INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

IDENTIFYING FAKE PROFILES ACROSS SOCIAL NETWORKS USING NEURAL NETWORK

P. Srinivasa Rao[1], Meshak Raaby S J[2], Abhinay Goud B[3], Vishwa Teja K[4]

[1]Associate Professor, CSE Dept, ACE Engineering College, Hyderabad, India.

[2,3,4]Student, CSE Dept, ACE Engineering College, Hyderabad, India

ABSTRACT

Social networking sites such as Facebook, Twitter, Instagram, etc. are extremely famous among people. Users always interact with their friends via these social network's sites or media. They share their personal and public information using these social networks. An immense number of people use social networking sites due to their attractiveness. In our proposed plan, we propose machine learning techniques such as Neural Networks and SVM for detecting fake accounts on Facebook, Twitter, or Instagram. Different data mining tools have been used for the simulation of the algorithm and the obtained results are presented by the proposed plan.

1. INTRODUCTION

Long range interpersonal communication has end up a notable diversion inside the web as of now, drawing in countless clients, burning through billions of minutes on such administrations. Online Social organization (OSN) administrations assortment from social cooperations based stages like Facebook or My Space, to understanding spread driven stages suggestive of twitter or Google Buzz, to social association trademark brought to introduce frameworks like Flicker.

The contrary hand, improving security concerns and safeguarding the OSN privateness actually connote a most significant bottleneck and saw mission. Having our singular expertise totally or to some degree uncovered to the overall population, makes us amazing focuses for exceptional kinds of attacks, the most exceedingly terrible of which could be ID burglary.

Ratings drives online account holders to understand newer approaches not naturally or manually to compete more with their neighbours. By these analogies, the maximum famous candidate in an election commonly get more number of votes. Happening of fake social media accounts and interests may be known. Instance is fake online account being sold on-line at a online market places for minimum price, brought from collaborative working offerings. More often feasible to have Twitter fans and Facebook media likes in online.

2. OBJECTIVES

In our project to identify fake profiles across social networks using neural networks, there are five main objectives:

- Develop a User-Friendly Interface
- Implement Automation Features
- Utilize Advanced Neural Network Architectures
- Ensure Security and Reliability
- Test and Validate the Model

3. METHODOLOGY

In developing our neural network-based system for identifying fake profiles, we begin by gathering data from various social networks to understand user behaviors and patterns. This data is then cleaned and organized to ensure accuracy. We design features to capture key aspects of profile authenticity and create a tailored neural network architecture for our model. Through training and optimization using labeled datasets, we fine-tune the system for real-world use. We assess its performance and make improvements before deploying it. Continuous monitoring and adaptation ensure it remains effective over time, meeting the evolving challenges of fake profile detection.

4. LITERATURE SURVEY

Title: Fake profile detection technique in large scale

Year: 2018

Author: Ramalingam D and Chinnaiah

Description: It emphasizes the importance of early detection of fake profiles and mentions the plethora of algorithms and methods proposed for this purpose, presenting a survey of existing and recent technical efforts in fake profile detection.

Purpose: It highlights the urgency of detecting fake profiles preemptively due to the significant harm they can cause to users.

Title: Spam profile detection in social networks based on public features

Year: 2017

Author: Ala'M, Al-Zoubi, Ja'far Alquatawna and Hossam Faris

escription: It describes a study focused on improving spam detection on Twitter through the development and evaluation of detection models using feature engineering and classification algorithms, showcasing promising preliminary results. Purpose: It outlines a study aiming to enhance social spam detection on Twitter by analyzing publicly available features and employing various classification algorithms, showing promising results in preliminary experiments.

Title: Clustering analysis of network for protocol and structure independent botnet detection

Year: 2016

Author: G.Gu, Perdisci R, Zhang and Lee W

Description: Introduces a general botnet detection framework that is protocol and structure agnostic, requiring no prior knowledge of botnet specifics. It utilizes clustering and cross-cluster correlation of communication and malicious activity patterns to effectively identify bot-infected hosts with minimal false positives.

