USER VERIFICATION USING KEYSTROKE DYNAMICS

BCI3006 – BIOMETRICS

PROJECT REPORT BY (TEAM 12)

1.Sanskriti Bansal (20BCE2634)

2.Mansha Gandhi (20BCI0049)

3.Megha Shashidhar (20BDS0376)

4.Dhairya Lunia (20BCE2178)

5.Shriram (20BCI0250)

ABSTRACT

Computing and communication systems have improved our way of life, but have also contributed to an increased data exposure and, consequently, to identity theft. A possible way to overcome this issue is by the use of biometric technologies for user authentication. Among the possible technologies to be analysed, this work focuses on keystroke dynamics, which attempts to recognize users by their typing rhythm. In order to guide future researches in this area, we have conducted a systematic review on keystroke dynamics here.

In this project, the training and testing data is the secondary data from CMU Keystroke Dynamics Benchmark Data set. We have verified the identity of users on the basis of their keystroke information. A model has been first trained by providing it with the typing patterns of the users to be enrolled, multiple patterns per user. The model is then provided with test patterns from the user as well as others posing as that user.

For the purpose, 300 inputs of each user are taken as genuine inputs for training our model. Rest 100 inputs of genuine user and 5 inputs from rest other users are taken as imposter inputs for testing our model.

The model rejects the imposters while accepting the genuine user based on the test pattern's similar to the trained model for the user. We have tested various detector algorithms which provide different ways of measuring this similarity and bring a comparison of all these algorithms based on their equal error rate (EER).

INTRODUCTION

Keystroke dynamics or typing biometrics refers to the automated method of identifying or confirming the identity of an individual based on the manner and the rhythm of typing on a keyboard. This is a behavioural biometric.

ORIGIN

The concept of Keystroke dynamic was introduced as late as World War II. The military transmitted messages through Morse Code. Using a methodology called "The Fist of the Sender", Military Intelligence identified that an individual had a unique way of keying in a message's "dots" and "dashes", creating a rhythm that could help distinguish ally from enemy.

WHAT MAKES IT DIFFERENT

Keystroke dynamics represent the latest in biometric authentication. Unlike other forms of biometric such as fingerprint scanning, iris scanning, and facial recognition, keystroke dynamics don't require an active input. Instead, it analyses the typing patterns of users; this can include typing rhythms, frequent mistakes, which shift keys they use for capitalization and pace.

So, instead of considering what words users type, the manner in which they type becomes of special interest. Thus, this depends on the behavioural biometrics of the person.

INDIVIDUAL CONTRIBUTIONS

1. Megha Shashidhar

- Introduction and Origin of Keystroke Dynamics
- Analysis of the requirements for user verification based on it-Hardware and Software
- National Survey- Keystroke Dynamics for Mobile Phones: A Survey Baljit Singh Saini 1,2, Navdeep Kaur1 and Kamaljit Singh Bhatia
- International Survey- A Survey on Keystroke Dynamics Biometrics: Approaches, Advances, and Evaluations Yu Zhong and Yunbin Deng
- Gaussian Mixture Model Algorithm- Code, errors and analysis
- Studied article- Keystroke Dynamics Analysis and Prediction by Karthik Shenoy

2. Dhairya Lunia

- Involved in analysing and interpreting the dataset that we got.
- Involved in loading and formatting of dataset
- Establishing Equal error rate (EER) algorithm code in python
- To analyse the performance/efficiency of all the keystroke dynamics algorithms that we have used using EER and ROC Curve
- Finding the objectives of our project and why Keystroke Dynamics
- To find out how it is implemented and the right way to use it by referring startups like
  1) keystrac
  2) Behaviosec

3. Sanskriti Bansal

- Structured layout of the project
- Studied 4 algorithms for the project
- Did code part of 4 algorithms (Manhattan detector, Manhattan scaled detector, Manhattan filtered detector, One class SVM model) in jupyter notebook
- Found dataset and did dataset analysis as well as interpretation by visualizing it in python matplotlib library
- Read 2 research papers based on keystroke dynamics
  1) A study and analysis of keystroke dynamics and its enhancement for proficient user authentication
  2) Study on continuous authentication using a combination of keystroke and mouse biometrics
- defined working and flow of our project
- Interpreted the results and final outcome of our project by comparing equal error rate (EER) of each algorithm
- Did project report creation

4. Shriram A

- Did required research on the feasibility of keystroke dynamics by looking out for advantages and disadvantages of working with this technology rather than adopting some other biometric technology.
- Studied Open SVM algorithm and worked on the code part but couldn't execute it because of syntax errors which were later fixed with the help of Sanskriti Bansal.
- Gathered information required to structure the project and found applications of keystroke in real life based on TypingDNA, IDcontrol startups
- Helped in making the final report.

5. Mansha Gandhi

- Manhattan filtered detector algorithm
- Gathered info on startup TypingDNA
- Tried implementing in django application but unable to do so as keystroke dynamics is a behavioral biometric, requires several attributes and sample data to implement
- Did ppt compilation for our project
- Finding the objectives of our project and why keystroke dynamics
- Studied article- Keystroke Dynamics Analysis and Prediction by Karthik Shenoy

DATASET

Dataset has 34 columns

Each row is timing data noted for each typist in a session

Column 1: Subject to uniquely identify them we use 51 subjects

Column 2: Session index to tell in which session this reading has been taken from 1 to 8

Column 3: ‘rep’ column name tells the nth time subject is typing password in each session. (For 1 session rep varies from 1 to 50 telling 50 times reading is taken)

Column 4 – 31: Gives timing readings as follows:

H for pressure or hold on key.

DD for keydown- keydown.

