National College of Ireland

User Authentication Based on the Keystroke Dynamics using Multi-Layer Perceptron

MSc Academic
MSc in Cyber Security

Lakshmi Bhargav Jetti
Student ID: X20113986

School of Computing
National College of Ireland

Supervisor: Prof. Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lakshmi Bhargav Jetti
Student ID:	X20113986
Programme:	MSc in Cyber Security **Year:** 2020-2021
Module:	Academic Internship
Supervisor:	Prof. Ross Spelman
Submission Due Date:	16/08/2021
Project Title:	User Authentication Based on Keystroke Dynamics using Multi-Layer Perceptron

Word Count:4131 **Page Count: 23**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Lakshmi Bhargav Jetti

Date: 16/08/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

User Authentication Based on the Keystroke Dynamics using Multi-Layer Perceptron

Lakshmi Bhargav Jetti
X20113986

Abstract

In today's age the use of computers has increased exponentially with a variety of online applications for public usage such as e-commerce, blogs, bank services etc. All of these existing online platforms require an authentication process to ensure the individuality of a genuine user. Despite of high-end encryption, it has become easy for intruders to enter the system and cause harm. Hence, arises the necessity of biometrics. One such example of biometrics is Keystroke dynamics which is known for its precise access control authentication process. The concept is known for its behavioral biometrics that makes use of typing patterns to analyze and gain insights of a user accessing the system. A total of 51 users are collected and asked to type a static password over a set of sessions repeated iteratively. Finally, the detection between a legit user and an intruder is being made based on the typing rhythms so obtained. The presented thesis directs its focus on the usage of deep learning model using the fundamentals of a Multilayer Perceptron to execute the working of keystroke dynamics.

Keywords: *biometrics, multi-layer perceptron, keystroke dynamics, verification, user authentication.*

1. Introduction

Our dependence on computers and digital platforms has been observed to be overwhelmingly increased to simplify our lives. The use of such automated information systems together has resulted in improved performance of the available networking services in the form of reliability and computational costs. With such efficient utilization the advances in technology have generated a collective interest in global access to such online platforms. However, at the same time there is a rise in the threats regarding to the security of computers. Usage of advanced methodologies to safeguard the system from attacks and frauds come under the topmost concerning priority of many research scientists. Hence there is a need to generate foolproof measures to prevent such unauthorized access is being worked upon. One such preventive method to give access to individuals by detecting their unique and behavioral pattern is an individual's typing rhythm. This unique typing rhythm tends to become a natural choice for security of computers and is commonly known as Keystroke Dynamics [1]. It is observed that when a person types; the placement of his fingers, applied pressure on the keys and the regularly typed strings appears to be consistent for a specific individual. Hence this concept is used to differentiate between an intruder and a legit user as they will be typing on the keyboard anyway. Therefore, making such kind of typing rhythms easily accessible to track computer activities.

1.1 Background and Motivation

The initial step to prevent any form of unauthorized control into a system is *user authentication*. This is the only operation that confirms the identity of an individual. This identity is matched with a pre-generated code or a registration number with the legit individual and is used in the process of identification. Once this identification is accomplished some form of indicator is shared to give access control to the user.

The information below depicts the grouping model.

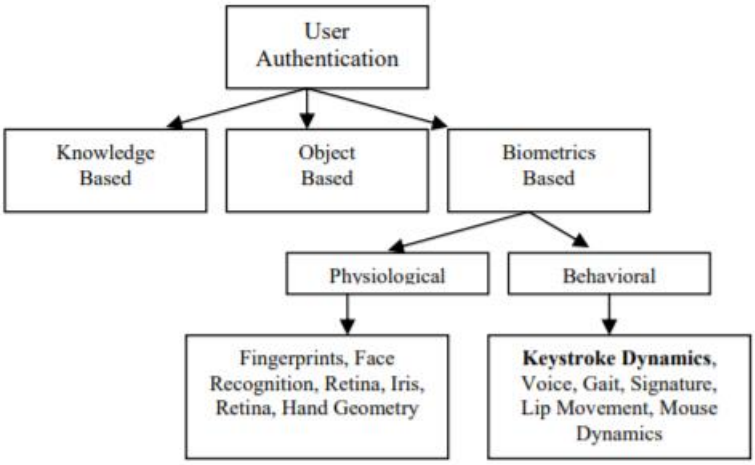


Figure 1.1: Classification of user authentications

Knowledge-based encryption is based on the idea of two people sharing their secrets. Username and Login credentials are two well-known examples. Whereas object-based authorization is characterized based on possession of specific things and relies on the concept of something which someone has. On the other hand, biometric based authentication is characterized by behavioral features an individual would have.

In real time, the classifications of object and knowledge based are merged to fulfil the process of authentications such bank passwords and their PINS. One major disadvantage with regards to this classification-based authentication is the ability to memorize and manage multiple such PINS and recalling them. Therefore, the usage of biometrics authentication is preferred as it overcomes these issues and makes use of automated methods to identify and verify the individual. Also, this form of authentication is gaining worldwide popularity as they provide an extra level of security.

1.2 Problem Statement

One of the major issues or a problem the digital world face is with regards to passwords used. Although, a few organizations do exist that provide secured authentication for users when they login but on the other hand there are still most sites that skip the process of identification leading to major crimes across the globe. Hence the research work in this thesis aims to suggest better security systems that could be analysed using the typing behaviour of individuals using keystroke dynamics. Main emphasis is being put on to detect the typing behaviour of individuals using the concepts of MLP to validate the features of users and give them a more secure and efficient system than ever before.

1.3 Research Questions

Biometrics has developed a space of natural choice for the purpose of identity verification, as it is based on detecting the unique characteristics of individuals such as physical and behavioral traits. It is seemed to be an excellent source of identification because all other forms of recognition methods might fail at some point of time either in the absence of physical damage or the data being lost or stolen. On the other hand, physiological characteristics of an individual such as fingerprints tend to become unique for every individual resulting in a precise authentication for access to systems. As a result, the primary goal of the project is to generate an environment for user credential safeguard and login security. Following are the primary research questions of the thesis:

- What are the benefits and drawbacks of using keystrokes for intrusion detection?
- What personality traits are extracted from the typing data?
- What algorithms are used for classification? What techniques are available in performance evaluation?
- What metrics are used to assess the performance?

The most important feature of this thesis is that Keystroke Dynamics possess the capability of recognizing an individual and encompasses both, physiological and behavioral personality traits. Physical attributes such as fingerprint images possess to have unique features resulting in accurate authentication, on the other hand behavioral characteristics provides Keystroke Dynamic authentications with regards to voice and signature based detection.

2. Literature Review

2.1 Survey of Existing System

Keystroke dynamics is observed to be an emerging field of interest in terms of security that is responsible to validate the authenticity of the system based on users typing rhythm. This authentication process has been overpowering other fields of technology due to the following reasons:

- Easy implementation as only the typing data is required
- Zero hardware requirement
- Low computational cost
- Does not require special permission from the user

Various statistical models have been proposed to implement the studies of keystroke dynamics [3-7]. Classifiers such as [8-12] have been built using machine learning approaches. Apart from this hybrid models are also taken into consideration [13-15] to build the model with accuracy. These models can distinguish and establish a certain pattern of

typing frequency and are also capable to access multitudes of data. The argument of keystroke dynamics is observed to differ in certain studies with regards to neuro-physiological behavior of an individual. This behavior exhibits a user's specific typing pattern [16]. In [17] the concept of keystroke was further classified using MLP and clustering algorithms. Alternatively, [18] designed a system that incorporates the features of applied pressure and typing latency time difference to create and organize a typing pattern for a user. To authenticate individual users, they made use of Artificial Neural Network as classifiers. With an average training time of 0.9094 seconds, the classification rate was 100%. Using the RBFN method, the same concept of pressure and time latency was established in [19]. [20] Made use of MLP with Radial Basis Function and achieved an authentication of 97%. The contribution in [21] achieved an accuracy of 97.5%. [24] Performed an experiment with a 17-digit password, where each individual typed in the password several times in each session and features such as the size of the finger and timing frames were taken into consideration to reduce the error rate. Similarly [25] made use of the time intervals between keystrokes utilizing the concepts of MLP and fundamentals of principal component analysis (PCA). This experimentation resulted in an accuracy of 80%. [26] Examined the dataset of keystrokes without the touchscreen features where the data was collected from Android devices and classification algorithms such as SVM, Naïve Bayes were used. This implementation resulted in a 10% percent increase than a normal authentication process.

