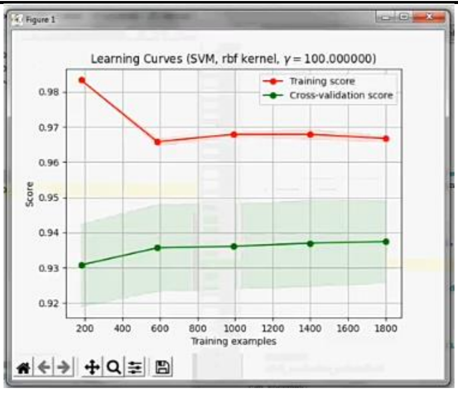
SVM

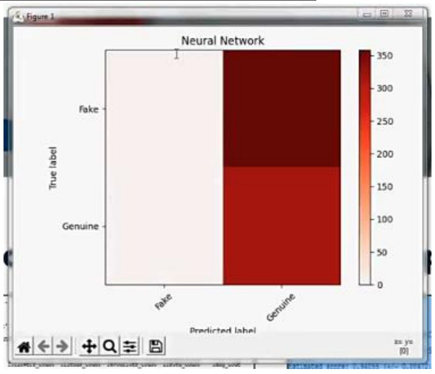
Random forest

Neural Network

Fig 4.2 Metrics of Algorithms







<u>SVM</u> **<u>Random forest</u>** **<u>Neural Network</u>**

Fig 4.3 ROC graphs

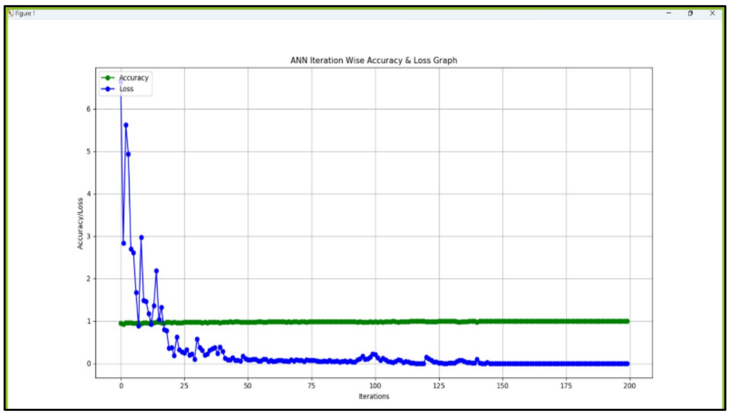
Data Collection:

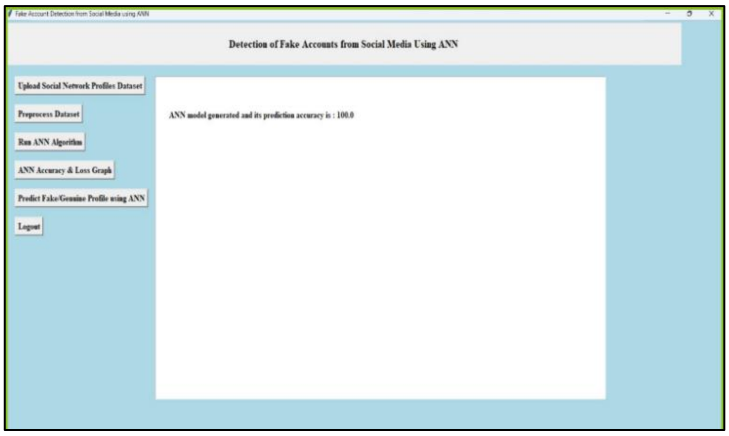
i. Uploading the data.

ii. Dataset pre-processing

iii. Choosing a feature

iv. A method for detecting fraudulent accounts and comparing the results.

