

# **KEYSTROKE DYNAMIC AUTHENTICATION**

**A SEMINAR REPORT**

*Submitted by*

**Gautham Gopan RA1911026020091**

**Ms. Devahema D**

**(Assistant Professor, Department of Computer Science and Engineering)**

*in partial fulfilment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**of**

**FACULTY OF ENGINEERING AND TECHNOLOGY**



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**RAMAPURAM CAMPUS, CHENNAI -600089**

**November 2021**

## **ABSTRACT**

As the technology and infrastructure are developing at such a remarkable rate, the human society around the world is being introduced to brand new personal devices on a weekly pace. Hence its safe to say that we are so addicted and dependent on our personal devices that a state has been breached were a person's whole character and actions can be predicted by hacking his persona device. Therefore, there is always a constant attempt to keep the data safe within them and hence the popularity of passwords, pin numbers and more advanced authentication methods like fingerprint scanner and retina scanner are being pushed to their limit. That's were our project "Keystroke Dynamic Authentication comes into play, with the help of this project instead of typing a password to validate a user, we can discriminate users by their typing rhythms. The typing rhythm of a person can be considered as unique identification of a user, and cannot be mimicked or recreated. The main aim will be to build a classification based on neural networks with keras library. The keystrokes will be collected by using 3 parameters the hold time [H], keydown-keydown time [DD], keyup-keydown time [UD]. With these parameters we can create an highly credible data set than can be used to train the system and thereby creating an Authentication system that is highly secure.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	Page
0	ABSTRACT	ii
	LIST OF FIGURES	iv
	LIST OF TABLES	iv

1.	INTRODUCTION	1
2.	LITERATURE SURVEY	4
	2.1 Journal 1	4
	2.2 Journal 2	4
	2.3 Journal 3	5
	2.4 Journal 4	5
	2.5 Journal 5	5
	2.6 Journal 6	6
	2.7 Journal 7	6
	2.8 Journal 8	7
	2.9 Journal 9	7
	2.10 Journal 10	8
	2.11 Journal 11	8
	2.12 Journal 12	8
	2.13 Journal 13	9
	2.14 Journal 14	9
	2.15 Journal 15	10

<b>3.</b>		<b>COMPARISON OF RESULTS</b>	11
<b>4.</b>		<b>ARCHITECTURE DIAGRAM</b>	14
<b>5.</b>	5.1	<b>Introduction to Artificial Immune System (AIS)</b>	15
	5.2	<b>Method</b>	17
	5.3	<b>GA Based Optimized Negative Selection Classification Algorithm (NSCA)</b>	21
<b>6.</b>		<b>Algorithm Description</b>	23
<b>7.</b>		<b>Model Execution</b>	27
<b>8.</b>		<b>Future Enhancements</b>	41
<b>9</b>		<b>Conclusion</b>	42
<b>10</b>		<b>Reference</b>	43

## LIST OF FIGURES

Figure 4.1 Architecture Diagram of the system

Figure 5.1 Flow of GA

Figure 6.1 Initialization procedure

Figure 6.2 Verification Phase

Figure 6.3 Registration model

Figure 6.4 Login Phase

Figure 6.5 Basic Working

Figure 7.1 Imported Data

Figure 7.2 Latency

Figure 7.3 Average Latency

Figure 7.4 Pressure data

Figure 7.5 Average Pressure Data

Figure 7.6 Keyup-keydown Variance

Figure 7.7 Keyup-keydown Average duration

Figure 7.8 Data Clustering

Figure 7.9 Negative selection

Figure 7.10 ROC Diagram

Figure 7.11 Nomenclature predictions

## **LIST OF TABLES**

Table 3.1 Comparison of results

Table 5.1 Initial results

Table 5.2 Comparison of results

# CHAPTER 1

## 1.1 Introduction

- Authentication systems are security measures put in place to secure data and systems by requiring additional input beyond username and password for users to access a system. Using authentication improves data security and prevents potential breaches. When multi-factor authentication is required to access a system, the system is less vulnerable to security issues like weak passwords or attacks like phishing. Authentication systems are ideal for businesses with sensitive data or systems that require secure user accounts.
- 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. Keystroke dynamics is a behavioral biometric. Keystroke dynamics uses a unique biometric template to identify individuals based on typing pattern, rhythm and speed. The raw measurements used for keystroke dynamics are known as “dwell time” and “flight time”. Dwell time is the duration that a key is pressed, while flight time is the duration between keystrokes. Keystroke dynamics can therefore be described as a software-based algorithm that measures both dwell and flight time to authenticate identity.
- The Intention of the project” Keystroke Dynamic Authentication” is to provide an alternative and more secure personal authentication system which will prove to be more efficient than the exiting methods like password verification and fingerprint scanners. With the help of this project instead of typing a password to validate a user, we can discriminate users by their typing rhythms. The typing rhythm of a person can be considered as unique identification of a user, and cannot be mimicked or recreated
- The roots of keystroke dynamics go back to the early days of the telegraph, when individuals developed distinctive patterns that identified them. This pattern was known as a telegraph operator's "fist." During World War II, a methodology known as the "first of the sender" helped to identify the source of Morse code and confirm that a particular message was, in fact, from a valid source.

## **1.2 Objective**

- The main aim will be to build a classification based on neural networks with keras library. The keystrokes will be collected by using 3 parameters the hold time [H], keydown-keydown time [DD], keyup-keydown time [UD]. With these parameters we can create an highly credible data set than can be used to train the system and thereby creating an Authentication system that is highly secure. Further illustrations of the working are portrayed through the Fig 1.1 below where the role of the parameters is highlighted.
- The raw measurements used for keystroke dynamics are dwell time and flight time.
  - Dwell time is the time duration that a key is pressed
  - Flight time is the time duration in between releasing a key and pressing the next key
- When typing a series of characters, the time the subject needs to find the right key (flight time) and the time he holds down a key (dwell time) is specific to that subject, and can be calculated in such a way that it is independent of overall typing speed. The rhythm with which some sequences of characters are typed can be very person dependent.
- Keystroke dynamics uses a unique biometric template to identify individuals based on typing pattern, rhythm and speed. The raw measurements used for keystroke dynamics are known as “dwell time” and “flight time”. Dwell time is the duration that a key is pressed, while flight time is the duration between keystrokes. Keystroke dynamics can therefore be described as a software-based algorithm that measures both dwell and flight time to authenticate identity.

## **1.3 Need for the study**

- Complying to the increased needs of more effective and reliable authentication systems, it is safe to say that keystroke dynamics is the future of authentications systems.
- The main advantage keystroke dynamics have over other conventional authentication system is that its highly immune to hacking and it is extremely difficult to manipulate an individual's keystrokes thereby increasing the efficiency of the system manifoldly.

- Considering other authentication systems like fingerprint time-log systems this technique proves to be much more hygienic and less time consuming as it requires no special equipment's and can be privately done in any individuals personal device.
- The amount and monetary supply for this system is extremely low as this system can be readily deployed to normal users without the need of any extra hardware to be integrated.

### **1.4 Scope of work**

- The development of Keystroke dynamics is currently focusing on the internet banking sector to provide a more reliable mode of user verification instead of using multiple passwords which the customer will also have trouble remembering
- There is a very competitive market for highly trained keystroke systems worldwide especially in tech-oriented countries like the US, India, UK etc.
- Keystroke dynamics can also play a major role in enhancing the user verification and account scrutiny of cryptocurrency systems. This will also lead to making cryptocurrency a more acceptable mode of transaction among the common public.
- This system can also be integrated as employee punching systems in the cooperative world, thereby replacing the conventional biometric system used by most MNCS.
- Another use is as a very specific form of surveillance. There exist software solutions which, often without end-users being aware of it, track keystroke dynamics for each user account. This tracking, historization of keystroke dynamics is then used to analyses whether accounts are being shared or in general are used by people different from the genuine account owner. Reasons for such an implementation could be verification of users following security procedures (password sharing) or to verify that no software licenses are being shared (especially for SAAS applications).



## CHAPTER 2

### Literature Survey

#### 2.1 Journal-1<sub>[1]</sub>

Author Name: M.Karnan ,M.akila

Paper Title: Biometric personal authentication using keystroke dynamics: A review

Objective of paper: Enhancing behavioral measurement using and utilizing manner and rhythm of typing.

Method: Data entry

Algorithm Used: The immune system is a remarkable information processing and self-learning system that offers inspiration to build artificial immune system (AIS).

Drawbacks: The efficiency of this system is highly questionable and the data set used is very limited and less diverse.

#### 2.2 Journal-2<sub>[2]</sub>

Author Name: Saurabh Singh ,K.V.Arya

Paper Title: Key classification: a new approach in free text keystroke authentication system.

Objective of paper: In this paper, a novel technique is proposed which is based on free text password. In this method user types randomly chosen password every time and he/she does not have to remember the password. We have used key board grouping technique to capture the keystroke pattern of the user, when user presses a key the group of that key is identified and the flight times among groups are recorded to form timing vector.

Method: Data entry

Algorithm Used: Support Vector Machines with Polynomial Kernel.

Drawbacks: If there is further development done in this field, this classification approach can be used in other behavioral identification and authentication systems like signature recognition, gait recognition etc.

### 2.3 Journal-3<sub>[3]</sub>

Author Name: Priyanka Namnaik Rajeshree Kurale, Sanyukta Mahindrakar

Paper Title: Keystroke dynamics for user authentication

Objective of paper: In this paper, authentication method is based on keystroke along with the current username and password System.

Method: Data entry

Algorithm Used: Deep Learning model with Extreme Gradient boosting.

Drawbacks: There is still room for further improvements that includes the identification of other parameters and the use of another algorithm to improve the level of security and implementation of keystroke on touch screen system (LAPTOP) and use of pressure keyboard.

### 2.4 Journal-4<sub>[4]</sub>

Author Name: Praveen Kumar, Shagun Seth, Kanishka Bajaj, Seema Rawat

Paper Title: Diverse security practices and comparison on key stroke dynamics.

Objective of paper: For better authentication behavioral biometrics methods have been introduced in which an insight into the behavioral aspects such as keystroke dynamics.

Method: Data entry

Algorithm Used: . Support Vector Machines with Linear Kernel.

Drawbacks: Not very feasible as compared to Iris Detection.

### 2.5 Journal-5<sub>[5]</sub>

Author Name: Nick Bartlow.

Paper Title: Username and Password Verification through Keystroke Dynamics

Objective of paper: The goal of this study is to establish the viability of keystroke dynamics with username and password input as a possible method of hardening authentication credentials. It should also result in a method which allows for a biometric system to be readily deployable both in an unsupervised and remote fashion. Finally, the study will attempt to establish the difference in performance of the biometric system associated with two significantly different types of passwords.

Method: Data entry

Algorithm Used: . Support Vector Machines with Linear Kernel

Drawbacks: Keystroke dynamics biometrics are inferior in terms of authentication accuracy due to the variations in typing rhythm that caused by external factors such as injury, fatigue, or distraction.

## 2.6 Journal-6<sub>[6]</sub>

Author Name: Jarmon Ilonen

Paper Title: Keystroke Dynamics

Objective of paper: This article is an introduction to keystroke dynamics. Keystroke dynamics is a biometric which is based on assumption that people type in uniquely characteristic manners. Keystroke dynamics is mainly used for verification, but also identification is possible. Also commercial products exist. Additionally, keystroke dynamics can be used to eavesdrop secure communications by guessing what was written based on timings between letters.

Method: Data entry

Algorithm Used: Support Vector Machines with Linear Kernel

Drawbacks: Many of the application papers use either normal words or slightly longer phrases than what is customary with passwords. If the password is used only in settings where the keystroke dynamics are also checked, for example, there is no chance to bypass the keystroke dynamics phase when logging in from a networked computer elsewhere, then it might be good enough to use a simple word as a password. Existing words are otherwise too easy to crack by dictionary attacks.

## 2.7 Journal-7<sub>[7]</sub>

Author Name: Saleh Bleha, Charles Slivinsky, & Bassam Hussien

Paper Title: Computer-Access Security Systems Using Keystroke Dynamics

Objective of paper: This correspondence describes a new approach to securing access to computer systems. By performing real-time measurements of the time durations between the keystrokes when a password is entered and using pattern recognition algorithms, three on-line recognition systems were devised and tested.

Method: Data entry

Algorithm Used: Support Vector Machines with Linear Kernel.

