User Authentication from Mouse Movement Data Using Multi Classifier

Masuda Karim

M. Phil Researcher

Department of Computer Science & Engineer

University of Dhaka.

As the candidate’s supervisor, I have approved this dissertation for submission.

Name: Professor Dr. Md. Hasanuzzaman

Signed:___________________________

Declaration

We declare that this thesis entitled "User Authentication from Mouse Movement Using Multi Classifier" isdone by me under the supervision of Dr. Md. Hasanuzzaman, Professor,Department of Computer Science and Engineering, University of Dhaka. We have confirmed that the full part of the work is done during this MPhil research study in the University of Dhaka, Bangladesh. We have also declared that any part of this thesis has not previously been submitted for awarding of any degree or other qualification. We have discussed related published works of others with appropriate references and this thesis work is done entirely by us and our contributions and enhancements from other works are clearly stated.

Candidate:

MasudKarim

Signed: ____________________________

Supervisors:

Professor Dr. Md. Hasanuzzaman

Signed: ____________________________

Abstract

User authentication is a process to verify the identity of someone who connects to system or resource. There are many technologies to authenticate a user. The biometric authentication system is becoming very popular due to its unique characteristics. This thesis presents a user authentication system from the mouse movement data. The mouse movement data are captured using our own developed user interface and two tools named Jitbit macro reader and Recording User Input (RUI). Raw data are sampled into blocks in two ways: one is based on specific number of action and another is based on specific duration. These data blocks are stored in database. From each blocks, twelve features are generated: Number of Points in the Trajectory, Delay Time, Number of Delay, Number of Action, Standard Deviation of Trajectory Length, Total Length of Trajectory, Standard Deviation of Slope, Standard Deviation of Difference Between Each of Slopes, Number of Curvatures, Curvature of Trajectory, Number of Changes in Horizontal Position and Number of Changes in Vertical Position. This system uses three classifiers: Support Vector Machine, K-Nearest Neighbor and Naïve Bayes separately to verify the proposed authentication system. The system is trained and tested using our captured dataset of 10 users and a benchmark dataset of 28 users. The experimental result shows that K-Nearest Neighbor based classifier performs better in terms of Average Receiver Operating Characteristic (ROC) Area, False Acceptance Rate (FAR) and False Rejection Rate (FRR). We have found FAR=2.78 and FRR=0 by using our collected own data and FAR=1 and FRR=1.2 by using benchmark data. This system is compared with S. Suganya, G. Muthumari, and C. Balasubramanian's research and found that both FAR and FRR is improved.

Acknowledgement

First of all, I wish to thanks to Almighty Allah who is the most Gracious and Merciful.

Special innumerous thanks to my supervisors Dr. Md.Hasanuzzaman, Professor, Department of Computer Science & Engineering for his extraordinary guidelines in my research work. It is a great privilege for me to work with such a devoted and inspiring professor. I am really lucky to work under supervision of Professor Dr. Md. Hasanuzzaman.

I would further like to send my heartfelt thanks to Professor Dr. Shabbir Ahmed, Chairman, Department of Computer Science and Engineering in the University of Dhaka for their exemplary technical assistance and invaluable support over the years.

I would like to thanks to honorable faculty members of department of Computer Science and Engineering, University of Dhaka. They have influence me in my research work. I have taken advice and suggestions from them. I have also shared my idea with them and have taken better idea from the department.

I am thankful to lab supervisor, office clerks, assistant and other staffs of department for their support. I thanks to staffs and officer of registrar section of the University. I am grateful too to students and other research fellow of the department.

I acknowledged ICT Division, Ministry of Posts, Telecommunications and IT, Government of the People's Republic of Bangladesh for awarding me a fellowship for my MPhil research work.

MasudKarim
September 2018

Table of Contents

List of Figures

List of Tables

List of Acronyms

FAR	:	False Acceptance Rate
FRR	:	False Rejection Rate
ROC	:	Receiver Operating Characteristic
RUI	:	Recording User Input
ThV	:	Threshold Value
ERR	:	Equal Error Rate
DI	:	Duplicate Identification
PIN	:	Personal Identification Number
MDA	:	Average Movement Speed Per Movement Direction
STDEV	:	Standard Deviation
NoSF	:	Number of Satisfied Feature
SVM	:	Support Vector Machine
kNN	:	k Nearest Neighbor
SMO		Sequential minimal optimization
AR	:	Root of Average
WEKA	:	Waikato Environment for Knowledge Analysis
ARFF	:	Attribute Relation File Format
ASCII	:	American Standard Code of Information Interchange
UI	:	User Interface
FERET	:	Face REcognition Technology
XM2VTSDB	:	The Extended Multi Modal Verification for Tele Service and Security Database

Chapter – 1
Introduction

1.1 Overview

User authentication is a verification process by which a user is identified based on identity. This process is a confirmation of a user's authenticity whenever accessing resources. This is very important in security issue due to vast increasing of network, systems and resources. Now a day, we are using password or PIN to authenticate a user. Password or PIN can be stolen or one can provide his/her password to others for completing task in evil way.Sometimes users use weak passwords which don't create strong protection and can be discovered easily. To make the system more authentic sometimes we use physical biometric authentication systems. Physical biometric (iris, fingerprint, retinal, face etc) authentication needs extra devices and these are complex system. To identify the theft and valid user is important issue for an organization. Sometimes it is difficult to identify false user and legitimate user based on password or card punching.