2.2 Challenges of Keystroke Dynamics

Since this concept is executed and based on an individual's typing on a keyboard, user validation is exhibited due to the following two reasons:

- Keystroke dynamics is not intrusive
- Since there is no requirement of hardware, it includes zero computational cost

Although other biometric features such as fingerprints and retinas remain fairly compatible over a longer period of time, typing patterns might become erratic. Even considering the fact that all forms of biometrics tend to change over a period of time, typing pattern tends to have small time frame for any changes.

In such a case not only the typing pattern varies and becomes inconsistent but also at times an individual might also experience sweat on his hands after elongated time intervals. This issue often results in pattern contrasts so obtained. Another crucial challenge in establishing typing patterns is the layout of keyboard used and the person's posture if standing or sitting.

Such kind of issues and challenges might further create additional inconsistencies in the output so generated.

Types	Description	Advantages/Disadvantages
Fingerprint	Performs analysis of pattern found on fingertips	Highly accurate method but very costly
Hand Geometry	Evaluates unique hand characteristics, including length of the finger, its surface and thickness	Highly accurate method but very costly
Face Recognition	Analyzes unique facial characteristics of eyes, lips, nose, etc.	Very costly
Voice Recognition	Captures unique voice features, including pitch, frequency and tone	Additional hardware required and Invasive
Retinal Recognition	Identifies unique blood vessel pattern at the rear part of the eye by directing infrared light of low-intensity through pupil	Quite accurate but very costly
Iris Scanning	Analyzes the fleck pattern on the iris, which are on the eye surface.	Expensive
Hand Signature	Analyzes how a user signs the name, including velocity, speed, and pressure.	Additional hardware required and Invasive
Keystroke Dynamics	Analyzes how a user types on the terminal, done by monitoring the keyboard input.	Inexpensive, no extra devices required, non-invasive

Table 2.2: Summary of biometric challenges

2.3 Existing State of Keystroke Dynamics

The method of keystroke verification follows to be either static or continuous. Static approach comes into picture to analyze features only at particular times such as the login progression. This approach however tends to be more robust in verification process but unfortunately does not provide continuous security to the user. It is not capable of recognizing the user's substitution after its initial verification. Continuous verification, on the other hand, monitors the user's typing behavior throughout the interaction. [22] Was the first to investigate the working of keystrokes to establish a habitual pattern in typing behaviors for identification. The experimentation in [22] was implemented by using a small population of dataset under the hypothesis of T-TEST to induce a digraph in various sessions. Similar examinations were conducted by [23] were developers made use of continuous approach for verification.

3. Research Methodology

In the thesis presented, a classification model is built and generated to differ between a legit user and an imposter. The methodology is implemented using the concepts of deep learning and MLP. Further the model so created is evaluated using the keystroke dataset.

3.1 Keystroke Analysis Approach

Various research has always been into existence [27-29] since the evolution of keystroke dynamics. However, statistical, and neural network techniques are the two main approaches to classify the verification process. In some cases, a combination of the two may

be used. Some of the approaches are hybrids of the two. The statistical method equates a reference set of typing attributes of one user to a test set of typing characteristics of the same user or a hacker. The difference between these two sets must be less than a certain criterion, or else the user will be labelled as an intruder. The Neural Networks workflow begins with the creation of a forecasting model from historical data, which is then used to predict the consequences of a new trial. Even though the approaches used in the studies differ, from what keystroke details make use of the pattern classification techniques, all have strived to solve the problem of providing a strong and low-cost alternative.

The table below depicts the research methodologies used.

Study	Classification Technique		Users	FAR (%)	FRR (%)
Joyce & Gupta (1990) [16]	Static	Statistical	33	0.25	16.36
Leggett et al. (1991) [18]	Dynamic	Statistical	36	12.8	11.1
Brown & Rogers (1993) [6]	Static	Neural Network	25	0	12.0
Bleha & Obaidat (1993) [27]	Static	Neural Network	24	8	9
Napier et al (1995) [23]	Dynamic	Statistical	24	3.8 (Combined)	
Obaidat & Sadoun (1997) [19]	Static	Statistical	15	0.7	1.9
		Neural Network		0	0
Monrose& Rubin (1999) [22]	Static	Statistical	63	7.9 (Combined)	
Cho et al. (2000) [7]	Static	Neural Network	21	0	1
Ord & Furnell (2000) [25]	Static	Neural Network	14	9.9	30
Bergadano et al. (2002) [5]	Static	Statistical	154	0.01	4
Guven & Sogukpinar(2003) [13]	Static	Statistical	12	1	10.7
Sogukpinar & Yalcin(2004) [28]	Static	Statistical	40	0.6	60
Dowland & Furnell (2004) [9]	Dynamic	Neural Network	35	4.9	0
Yu & Cho (2004) [10]	Static	Neural Network	21	0	3.69
Gunetti & Picardi (2005) [12]	Static	Neural Network	205	0.005	5
Clarke & Furnell (2007) [8]	Static	Neural Network	32	5 (Equal Error Rate)	
Lee and Cho (2007) [14]	Static	Neural Network	21	0.43 (Average Integrated Errors)	
Pin shen The et al (2008) [27]	Static	Statistical	50	6.36 (Equal Error Rate)	

Table 3.1: Approaches in keystroke analysis

3.2 Deep Learning

Usages of deep learning in keystrokes offer multiple advantages in multi-tasking and feature selection properties. This is primarily achieved due to the structure and the featured characteristics of representation of neurons. Hence it not only enhances the system but also increases the efficiency to a certain extent. Thus, it is said that the usage of deep learning optimizes the parameters to improve the presentation of the thesis. The concept of multi-layer perceptron in deep learning refers to a networking layer consisting of neurons connected to each other in a cyclic manner. In general input, hidden and output are the three major layers of this network. The input layer is responsible to receive input values from the training set of data, assigns a weight to them, and forwards them to the hidden layer. It is in this layer that mathematical calculations are performed, and an output is generated to predict the values of

the training tuple. This output is embedded with an activation function and is further transformed using matrix multiplication.

4. Implementation

With an effort to provide the users an efficient and a fool proof method for verifying an individual's identity, this thesis makes use of keystroke analysis [30]. The design of the proposed typing recognition system traditionally involves *representation, classification, and extraction.* In cases when the output is in the form of a set of real numbers, it is often liable to create a vector pattern following the concepts of Euclidean space. Hence the pattern so generated is *represented* using input data measurement characteristics. Next, the characteristic features of input data are reduced in terms of dimensionality using features of *extraction.* The resulting pattern vector is then used for processing the model in the later stages. Finally, *classification and identification* procedures are performed to yield optimum decisions. Once the data from the extracted patterns are recognized, they are further expressed as measurement vectors. The figure below gives an overview of the authentication system.

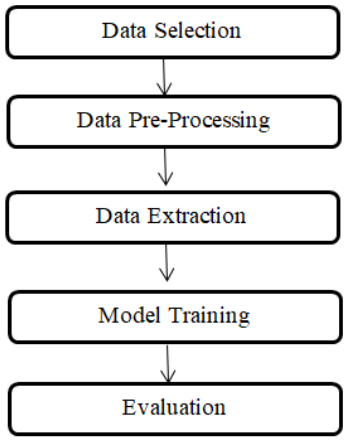


Figure 4: Specifications of the proposed system

4.1 Data Selection

The dataset chosen for this thesis' implementation consists of keystroke system based on the information accumulated from 51 users; wherein each user was given a task of typing a protected encryption key (. tie5Roanl) 400 times over the course of 8 sessions (50 repetitions per session), yielding a total of 34 parameters and 20400 inferences. The password was chosen in such a manner that it was a representative of a strong characteristic of 10

words. The dataset so obtained for keystrokes sets a CMU benchmark [35], consists of keystroke parametric functions (H, DD and UD) collected from 51 users.