Purpose: Implemented as the BotMiner prototype, it demonstrates capability in detecting various real-world botnets with minimal false positives.

5. PROPOSED SYSYTEM

- This proposed work uses the techniques like neural networks i.e. NN and a support vector machine called SVM for the classification of real and fake accounts.
- The machine learning techniques are neural network and support vector machine that provides accurate results.

6. HARDWARE AND SOFTWARE REQUIREMENTS

6.1 HARDWARE REQUIREMENTS:

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 512 Mb.

6.2 SOFTWARE REQUIREMENTS:

- Web technology: HTML, CSS, Java scripts, AJAX
- Python Web Server: Django
- Programming Language: Python
- Algorithm: Deep Neural Network
- Database: MYSQL

7. PACKAGES USED

TensorFlow

TensorFlow is a popular open-source Python machine learning toolkit for creating and training deep neural networks. It has a versatile architecture and supports a variety of platforms, including CPU, GPU, and TPU. TensorFlow simplifies the implementation of complicated algorithms and models, allowing developers to create scalable and efficient machine learning systems.

Keras

Keras is a Python-based high-level neural network API that operates on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. It offers an easy-to-use interface for building and training deep learning models, letting users to easily experiment with alternative architectures and hyperparameters. Keras also provides pre-trained models as well as a huge collection of building blocks for developing sophisticated models.

Scikit-learn

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Scipy

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

Scipy is a Python package for scientific and engineering computing. It includes modules for optimization, integration, linear algebra, signal processing, and other tasks. Scipy is built on top of Numpy, another famous Python package for scientific computing, and the two combined constitute a strong data analysis and numerical calculation tool.

Numpy

NumPy is an important Python package for scientific computation. It supports huge, multidimensional arrays and matrices, as well as a diverse collection of high-level mathematical operations for these arrays. NumPy is a popular choice for numerical operations in scientific research and data analysis due to its efficient and user-friendly interface.

Pandas

Pandas is a popular open-source Python data analysis and manipulation package. It offers sophisticated data structures and tools for working with structured data, including as data frames and series, and it allows for quick data processing, cleaning, merging, and reshaping. Pandas also supports reading and writing a variety of file types, including CSV, Excel, and SQL databases.

Matplotlib

Matplotlib is a popular Python data visualization package. It includes line graphs, scatter plots, bar plots, and histograms among its 2D and 3D displays. Matplotlib is a useful tool for data exploration and communication since it is extremely customizable and supports extensive labelling, annotations, and text formatting.

Tkinter and NLKT

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications. NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers.

8. TECHNOLOGY DESCRIPTION

Python is an interpreted high-level programming language that is simple to learn and use. It features a basic and clear syntax that makes it suitable for both beginners and professionals. Python is utilized in many different areas, such as web development, scientific computing, data analysis, and artificial intelligence.

9. SOURCE 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 keras.optimizers import Adam
from keras.utils.np_utils import to_categorical
main = Tk()
main.title("Identifying of Fake Profiles Across Online Social Networks Using Neural Network")
main.geometry("1300x1200")
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\nDataset contains total profiles : "+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']
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

```
acc = acc[199] * 100
outputarea.insert(END,"ANN model generated and its prediction accuracy is : "+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)
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])+" "+msg+"\n\n")
def close():
main.destroy()
font = ('times', 15, 'bold')
title = Label(main, text='Identifying of Fake Profiles Across Online Social Networks Using Neural Network')
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)
```

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

```
annButton = Button(main, text="Build ANN", command=executeANN)
annButton.place(x=20,y=200)
annButton.config(font=ff)
graphButton = Button(main, text="Accuracy & Loss Graph", command=graph)
graphButton.place(x=20,y=250)
graphButton.config(font=ff)
predictButton = Button(main, text="Predict Fake/Genuine Profile", 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()
```