Fingerprint is widely used for user authentication. Its accuracy level is very high. But it needs a different setup of hardware and software. Moreover in Internet with large number of user communication system, it is quite impossible to take fingerprint frequently to detect user.

Another suitable user authentication system is face recognition technique.It identifies or verifies a person automatically from a digital image or a video frame from a video source. It is the most natural mean of biometric identification. But such technique may suffer from the rise of wrong identifications owing to the surrounding environment and lighting affecting the quality of images acquired. Iris technology is strong to detect user but it also need extra setup of hardware and software.

In computers finger-print scanning, voice recognition etcare used for security. Few companies have included those systems with their computer products but most of the machines are unequipped for biometric recognition systems due to the limitation of specific hardware devices.

Hand geometry technology is based on the fact that nearly every person's hand is shaped differently and that the shape of human hands does not change after a certain age. In this approach false data are found to place their hands on a flat surface or a digital scanner. This technology has a hygienic issue. This technique is needed extra equipment and these are not portable.

For this reason, the companies who deal with a large number of varied populations are not getting the scope of using biometric verification systems. In this consequence, Mouse Dynamics and its actions come with its importance. This counts as human computer interactions for verification. Though many researchers have tried to work on this problem but there is no standard general interface for this problem. In this research, we'll discuss the extraction of individual behavior through mouse movements to enhance the computer security.

The biometric technology is not a recent concept. The utilization of it was first found in Egyptian architectures like the pyramids. The Egyptians also used their thumbprint for authentication. The practice of identifying individuals was started in late 1800s. In 1880 Scottish doctor Henry Faulds published some ideas suggesting to use biometric for the identification of criminals. In 1900 Galton-Henry fingerprint classification was published[39]. In 1960s Miller Brothers in New Jersey, USA launched a device for measuring finger size which gave the research a new pace [40]. The voice and signature recognition were developed in the 1960s and 70s. By the interest from the US armed forces the Automated Fingerprint Identification System (AFIS) was created. After 9/11 the biometric system spread widely. For privacy concern the biometric standards were revised. The first biometric standards were adopted in 2002 in U.S [1][2] [39].

In this thesiswe have proposed a user authentication method using mouse movement data. Many researchers observed that every user has separate pattern of mouse movement and it is easier to find from a collection of mouse movement data. In this thesis mouse movement data are collected, preprocessed and stored in database. Twelve features are generated from the collected data. A user can be identified based on these features value. We have also applied multi classifiers separately to classify the user based on twelve features. In performance measurement we have find FAR=2.78 and FRR=0.

1.2Motivation

Cloud computing is increasing rapidly day by day. As a result security is getting top priority to use systems and access resources. So, strong authentication is essential to determinewhether someone or something is. Authentication technology provides access control for systems by checking to see if a user's credentials match the credentials in a database of authorized users or in a data authentication server.

Most of the available authentication systems are able to identify a user information but unable to identify original user. Our proposed method ensures the presence of original user. Because in proposed method, user access information is so critical that it cannot be stolen, hacked or known by force.

Authentication is important because it enables organizations to keep their networks secure by permitting only authenticated users to access its protected resources, which may include computer systems, networks, databases, websites and other network-based applications or services.

After the authentication user is authorized for several resources. Sometimes user is again authenticated for authorization. The conventional authentication systems are unable for continuous authentication. Our proposed system can handle the continuous authentication based on user mouse movement behavior.

Though security is a vital issue now a day so researchers and industries are concern about the development of this sector. By this research we have developed such a system. This proposed system able to continuous user verification.

Our main motivation behind this proposed model is to establish high security in authentication where its identity cannot be shared or stolen. Tracking the right person is another motivation of this research. Practices those are available in market to authenticate user are cost effective or weak to share hidden information. Our proposed system can minimize this security problem.

1.3 Aims and Objectives

The importance of robust user access control system and security system is growing continually to avoid the unauthentic use of essential data. New techniques are being created and applied for shielding the privacy and reliability of data. Biometrics authentication is such an approach.

Primary aim of this research is to develop asimple, robust and accurate user authentication system.In this research we have proposed mouse movement based user authentication system. Thissystem does not require any extra hardware or devices. So the proposed system is simple to use. This thesis shows deep analysis of mouse movement behavior of user. Characteristics and patterns of this behavior can only be found from valid user. Proposed method establishes high accuracy level of user authentication.

Core aims and objectives of this research are as the below:

- Use mouse movement data to avoid additional hardware or device.
- Collect or acquisition real time mouse movement data.
- Sample data in to block based on number of action and time.
- Generate new features (Number of Delay, Standard Deviation of the Trajectory Length, Standard Deviation of Slope (m), Standard Deviation of Difference Between Each of Slopes, Number of Curvatures, Number of Changes in Horizontal Position and Number of Changes in Vertical Position.) from sampled blocks.
- Train and test the system using collected real time data and benchmark data [44].
- Use three classifier (SVM, k Nearest Neighbor and Naive Bayes)algorithm separately to verify the system.
- Compare the result of the proposed system with existing related research [4].