The dataset consists of 34 columns in all, with first three columns termed as subject_ID, session, and repetition and the remaining 31 columns used to describe the timing data and information for the password. This timing information includes 3 types of data namely:

- (H): represents the time frame from pressing the key to the time frame of releasing it and is known as Hold Time.
- (DD): represents the time frame from when key1 was pressed to when key2 was pressed and is commonly called as keydown-keydown.
- (UD): represents the time frame to when key1 was released to the time frame when key2 was pressed and is known as keyup-keydown.

In this dataset of 51 users with 400 records 70% of the data has been made use for training and the rest was used for the purpose of testing. The description of the dataset used is given in the Table below.

No.	Variables	Details
1	Subject	Subject ID or class label for 51 users involved in typing task.
2	sessionIndex	A number of the session in the typing task; consists of 8 sessions in total.
3	Rep	A number of repetition in the typing task; consists of 50 repetitions for each session.
4	H.period	The duration between pressing and releasing '.' key.
5	DD.period.t	The duration between pressing '.' key and pressing 't' key.
6	UD.period.t	The duration between releasing '.' key and pressing 't' key.
7	H.t	The duration between pressing and releasing 't' key.
8	DD.t.i	The duration between pressing 't' key and pressing 'i' key.
9	UD.t.i	The duration between releasing 't' key and pressing 'i' key.
10	H.i	The duration between pressing and releasing 'i' key.
11	DD.i.e	The duration between pressing 'i' key and pressing 'e' key.
12	UD.i.e	The duration between releasing 'i' key and pressing 'e' key.
13	H.e	The duration between pressing and releasing 'e' key.
14	DD.e.five	The duration between pressing 'e' key and pressing 'five' key.
15	UD.e.five	The duration between releasing 'e' key and pressing 'five' key.
16	H.five	The duration between pressing and releasing '.' key.
17	DD.five.shift.r	The duration between pressing 'five' key and pressing 'shift.r' key.
18	UD.five.shift.r	The duration between releasing 'five' key and pressing 'shift.r' key.
19	H.shift.r	The duration between pressing and releasing 'r' key.
20	DD.shift.r.o	The duration between pressing 'shift.r' key and pressing 'o' key.
21	UD.shift.r.o	The duration between releasing 'shift.r' key and pressing 'o' key.
22	H.o	The duration between pressing and releasing 'o' key.
23	DD.o.a	The duration between pressing 'o' key and pressing 'a' key.
24	UD.o.a	The duration between releasing 'o' key and pressing 'a' key.
25	H.a	The duration between pressing and releasing 'a' key.
26	DD.a.n	The duration between pressing 'a' key and pressing 'n' key.
27	UD.a.n	The duration between releasing 'a' key and pressing 'n' key.
28	H.n	The duration between pressing and releasing 'n' key.
29	DD.n.l	The duration between pressing 'n' key and pressing 'l' key.
30	UD.n.l	The duration between releasing 'n' key and pressing 'l' key.
31	H.l	The duration between pressing and releasing 'l' key.
32	DD.l.return	The duration between pressing 'l' key and pressing 'return' key.
33	UD.l.return	The duration between releasing 'l' key and pressing 'return' key.
34	H.return	The duration between pressing and releasing 'return' key.

Table 3.4: Summary of dataset used

4.2 Data Pre-Processing

Data pre-processing is the foremost step in the KD system wherein the data is collected from multiple systems and processed in later stages. The data which is used in systems is acquired from various inputs ranging from sensitive keyboards to desktop keyboards. Some examinations so conducted also make use of special purpose number pad or a smart phone with touch screen [31]. Due to the shortage of globally available datasets, researchers started to generate their own datasets. [32] Made use of 1254 datasets out of which only 118 had

long passwords set for their systems, the rest made use of short codes. In [33] the input was divided into two groups: long and short.

The dataset used in the thesis however does not possess a missing value, still a certain set of outliers are found due to timing features. The occurrence of theses outliers is due to the presence of various styles in typing the keyboard. For example, an individual with a higher experience in typing tasks might type faster than compared to those who do not have one. Unfortunately, the dataset used in this thesis does not provide information regarding the typing efficiency of an individual. In the development stage, the processing of this model consists of 34 columns and 6000 rows, where each row reveals the timing information regarding to the password generated by a single subject for a single repetition.

4.3 Data Extraction

Once the data is put-together it must further undergo the stage of classification. Various feature extraction methods are applicable in the KD method [33]. The most used feature when it comes to KD implementation is the use of time measurements. When a key is used on the keyboard, a part of hardware is interrupted in the processor and a timestamp is generated. This time- stamp is generally measured in microsecond precision and contains information about the duration and time intervals between the keystrokes. This information in the timestamp between keystrokes is calculated and represented using; *di and n-graphs.*

1. A *di-graph* contains information of time execution between two consecutive keystrokes that are measured using parameters such as Dwell and Flight Time.
    - The duration between key presses and key releases is the dwell time; denoted by **(H).**
    - The timing between releasing and pressing the key is called as the flight time; signified by **(UD).** Latency is the by-product of the combination of the two related definitions mentioned above.

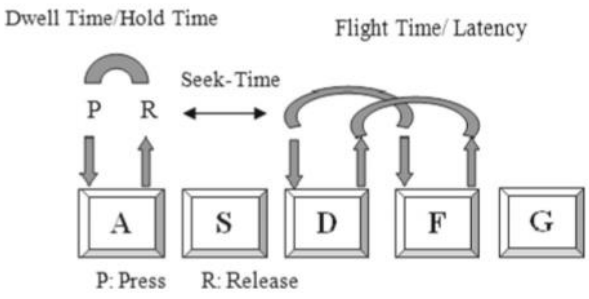


Figure 4.2 Dwell and flight time

2. An *n-graph* contains information of time execution three or more consecutive keystrokes.

Pressure, location on the touchpad, distance and relapse of term error are among the other variance extracted from keystroke data.

Authenticator Based on PCA

Since the keystroke differs for a valid user and an intruder it becomes easy to analyze intruder behavior and generate their respective vectors. This behavior vectors are known as outliers that is used as a measure of differentiation with respect to the behavior of a valid user. To detect these outliers in the thesis submitted, the concept of *Principal Component Analysis (pica)* is used in the 3rd and 10th neighborhood of the algorithm to generate vectors *pca3* and *pca10* respectively.

4.4 Data Visualization

When all the occurring events that a user types are recorded and monitored using parametric functions of a time frame; it is termed as latency. The outputs obtained below are the latency graphs of 6 classes consisting of durations between keys, denoted by H, DD and UD.