Drawbacks: there is no chance to bypass the keystroke dynamics phase when logging in from a networked computer elsewhere, then it might be good enough to use a simple word as a password. Existing words are otherwise too easy to crack by dictionary attacks.

## 2.8 Journal-8<sub>[8]</sub>

Author Name: Fabian Montrose, Michael K. Reiter & Susanne Wetzel

Paper Title: Password hardening based on keystroke dynamics

Objective of paper: We present a novel approach to improving the security of passwords. In our approach, the legitimate user's typing patterns (e.g., durations of keystrokes and latencies between keystrokes) are combined with the user's password to generate a hardened password that is convincingly more secure than conventional passwords alone. In addition, our scheme automatically adapts to gradual changes in a user's typing patterns while maintaining the same hardened password across multiple logins, for use in file encryption or other applications requiring a long-term secret key. Using empirical data and a prototype implementation of our scheme, we give evidence that our approach is viable in practice, in terms of ease of use, improved security, and performance.

Method: Data entry

Algorithm Used: Support Vector Machines with Linear Kernel.

Drawbacks: If there is further development done in this field, this classification approach can be used in other behavioral identification and authentication systems like signature recognition, gait recognition etc.

## 2.9 Journal-9<sub>[9]</sub>

Author Name: Paulo Henrique Pisani & Ana Carolina Lorena

Paper Title: A systematic review on keystroke dynamics

Objective of paper: This paper discusses the process involved in the review along with the results obtained in order to identify the state of the art of keystroke dynamics. We summarized main classifiers, performance measures, extracted features and benchmark datasets used in the area.

Method: Data entry

Algorithm Used: Deep Learning model with Nadam (Nesterov-accelerated Adaptive Moment Estimation) Optimizer.

Drawbacks: From the studies we have learned that it is unlikely that keystroke dynamics alone will be robust enough to uniquely identify users, but it shows great promise as a part of a larger multimodal biometric authentication method.

## 2.10 Journal-10<sub>[10]</sub>

Author Name: Pin Shen Teh ,Andrew Beng Jin Teoh , and Shigang Yue

Paper Title: A Survey of Keystroke Dynamics Biometrics

Objective of paper: The objective of this paper is to provide an insightful survey and comparison on keystroke dynamics biometrics research performed throughout the last three decades, as well as offering suggestions and possible future research directions.

Method: Data entry

Algorithm Used: Support Vector Machines with Linear Kernel.

Drawbacks: The literature study suggested that keystroke dynamics biometrics are unlikely to replace existing knowledge-based authentication entirely and it is also not robust enough to be a sole biometric authenticator.

## **2.11 Journal-11<sub>[11]</sub>**

Author Name: Paulo Henrique Pisani & Ana Carolina Lorena

Paper Title: A systematic review on keystroke dynamics

Objective of paper: This paper discusses the process involved in the review along with the results obtained in order to identify the state of the art of keystroke dynamics. We summarized main classifiers, performance measures, extracted features and benchmark datasets used in the area.

Method: Data entry

Algorithm Used: Deep Learning model with Nadam (Nesterov-accelerated Adaptive Moment Estimation) Optimizer.

Drawbacks: Recognition precision by keystroke dynamics may be affected in the presence of keyboards with different characteristics in the same environment. The sheer amount of time required to train the model and twitch it to attain pinnacle efficiency was extremely long and exhausting in nature thereby effecting the overall efficiency of the system. But it shows great promise as a part of a larger multimodal biometric authentication method.

## **2.12 Journal-12<sub>[12]</sub>**

Author Name: H. Saevanee, P.Bhattarakosol

Paper Title: Authenticating User Using Keystroke Dynamics and Finger Pressure

Objective of paper: In this paper, we proposed behavioral manners of users over the touch pad acting like touch screen that is able to detect finger pressure. We study the potential of each biometrics behavioral by individual and couple, comprise with hold-time, inter-key and finger pressure. The finding has shown that, the finger pressure gives the discriminative information more than keystroke dynamics with the PNN analytical method. Moreover, using only the finger pressure produces high accuracy rate of 99%. Data collected from 20 participants. Each participant was

asked to register himself/herself and then each was invited to for login trail 5 times as legitimate user and 5 times as impostor randomly. Participants were final year engineering students of age group 20–28 Y.

Method: Data entry

Algorithm Used: Support Vector Machines with Linear Kernel.

Drawbacks: Recognition precision by keystroke dynamics may be affected in the presence of keyboards with different characteristics in the same environment.

## **2.13 Journal-13<sub>[13]</sub>**

Author Name: Salil P. Banerjee, Damon L. Woodard

Paper Title: Biometric Authentication and Identification using Keystroke Dynamics

Objective of paper: in this paper, we provide a basic background of the psychological basis behind the use of keystroke dynamics. We also discuss the data acquisition methods, approaches and the performance of the methods used by researchers on standard computer keyboards. In this survey, we find that the use and acceptance of this biometric could be increased by development of standardized databases, assignment of nomenclature for features, development of common data interchange formats, establishment of protocols for evaluating methods, and resolution of privacy issues

Method: Data entry

Algorithm Used: Deep Learning model with Nadam (Nesterov-accelerated Adaptive Moment Estimation) Optimizer.

Drawbacks: Majority of the work on keystroke dynamics involves English as the primary language of communication. However, differences in language can lead to drastically different results even with the same algorithm.

## **2.14 Journal-14<sub>[14]</sub>**

Author Name: Xiao feng Lu, Sheng fei Zhang, Pan Hui, Pietro Lio

Paper Title: Continuous authentication by free-text keystroke based on CNN and RNN

Objective of paper: The method proposed in this paper authenticates users via their keystrokes when they type free text. The user keystroke data is divided into a fixed-length keystroke sequence, which is then converted into a keystroke vector sequence according to the time feature of the keystroke. The main computations is enforced by using CNN and RNN. The model is tested using two open datasets, and the best false rejection rate (FRR) is found to be (2.07%,6.61%), the best false

acceptance rate (FAR) is found to be (3.26%, 5.31%), and the best equal error rate (EER) is found to be (2.67%, 5.97%).

Method: Data entry

Algorithm Used: CNN with RNN model is used to get the data of free texts.

Drawbacks: There are still some deficiencies in this research; for example, the small amount of data may lead to the inadequacy of the personal keystroke mode. There is no chance to bypass the keystroke dynamics phase when logging in from a networked computer elsewhere, then it might be good enough to use a simple word as a password. Existing words are otherwise too easy to crack by dictionary attacks.

## **2.15 Journal-15<sub>[15]</sub>**

Author Name: Margit Antal , László Zsolt Szabó , Izabella László

Paper Title: Keystroke Dynamics on Android Platform

Objective of paper: Touchscreen allows adding features ranging from pressure of the screen or finger area to the classical time-based features used for keystroke dynamics. In this paper we examine the effect of these additional touchscreen features to the identification and verification performance through our dataset of 42 users. User Measurements were performed using WEKA, which is a popular machine learning Software. Significant differences in the results were determined using corrected period t-test at 0.05 significance level.

Method: Data entry

Algorithm Used: several machine learning classification algorithms, of which the best performers were Random forests, Bayesian nets and SVM, in this order.

Drawbacks: The amount of time required to train the model Is extremely long and proves to be extremely useless in this modern era. Many of the application papers use either normal words or slightly longer phrases than what is customary with passwords. If the password is used only in settings where the keystroke dynamics are also checked

## CHAPTER 3

### Comparison Of Results

Table 3.1 Comparison of results

SI.NO	NAME OF ALGORITHM	DESCRIPTION	ACCURACY
1.	Support Vector Machines with Polynomial Kernel	Support vector machines so called as SVM is a <b><i>supervised learning algorithm</i></b> which can be used for classification and regression problems as support vector classification (SVC) and support vector regression (SVR). It is used for smaller dataset as it takes too long to process. SVM is based on the idea of finding a hyperplane that best separates the features into different domains. The polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of <u>regression analysis</u> , such combinations are known as interaction features. The (implicit) feature space of a polynomial kernel is equivalent to that of <u>polynomial regression</u> .	The accuracy is around 2.35%.
2.	Support Vector Machines with RBF (Radial Basis Function) Kernel	The kernel functions are used to map the original dataset (linear/nonlinear) into a higher dimensional space with view to making it linear dataset. Usually linear and polynomial kernels are less time consuming and provides less accuracy than the RBF or Gaussian kernels	The accuracy is around 30.07%.



SI.NO	NAME OF ALGORITHM	DESCRIPTION	ACCURACY
3.	Support Vector Machines with Linear Kernel	<b>Linear Kernel</b> is used when the data is linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a large number of Features in a particular Data Set. Training a SVM with a Linear Kernel is <b>faster</b> than with any other Kernel.	The accuracy is around 71.15%.
4.	. Deep Learning model with Nadam (Nesterov-accelerated Adaptive Moment Estimation) Optimizer	The <b>Nadam</b> algorithm is an extension to the Gradient Descent Optimization algorithm (ADAM to be specific). Momentum adds an exponentially decaying moving average (first moment) of the gradient to the gradient descent algorithm. This has the impact of smoothing out noisy objective functions and improving convergence. <b>Nadam</b> uses a decaying step size (alpha) and first moment (mu) hyper parameters that can improve performance. For the case of simplicity, we will ignore this aspect for now and assume constant values.	The accuracy is around 92.06%.
5.	Deep Learning model with Extreme Gradient boosting(XG boost)	XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now	The accuracy is around 93.59%.

SI.NO	NAME OF ALGORITHM	DESCRIPTION	ACCURACY
6.	Artificial Immune Systems [AIS]	This was the latest and most innovative approach we found after going through 15 literature surveys. AIS is an biologically inspired system which has the capability of accepting and blocking unwanted data entries analogues to how our human immune system works.	The accuracy is around 82.12%.

## CHAPTER 4

### Architecture Diagram

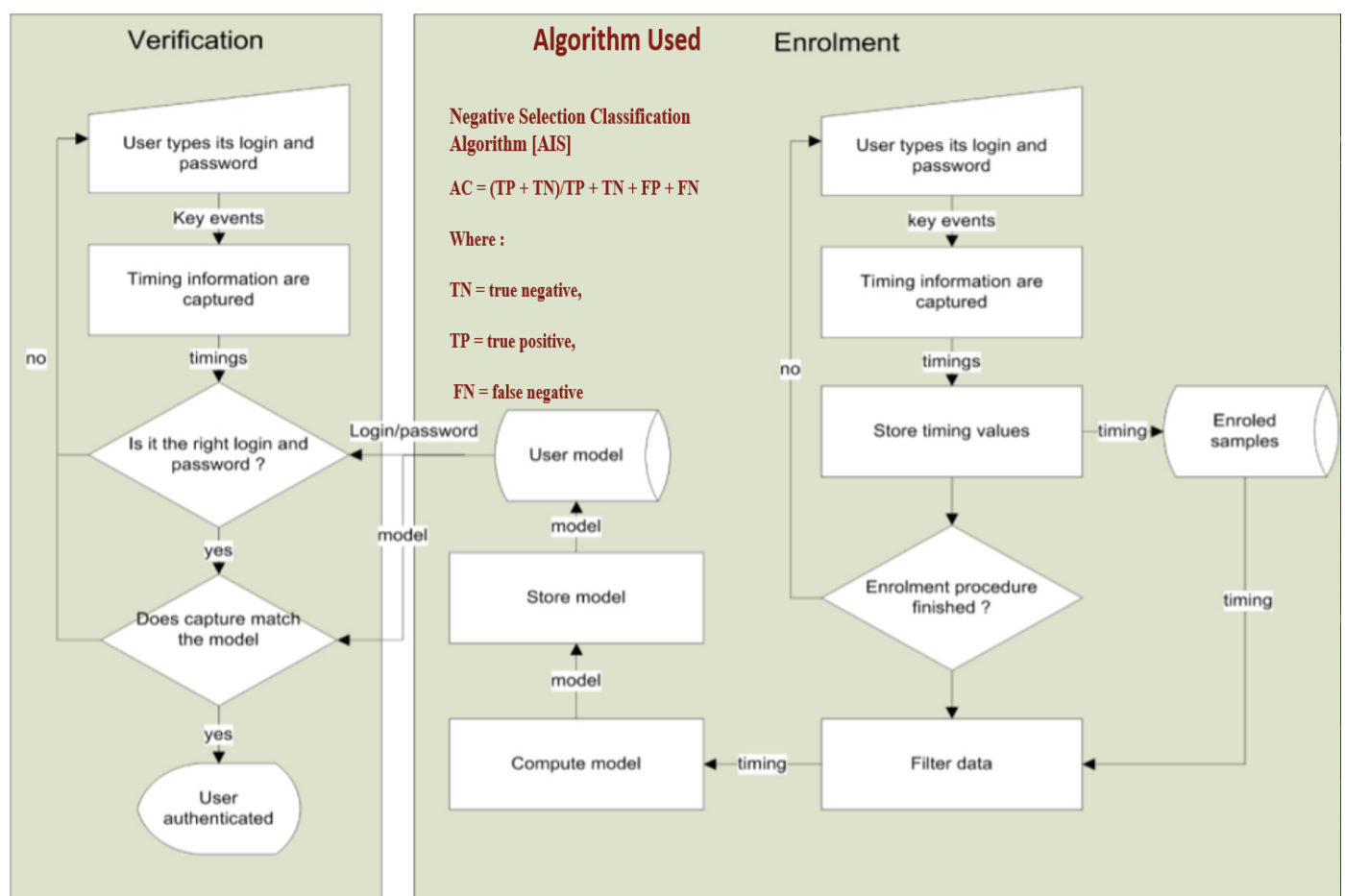


Figure 4.1 Architecture Diagram of the system

With Respect to the above Architecture diagram the results are produced via calculation of the classification accuracy using equation:

$$AC = (TP + TN) / (TP + TN + FP + FN)$$

Where :

TN = true negative,

TP = true positive,

FN = false negative

negative, and FP = false positive. Also note that these free parameters are interdependent, as discussed in the conclusion section in more detail. Briefly, a search across the entire parameter space was required for each calculation.