UD for keyup- keydown

Example: H.i tells time taken in pressing key ‘i’, DD.t.i and UD.t.i tell keydown-keydown and keyup- keydown time taken in pressing key from ‘t’ to ‘i’

Password used in the dataset: .tie5Roanl

DATASET LINK

http://www.cs.cmu.edu/~keystroke/

MODULES USED

- from scipy.spatial.distance import cityblock
  This is used to calculate the Manhattan distance

- import numpy as np

- np.set_printoptions(suppress = True)
  This determines the way floating point numbers, arrays and other NumPy objects are displayed.

- import pandas as pd
  This is used to load the dataset in csv file format

- from sklearn import metrics
  This implements several loss, score, and utility functions to measure classification performance

- from scipy.spatial.distance import Euclidean
  this is used to calculate the eucledian distance in Manhattan filtered detector
- from sklearn.svm import OneClassSVM

- from sklearn.mixture import GaussianMixture

- import warnings
- warnings. filterwarnings('ignore')
  This is used to suppress the warnings in python jupyter notebook

- import matplotlib.pyplot as plt
  this is used to plot and visualize the dataset

ERRORS

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-1-0128d74043a5> in <module>
      1 #keystroke_GMM.py
      2
----> 3 from sklearn.mixture import GMM
      4 import warnings
      5 warnings.filterwarnings("ignore")

ImportError: cannot import name 'GMM' from 'sklearn.mixture' (C:\Users\megha\anaconda3\lib\site-packages\sklearn\mixture\__i
nit__.py)
```

```
    559                 raise ValueError("Validation should be done on X, y or both.")
    560             elif not no_val_X and no_val_y:
--> 561                 X = check_array(X, **check_params)
    562                 out = X
    563             elif no_val_X and not no_val_y:

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in check_array(array, accept_sparse, accept
_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_
min_features, estimator)
    759                 # If input is 1D raise error
    760                 if array.ndim == 1:
--> 761                     raise ValueError(
    762                         "Expected 2D array, got 1D array instead:\narray={}.\n"
    763                         "Reshape your data either using array.reshape(-1, 1) if "

ValueError: Expected 2D array, got 1D array instead:
array=[ 0.1016  0.287   0.1854  0.0718  0.1214  0.0496  0.0576  0.1252  0.0676
  0.081   0.6911  0.6101  0.0694  0.3592  0.2898  0.1212  0.1711  0.0499
  0.0784  0.1569  0.0785  0.1096  0.0864 -0.0232  0.08    0.1993  0.1193
  0.0845  0.2359  0.1514  0.0766].
Reshape your data either using array.reshape(-1, 1) if your data has a single feature or array.reshap
e(1, -1) if it contains a single sample.
```

After using reshape function, below error message showed up:

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-84b56cb53ef9> in <module>
     72 subjects = data["subject"].unique()
     73 print ("average EER for Manhattan filtered detector: ")
---> 74 print(ManhattanFilteredDetector(subjects).evaluate())

<ipython-input-7-84b56cb53ef9> in evaluate(self)
     64
     65                 self.training()
---> 66                 self.testing()
     67                 eers.append(evaluateEER(self.user_scores, self.imposter_scores))
     68             return np.mean(eers)

<ipython-input-7-84b56cb53ef9> in testing(self)
     37         def testing(self):
     38             for i in range(self.test_genuine.shape[0]):
---> 39                 j = self.test_genuine.iloc[i].values
     40                 cur_score = self.gmm.score(j)
     41                 self.user_scores.append(cur_score)

AttributeError: 'numpy.ndarray' object has no attribute 'iloc'
```

RESULT

Algorithm	Equal Error Rate (EER)
Manhattan Detector	0.158807700030822
Manhattan Filtered Detector	0.13441161913402994
Manhattan Scaled Detector	0.10932574138929727
One Class SVM	0.10818661068682427

The above table shows the Equal Error Rates (EER) found for each algorithm. It was noticed that EER declined considerably from Manhattan detector to One Class SVM model. By calculating cityblock distance in Manhattan detector, EER obtained is approximately 0.1588. After filtering out dataset by removing the outliers using standard deviation matrix in Manhattan filtered detector, EER obtained is approximately 0.1344. In Manhattan scaled detector, the values were scaled using mean absolute deviation vector after which EER reduced to approximately 0.1093. Finally, after implementation of One Class SVM model, EER obtained is approximately 0.1082 which is the lowest value achieved in this project.

Hence, we have found that keystroke biometric has a high potential and scope in the future as it can be used in 2 – user verification system.

REFERENCES

Research Papers:

- https://www.researchgate.net/publication/273382241_A_Study_And_Analysis_of_Keystroke_Dynamics_And_Its_Enhancement_For_Proficient_User_Authentication
- https://www.researchgate.net/publication/310733212_A_study_on_Continuous_Authentication_using_a_combination_of_Keystroke_and_Mouse_Biometrics

ROC Curve and EER:

- https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
- https://stackabuse.com/understanding-roc-curves-with-python/
- https://stackoverflow.com/questions/28703395/how-to-compute-plot-equal-error-rate-eer-from-far-frr-values-using-matlab

Start-ups:

- https://www.keytrac.net/en/
- https://www.behaviosec.com/
- https://www.typingdna.com/

Article:

- https://towardsdatascience.com/keystroke-dynamics-analysis-and-prediction-part-1-eda-3fe2d25bac04

Code:

- https://appliedmachinelearning.blog/2017/07/26/user-verification-based-on-keystroke-dynamics-python-code/

Surveys:

- National: https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2016/Issue-6/Article6.pdf
- International: https://www.researchgate.net/publication/271384778_A_Survey_on_Keystroke_Dynamics_Biometrics_Approaches_Advances_and_Evaluations