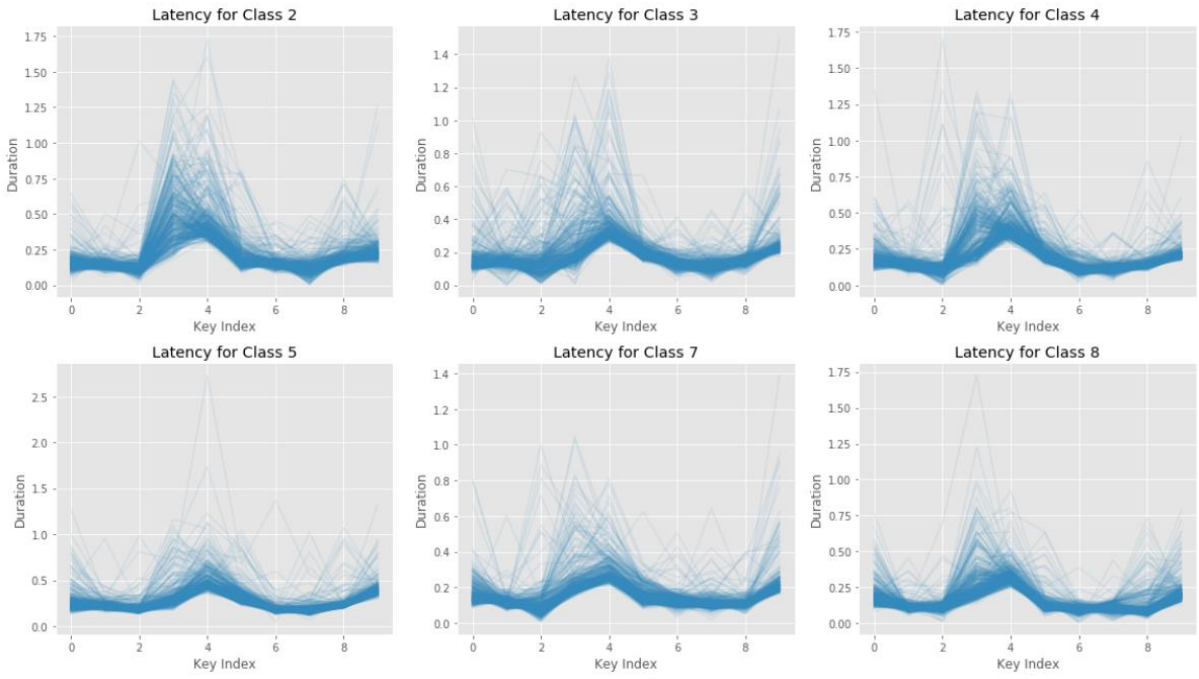


Figure 4.4 (a): Latency duration between keydown-keydown (DD)

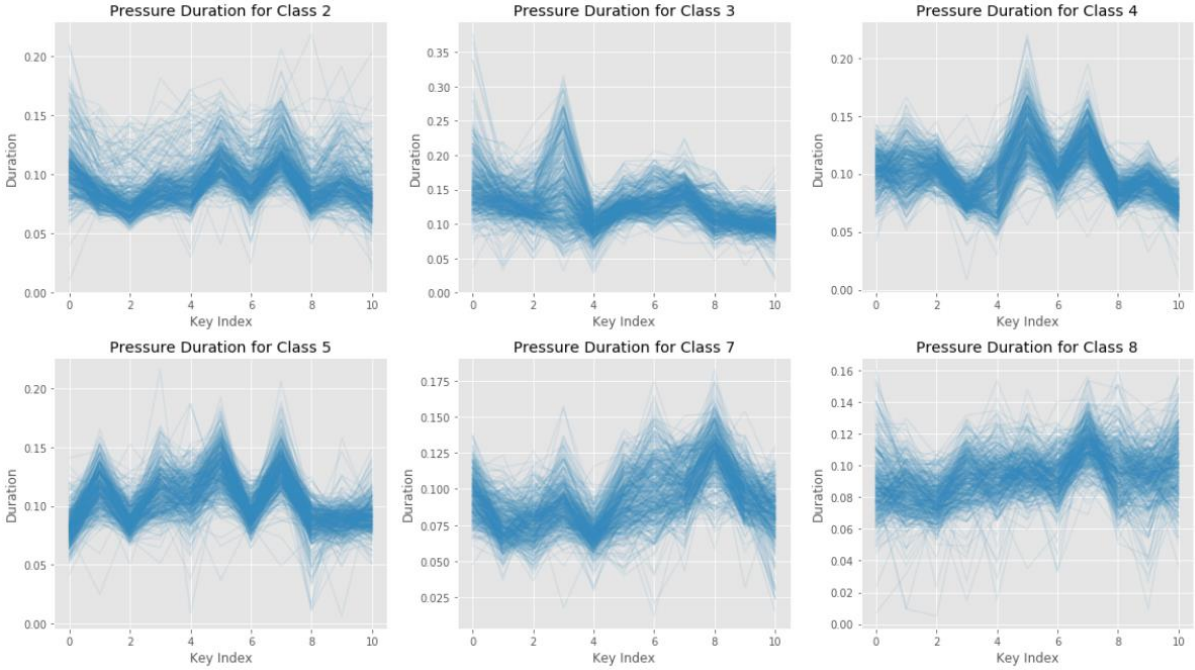


Figure 4.4 (b): Latency duration between keydown-keyup (H)

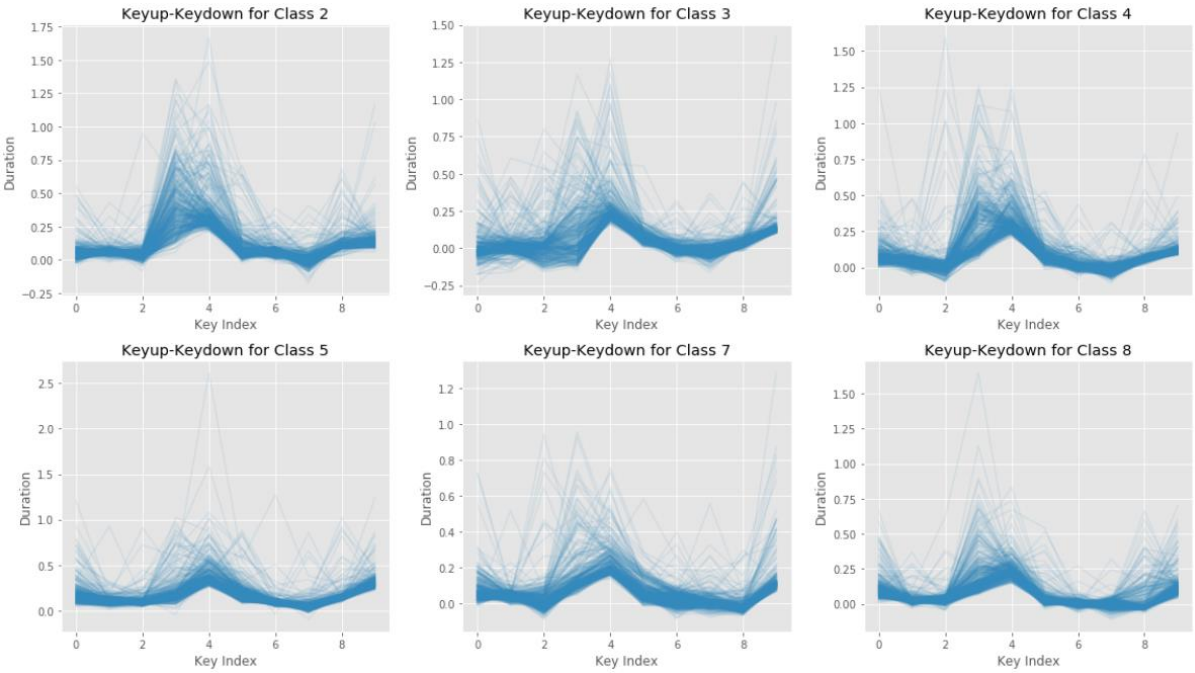


Figure 4.4 (c): Latency duration between release of one key and press of next (UD)

4.5 Model Training

4.5.1 K Nearest Neighbor

Nearest neighbor's is an example classification algorithm used to train the model in which the k closest neighbors decide on a new instance label. Because it reserves training samples and their labels, this algorithm does not require explicit training. In the submitted thesis, the KNN procedure is carried out in a straightforward way by computing simple majority votes of k-nearest neighbors at each point and a result is generated using the k-neighboring classifier. Accuracy is calculated for H, UD, and DD along with pca values for 3^{rd} and 10^{th} neighbors. Accuracy and ROC Curves so generated is mentioned in the results section.