The Algorithm that has been used in a part of Artificial immune systems (AIS). They are intelligent algorithms derived from the principles inspired by the human immune system. In this study, electroencephalography (EEG) signals for four distinct motor movements of human limbs are detected and classified using a negative selection classification algorithm (NSCA). For this study, a widely studied open-source EEG signal database (BCI IV–Graz dataset 2a, comprising nine subjects) has been used. Mel frequency cepstral coefficients (MFCCs) are extracted as selected features from recorded EEG signals. Dimensionality reduction of data is carried out by applying two hidden layered stacked auto-encoder. Genetic algorithm (GA) optimized detectors (artificial lymphocytes) are trained using negative selection algorithm (NSA) for detection and classification of four motor movements.

The trained detectors consist of four sets of detectors, each set is trained for detection and classification of one of the four movements from the other three movements. The optimized radius of detector is small enough not to mis-detect the sample. Euclidean distance of each detector with every training dataset sample is taken and compared with the optimized radius of the detector as a nonself detector. Our proposed approach achieved a mean classification accuracy of 86.39% for limb movements over nine subjects with a maximum individual subject classification accuracy of 97.5% for subject number eight.

## CHAPTER 5

### 5.1 Introduction to Artificial Immune System (AIS)

#### 5.1.1 Definition:

- A variety of computational models have their roots in biological processes: these include artificial neural networks, genetic algorithms, particle swarms, and more recently artificial immune systems. The attraction of these biological processes is probably derived from their apparent information processing capabilities. These systems have the innate ability
- To perform classification-based activities, are distributed, and are adaptable. In addition, these biological computation capacities appear to operate automatically and autonomously– a very desirable yet debatable topic. In the present case, the artificial immune system contains these same properties: they are distributed, adaptable systems with memory that provide the organism with the basic ability to distinguish self from non-self.

#### 5.1.2 Goal:

- The operational goal of the immune system is to eradicate any non-self-matter that enters the organism's biological domain. The end result of this process is the destruction of that which is deemed to be non-self through a series of chemical reactions. In the present context, we would like to produce an AIS that is able to perform the essential functions of biological immune systems: distinguishing self (authentic users) from non-self (imposters).
- In the keystroke dynamics domain, self is not a fixed point, but rather a set of entries that the user has successfully been authenticated with. We generally do not repeat the same typing pattern precisely. As a matter of fact, such perfect fidelity may alert a 'replay attack' module that may reject the authentication attempt outright. So, variation is expected – the issue is how much can we incorporate

into our authentication system in order to maintain false acceptance and false rejection rates within desirable levels.

### **5.1.3 Ideology :**

- An artificial immune system is simply an implementation of a biological immune system in silico basically. It must implement the salient features of the biological immune system, at some level. In this work, the concept of distinguishing self from non-self is implemented in a fashion that certainly has biological realism, but is not complete in all levels of detail. The AIS presented here implements the antigen-antibody concepts which form the cornerstone of immunology.
- The antigen is the foreign object which may be recognised by the immune system – if it is, it will be destroyed if possible. The foreign object is initially encountered by the human host through interactions with host generated molecules termed antibodies. Human immune systems contain literally billions of antibodies each capable of interacting with a bewildering array of antigens. In some instances, these antigens may be part of the host, yielding an auto-immune response. This is considered a mistake so to speak, but yields serious repercussions as people with arthritis and related diseases are well known. More typically, the circulating antibodies identify a substance as foreign, which is truly foreign, and mount an attack which attempts to destroy the host. In this sense, the immune system operates in a distributed and parallel fashion. This feature must clearly be incorporated into an AIS model, which is true in the current case.
- Lastly, the system must be adaptable if it is to respond to antigens it has not been previously exposed to – that is like ANNs, it must be able to generalize. This ability should manifest over variable time windows – locally by differentiating into variants that can attack antigens that are also adaptable, and also over the long term, so that a repeat attack will be acted upon more rigorously.
- This long term aspect of the immune system simply indicates that it has actually learned something from the previous interaction. The distinction is like guessing the answer to a question by shouting

out random answers, compared to solving the problem analytically. The next section describes the basis of the experiment and describes how an implementation of the AIS was deployed.

## **5.2 Method**

### **5.2.1 Data Preparation:**

- There were 20 participants in this study, all from computer science undergraduate students from a Polish University. The users were provided with 8-character login IDs and 8-character passwords, generated randomly by a computer programmer. The characters consisted of all upper and lowercase alphabetic characters and the digits. The enrolment process required users to enter their login ID/password 10 times successfully. Each participant enrolled on to a single machine located on campus - for both enrolment and subsequent login attempts.
- After successfully enrolling (10 trials), the participants were asked to perform 100 self-logins (for FRR) and 100 attacks on other accounts for each account including their own, which was not utilised, for FAR data). The following regime was used for non-enrolment logins: each participant was asked to self-login 100 times over a 7-day period. Therefore, each participant logged into their own account approximately 15 times/day. In addition, students were instructed to login at 3 different periods of the day: morning (09:00-10:00), noon (12:00-13:00) and early evening (17:00-18:00). At each period, students were asked to either perform self-login or non-self-login 5 times. This simulates the way users would normally access their computer systems, logging in at various periods during the course of a workday.

### **5.2.2 Data Extraction:**

- The data that was extracted from the login ID and password combination were simply keypress di-graphs – that is, the time (in Ms) between depressing successive keys. There was a total of 14 di-graphs in the login IDs and passwords that were recorded, and stored in a vector of floats (normalised to [0,1]), along with the actual di-graph characters. This vector of di-graph times serves as the shape space of the authentication attempt (the antigen). The enrolment process provides a sample of self-logins, which serve as the means of providing a reasonable set of examples of self.
- The enrolment samples then serve as the basis for fine tuning the immune system such that it is able to differentiate self (those samples similar to the enrolment entries) from non-self (samples that differ significantly from the enrolment samples). The enrolment data di-graph vector becomes the reference vector for each user of the system.

- More specifically, a random set of antibodies is produced (1,000 in this study), with a shape space identical to that of the enrolment vectors – containing 14 floats, each element of which is assigned a random number on the interval of [0..1]. The antibodies are then allowed to match up in a lock and key fashion with each of the enrolment vectors and a matching score is obtained. This matching score describes the affinity between the antibody and the antigen.

### 5.2.3 Data Processing:

- As a first processing step to generate usable antibodies, those that react with self must be eliminated in order to prevent auto-immune reactions. After the 10 enrollment entries are generated, they are exposed to the antibody pool and the GMS values are computed with respect to each antibody. Those antibodies with GMSs above a threshold,  $\alpha$ , are considered to be activated by ‘self’ antigens (i.e. the authentic user’s enrollment data), and are removed from the antibody pool.
- Only those antibodies that are non-reactive to the enrollment data are kept for subsequent use in the AIS system. Note that this process occurs for each user in the system, as each will produce their own enrollment data. This activity engenders the maturation of the immune system, in which lymphocytes in the thymus become activated by a process of culling out hyper and self-reactive B-cells. Once a processed set of antibodies has been produced, users will enter the authentication (testing) phase, where they will be asked to enter their login ID/password details.
- The same features will be extracted during authentication, and utilised for the authentication purposes. When the user attempts to authenticate, the digraphs will be collected and form their antigen surface that has the same structure as those collected during enrollment. The authentication sample will be exposed to the pool of primed antibodies and the GMS will be obtained for each antibody. If the score is above a threshold (the same  $\alpha$  deployed in the priming stage), then the authentication attempt is classified as rejected, otherwise it will be accepted.

### 5.2.4 Feature Extraction:

- Once a processed set of antibodies has been produced, users will enter the authentication (testing) phase, where they will be asked to enter their login ID/password details. The same features will be extracted during authentication, and utilised for the authentication purposes. When the user attempts to authenticate, the digraphs will be collected and form their antigen surface that has the same

structure as those collected during enrollment. The authentication sample will be exposed to the pool of primed antibodies and the GMS will be obtained for each antibody.

- If the score is above a threshold (the same  $\alpha$  deployed in the priming stage), then the authentication attempt is classified as rejected, otherwise it will be accepted. The reason for this decision is that if the antigen (the authentication sample) is identified by the antibodies (via the GMS score being above threshold), then it must be significantly different from the enrollment samples, as the antibodies were selected based on their low GMS scores.
- If the authentication attempt is not recognised by the antibody pool (that is the GMS is below the threshold), then this sample is considered to be similar to those contained within the enrollment pool. If we stop at this point, then each authentication sample is classified as being produced either by the actual owner (doesn't activate the immune system) or by an imposter (activates the immune system). Knowing the actual identity of the person entering the authentic login details, we can calculate the FAR and FRR of this stage in the AIS system. Further, by varying the acceptance threshold,  $\alpha$ , we can calculate the equal error rate.

#### **5.2.4 Data Classification:**

- Now, those login attempts that are generally classified as accepted fall into one of two categories: true positive (TP) and false positives (FP). Likewise, those attempts rejected fall either into the true negative (TN) or the false negative (FN) class, which can be summarised conveniently by a confusion matrix.
- The true classification rate can be calculated from this data, which is presented in the confusion matrix below. Note this confusion matrix was calculated from a single login account that was checked for FAR and FRR 100 times, selected randomly from the pool of 20 users. Although deploying a supervised approach is not ideal (we would like to make the system as unsupervised as possible), the purpose of this study is to examine how large the antibody pool must be in order to acquire sufficiently high classification accuracy. That is, how useful is the antibody self-reactivity selection process in the deployment of an AIS? To address this question, the classification accuracy was assessed with respect to the number of antibodies, DOC and GMS .
- The number of antibodies varied from 100 to 1,000,000 in (10-fold) increments (results presented in table 3). In order to estimate For this effect, values for the other free parameters are required first.
- The acceptance threshold ( $\alpha$ ) was varied from 0 (requiring a perfect match) down to 0.5 in increments of 0.1, and the resulting classification accuracy was computed. These results are presented in table 5.1.



<b>0.0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>
87.0%	92.4%	84.6%	77.2%	73.9%	62.1%

Table 5.1 Initial results

### **5.3 GA Based Optimized Negative Selection Classification Algorithm (NSCA)**

Proposed selection of features and an optimal reduction in dimensionality of data provides us the most suitable combination for classification of four class motor imagery using the negative selection classification algorithm (NSCA). The reduced feature data is normalized, and the detectors are generated in the reduced low dimensional self-sample space. Otherwise, high dimensional data space can lead to an exponential increase in computational load. Since we have EEG signals pertaining to four classes of human limb movement, we have to generate, optimize, and train four sets of detectors corresponding to each movement class. The detectors are trained using the self-sample (training sample), and the best optimized detector set for the particular class is saved.

The selection of the best set of detectors is based on accuracy. As the best sets of detectors for each class are obtained through training (self) samples, our algorithm computes the classification accuracy for the test samples (unknown data) for all classes.

