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("IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY USING NEURAL NETWORK")

main.geometry("1300x1200")

global filename

global X, Y

global X\_train, X\_test, y\_train, y\_test

global accuracy

global dataset

globalmodel

1

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

1

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 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()

1

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(test)

classes = np.argmax(predict,axis =1)

print(predict)

for i in range(len(test)):

msg = ''

if str(classes[i]) == '0':

msg = "Given Account Details Predicted As Genuine"

if str(classes[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='IDENTIFING OF FAKE PROFILES ACROSS ONLINE SOCIAL NETWORKS BY 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')

1

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()