The primary reason to choose this algorithm for keystrokes is that; it executes well on huge datasets where the incoming values might vary each time on execution. Since the thesis focus only on two features of the input; the releasing and pressing of keys; this algorithm serves as an advantage to keystrokes. The output of this algorithm depends largely on the hyper parameters used. Some of the most important hyper parameters are:

- Neighbours
- Weights
- Algorithms
- Metrics

Algorithm tuning is the final step in the entire process of machine learning prior to yielding an output. This process is generally termed as *hyperparameter Optimization*. Hence multiple search strategies are used depending on the algorithms. The one that this thesis aims to focus is Grid Search CV. This process is carried out to select the best features of KNN generated entities and to increase its accuracy. After the execution of Grid Search CV, top three models are ranked, and the number 1 model is selected for accuracy generation.

4.5.1 Multilayer Perceptron

MLP is a network pattern that is responsible to map the input data to a set of outputs. It comprises of:

- Input Layer
- Hidden Layer
- Output Layer

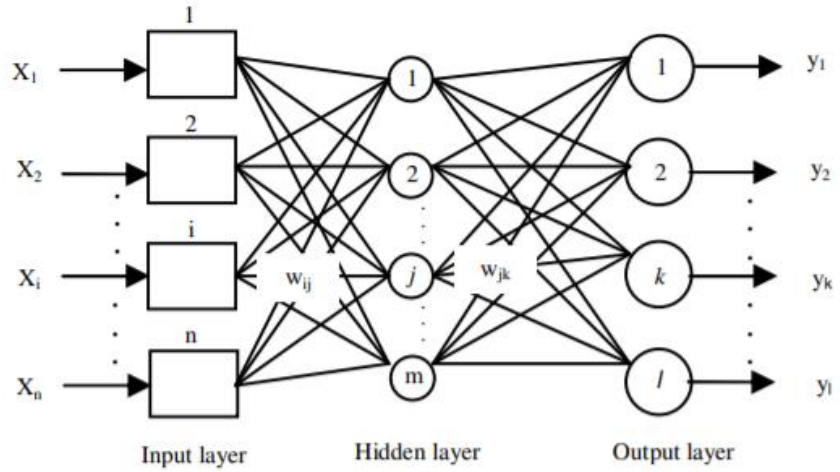


Figure 4.5.1: Overview of an MLP Network

- The input layer consists of input points along with assigned weights (x_1, x_2….x_n) which are further fed to the hidden layer.
- At each neuron in the hidden layer and activation function is triggered which travels from the current layer to the output layer. Multiple activation function exists such as ReLu, Sigmoid and tanh.
- Once the output of the hidden layer is generated via the activation function; a dot product is iteratively calculated with corresponding updated weights (y_1, y_2…..y_n).

In this thesis, the deep learning based MLP concept makes use of 2 hidden layers, 1 Input layer and 1 Output Layer. The number of neurons to be selected in the hidden layer is based on a thumb rule [34] depending on the number of units in both the layers. In this model 31 set of input neuron units were used in the input layer and 23 set of neurons were used in the hidden layer. These sets of units were decided based on trial and error while searching the optimal precision of the classifier. Therefore 15 units of neurons were selected for the output layer. The model so generated in this thesis made use of ReLu and Softmax as an activation function to avoid the issue of overfitting in the middle layers. Further this issue was also employed by a drop out layer with a dropout rate of 20%. The network was tested for 50, 100, 150, 200, 250, 300 nodes for each hidden layer. The entire working takes place on Python with Adam as an optimizer. The entire code was initially tested on 100 epochs with the preparatory dataset split into 70% training and 30% testing.

4. Results and Discussion

4.1 Evaluation

The process of generating new knowledge through the identification of distinct patterns is known as evaluation. The model's output is interpreted and transformed into knowledge in this step. Statistical inference is one method for interpreting the result. The receiver operating characteristic (ROC) curve is commonly used to describe the productivity of a biometric verification process. The transaction between false acceptance rate and false rejection rate is measured by the ROC. The equal error rate (EER), the point at which FAR is equal, describes the overall system performance.

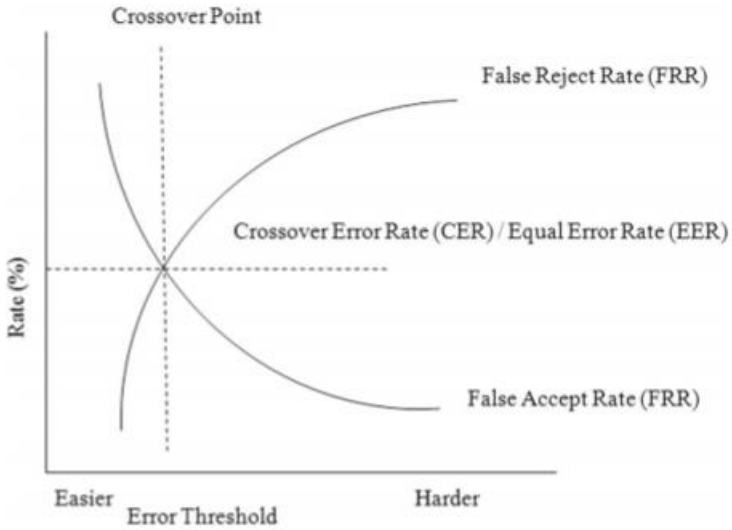


Figure 4.1: Relationship between FAR, FRR and EER

4.2 Results

```
==============================KNN Model=====================================
Accuracy for total : 0.69 (+/- 0.25)
Accuracy for H     : 0.65 (+/- 0.12)
Accuracy for DD    : 0.55 (+/- 0.20)
Accuracy for UD    : 0.60 (+/- 0.23)
Accuracy for pca3  : 0.19 (+/- 0.09)
Accuracy for pca10 : 0.58 (+/- 0.22)
```

The values of H, DD and UD are obtained using simple KNN as an algorithm. This KNN model is also being evaluated on the basis of Principal Component Analysis (pca) for the 3rd and 10 neighbors.

The total accuracy so obtained tends to be 69%.

==========**Grid Search CV**==========

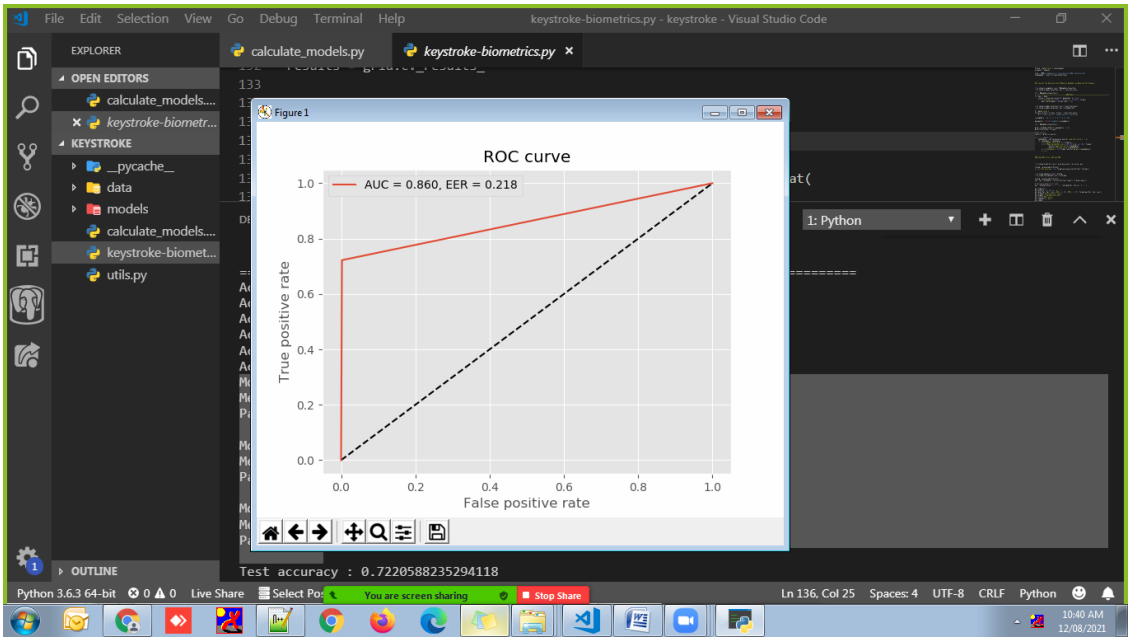
Below are the ranked models based on best feature selection

```
Model with rank: 1
Mean validation score: 0.696 (std: 0.008)
Parameters: {'n_neighbors': 3}

Model with rank: 2
Mean validation score: 0.665 (std: 0.008)
Parameters: {'n_neighbors': 5}
Test accuracy : 0.7220588235294118
Model with rank: 3
Mean validation score: 0.644 (std: 0.007)
Parameters: {'n_neighbors': 7}
```

Further the thesis provides the implementation of Grid Search CV. The above model of KNN was extended to this concept of Grid Search to increase the overall accuracy by generating three more models and ranking them based on selecting a model which portrays the best features.