1.4 Contributions

This research proposed and develops a simple and robust user authentication system. This system introduces new mouse movement features those are important to identify user uniquely. The proposed methodis tested by capturing data *JitBit macro reader* and Recording user Input (RUI). We have applied this model over benchmark data and finally we have built a user interface to capture data. This model is trained and tested using three classifier algorithms: Support Vector Machine (SVM), k-Nearest Neighbor (kNN) and Naïve Bayes separately. We have found the best performance of using kNN.

Collected raw data of user mouse movement are sampled in blocks. This sampled is done on two ways: based on number of specific action and based on specific duration. Data collected by JitBit and RUI are sampled in blocks based on 50 numbers of actions. On the other hand to observe and analysis the output of the proposed system in different environment, data are collected by a developed user interface. These data are sampled in blocks based on 10 mile seconds intervals.

The proposed authentication system using mouse movement data introduces seven new features. These are: Number of Delay, Standard Deviation of the Trajectory Length, Standard Deviation of Slope, Standard Deviation of Difference Between Each of Slopes, Number of Curvatures, Number of Changes in Horizontal Position and Number of Changes in Vertical Position. These seven features are used with five existing features. These are:Number of Points in the trajectory, Delay Time, Number of Action, Total Length of Trajectory and Curvature of the Trajectory. Some researchers have used these five features [10], [26], [38] in their research. These features are used with combination of our new seven features. On the other hand though some of these existing features are used by some researchers before,but we have used these existing features in different way. Our calculation and finding the feature's values are different from othersresearchers. Considering this case all twelve features and feature generation procedures in this research are new concept.

The features those are used in this research, creates a uniqueness of biometric behavioral authentication system. In order to determine the contribution of the proposed approach by using mouse movement features four types of experiments were conducted: *Free style data tracer using JitBit tool, Free style data tracer using RUI tool, A specific guided user interface* and *Using the benchmark data*.

We have found better result of proposed system by comparing existing related research (details in 4.9). Not only that, FAR and FRR of others related researchers are larger than ours where we have found very low FAR and FRR. We have brought the lowest FAR=2.78 and FRR=0 by applying and comparing different methods (details in 4.5.1).

1.5 Outline of the Thesis

The rest of the paper is organized as follows:

Chapter 2 describes the literature review in same field. In this chapter we have classified mouse movement behavior. We have find out existing features of mouse movement. Limitations of some existing systems are described. We have also focused challenges and target of proposed system in this chapter. Features and benifits of this thesis are illustrated and briefly described here.

Chapter 3 shows the description of proposed system. We have proposed and describe new features of mouse movement. We have shown hot to collect data and sample the collected data into blocks. Descriptions of different blocks with examples are illustrated in this chapter.

Chapter 4 presents experimental results and performances of the proposed system. We have find out best process to uniquely identify user based on our proposed mouse movement features. Classification algorithms and their output results are described in this chapter.

Chapter 5 describes the future plan for improvement and conclusion.

Chapter –2
Literature Review

Companies which are working on information system security need to find the right balance between cost and functionality. It's important to start by determining whether to build a solution from scratch, buy an out-of-the-box solution, or a combination of both. In reality, most projects will require some system tailoring to meet business requirements. Decision-makers must understand how much software development is enough and craft a detailed implementation plan to ensure the project's success.Biometrics system can solve these issues. But the existing biometric system is not up to the mark reliably consideringrobustness accuracy and trust. The authentication using mouse movement data is an alternative that can identify user behavior accurately without any cost of additional device.

There are several researchers working in the field of user authentication using mouse movement data. Different researchers have presented their idea in different way but not studied as whole in terms of mouse movement based authentication. Actually mouse dynamic based authentication is a new concept [5], [21]. Researchers are trying to formalize this concept.

2.1 Brief Description of Related Research

Many researchers have worked with mouse dynamic. Their focusing criteria/issues are also differing from one to another. It is very difficult to formalize all concepts together in single platform. We can divide previous finding in three main categories:

i. Based on Free Style Mouse Movement
ii. Based on Guided Mouse Movement and
iii. Combination of Mouse Movement and Keyboard Pattern

We have illustrated related researches insections below.

2.1.1 Based on Free Style Mouse Movement

Ross A. J. Everittetc [1] gathered initial data from an experiment involving 41 participants between the ages of 20 and 30, with each participant registering with the system via a Web-based applet. Each of the 41 participants was requested to register using their own details and also asked to provide a test set of forged samples for a random selection of other users. This allowed an assessment of the authentic user consistency levels and the degree of variation between authentic and impostor samples, to determine uniqueness.The main issues encountered during data analysis relate to the direct reliance upon the local Java VM (Virtual Machine) to accurately sample information. X and Y points are collected from the raw data. The Euclidean distance formula is used to create a user profile. By this method it is possible to authenticate an individual based upon whether their input data falls within this profile space region. In this system, the means and standard deviations extracted from the data supplied at the time of registration (hold/latency times or angle/distance relationships) are used to provide an approximate model of the profile space.

S. Suganya and others [4] have addressed a method that creates a database of containing mouse dynamic data like co-ordinate value time stamp value and mouse operation. From these features, features vectors are obtained. The dataset contains the static mouse behavior data of 20 users. The dataset is collected from openly available data set. Diffusion map algorithm was proposed to reduce the dimension of future vectors. The author used neural network classes which made up of a number of simple and highly interconnected processing elements. Performance analysis is shown by comparing existing of proposed system as FAR and FRR where the proposed system shows FAR=4.15% and FAR=5.05%. The proposed system shows better performance.