For training of detectors, knowledge of only self-sample is given to NCSA, and after the detectors are optimized, the best selected set is used for the classification of the test sample. Classification of motor imagery based four human limb movements using NCSA is a four-stage process:

- (1) Selection of combination of neurons in hidden layers;
- (2) Detector generation and optimization;
- (3) Selection of best set of the optimized detectors;
- (4) Classification of the test data set.

H1/H2	Max detection accuracy	Number of detectors
15/6	0.4375	40
15/7	0.4861	30
15/8	0.5069	30
15/9	0.5694	40
15/10	0.6806	30
16/6	0.5694	30
16/7	0.7708	30
<b>16/8</b>	<b>0.8194</b>	<b>40</b>
16/9	0.4931	40
16/10	0.5903	30
17/6	0.4931	30
17/7	0.6181	40
17/8	0.5625	40
17/9	0.4375	40
17/10	0.4861	40
18/6	0.5833	40
18/7	0.5278	30
18/8	0.75	40
18/9	0.5347	30
18/10	0.50	30
19/6	0.5625	30
19/7	0.7986	30
19/8	0.6528	30
19/9	0.5278	40
19/10	0.7083	40

Table 5.2 Comparison of results

Table 5.1 shows the detection accuracies for each combination of hidden layers only for subject 1, bold values show maximum detection accuracy for all combinations of hidden layers. H1 are the number of neurons in first hidden layer and H2 in second hidden layer. The combination of 16/8 is selected with the maximum detection accuracy.

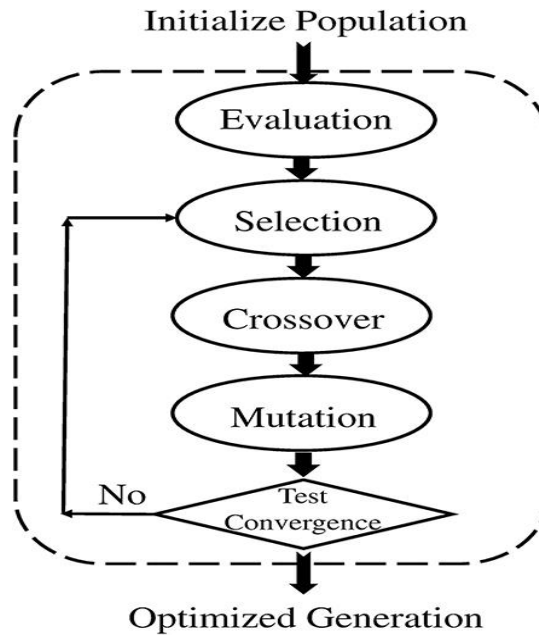


Figure 5.1 Flow of GA

The output of training algorithm is the set of detectors (antibodies) that is used by the classification algorithm.

## CHAPTER 6

### 6.1 Algorithm description

Various features to be extracted to implement the algorithm is calculated using following equations:

- Session time = Starting time – User response time
- Flight time =  $\Sigma$ Releasing time of key 1 –  $\Sigma$ pressing time of key 2
- Dwell time =  $\Sigma$ Releasing time of key –  $\Sigma$ pressing time of same key.
- Keystroke latency =  $\Sigma$ Releasing time of key 1 –  $\Sigma$ pressing time of key 2 per sentence.

After the feature extraction and feature selection phase, the next phase is the classification phase where the matching between the template stored and sample provided during the session takes place. There are various methods used for classification. These classification algorithms are pattern reorganization-based algorithms and decision tree.

#### 6.1.1 New User Registration:

- The process is shown in Figure to the right, for the registration of the first time user. User has to set the username and password along with a long Sample Text already presented to the user.
- The username – password binding will be stored in the database. Also, the Dwell Time and Flight Time of the password entered will also be stored.
- The password, keystroke dynamics (dwell time and flight time) of the password and the sample text will be used for authentication of the user later.

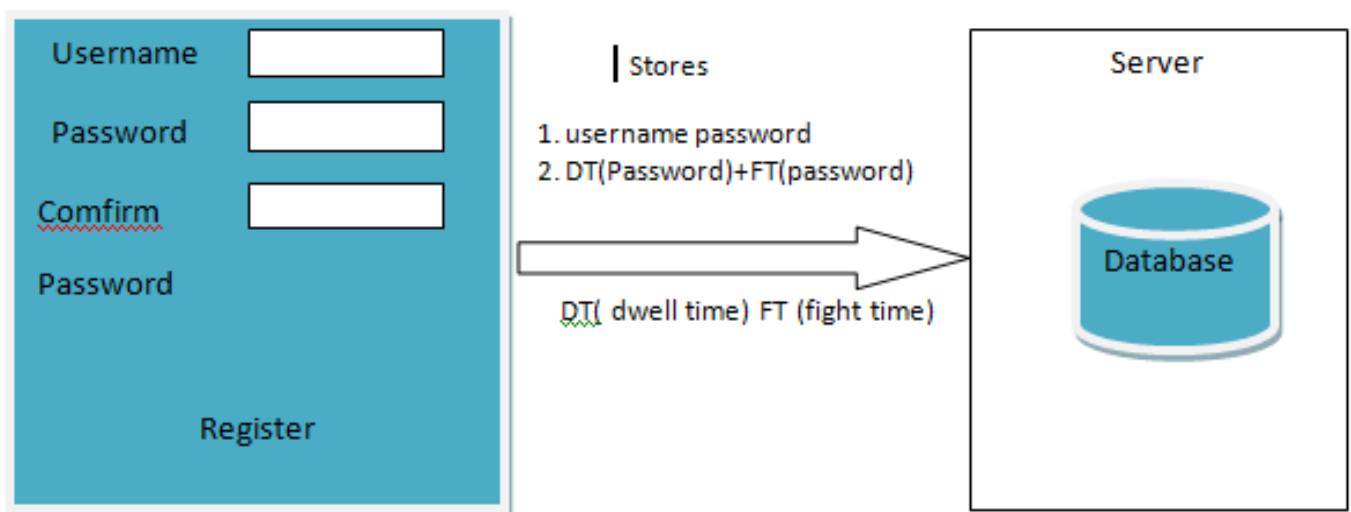


Figure 6.1 Initialization procedure

### 6.1.2 User login procedure:

- When a user wants to login to the system, he needs to enter the username and password which will be checked by the system. Apart from this, the system will also calculate the values of the keystroke dynamics of the entered password and will compare it with the ones stored in the database.
- If the match is within an acceptable limit, then the second level of authentication is done. For the third level of authentication, the user has to type a phrase displayed on the screen in a window which will be a random challenge to the user. The keystroke dynamics of this challenge will be compared with that of stored values of the Sample Text from the Registration phase.

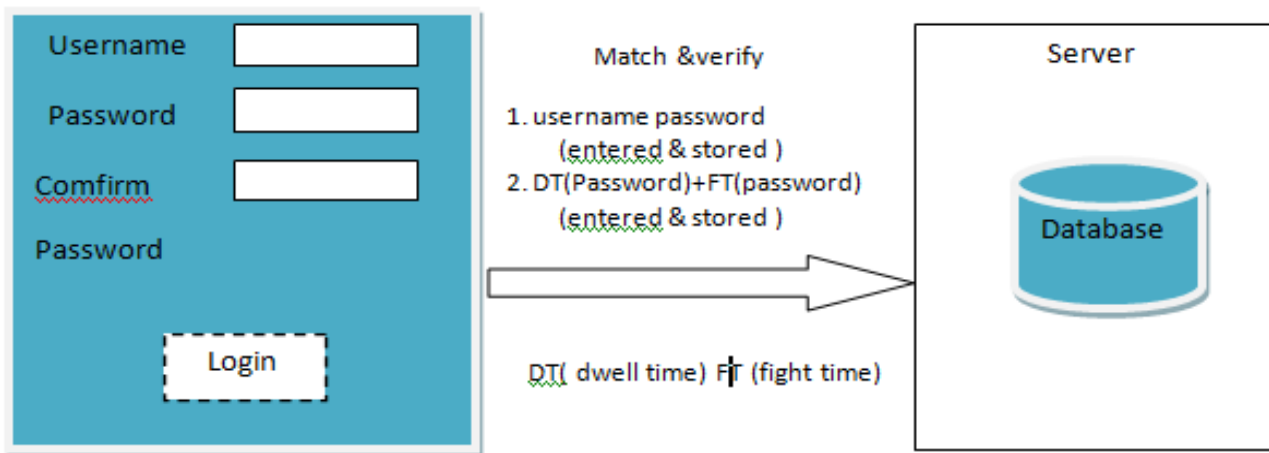


Figure 6.2 Verification Phase

## 6.2 Implementation description

### 6.2.1 Registration Module:

- Enter Username: It is checked in the username module and if username already exists, user has to change its username and register with a username which is available.
- Enter Password: After entering a unique username, user is supposed to enter his choice of password and the password is encrypted and stored in a module known as password module

and along with password it also stores the factors (flight time, dwell time and total time taken to enter the password) of keystroke authentication.

- Registration: A registration module is created which links to various modules that are username, password.

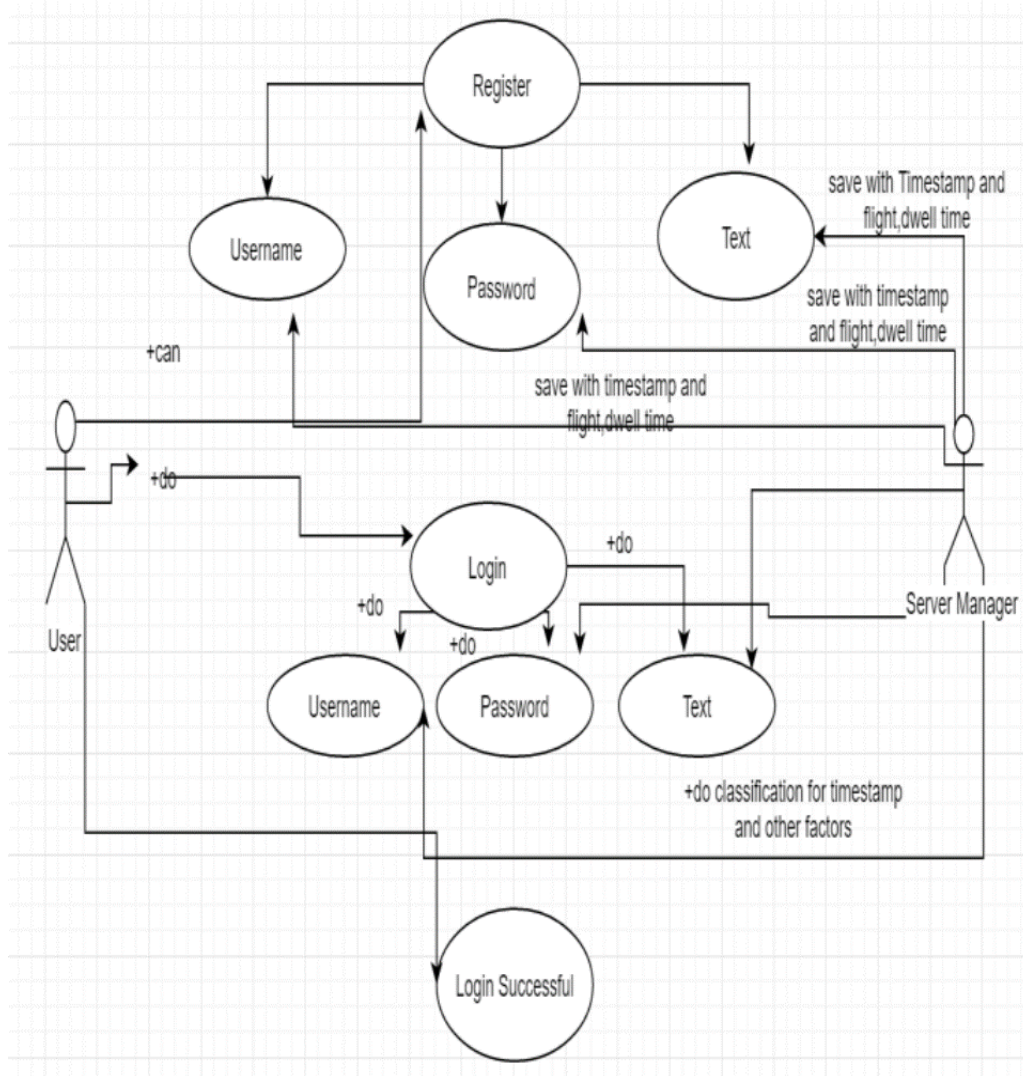


Figure 6.3 Registration model

### 6.2.2 Login Phase:

- User enters his username and password: Username and password are matched with username and password module, if username matches, corresponding password is checked and if it matches the biometric factors of keystroke authentication stored in module, it proceeds to next step of login.