Hence, the total accuracy obtained using Grid Search CV tends to be 72%.



Below is the generated ROC and AUC curve of Grid Search CV.

==========**MLP Model**==========

```
Data : total, Nodes : 300
  Accuracy for train set : 0.9871
  Accuracy for test set  : 0.8973
Data :     H, Nodes : 300
  Accuracy for train set : 0.7573
  Accuracy for test set  : 0.6260
Data :    DD, Nodes : 300
  Accuracy for train set : 0.7688
  Accuracy for test set  : 0.6392
Data :    UD, Nodes : 300
  Accuracy for train set : 0.8962
  Accuracy for test set  : 0.7225
Data :  pca3, Nodes : 300
  Accuracy for train set : 0.2106
  Accuracy for test set  : 0.2130
Data : pca10, Nodes : 300
  Accuracy for train set : 0.9381
  Accuracy for test set  : 0.7350
```

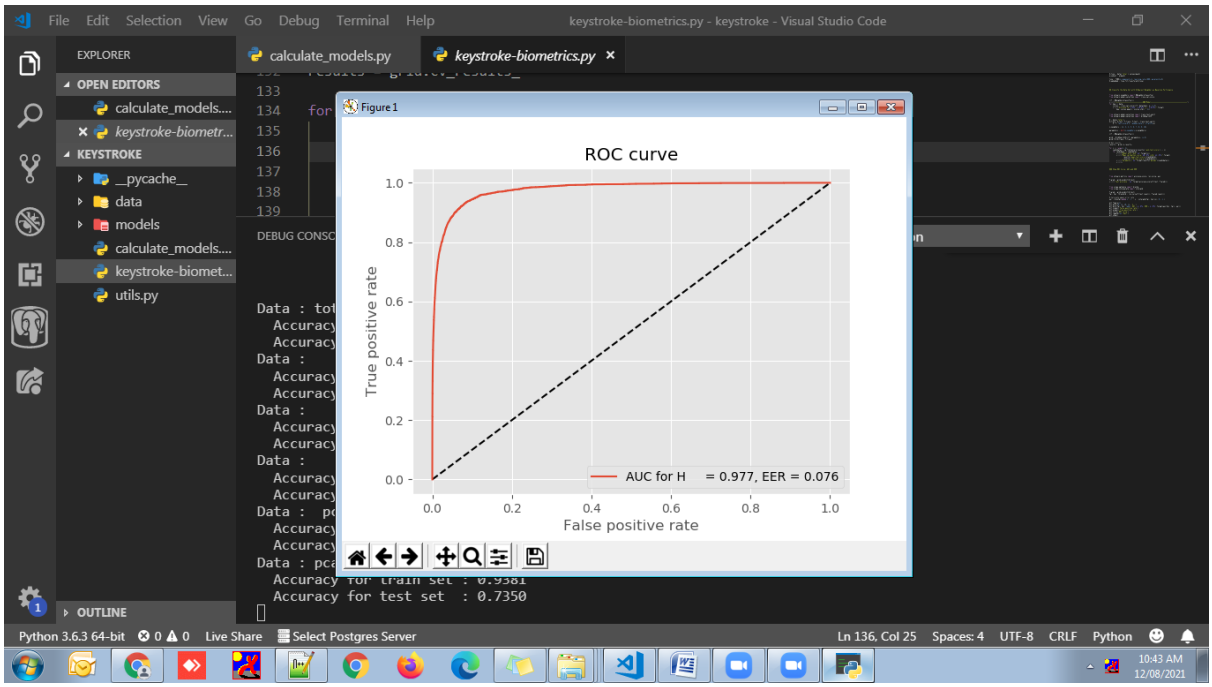
Output 4.2: ROC and AUC Curve of Grid Search Cv

After the execution of Grid Search CV with an increased accuracy over simple KNN, implementation of MLP model takes place. This execution is carried on H, DD, and UD along with PCA being performed on 3rd and 10th neighbours. The entire fundamentals of MLP are based on neurons and hence execution has taken place on nodes.

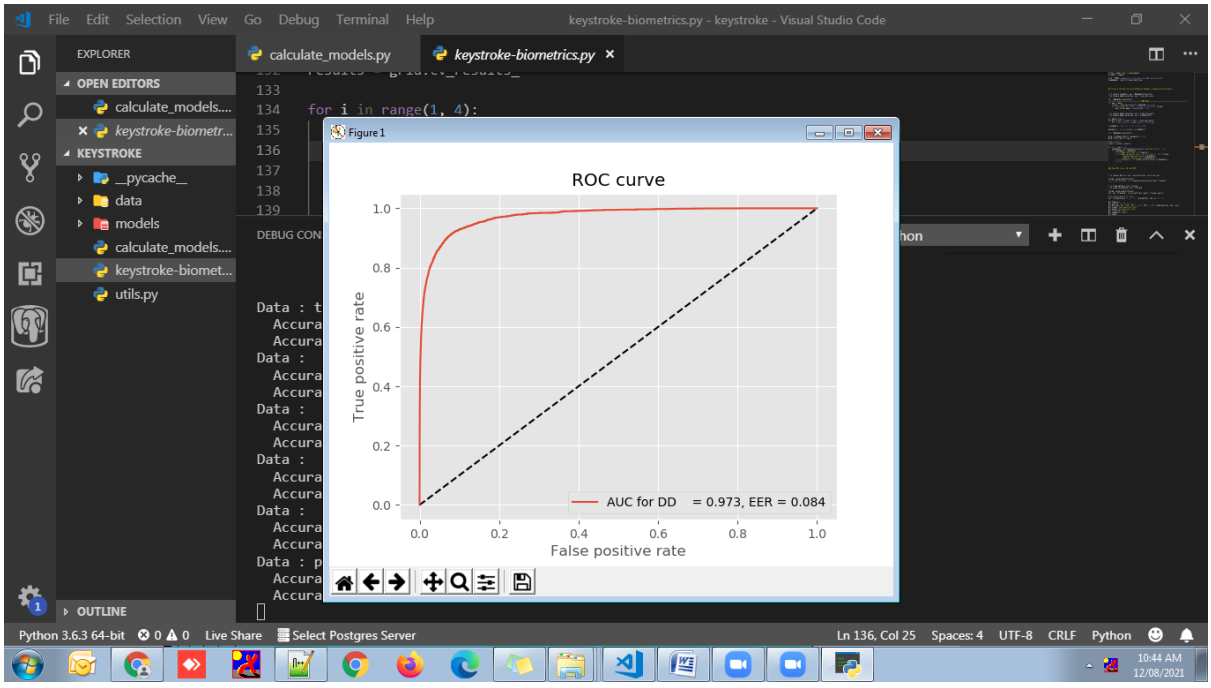
The accuracy on execution of MLP model is observed to be 73%.