Research of G. Muthumari and others [10] is based on classification. The extracted features are analyzed and performs authentication using Learning Vector Quantization (LVQ). This LVQ Algorithm is used as a classifier, to find whether the given sample is authorized or unauthorized identity. Data are captured from user mouse movement behavior. In this work a drawing interface is used to capture data. The mouse dynamic data consists of x-y coordinates and elapsed time of users when drawing some predefined letters on drawing area. In this method LVQ is compared with Support Vector Machine (SVM) by calculating FAR and FRR. The performance of the proposed system provides 7.25% FAR 6.25% FRR. The test result proves that, the proposed method LVQ provides low error rates with good accuracy than the existing method SVM classifier.

ShivaniHashia[20] shows in the initial stage, various mouse movement patterns are recorded for different users and then these recorded statistics are used to verify the users. The idea is to check how the user moves the mouse when he moves from one position on the screen to another. Based on user's mouse movements, the coordinates of the mouse are recorded after every 50ms. Using the recorded coordinates, they calculate the speed, deviation from a straight line and angle. Speed is calculated by dividing distance by 50. Deviation is computed by finding the length of the perpendicular from the point where the mouse is currently located (while moving) to the line formed between two of the ten given points between which the mouse is moving. Angle gives us the angle formed between the point where the mouse is currently located, and the two points given by the two dots between which the mouse is moving. Angle is further separated into positive angle (angle which lies between 0 to 180 degrees) and negative angle (angle which lies in the range 0>angle ≥-180 degrees. These parameters are stored in a file for that user. The user has to move five times on the ten points to complete the registration. It is done to get as much information about the user's mouse movements as possible. Between each pair of points, they find the average, standard deviation, minimum and maximum of the four parameters speed, deviation, positive angle and negative angle. Thus, there are sixteen parameters for a pair of points. There are one hundred and forty four parameters (vectors) stored in the template file for nine pairs of points. This procedure is repeated for all users who register. After finding all vectors for a user, our next task was to normalize them.

Normalization was necessary because the variation in all four parameters was different. A small variation in speed meant a lot more percentage change than a large increase or decrease in angle or deviation because the speed didn't vary much whereas angle and deviation varied a lot. To normalize the vectors, they found the maximum and minimum of all 4 vectors for all registered users. Difference between the maximum and minimum value of a parameter gave the range for that parameter. They multiplied each of the 144 vectors by 100 and divided by the range of the corresponding parameter i.e., for all the parameters like average speed, standard deviation of speed, etc that are derived from speed parameter, they multiply by 100 and divide by the range (maximum –minimum of speed for all registered users) of the speed parameter. They stored all the normalized vectors in a file Vector.txt. It contains 144 vectors of all users who register. The file will then be used in training and verification phases. Enrollment phase is like mean of the vectors and forms the base from which they compare the vectors in verification phase.

The idea of the experiment was to find how to use the four parameters (speed, deviation, positive angle and negative angle) so that they could verify the authenticity of users. Angle was separated into positive and negative angles to avoid the possibility of averaging out the positive and negative values of the angle. For four parameters they find average, standard deviation, maximum and minimum between each pair of points i.e. from 1 to 2, 2 to 3 and so on. So between a pair of points they found 16 parameters. They had 9 pairs of points, which gave us 144 parameters for every user.

Zach Jorgensen and Ting Yu [34] studied existing authentication techniques in this paper. Some limitation also discussed in existing authentication. A new technic is proposed which solves the limitation of existing methods. They present the results of several experiments that they conducted to illustrate our observations and suggest guidelines for e-valuating future authentication approaches based on mouse dynamics.

Like other biometric authentication systems, those based n mouse dynamics are typically evaluated with respect to the following metrics:

- False Acceptance Rate (FAR): the probability that the system will incorrectly label the active user as the same user that produced the enrollment signature.
- False Rejection Rate (FRR): the probability that the system will incorrectly label the active user as an impostor, when in fact it is not.
- Equal Error Rate (EER): the error rate when the system's parameters (such as the decision threshold) are set such that the FRR and FAR are equal. The lower the EER the more accurate the system.
- Verification time: the time required by the system to collect sufficient behavioral data to make an authentication decision.

To collect data some computer science student are selected. They have used both USB optical mouse and USB touch pad. The subjects were given a specific web browsing task designed to last 30 minutes and were asked to perform that task once for each of the two pointing devices. Their custom mouse authentication software was installed in every PC which is written in C#. The data collection procedure produced two sets of raw data: one from the optical mouse and the other from the touchpad. Two versions of these datasets were created by preprocessing the data according to the requirements of the two MDA approaches used in this study.

In that paper, the feature selection step was performed using each user's test set, which is known to bias the classier to the test data and produce unrealistic results. They have found that using the training set for feature selection gave comparable or worse results than doing no feature selection at all and that using part of the data as a validation set was not feasible due to the limited amount of data available to us. Therefore, feature selection is omitted and all features for every user are used.