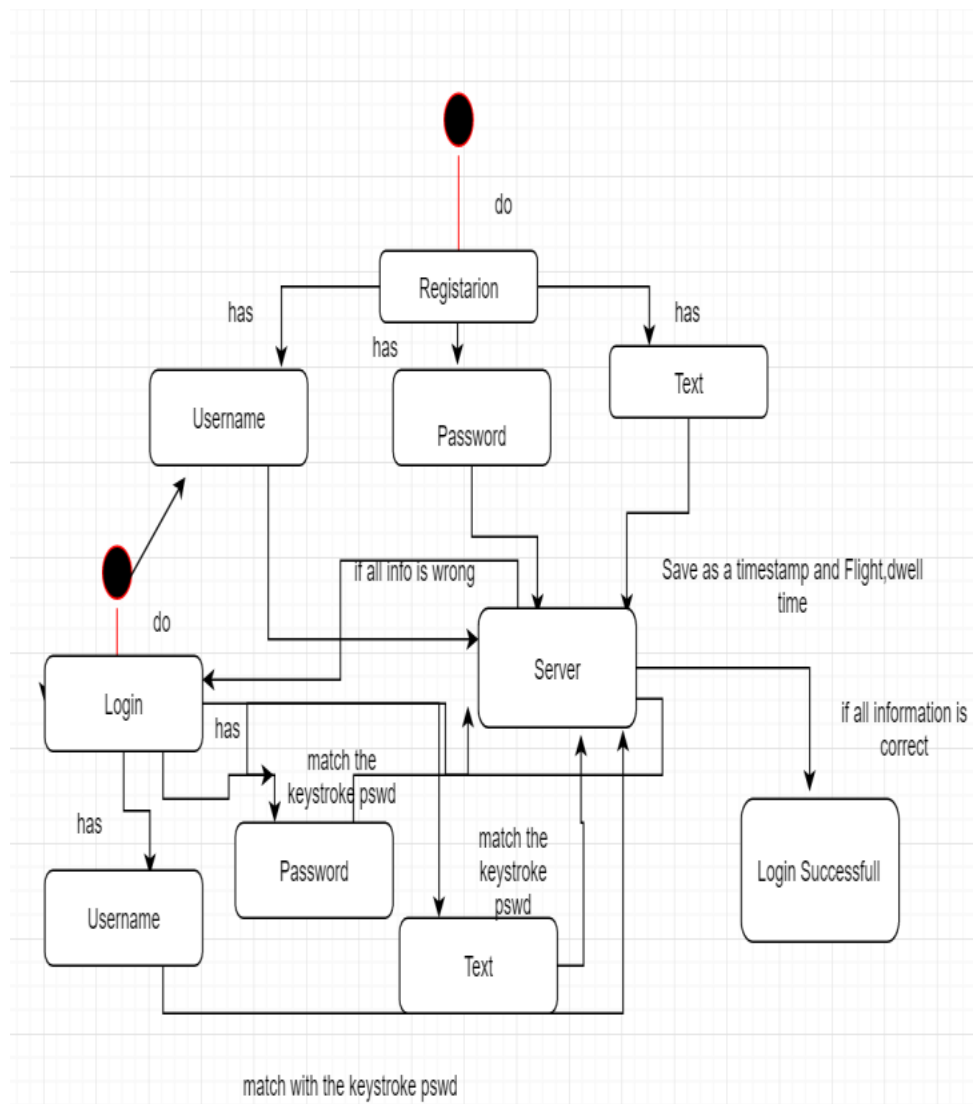


Figure 6.4 Login Phase

- . In order to match the factors, we need to temporarily store the flight time, dwell time and total time when user enters his password during login in the login module.
- If all factors and details match, login is successful

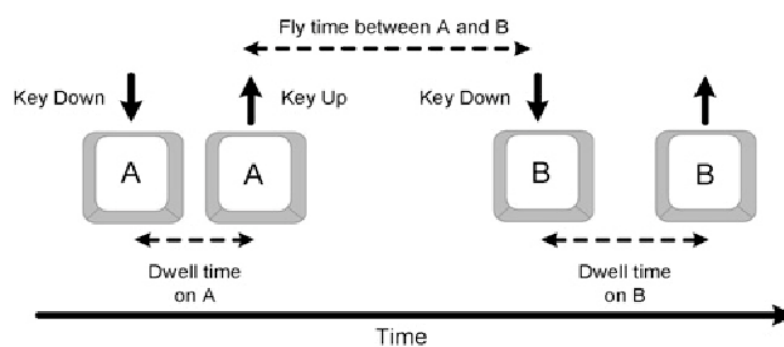


Fig 6.5 Basic working

## CHAPTER 7

### 7.1 Model Execution

#### 7.1.1 Data Import:

The following lines of code imports important libraries and the dataset to be used for the Dynamic Authentication using key-strokes.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import utils

plt.style.use('ggplot')
%matplotlib inline
df = pd.read_csv('data/DSL-StrongPasswordData.csv')
subject = df['subject']
df.head()
```

	subject	sessionIndex	rep	H.period	DD.period.t	UD.period.t	H.t	DD.t.i	UD.t.i	H.i	...	H.a	DD.a.n	UD.a.n	H.n	DD.n.l	UD.n.l	H.l	DD.l.Return	UD.l.Retu
0	2	1	1	0.1491	0.3979	0.2488	0.1069	0.1674	0.0605	0.1169	...	0.1349	0.1484	0.0135	0.0932	0.3515	0.2583	0.1338	0.3509	0.21
1	2	1	2	0.1111	0.3451	0.2340	0.0694	0.1283	0.0589	0.0908	...	0.1412	0.2558	0.1146	0.1146	0.2642	0.1496	0.0839	0.2756	0.19
2	2	1	3	0.1328	0.2072	0.0744	0.0731	0.1291	0.0560	0.0821	...	0.1621	0.2332	0.0711	0.1172	0.2705	0.1533	0.1085	0.2847	0.17
3	2	1	4	0.1291	0.2515	0.1224	0.1059	0.2495	0.1436	0.1040	...	0.1457	0.1629	0.0172	0.0866	0.2341	0.1475	0.0845	0.3232	0.23
4	2	1	5	0.1249	0.2317	0.1068	0.0895	0.1676	0.0781	0.0903	...	0.1312	0.1582	0.0270	0.0884	0.2517	0.1633	0.0903	0.2517	0.16

5 rows × 34 columns

Figure 7.1 Imported Data

#### 7.1.2 Data Set Exploration:

In this section we will explore the various feature sets of the keystroke dynamics data set.

##### 7.1.2.1 Latency (Keydown-Keydown)

A classical keystroke feature vector which consists of the durations between the key-down of one key to the key-down of the next key, denoted in this data set as DD.

```

df_plot = data['DD'].copy()

plt.figure(figsize=(16, 9))

for i, y0 in enumerate(np.unique(y)[:6]):
    plt.subplot(2, 3, i + 1)
    for x in df_plot[y == y0].values:
        plt.plot(x, color='C1', alpha=0.1)
    plt.title('Latency for Class {}'.format(y0))
    plt.xlabel('Key Index')
    plt.ylabel('Duration')
plt.tight_layout()

```

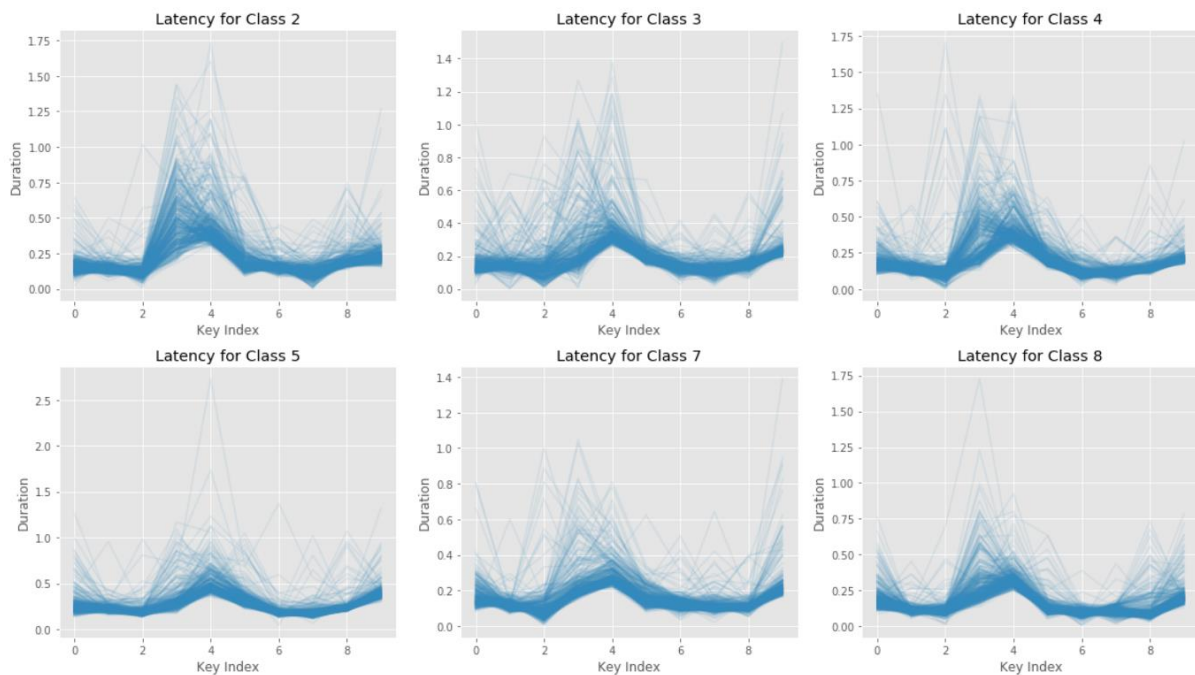


Figure 7.2 Latency

### 7.1.2.2 Average Latency per Class

We calculate the average latency per class to understand the difference between the different classes of users.

```

df_plot = data['DD'].copy()
df_plot['subject'] = y
df_plot = df_plot.groupby('subject').mean()

```



```
df_plot.iloc[:6].T.plot(figsize=(16, 9), title='Average Latency')
plt.xlabel('Key Index')
plt.ylabel('Duration');
```

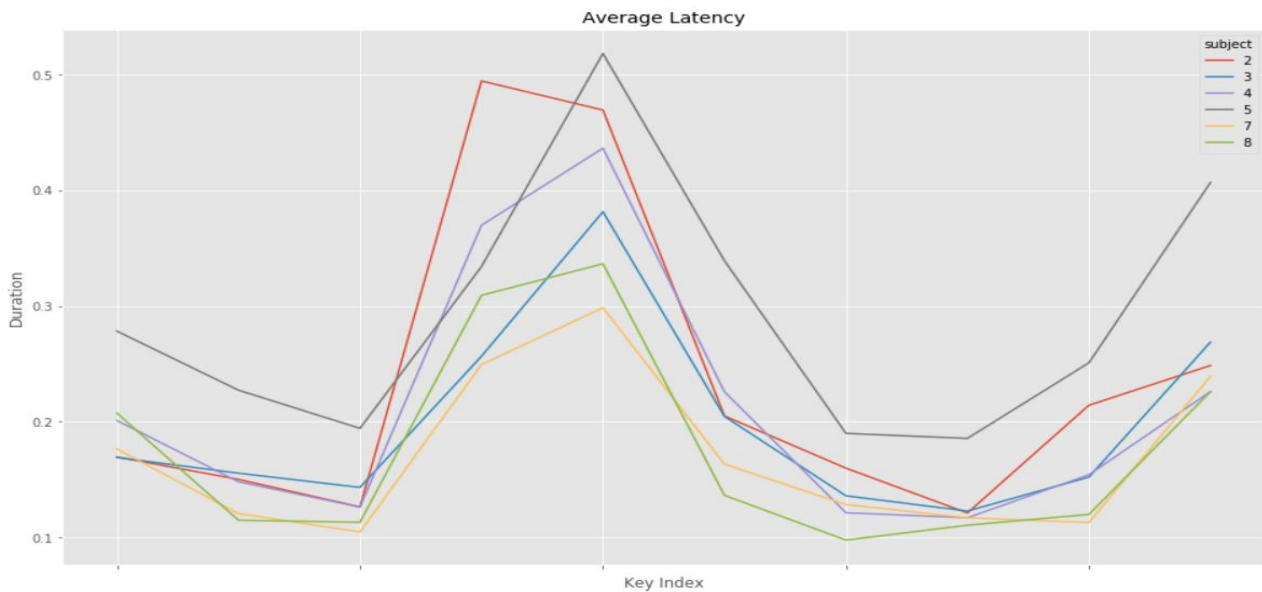


Figure 7.3 Average Latency

### 7.1.2.3 Pressure Duration (Hold)

This feature vector consists of the durations of key-down to key-up of a single key and is denoted as H in the data set.