10. OUTPUT

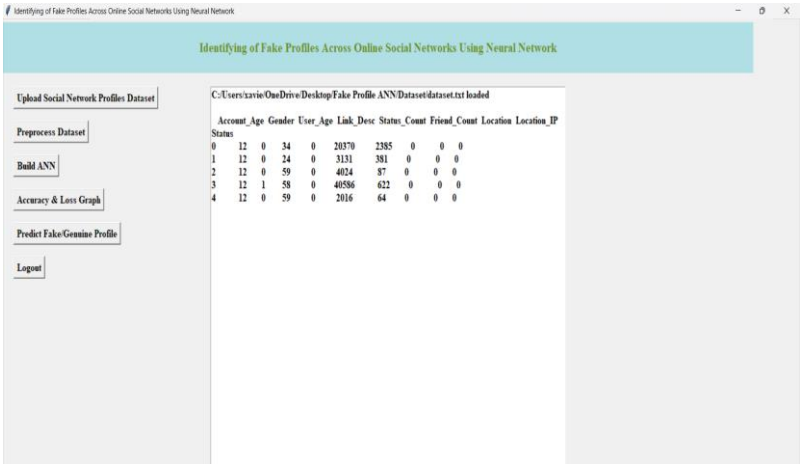

Fig:10.1 Data sets

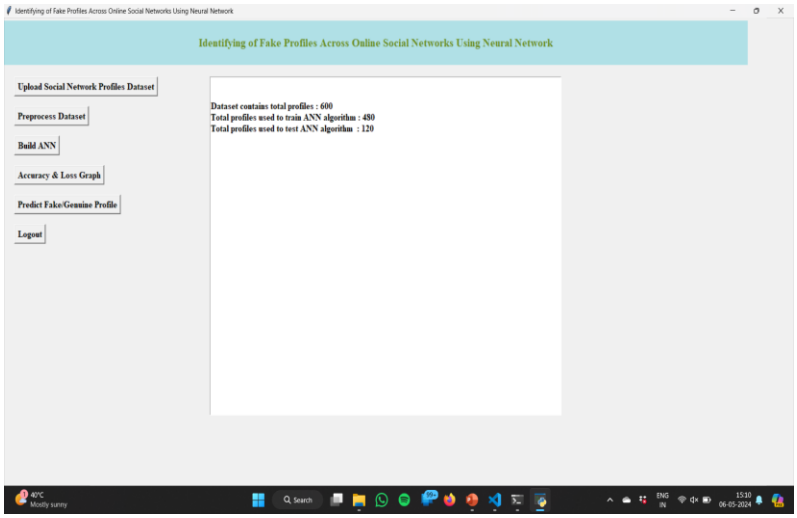

Fig 10.2 Total number of data sets

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

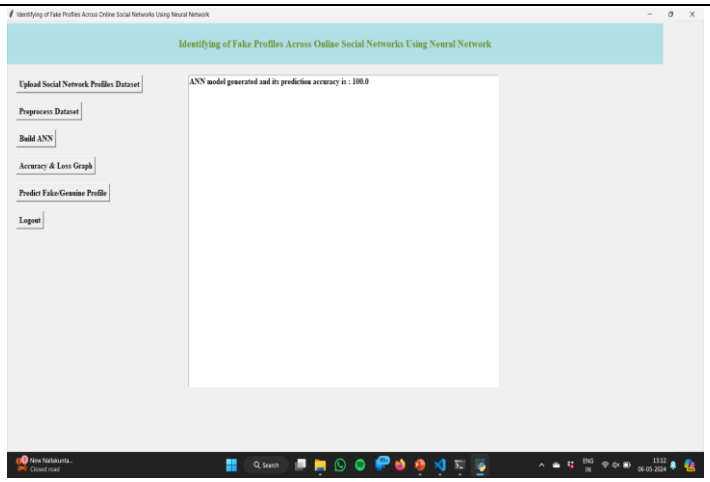

Fig:10.3 Accuracy

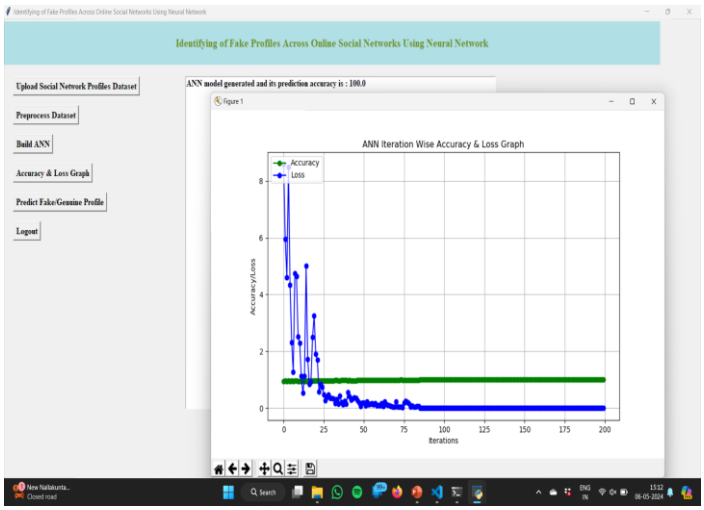

Fig:10.4 Loss Graph

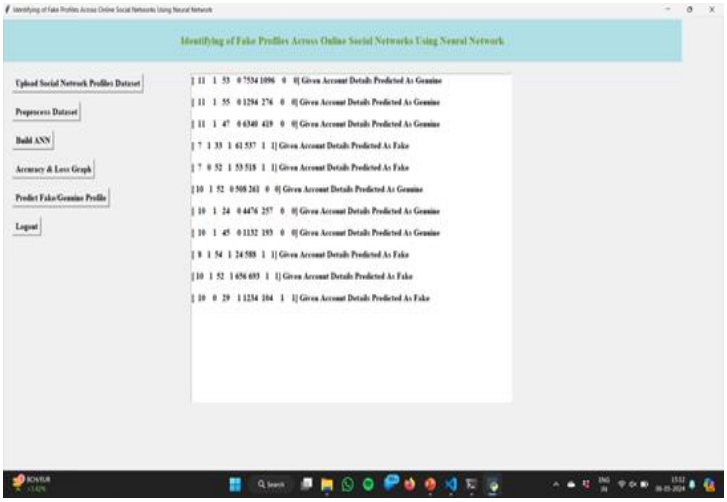

Fig 10.5: Fake profiles

11. CONCLUSION

In this paper, 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 backpropagation, minimizing the final cost function and adjusting each neuron's weight and bias. In this paper, we outline the classes and libraries involved. We also discuss the sigmoid function and how are the weights determined and used. We also consider the parameters of the social network page which are the most important to our solution

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

e-ISSN : 2583-1062

Impact Factor: 5.725

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1754-1761

12. FUTURE SCOPE

Each input neuron would be a different, previously chosen feature of each profile converted into a numerical value (e.g., gender as a binary number, female 0 and male 1) and if needed, divided by an arbitrary number (e.g., age is always divided by 100) to minimize one feature having more influence on the result than the other. The neurons represent nodes. Each node would be responsible for exactly one decision-making process

13. REFERENCES

[1] Ramalingam, D. and Chinnaiah, V. Fake profile detection techniques in large-scale online social networks: A comprehensive review. Computers & Electrical Engineering, 2018.

[2] Ala'M, Al-Zoubi, Ja'far Alqatawna, and Hossam Faris. "Spam profile detection in social networks based on public features." 8th International Conference on information and Communication Systems (ICICS). IEEE, 2017.

[3] G. Gu, R. Perdisci, J. Zhang, and W. Lee, ``BotMiner: Clustering analysis of network traf c for protocol-and structure- independent botnet detection," in Proc. USENIX Secure. Symp., vol. 5. 2008