Three experiments are performed in this study. The first experiment investigated the effectiveness of the two MDA approaches when all environmental variables were constant across users. The procedure was the same for both of the approaches and was performed. The experimental procedure was performed separately on the mouse and touchpad datasets and the results of Far/FRR are found out. The purpose of Experiment 2 was to investigate whether data collected from a given user on one of the pointing de-vices could be used to verify that user's identity based on da-ta collected from the same user on the other pointing device. The retrained classier is then tested both on valid user and invalid user in test sessions from the touchpad dataset. This whole procedure is repeated a second time using the touchpad data set for training and the mouse da-ta set for testing. The thresholds for determining the error rates in this experiment were set to be the same as those determined for the same user in first experiment.

In the final experiment, the researchers sought to determine whether the two approaches could be used to identify the pointing device that generated a given session. They first divided the 17 users into two groups with eight users for training and nine for testing, such that no user was represented in both the training and testing set. Researchers then labeled all of the sessions in the training and test groups according to pointing device.

A classier was then trained on the labeled training sessions and subsequently tested on the test sessions. Using the results from the first experiment as a baseline, it can be observed from the results of Experiment 2 that the average error rates rose substantially for both techniques when training and testing on data from different pointing devices. For nearly all of the users, either the FRR or FAR rose above 50%. These results suggest that pointing device hardware itself exhibits in an influence on mouse dynamics strong enough to overshadow the unique behavioral patterns of most users. The results of Experiment 3 pro-vide further evidence of this influence. In the experiment, both techniques were able to correctly determine, with al-most perfect accuracy, which of the two pointing devices generated a given data session. The results further indicate that the behavioral variations caused by the pointing device hardware exhibit a distinct, user-independent pattern.

They showed that when enrollment data and verification data for the same user are collected under two different pointing devices, existing techniques are not likely to be able to accurately verify the user's identity. This finding suggests that mouse dynamics may not be a good choice for authentication in web-based applications or other remotely accessed resources.

MajaPusara and Carla E. Brodley [35] proposed supervised learning method where users are differentiated based on their mouse movement behavior with a false positive rate of 0.43% and a false negative rate of 1.75%. The researchers collected mouse data from eighteen users all working with Internet Explorer. We applied a supervised learning algorithm to determine whether we could discriminate users based on their mouse movements. The results show that for users that utilize the mouse we obtain a false positive rate of 0.43% and a false negative rate of 1.75%. These initial results indicate that data from the user's mouse movements provide a strong signal of normalcy.

A model is built by a training session and at the time of authentication user’s mouse movement compared with that model. If the current behavior of a user deviates significantly from the model of normal behavior, the system flags this behavior as anomalous and does one of two things: asks the user to authenticate again or reports the anomaly to a system administrator. Due to unavoidable gradual changes in the user's behavior over time, the pro le should, ideally, be updated with new training instances periodically.

To extract features from the mouse movement data set, the researchers first compute the distance, angle, and speed between pairs of data points. These pairs can either be consecutive or they can be separated by k data points. We call the parameter k the frequency. This parameter is customized for each user. After we obtain our raw features, we extract their mean, standard deviation and the third moment values over a window of N data points.

The extracted features for each observation in a user's profile are:

1. For each event category in the mouse event hierarchy, we count the number of observed events in the window. This creates six features.
2. For each event category and for the cursor movement data, we compute the mean, standard deviation and third moment of the distance, angle and speed between pairs of points. This creates 63 features.
3. For each event category and for the cursor movement data, we compute the mean, standard deviation and third moment for the X and for the Y coordinates. This gives a rough measure of the location of the events for that window and the location of the cursor in that window.

If collecting data from all individuals who have physical access is not feasible, then re-authentication is best viewed as an unsupervised learning problem. In this case, the mouse data is collected for each individual user whose behavior we are trying to model. During the user re-authentication process, a normal" user's profile, generated from the mouse data during the training phase, is compared against a current user's behavior.

Data are collected from 18 user, all are students using internet explorer and windows. A decision tree is created based on behavior of collected data. Researchers used C5.0 algorithm. From the collected data first two quarters of data for training, the third quarter for parameter selection, and the last quarter for testing.

Recall that the parameter selection data set is used to select the frequency value for each category and the window size for feature extraction. To search for the frequency values from the candidate are chosen. Their choice of values was based on observations about the data. For example, mouse wheel points are frequent (approximately 25 per second), whereas single-click points are scarce (approximately one every five to ten seconds). Larger frequency values generate a more accurate profile when coupled with mouse wheel events and smaller values are better for single click events. In our experiments we set the frequency value to be equal to one for single and double clicks, and used the parameter data set to select the frequency value for both the mouse movement data and the remaining four event categories that minimizes the false positive and false negative rates for the user.

Ahmed AwadE.Ahmed and IssaTraore [40] have developed a technique that can be used to model the behavioral characteristics from the captured data using artificial neural networks. In addition, they present an architecture and implementation for the detector, which cover all the phases of the biometric data flow including the detection process. Experimental data illustrating the experiments conducted to evaluate the accuracy of the proposed detection technique are presented and analyzed. They collect the mouse coordinate and generate several features such as speeds, accelerations, angular velocities, curvatures and the derivative of the curvature of curve.

Chinmayee.KS and Vanishree C [42] have proposed a framework to authenticate user from mouse movement data. The framework covers four modules: gesture creation, data acquisition and preprocessing, feature extraction and classification. They have worked for static authentication. Mouse movement data are analyzed using a Hidden Markov Model. They have captured mouse movement data by asking user to draw a mouse gesture. First gestures are stored as template. In next time user replaced gesture several times and these compared with template. The researchers have recommended to use their proposed system in complex environment like e commerce or e learning.