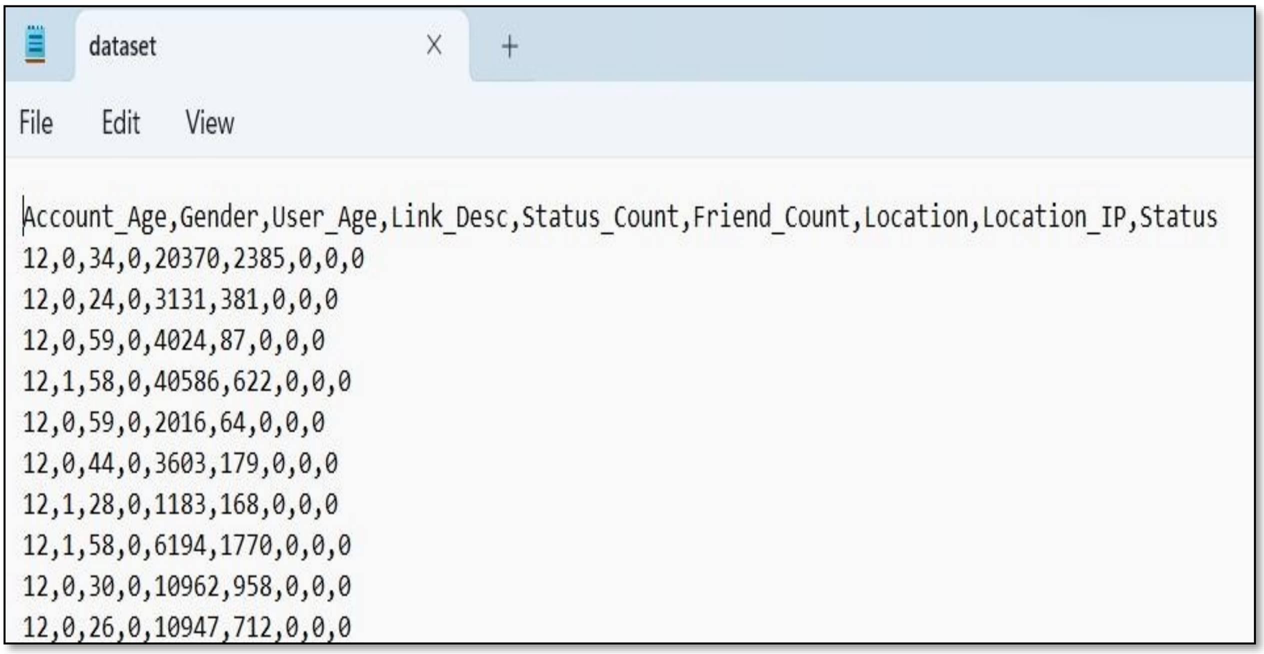



Fake Account Detection from Social Media using ANN
Upload Social Network Profiles Dataset
Preprocess Dataset
Run ANN Algorithm
ANN Accuracy & Loss Graph
Predict Fake/Genuine Profile using ANN
Logout

Detection of Fake Accounts from Social Media Using ANN

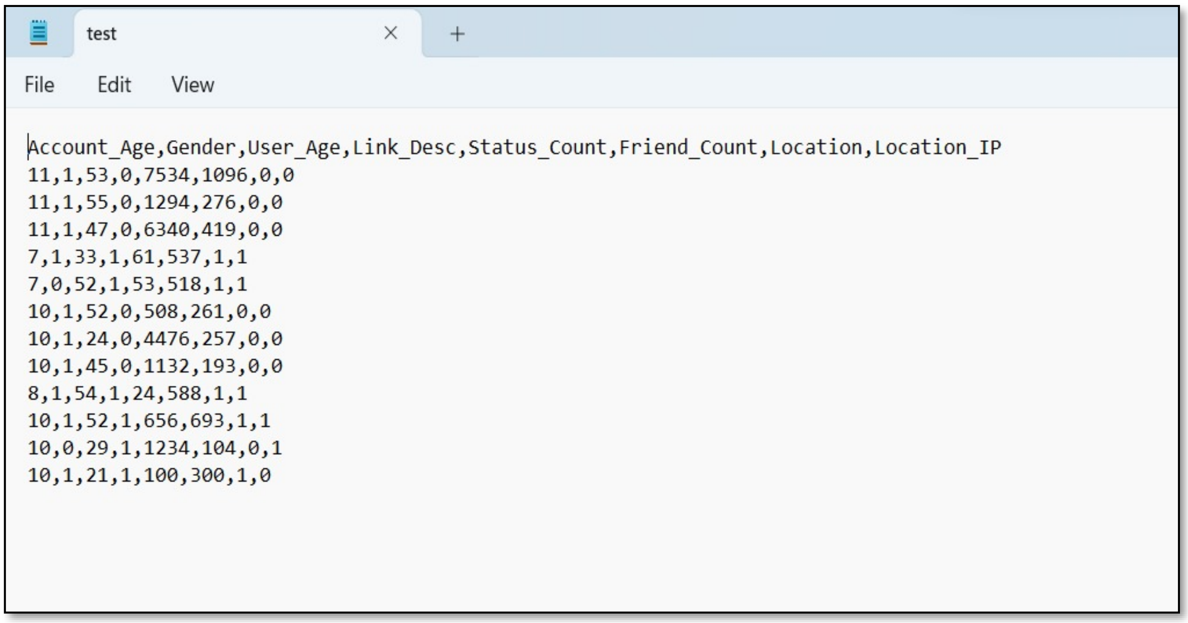
ANN model generated and its prediction accuracy is : 100.0