Hence in comparison to the KNN based Grid Search CV, MLP model has proved to be better in terms of execution with an overall increase in accuracy.

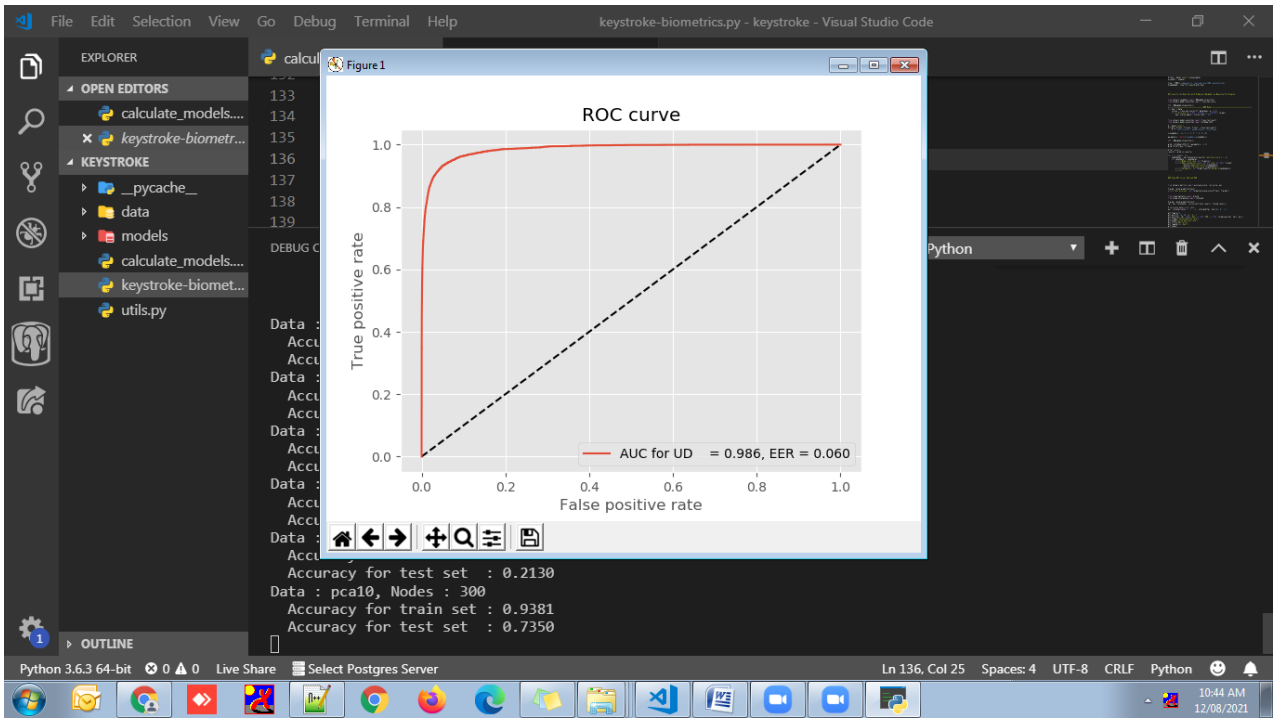
Below are the generated ROC and AUC Curves for H, DD, UD and PCA.



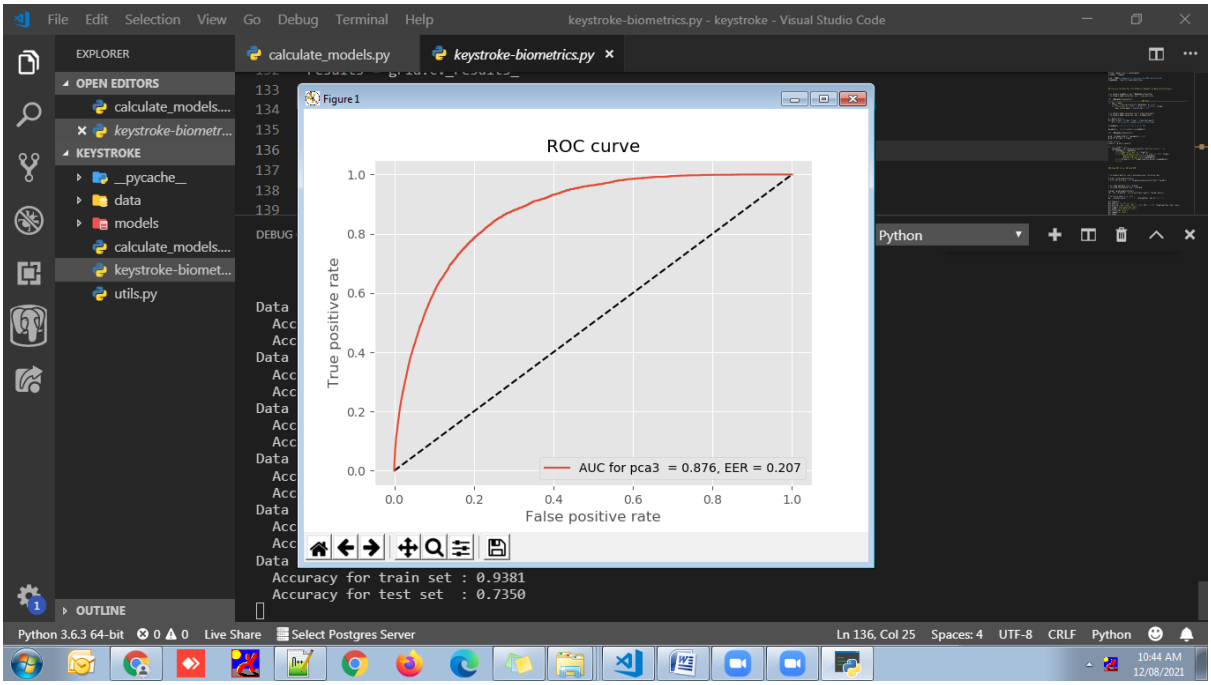
Output 4.2 (a): ROC Curve and AUC for H



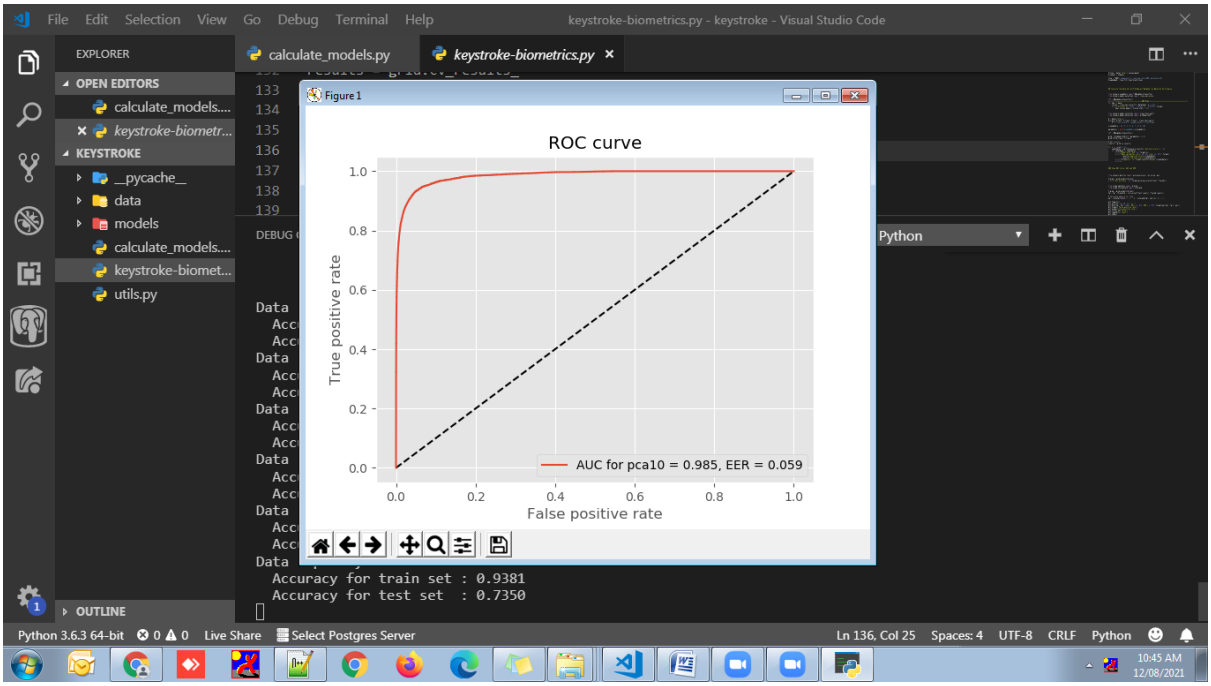
Output 4.2 (b): ROC Curve and AUC for DD



Output 4.2 (c): ROC Curve and AUC for UD



Output 4.2 (d): ROC Curve and AUC for pca3



Output 4.2 (e): ROC Curve and AUC for pca10

References

[1] Gentner, Keystroke timing in transcription typing, Cognitive Aspects of Skilled Typewritting, 1993, pp. 95–120.