2.1.2 Based on Guided Mouse Movement

NazirahAbd Hamid etc. [9] captured data by using mouse clicks on several buttons. When user clicks a button then it vanishes and appears new one randomly. A user was asked to perform this experiment six times to establish the user personal profile consisting of user behaviors and characteristics. Raw data consisted of coordinate X and Y as the user clicked on a button that set in a certain grids. The other attribute that involved in the process was time in milliseconds. The raw data, x and y coordinates and time from a data file was applied with five different formulas to create a user's mouse profile. Then the average and standard deviation of each result would be determined as the mouse profile measurement.

Nan Zheng, Aaron Paloski, and Haining Wang[23]intentionally set a normal environment for these users and inform them to behave as naturally as possible. Mouse movement data are recorded during their routine computing activities. These activities range among word processing, surfing the Internet, and programming, online chatting, and playing games. They make use of a logging tool RUI to record their mouse movement activities.

For the field set, more than 1,000 unique forum users' mouse movements are recorded by JavaScript code, and submitted passively via AJAX requests to the they server. On one hand, these users are anonymous but identifiable through unique login names. However, there is no guarantee on the amount of data collected for a certain user. A forum user could be logged in for a long time with frequent mouse activities, or could perform just one click and then leave. On the other hand, the breadth of this corpus of users is utilized to serve as the base profile for both training and testing purposes.

To analyze the mouse movement data, they define three fine-grained angle-based metrics: direction, angle of curvature, and curvature distance. These newly-defined metrics are different from the conventional metrics, such as speed and acceleration, and can accurately characterize a user's unique mouse moving behaviors, independent of its running plat form. As a comparison, they list the definition of two traditional mouse movement metrics, speed and pause-and-click.

They choose Support Vector Machines (SVMs) as our classifier to differentiate users based on their mouse movement dynamics. SVMs have been successfully used in resolving real-life classification problems, including handwritten digit recognition object recognition text classification and image retrieval. In general, SVMs are able to achieve comparable or even higher accuracy with a simpler and thus faster scheme than neural networks. In the two-class formulation, the basic idea of SVMs is to map feature vectors to a high dimensional space and compute a hyper plane, which separates the training vectors from different classes and further maximizes this separation by making the margin as large as possible. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs outlining a hyper plane in feature space.

They compare our evaluation results with those of existing works in terms of verification accuracy and time. The verification time is highly dependent on the number of mouse events needed to make a decision, the type of mouse events used (mouse click, mouse move, or drag-and-drop), as they'll as how fast a user generates mouse events. Even for the same user at different times, the number of mouse events per unit time varies a lot.

To be truly portable, this system makes minimal assumptions about the available technology at the point of use. The main issues encountered during data analysis relate to the direct reliance upon the local Java VM (Virtual Machine) to accurately sample information. At present, the Java VM technology produces inconsistent and no uniform sampling rates due to its reliance upon the underlying hardware and software of the client machine. This inconsistency means that noise is introduced into any sampled data. This system has therefore been designed to be tolerant of this noise by using neural networks trained with noisy genuine data and appropriately noisy automatically generated forged data samples.

No two signatures are identical, even when signed by the same person. The lengths of the signature trace (in terms of the total number of sampled points N), the spatial size and temporal information will all vary. The input signature trace therefore needs to be preprocessed to reduce the effect of these differences and to convert it into a standard format. Signature traces are preprocessed to normalize the total arc length (disjoint segments are joined to produce a single continuous arc). Next, the total time taken to produce the trace is normalized. Finally, the traces are linearly time warped so that the N points contained in the signature trace are replaced by N0 temporally equidistant points using the process. This step is necessary because the Java VM is unable to sample at a constant rate, thus adding excess noise to the input signature and exaggerating differences in input.

The Euclidean distance, dij, is between two points Si=(xi,yi) and Sj=(xj,yj). N0 is the number of temporally equidistant points contained in the signature after performing linear time warping.

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2},$$

Where $[(0 \le i < j \le N') \vee (i = N', j = 0)]\, N' = 100$.

The vector associated with a point, Si, is obtained by the function V(Si), which returns either a vector from the previous point to the current, or from the last point to the first in the trace.

It is possible to authenticate an individual based upon whether their input data falls within this profile space region.The difficulty for any system, therefore, is determining the size and shape of the authentic user’s profile space. In this system, the means and standard deviations extracted from the data supplied at the time of registration (hold/latency times or angle/distance relationships) are used to provide an approximate model of the profile space. In each dimensional plane, we assume a circular distribution; the mean values provide the profile space center, while standard deviation values provide the radius.

The most difficult forgeries to recognize are those that are very similar to authentic samples, lying close to the authentic user’s profile space. Forgeries which reside further from the profile space can more easily be rejected by a verification system and need fewer training examples. In training a high proportion of false samples lying close to the profile space boundaries, within a boundary space region were used along with a few outlying samples to ensure a correct modeling of the problem domain. The boundary space region is an enclosing subspace (around the profile space region) whose radius is the same as the profile space radius (in each dimension) and extends a further distance of 0.25 times the relevant radius. False samples are generated within the boundary space using pseudorandom values for each axis, based upon the authentic user’s characteristic patterns.