Fig 4.4 Metrics of ANN



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

Table 4.1 Train Dataset



```
Account_Age,Gender,User_Age,Link_Desc,Status_Count,Friend_Count,Location,Location_IP
11,1,53,0,7534,1096,0,0
11,1,55,0,1294,276,0,0
11,1,47,0,6340,419,0,0
7,1,33,1,61,537,1,1
7,0,52,1,53,518,1,1
10,1,52,0,508,261,0,0
10,1,24,0,4476,257,0,0
10,1,45,0,1132,193,0,0
8,1,54,1,24,588,1,1
10,1,52,1,656,693,1,1
10,0,29,1,1234,104,0,1
10,1,21,1,100,300,1,0
```

Table 4.2 Test Dataset

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION

5.1 Conclusion and Future Enhancement

In this Project we have presented a machine learning pipeline for detecting fake accounts in online social networks. Rather than making a prediction for each individual account, our system classifies clusters of fake accounts to determine whether they have been created by the same actor. Our evaluation on both in-sample and out- of-sample data showed strong performance, and we have used the system in production to find and restrict more than 250,000 accounts. In this work we evaluated our framework on clusters created by simple grouping based on registration date and registration IP address. In future work we expect to run our model on clustering that are created by grouping on other features, such as ISP and other time periods, such as week or month.

We use machine learning, namely an artificial neural network to determine what are the chances that a friend request is authentic are or not. Each equation at each neuron (node) is put through a Sigmoid function. We use a training data set by Facebook or other social networks. This would allow the presented deep learning algorithm to learn the patterns of bot behavior by back propagation, minimizing the final cost function and adjusting each neuron's weight and bias.

FUTURE SCOPE

1. This technique can also be used for other social networking sites such as Facebook or Twitter and LinkedIn with minor changes.
2. The accuracy of the proposed technique can also be improved using different feature selection techniques.
3. Finally, we intend to enrich the dataset further and look forward to observing the results of other elements of the boosting methods.

CHAPTER 6
REFERENCES

CHAPTER 6
REFERENCES

[1]. Estee Van Der Walt and Jan Eloff, "Using Machine Learning to Detect Fake Identities: Bots vs Humans "IEEE Trans. Emerg.TopicsComput. Intell., vol. 1, no. 1, pp. 61–71 March 2018.

[2]. Loredana Caruccio,DomenicoDesiato and Giuseppe Polese"Fake Account Identification in Social Networks" IEEE International Conference on Big Data.,, vol. 9, no. 6, pp. 811–824,2018.

[3]. SarahKhaled, Neamat El-Tazi and Hoda M. O. Mokhtar "Detecting Fake Accounts on Social Media" IEEE International Conference on Big Data.., vol.6 pp 101-110 ,2018.

[4]. SuyashSomani and Somya Jain "Resolving Identities on FaceBook and Twitter" Tenth International Conference on Contemporary Computing (IC3), 10- 12 August 2017.