[2] https://arxiv.org/ftp/arxiv/papers/0910/0910.0817.pdf

[13] J. V. Monaco and C. C. Tappert, "The partially observable hidden Markov model and its application to keystroke dynamics," Pattern Recognition, 2018.

[4] S. Roy, U. Roy, and D. D. Sinha, "Security enhancement of knowledge-based user authentication through keystroke dynamics," in Proc. MATEC Web Conf., 2016, vol. 57.

[5] K. S. Killourhy. (2012). A scientific understanding of keystroke dynamics. [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/2012/CMU-CS-12-100.pd f

[6] A. Messerman, T. Mustafić, S. A. Camtepe, and S. Albayrak, "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," in Proc. Int. Jt. Conf. Biometrics, 2011.

[7] P. S. Teh, B. J. Andrew Teoh, T. S. Ong, and H. F. Neo, "Statistical fusion approach on keystroke dynamics," in Proc. Int. Conf. Signal Image Technol. Internet Based Syst., January 2007, pp. 918–923.

[8] J. Ho and D. K. Kang, "Mini-batch bagging and attribute ranking for accurate user authentication in keystroke dynamics," Pattern Recognit., vol. 70, pp. 139–151, 2017.

[9] S. Maheshwary and V. Pudi, "Mining keystroke timing pattern for user authentication," Lect. Notes Comput. Sci., vol. 10312, pp. 213–227, 2017.

[10] A. Darabseh and A. S. Namin, "On accuracy of classification-based keystroke dynamics for continuous user authentication," in Proc. 2015 Int. Conf. Cyberworlds, 2016, pp. 321–324.

[11] P. H. Pisani, A. C. Lorena, and A. C. P. L. F. de Carvalho, "Adaptive Positive Selection for Keystroke Dynamics," J. Intell. Robot. Syst. Theory Appl., vol. 80, pp. 277–293, 2015.

[12] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics with low constraints SVM based passphrase enrollment," in Proc. IEEE 3rd Int. Conf. Biometrics Theory, Appl. Syst., 2009.

[13] H. Mohabeer and S. K. M. Soyjaudah, "Application of predictive coding in neuroevolution," Int. J. Comput. Appl., vol. 114, no. 2, pp. 41–47, 2015.

[14] J. Nisha and R. P. Kumar, "User authentication based on keystroke dynamics analysis," Int. J. Eng. Res. Appl., vol. 4, no. 3, pp. 345–349, 2014.

[15] H. B. K. Bharadi, P. S. Shah, and A. Ambardekar, “Keystroke dynamic analysis using relative entropy & timing sequence euclidian distance,” in Proc. Int. Conf. Work. Emerg. Trends Technol., 2011, p. 220

[16] R. Joyce, G. Gupta, Identity authorization based on keystroke latencies, Commun. ACM 33 (2) (1990) 168–176.

[17] L. K. Maisuria , C. S. Ong and W. K. Lai, “ A comparison of artificial neural network and cluster analysis for typing biometrics authentication”, International Joint Conference on Neural Network, IJCNN’99, vol.5, pp 3295-3299, 1999.

[18] H. Ali, Wahyudi and Momoh J.E Salami, “Intelligent Keystroke Pressure-Based Typing Biometrics Authentication System by Combining ANN and ANFIS-Based Classifiers”, International Colloquium on Signal Processing & Its Applications (CSPA), pp198, 2009.

[19] A. Sulong, Wahyudi and M.U Siddiqi, “Intelligent Keystroke Pressure-Based Typing Biometrics Authentication System by Using Radial Basis Function Network”, International Colloquium on Signal Processing & Its Applications (CSPA), pp151, 2009.

[20] N. Capuano, M.Marsella, S.Miranda and S. Salerno, “User Authentication with Neural networks”, Univerity of Salerno Italy. http://www.capuano.biz/Papers/EANN_99.pdf

[21] M.S. Obaidat and D.T Macchairolo, “A multilayer neural network system for computer access security”, IEEE transactions on Systems, Machine and Cybernetics, vol 24(5), 1994.

[22] R. Gaines, W. Lisowski, S. Press, N. Shapiro, Authentication by keystroke timing: some prelimary results. Rand Rep. R-2560-NSF, Rand Corporation, 1980.

[23] J. Leggett, G. Williams, Verifying identity via keystroke characteristics, Int. J. Man-Mach. Stud. 28 (1) (1988) 67–76.

[24] Trojahn, M., Arndt, F., & Ortmeier, F. (2013). Authentication with keystroke dynamics on Touchscreen Keypads—effect of different N-Graph combinations. In MOBILITY 2013, The third international conference on mobile services, resources, and users (pp. 114–119).

[25] Harun, N., Dlay, S.S., & Woo, W.L. (2010). Performance of keystroke biometrics authentication system using multilayer perceptron neural network (mlp nn). In 7th International symposium on communication systems networks and digital signal processing (CSNDSP’ 2010) (pp. 711–714). IEEE, Newcastle upon Tyne.

[26] Antal, M., Szabo, L.Z., & Laszlo, I. (2014). Keystroke dynamics on android platform. In INTER-ENG 2014, 8th international conference inerdisciplinarity in engineering (pp. 114–119). Tirgu Mures: Elsevier

[27] Bergando et al, “User Authentication through keystroke Dynamics”, ACM transaction on Information System Security” Vol.No. 5, pg 367-397, Nov 2002.

[28] Cho et al , “Web based keystroke dynamics identity verification using neural network”, Journal of organizational computing and electronic commerce, Vol. 10, No. 4, 295-307, 2000.

[29] Enzhe Yu, Sungzoon Cho, “Keystroke dynamics identity verification and its problems and practical solutions”, Computers & Security, 2004.

[30] K. S. Killourhy. (2012). A scientific understanding of keystroke dynamics. [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/2012/CMU-CS-12-100.pdf

[31] Pavaday, N., & Soyjaudah, K.M.S. (2007). Investigating performance of neural networks in authentication using keystroke dynamics. In Proceedings of the IEEE AFRICON 2007 conference (pp. 1–8).

[32] El-Abed, M., Dafer, M., & El Khayat, R. (2014). Rhu keystroke: a mobile-based benchmark for keystroke dynamics systems. In 2014 International Carnahan conference on security technology (ICCST) (pp. 1–4). IEEE.

[33] Teh, P.S., Teoh, A.B.J., & Yue, S. (2013). A survey of keystroke dynamics biometrics. The Scientific World Journal, 2013, 1–24. doi:10.1155/2013/408280.

[34] A. Blum, Neural Networks in C++: An Object-Oriented Framework for Building Connectionist System, New York: John Wiley & Sons, 1992

[35] K. S. Killourhy. (2012). A scientific understanding of keystroke dynamics. [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/2012/CMU-CS-12-100.pdf