Joseph S. Valacich and other [24] researchers have proposed a polygraph technique called the concealed information test (CIT). It can be automated and implemented within an online survey environment for mass deploy ability to help identifying individuals involved in specific insider threat activities. This CIT detects if a person has ‘inside’ or ‘concealed’ knowledge of a target. It can be used sectors of a crime, event, person, object, etc. The objective of the CIT is to detect if a person has ‘inside’ or ‘concealed’ knowledge of a target e.g., a crime, event, person, object, etc.

CIT can be automated and conducted in an online survey for mass deployment. Similarly to a face-to-face polygraph CIT, a survey can present a question followed by several plausible answers, including one key-item. The researchers have first developed propositions of how an insider threat would respond to an online CIT and what mouse movements would be indicative of those responses. The propositions are:

1. Presentation of the key-item will trigger an orienting response for guilty insiders.
2. Prior to the presentation of the key-item, guilty insiders will engage in a search process.
3. Deceiving on the key-item will cause increased cognitive effort.
4. Guilty insiders will exhibit greater caution when responding to the key-item than when responding to non-key-items.
5. Guilty insiders will experience greater emotional arousal than innocent people.

As people become accustomed to the online CIT format, they develop a mental schema or representation of the CIT and respond habitually to the items. For example, individuals may develop a pattern of 1) seeing a non-key-item, 2) indicating that they have no knowledge of the non-key-item, and 3) proceeding to the next item. This pattern becomes habitual and respondents’ mouse movements become consistent as they go through this semi-automatic process. Thus all propositions have been developed and only general propositions are proposed but not the hypotheses. In the exploratory research there are two different scenarios as a) unsanctioned deception scenario and b) sanctioned deception scenario. Using these scenarios mouse dynamic is identified.

Saurabh Singh and other [33] propose a new approach in behavioral biometric authentication system using mouse interaction. In the work a new idea of classifying the distances travelled by the mouse is presented with supporting results. The study was done on limited number of participants and it is possible that if different population is used results may vary from those presented here.

User's behavior on mouse is unique and can be taken as biometric property. A graphical panel has been designed for acquiring user pattern and a grouping approach has been applied to reduce the size of the pattern. Study says that mouse speed increases with the distance travelled by the mouse and highly dependent on the direction of the mouse movement. Based on this two hypothesis are created as *Mouse speed increases with the distance travelled* and *Mouse speed is different in different directions considerably*.

To capture the features, a nine button panel was considered and distances between any pair of buttons on the panel are classified as below:

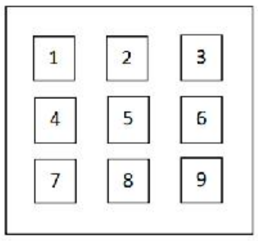


Figure 2.1: Button panel used in experiment [33]

Short- For the buttons adjacent (1-2, 1-5, 2-5, 2-6 etc.).
Long- For the buttons neither adjacent and nor at largest diagonals (1-3, 1-6 etc.).
Very Long- For the buttons that are at the ends on largest diagonals (1-9, 3-7).

To establish the method a group of five users participated in experiments. Speeds against distance and directions are taken from different users. Average FAR and FRR are calculated from the user to verify the authentication. Worst FAR is 1.53 and worst FRR is 5.65 which ensure the acceptability of the system. Low value of FAR is more important than FRR because the low value of FAR shows the strictness of secured system. In the work a new idea of classifying the distances travelled by the mouse is presented with supporting results.

A. Kaklauskas, E. K. and others [13] observe activities of employees. It is a web based biometric mouse decision support system. The researchers develop a Web-based Biometric Mouse Decision Support System for User’s Emotional and Labor Productivity Analysis (MDSS-UELPA) able to analyze data from biometric mouse usage and e-self-reports. We believe that different biometric parameters, including physiological (skin conductance, amplitude of hand trembles, skin temperature), psychological and behavioral/motor-behavioral (mouse pressure, speed of mouse pointer movement, acceleration of mouse pointer movement, scroll wheel use, right- and left-click frequency), correlate with the user’s emotional state and labor productivity.

Behavioral measurement methods are based on the fact that the body usually responds physically to an emotion (e.g. changes in muscle tension, coordination, strength, frequency) and that the motor system acts as a carrier for communicating affective state.

This study is a web-biometric mouse decision system where different biometric parameters, including physiological (skin conductance, amplitude of hand trembles, skin temperature), psychological (e-self-reports) and behavioral/motor behavioral (mouse pressure, speed of mouse pointer movement, acceleration of mouse pointer movement, scroll wheel use, right- and left-click frequency), correlate with the user’s emotional state and labor productivity.

MDSS-UELPA consists of seven subsystems:

- Data Capture and Collection Subsystem;
- Feature Extraction Subsystem
- Database Management Subsystem;
- Model-base Management Subsystem;
- Equipment Subsystem;
- e-Self-Reports Subsystem;
- Graphic Interface.

The collection of data by the Data Capture and Col-lection (DCC) Subsystem is handled by a back-ground process and therefore hidden from the user. When a user works with a computer, all measured parameters are captured and stored in a CSV format in an excel file as raw data. Raw data consists of mouse clicks and movements, as well as moments when a palm simply rests on the mouse. Each time the user clicks and moves the mouse or holds his/her palm on the mouse, the DCC Subsystem collects and stores the raw data respectively in a .csv file. Any event of mouse use is recorded, whether it is a move, drag and drop, a click (pressed or released) or simply palm held on the mouse. Time of the event in milliseconds is recorded. X and Y coordinates of the mouse pointer on the screen are recorded. A text file is being generated in a CSV format continually.