```
df_plot = data['H'].copy()

plt.figure(figsize=(16, 9))
for i, y0 in enumerate(np.unique(y)[:6]):
    plt.subplot(2, 3, i + 1)
    for x in df_plot[y == y0].values:
        plt.plot(x, color='C1', alpha=0.1)
    plt.title('Pressure Duration for Class {}'.format(y0))
    plt.xlabel('Key Index')
    plt.ylabel('Duration')
plt.tight_layout()
```

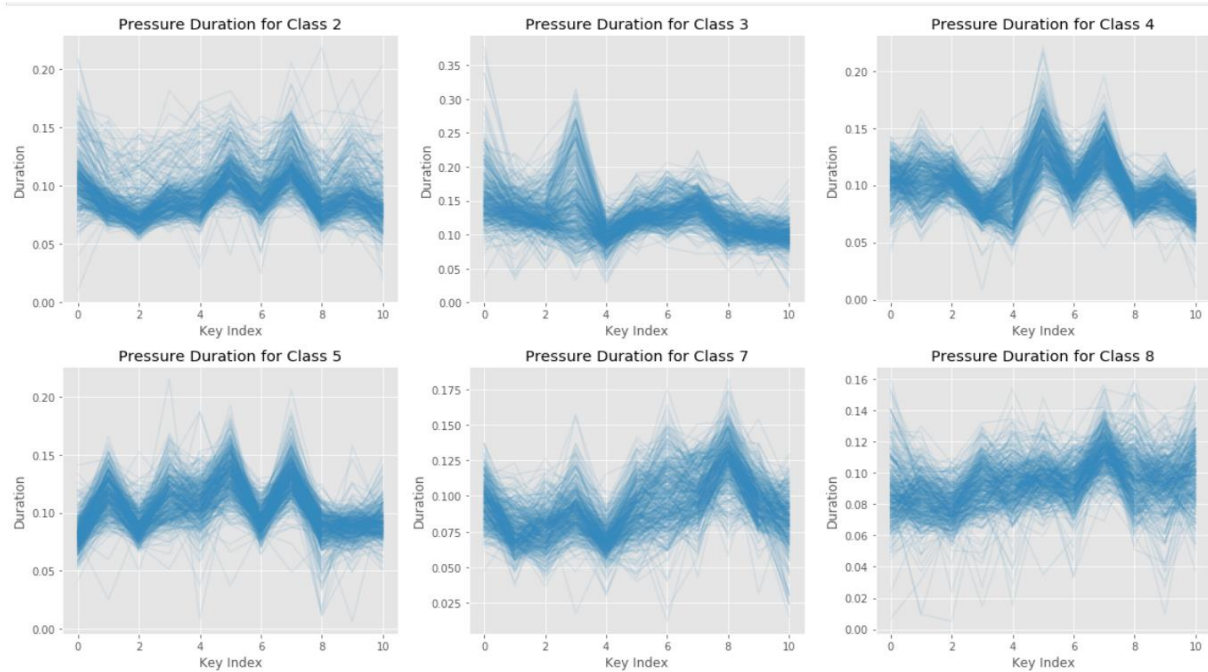


Figure 7.4 Pressure data

#### 7.1.2.4 Average Pressure Duration per Class

```
df_plot = data['H'].copy()
df_plot['subject'] = y
df_plot = df_plot.groupby('subject').mean()
df_plot.iloc[:6].T.plot(figsize=(16, 9), title='Average Pressure Duration')
plt.xlabel('Key Index')
plt.ylabel('Duration');
```

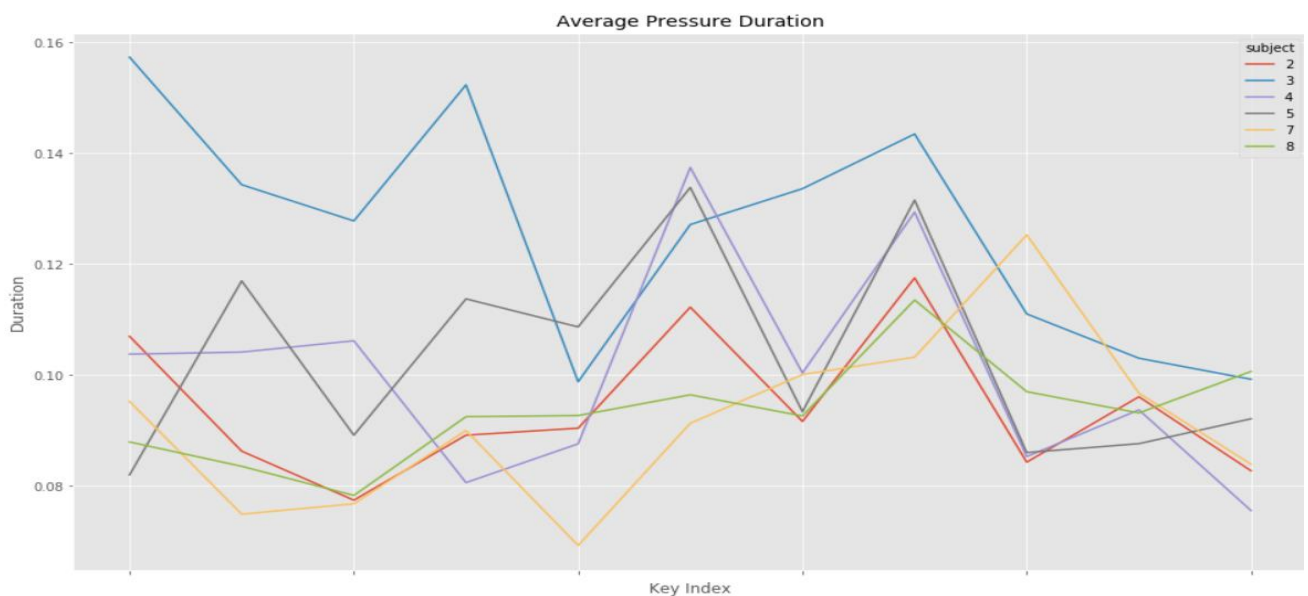


Figure 7.5 Average Pressure Data

### 7.1.2.5 Keyup-Keydown

Duration between the release of one key and the press of the next key.

```
df_plot = data['UD'].copy()
```

```
plt.figure(figsize=(16, 9))
```

```
for i, y0 in enumerate(np.unique(y)[:6]):
```

```
    plt.subplot(2, 3, i + 1)
```

```
    for x in df_plot[y == y0].values:
```

```
        plt.plot(x, color='C1', alpha=0.1)
```

```
    plt.title('Keyup-Keydown for Class {}'.format(y0))
```

```
    plt.xlabel('Key Index')
```

```
    plt.ylabel('Duration')
```

```
plt.tight_layout()
```

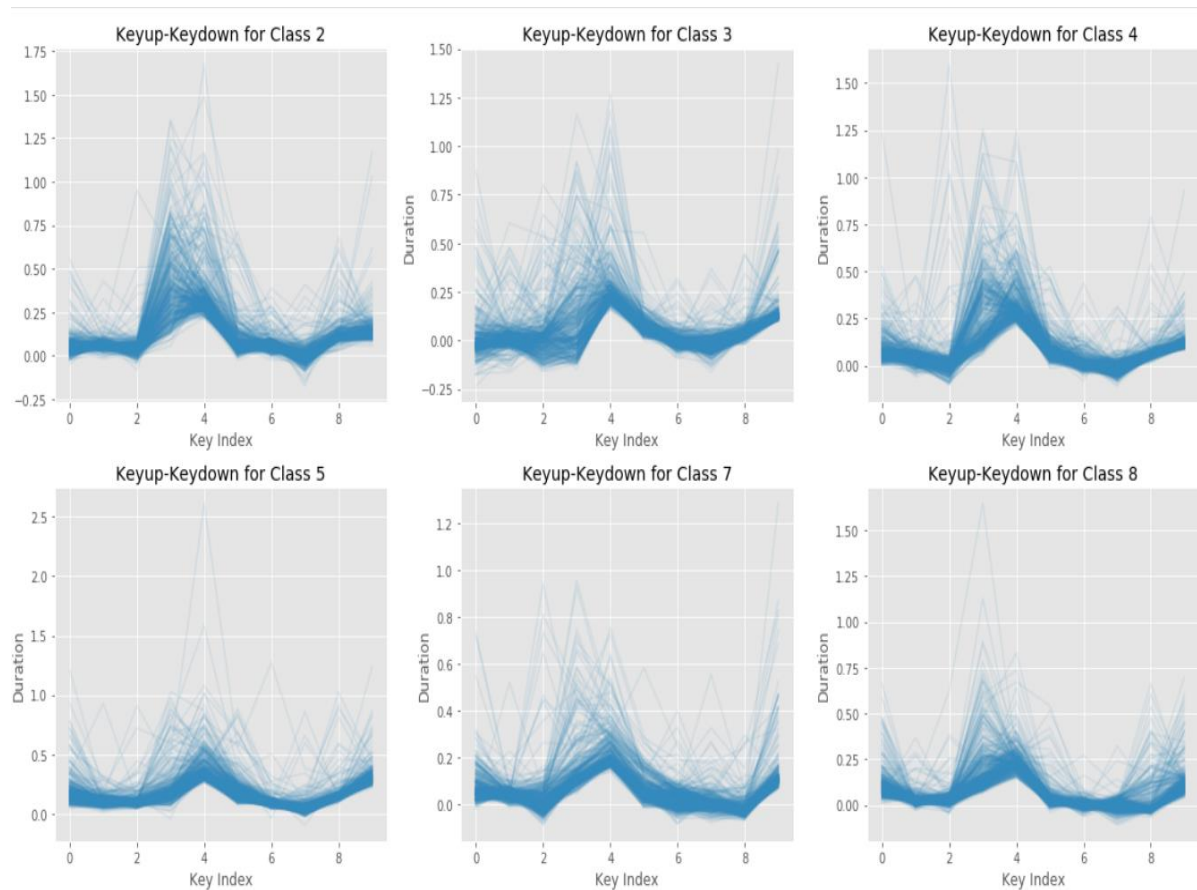


Figure 7.6 Keyup-keydown Variance

### 7.1.2.6 Average Keyup-Keydown Duration per Class