[5]. FrancescoBuccafurri, Gianluca Lax,Denis Migdal, Serena Nicolazzo, Antonino Nocera and Christophe Rosenberger"Contrasting False Identities in Social Networks by Trust Chains and Biometric Reinforcement " International Conference on Cyberworlds vol 5,2017.

[6]. SuprajaGurajala, Joshua S White, Brian Hudson,Brian R Voter and Jeanna N Matthews"Profile characteristics of fake T witter accounts"Big Data &Society,July–December 2016: 1–13.

[7]. Simranjit. Kaur. Tuteja, ‘‘A survey on classification algorithms for email spam filtering,’’ International Journal Eng. Sci., vol. 6, no. 5, pp. 5937–5940, 2016.

[8]. M.A.Devmane and N.K.Rana "Detection and Prevention of Profile Cloning in Online Social Networks"IEEE International Conference on Recent Advances and Innovations in Engineering,May 09-11, 2014.

[9]. SaraKeretna ,Ahmad Hossny and Doug Creighton "Recognising User Identity in T witter Social Networks via Text Mining"IEEE International Conference on Systems, Man, and Cybernetics,2013.

[10]. MohamedT orky, Ali Meligy and Hani Ibrahim"Recognizing Fake Identities In Online Social Networks Based on a Finite Automaton Computer Approach"International Journal of Applications,2016.

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

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

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

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

[15]. 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.

[16]. 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

[17]. 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.2017.05.0 20.

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

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

[20]. https://stackoverflow.com/questions/40758562/c an-anyone-explain-mestandardscaler

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

[22]. https://www.tutorialspoint.com/python_pandas/index.html

[23]. M. Egele, G. Stringhini, G. Stringhini, and G. Vigna, "Towards Detecting Compromised Accounts on Social Networks," IEEE, vol. 5971, no. c, 2015.

[24]. D. M. Freeman and T. Hwa, "Detecting Clusters of Fake Accounts in Online Social Networks Categories and Subject Descriptors," AISec, 2015.

[25]. K. B. Kansara, "Security against sybil attack in social network," ICICES, no. Icices, 2016.

[26]. A. M. Meligy, "Identity Verification Mechanism for Detecting Fake Profiles in Online Social Networks," IJCNIS, no. January, pp. 31–39, 2017.

[27]. Ashraf Khalil, Hassan Hajjdiab, and Nabeel Al-Qirim , Detecting Fake Followers in Twitter: A Machine Learning Approach, International Journal of Machine Learning and Computing, Vol. 7, No. 6, December 2017.

[28]. S. Khaled, N. El-Tazi and H. M. O. Mokhtar, "Detecting Fake Accounts on Social Media," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 3672-3681.

[29]. Nazir, Atif, Saqib Raza, Chen-Nee Chuah, Burkhard Schipper, and C. A. Davis. "Ghostbusting Facebook: Detecting and Characterizing Phantom Profiles in Online Social Gaming Applications." In WOSN. 2010.

[30]. Adikari, Shalinda, and Kaushik Dutta. "Identifying Fake Profiles in LinkedIn." In PACIS, p. 278. 2014.

[31]. Chu, Zi, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. "Who is tweeting on Twitter: human, bot, or cyborg?." In Proceedings of the 26th annual computer security applications conference, pp. 21-30. ACM, 2010.

[32]. Stringhini, Gianluca, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. "Follow the green: growth and dynamics in twitter follower markets." In Proceedings of the 2013 conference on Internet measurement conference, pp. 163-176. ACM, 2013.

[33]. Thomas, Kurt, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. "Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse." In Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13), pp. 195-210. 2013.

[34]. Farooqi,Gohar Irfan, Emiliano De Cristofaro, Arik Friedman, Guillaume Jourjon, Mohamed Ali Kaafar, M. Zubair Shafiq, and Fareed Zaffar. "Characterizing Seller-Driven Black-Hat Marketplaces." arXiv preprint arXiv:1505.01637 (2015).

[35]. Viswanath, Bimal, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. "Towards detecting anomalous user behavior in online social networks." In 23rd {USENIX} Security Symposium ({USENIX} Security 14), pp. 223-238. 2014.

GitHub Link:

https://github.com/Monu-06/Fake-Account-Detection-using-ANN.git