```
df_plot = data['UD'].copy()
df_plot['subject'] = y
df_plot = df_plot.groupby('subject').mean()
df_plot.iloc[:6].T.plot(figsize=(16, 9), title='Average Keyup-Keydown Duration')
plt.xlabel('Key Index')
plt.ylabel('Duration');
```

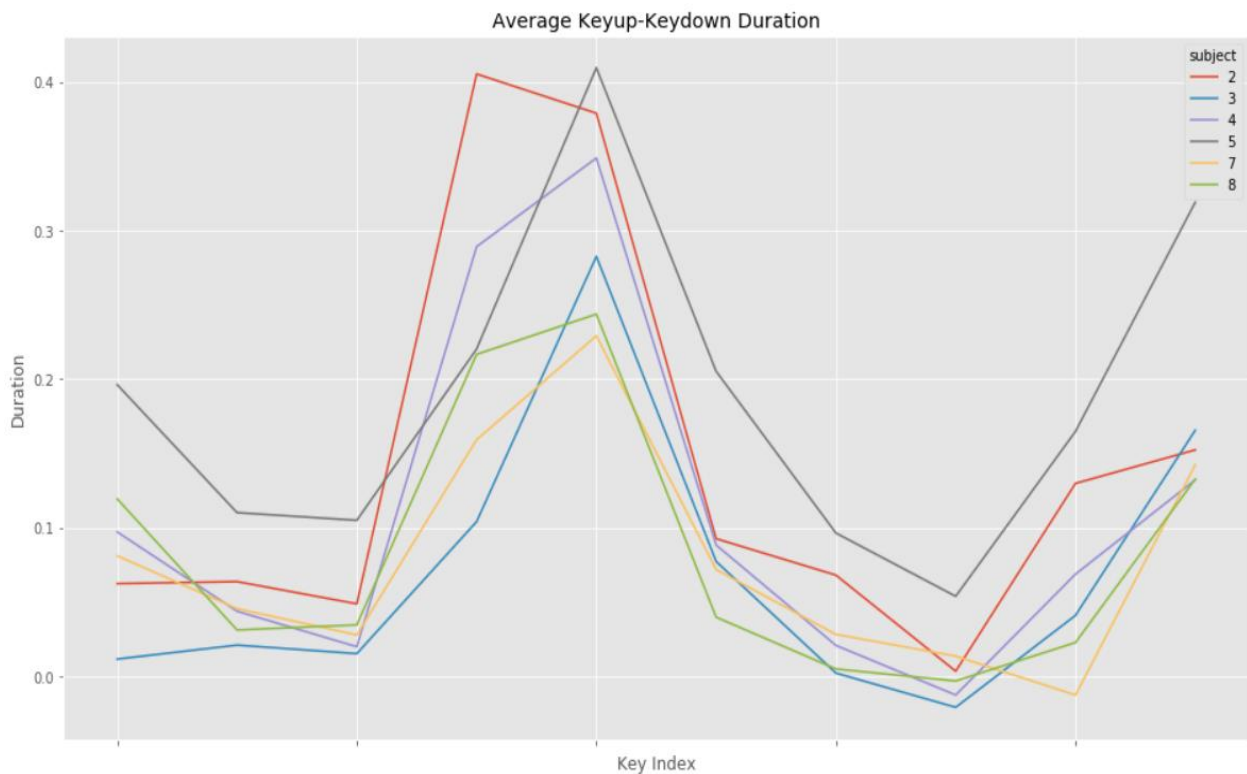


Figure 7.7 Keyup-keydown Average duration

### 7.1.3 Data Clustering With T-SNE

T-Distributed Stochastic Neighbour Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets.

```
from sklearn.manifold import TSNE
```

```
# Create the subset of the first 10 classes
```

```

classes = np.unique(y)[:10]

mask = [y0 in classes for y0 in y]

X_tsne = data['total'].values[mask]

y_subset = y[mask]

tsne = TSNE(n_components=2, learning_rate=1000, perplexity=8)

X_embedded = tsne.fit_transform(X_tsne)

plt.figure(figsize=(16, 12))

for y0 in classes:
    label = 'Class {}'.format(y0)
    plt.plot(X_embedded[y_subset == y0][:, 0],
             X_embedded[y_subset == y0][:, 1],
             'o', label=label)

plt.title('T-SNE of all features for the first 10 Classes')

plt.legend();

```

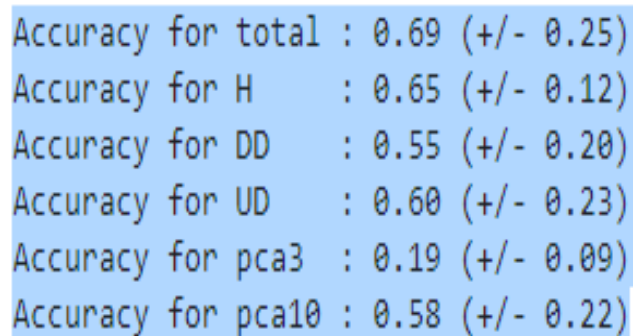


Figure 7.8 Data Clustering

### 7.1.4 Classify the Data Set with Negative Selection Classification Algorithm

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
clf = KNeighborsClassifier()

for key in data:
    scores = cross_val_score(clf, data[key], y, cv=5)
    print("Accuracy for {s} : {0.2f} (+/- {0.2f})".format(
        key, scores.mean(), scores.std() * 2))
```



```
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)
```

Figure 7.9 Negative selection

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV

X = data['total']
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=1, stratify=y)

n_neighbors = [2, 3, 4, 5, 6, 7, 8, 9, 10]

parameters = dict(n_neighbors=n_neighbors)
```

```

clf = KNeighborsClassifier()

grid = GridSearchCV(clf, parameters, cv=5)
grid.fit(X_train, Y_train)

# Get results
results = grid.cv_results_

# Following code snippet adapted from:
# http://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_randomized\_search.html
for i in range(1, 4):
    candidates = np.flatnonzero(results['rank_test_score'] == i)
    for candidate in candidates:
        print("Model with rank: {}".format(i))
        print("Mean validation score: {0:.3f} (std: {1:.3f})".format(
            results['mean_test_score'][candidate],
            results['std_test_score'][candidate]))
        print("Parameters: {}".format(results['params'][candidate]))
        print()

```

### 7.1.5 Show ROC Curve, AUC and EER

```

from sklearn.metrics import accuracy_score, roc_curve, auc

Y_pred = grid.predict(X_test)
print('Test accuracy : {}'.format(accuracy_score(Y_test, Y_pred)))

from scipy.optimize import brentq
from scipy.interpolate import interp1d

Y_pred = grid.predict(X_test)
fpr, tpr, threshold = roc_curve(Y_test.ravel(), Y_pred.ravel())

# Calculate equal-error-rate

```

```
eer = brentq(lambda x : 1. - x - interp1d(fpr, tpr)(x), 0., 1.)
```

```
plt.figure(1)
```

```
plt.plot([0, 1], [0, 1], 'k--')
```

```
plt.plot(fpr, tpr, label='AUC = {:.3f}, EER = {:.3f}'.format(auc(fpr, tpr), eer))
```

```
plt.xlabel('False positive rate')
```

```
plt.ylabel('True positive rate')
```

```
plt.title('ROC curve')
```

```
plt.legend(loc='best')
```

```
plt.show()
```

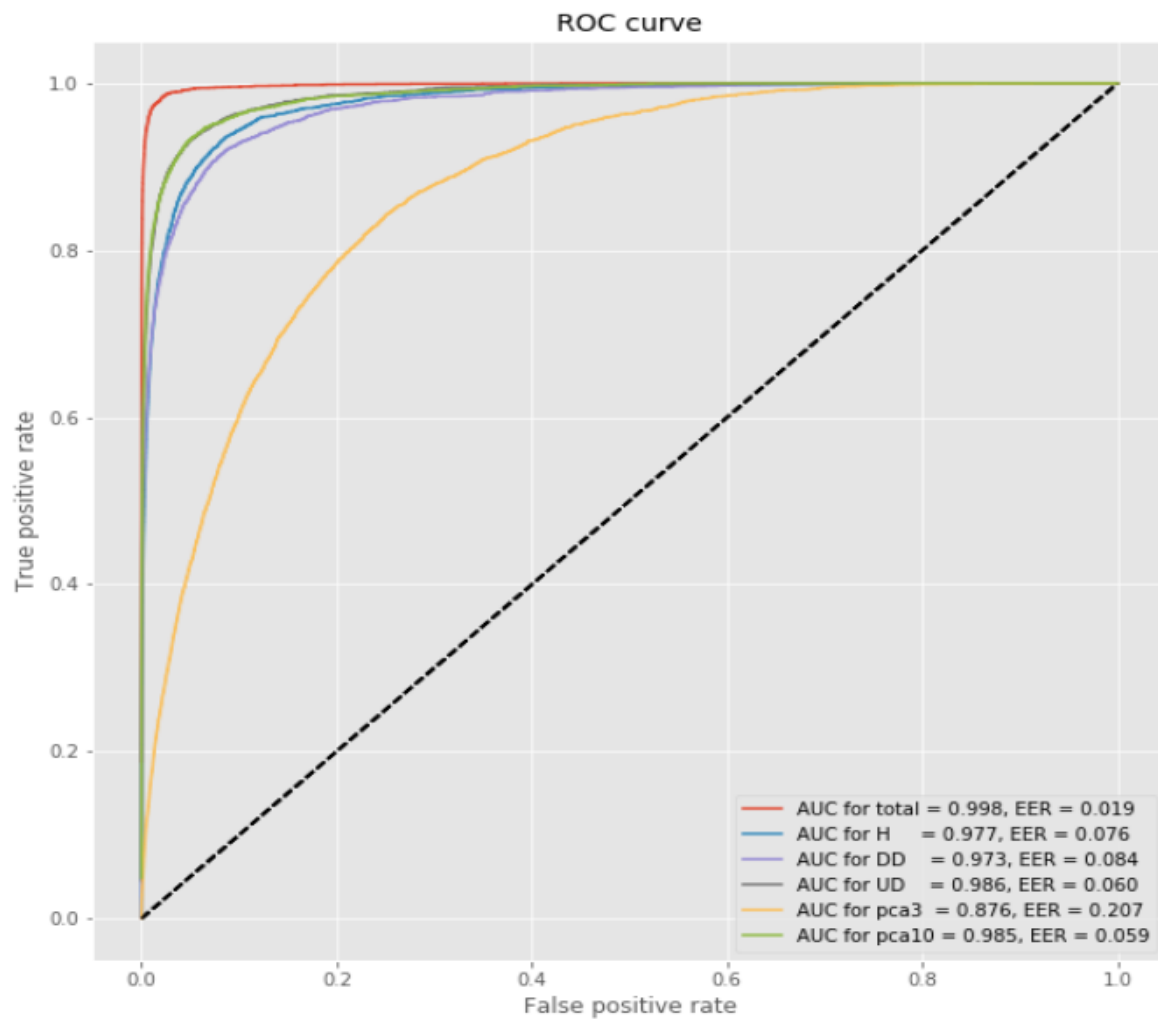


Figure 7.10 ROC Diagram



### 7.1.5.1 ROC Curve for each Class

Here we want to compare the ROC Curves for each class and how much variation can be seen within each data set.

```
plt.figure(figsize=(15, 20))
for k, key in enumerate(key_list):

    X = data[key]
    X_train, X_test, Y_train, Y_test = train_test_split(
        X, Y, test_size=0.2, random_state=1, stratify=y)

    model = models_dict[key][300]
    Y_pred = model.predict(normalize(X_test))

    fpr_dict, tpr_dict = {}, {}
    for i in range(n_classes):
        fpr_dict[i], tpr_dict[i], threshold = roc_curve(Y_test[:, i], Y_pred[:, i])

    plt.subplot(3, 2, k + 1)
    for i in range(n_classes):

        plt.plot([0, 1], [0, 1], 'k--')
        plt.plot(fpr_dict[i], tpr_dict[i], color='C1', alpha=0.4)

    plt.xlabel('False positive rate')
    plt.ylabel('True positive rate')

    plt.title('ROC curve for {}'.format(key));
plt.tight_layout()
```

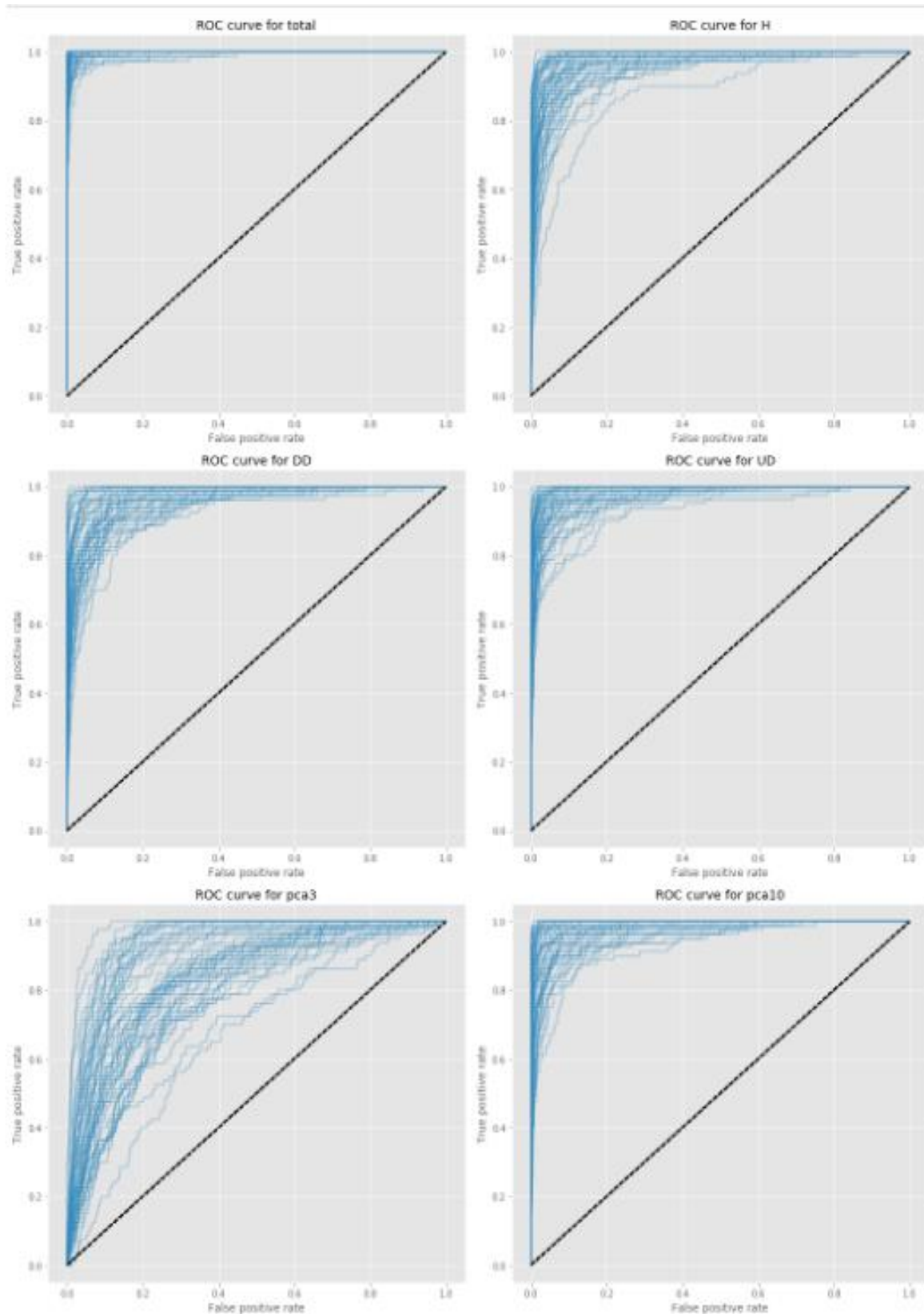


Figure 7.12 ROC Of each Class

### 7.1.6 Normalized Confusion Matrix

Calculate the normalized confusion matrix of the data set. The diagonal represents the normalized number of points where the predicted label is equal to the true label. The off-diagonal elements are the mislabelled elements by the model. The normalization is based on the number of elements for each class. In this case it would not make visually a difference since the data set is perfectly balanced for each class.

```
from sklearn.metrics import confusion_matrix

plt.figure(figsize=(15, 20))
for i, key in enumerate(key_list):
    X = data[key]
    X_train, X_test, Y_train, Y_test = train_test_split(
        X, Y, test_size=0.2, random_state=1, stratify=y)

    model = models_dict[key][300]
    Y_pred = model.predict(normalize(X_test))

    C = confusion_matrix(Y_test.argmax(axis=1), Y_pred.argmax(axis=1))

    # Normalize confusion matrix
    C = C.astype('float') / C.sum(axis=1)[:, np.newaxis]

    plt.subplot(3, 2, i + 1)
    plt.imshow(C, vmin=0, vmax=1)
    plt.colorbar()
    plt.title('Confusion Matrix for {}'.format(key))
plt.tight_layout()
```

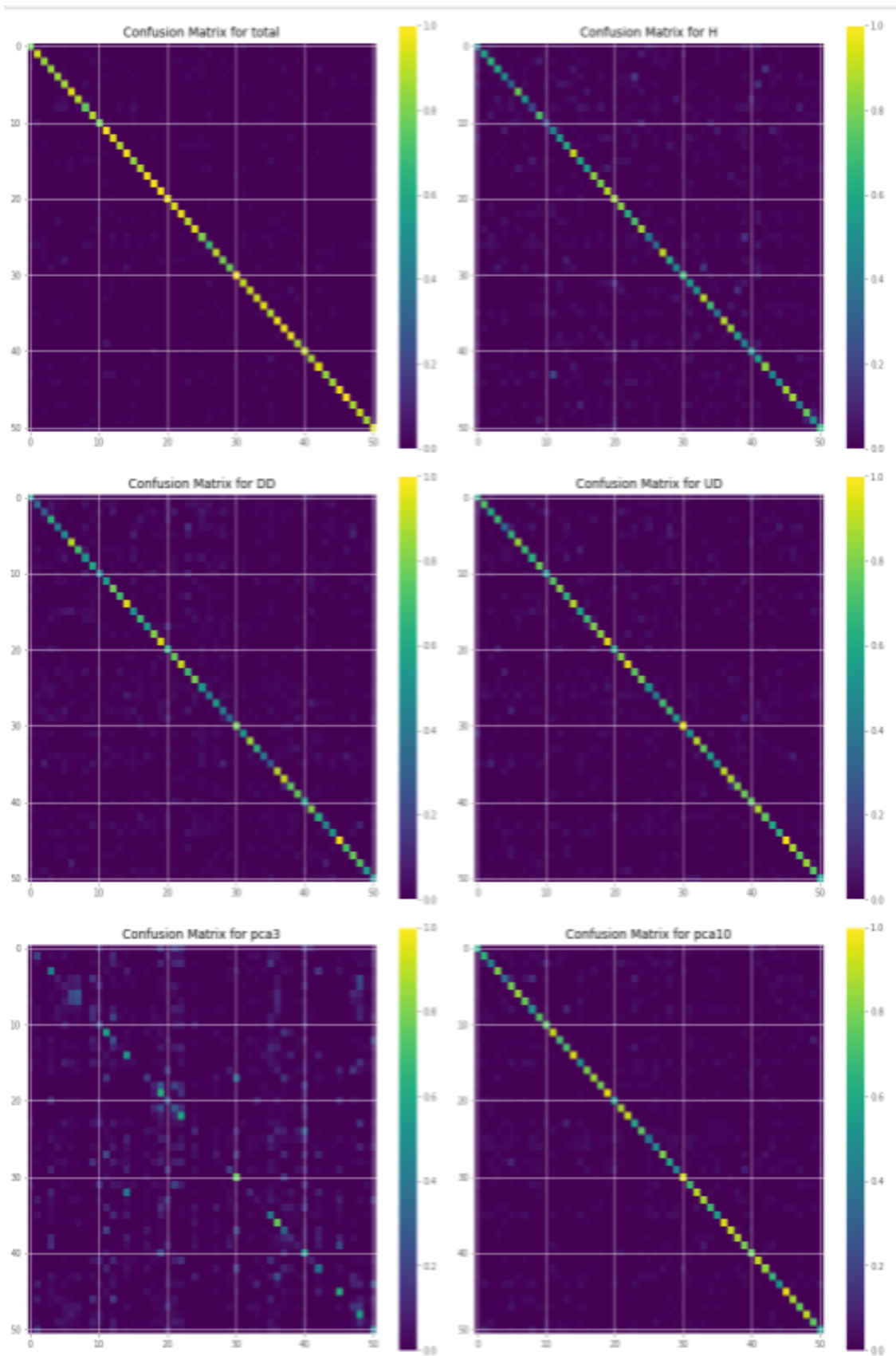


Figure 7.11 Nomenclature predictions

## CHAPTER 8

### 8.1 Future Enhancements

Extensive researches have to be undertaken to ensure that the usability and flexibility of Keystroke based authentication systems are at par or better than existing authentication systems like Fingerprint, Pin code and Face id system.

The major factors to be scrutinized are: -

- The initial time taken to acclimatize with the system specially to train the system is what is holding back Keystroke authentication systems compared to its major opponents like Fingerprint systems which can be initially set up within 3 mins, the average time taken to set up keystrokes is almost 15 mins. Therefore, as a part of future enhancements its completely necessary that we address this issue and find out possible solutions to decrease the initial set up time.
- As a part of future researches, we are also interested in pondering about innovative methods to integrate keystroke to the world of cryptocurrencies and blockchains, by doing this we are attempting to make the common man feel a bit more secured regarding his credentials while using crypto applications, moreover we are trying to enhance an already well-built security system.
- The need promotes a new technology always plays a major role in ensuring that the newly created initiative finds its roots deep into the society, with that thought in mind we have made it our pivotal priority to promote the importance's and advantages of keystroke-based authentication systems to the common public by providing various free trials and by trying to establish partnerships with companies that focusses on cybersecurity and integrating keystrokes with their existing solutions.

## **CHAPTER 9**

### **Conclusion**

Adding to our initial problem Statement, after referring to more than healthy number of Literature Surveys, We have identified the following Limitations: -

- Most of the existing Systems are extremely hard to train and the Data set required for training the system are really complex and are needed in a large number. In most of the Literature Surveys we referred the systems developed needed at least 50 or more varying inputs from a single user to have him registered in the system.
- Such a tedious initial registration procedure will make this new authentication system less desirable to the modern tech World.
- Therefore, through our Project we are aiming to develop a system that has a much more efficient training setup. A System that will be able to register a user at the same speed and efficiency as other existing authentication systems
- Another major issue we were able to dig out was that there is very little amount of research done in Implementing Keystrokes in the Android and IOS operating systems.
- Hence, we are also making an effort to develop a system that will be hybrid enough to be implemented in the Android and IOS platforms as doing this will help us to extend our possibilities in the business world
- Moreover, the Need of the hour is to create system that will be able to work seamlessly with the existing hardware with requiring any extra technical support systems, Through this we aim to deliver our solution to everyone who has a normal keyboard or a touchpad.

## CHAPTER 10

### References

- [1] M.Karnan, M.akila (March 2011) biometric personal authentication using keystroke dynamics, *Applied Soft Computing*, Volume 11,pp. 1565–1573.
- [2] Saurabh Singh ,K. V. Arya (June 2012), key classification: a new approach in free text keystroke authentication system, *Third Pacific-Asia Conference on Circuits, Communications and System (PACCS)*.
- [3] Yu Zhong , Yunbin Deng , Anil K. Jain (July 2012), keystroke dynamics for user authentication , *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*.
- [4] Praveen Kumar, Shagun Seth, Kanishka Bajaj, Seema Rawat (November 2019), diverse security practices and comparison on key stroke dynamics, *8th International Conference System Modeling and Advancement in Research Trends* .
- [5] Nick Bartlow ( March 2015 ), username and password verification through keystroke dynamics, *West Virginia University Graduate thesis*.
- [6] Akshita Gupta , Arun Kumar (December 2021), keystroke dynamics , *Lappeenranta University of Technology , finaland journal ,vol 36 ,pp. 123-142*.
- [7] Saleh Bleha, Charles Slivinsky, and Bassam Hussien (January 2000), *Computer-Access Security Systems Using Keystroke Dynamics , IEEE Transactions on Pattern Analysis and Machine Intelligence ,Volume: 12, Issue: 12,pp.1217-1222*.
- [8] Fabian Montrose, Michael K. Reiter & Susanne Wetzel (April 2016), Password hardening based on keystroke dynamics, *International Journal of Information Security ,Volume: 1, pp.69–83*.
- [9] Heather Crawford (April 2015), *Keystroke Dynamics: Characteristics and Opportunities*, *Eighth International Conference on Privacy, Security and Trust*.
- [10] Pin Shen Teh , Andrew Beng , Jin Teoh , and Shigang Yue ( Feb 2015 ), *A Survey of Keystroke Dynamics Biometrics*, *The Scientific World Journal*,Volume 2013,pp.1-24.
- [11] Paulo Henrique Pisani & Ana Carolina Lorena (April 2016),*A systematic review on keystroke dynamics*, *Journal of the Brazilian Computer Society*, volume-19 pp.573–587 .

- [12] H. Saevanee , P. Bhattarakosol (April 2015), Authenticating User Using Keystroke Dynamics and Finger Pressure, 6th IEEE Consumer Communications and Networking Conference.
- [13] Salil P. Banerjee , Damon L. Woodard ( November 2015 ), Biometric Authentication and Identification using Keystroke Dynamics, Journal of Pattern Recognition Research, Volume-7(1) pp.116-139.
- [14] Xiao feng Lu , Sheng fei Zhang , Pan Hui,Pietro Lio (April 2019), Continuous authentication by free-text keystroke based on CNN and RNN,Procedia Computer Science,Volume 147, 2019, pp. 314-318.
- [15] MargitAntal , László Zsolt Szabó , Izabella László ( Feb 2015 ), Keystroke Dynamics on Android Platform, Procedia Technology,Volume -19, pp. 820-826.