The speed of mouse pointer is the distance covered by the pointer during a fixed period of time. It is possible to use Euclidean or Minkowski distance. The acceleration is calculated as difference between the current speed, and the speed measured during the previous time period. Hand shaking parameter indicates the oscillation of the mouse pointer. This parameter is calculated as follows:

- Ten values of mouse pointer’s coordinates are stored.
- The differences between neigh boring coordinate values are calculated.

- The minimum and the maximum differences are estimated.
- If the minimum and the maximum differences are of different signs, then the average value of stored coordinate values is calculated and the maximum deviation from the average is estimated.
- If the deviation is small enough, then we treat the movement of the pointer as shaking.
- The value of shaking parameter can be changed according to the user and the type of mouse.

The DCC Subsystem collects data on the following parameters for correlations with user's emotional and labor productivity state:

- Mouse pressure when a user presses his/her mouse and a button (power sensors);
- Electro galvanic skin conductance (electro galvanic skin response sensors);
- Skin temperature (thermistor);
- Speed of mouse pointer's movement;
- The acceleration speed of mouse pointer's movement;
- Amplitude of hand tremble;
- Scroll wheel use;
- Right- and left-click frequency;
- Idle time.

The Feature Extraction (FE) Subsystem receives the raw data from the DCC Subsystem and extracts the characteristics that signify a user's emotional and labor productivity behavior. Before the biometric mouse tracking, a user should evaluate his/her emotional state, stress, desire to work and productivity on a ten-point scale and temperament (choleric, sanguine, and phlegmatic) through e-Self-Report Subsystem.

Monitoring of all mouse actions, hand characteristics and e-self-report produced by the user during interaction with the Graphical Interface allows generation of a unique profile, which can be used for analysis of user's emotional state and labor productivity.

Factors that can make a normal human hand tremble more noticeable include stress, anxiety, fatigue, withdrawal of alcohol or certain other drugs (such as opioids), an overactive thyroid gland (hyperthyroid-ism), consumption of caffeine, and use of certain drugs. MDSS-UELPA is under development in order to evaluate user's emotional state such as anger, fear, sadness, disgust, happiness, surprise etc.

2.1.3 Combination of Mouse Movement and Keyboard Pattern

AnandMotwani and others [7] has presented two phases. One is enrollment phase and another is verification phase. At enrollment different mouse and keyboard events data are captured. Features are extracted from these captured data and a profile database is created. At verification phase user's features are matched with the profile database. A probability and knowledge generation step works here. The probability phase identifies who is the most likely user. In knowledge generation phase the users are classified valid or invalid user. It shows the accuracy of the system. The researchers takes three sample of each user and 27 features are worked.

Raina K. Jain etc. [8] has illustrated various Biometric Systems and mainly focuses on keyboard and mouse dynamics. Various problems of these systems are described. Possible solutions also are shown to increase performance of existing system. Different biometric techniques are described in different ways but author has classified the behavioral biometric techniques further into 5 categories. There is a comparison of mouse movement based authentication based on data collection environment feature source, FAR and FRR. The authors mentioned large size of data, more authentication time, and large training period as basic limitation of existing system. The authors suggested to combine mouse/keyboard based authentication with existing systems called multimodal behavioral biometric.

SoumikMondol and PatricBours [41] have recently published a biometric authentication based on mouse dynamic and keystroke. By using software they have collected mouse movement and keystroke data from 53 users. They have worked on continuous authentication system by developing robust trusted algorithm. Some new mouse movement features are introduced. A windows based login tools is created to collect data. Data are collected in csv format in local machine. They have converted mouse events into four different actions, single click, double click, move and drag-drop. Collected data are separated into three parts as, for training, to adjust parameter and for testing. Researchers have compared their result with related research and found their proposed system shows better result. They have also illustrated comparative result between continuous authentication and periodic authentication. Separate classification technics for different modalities along with different verification processes are applied. Researchers have used three different classifier models in Multi Classifier Fusion architecture for all the different actions. The regression models are Artificial Neural Network (ANN) and Counter Propagation Artificial Neural Network and the prediction model is Support Vector Machine (SVM).

2.2Existing Features of Mouse Movement Data

Different researchers have derived different features regarding mouse movement pattern. Calculations of these features also differ from researchers to researchers. Lists of such features are shown in table 2.1.

Table 2.1 : List of existing features of mouse movement

Existing Features	Reference
• Pattern of pressing keyboard buttons.	[1]
• Time • Speed • Acceleration • Deviation • Angle of deviation	[9]
• x-coordinate • y-coordinate • time remaining features • Horizontal velocity • Vertical velocity • Tangential velocity • Tangential acceleration • Tangential jerk • Path from the original pixel • Slope angle of the tangent • Curvature • Curvature rate of change	[10]
• Average between each pair of points • Standard deviation between each pair of points • Maximum between each point between each pair of points • Minimum between each point between each pair of points	[20]
• Direction • Angle of curvature • Curvature distance	[23]
• Horizontal coordinate • Vertical coordinate • Absolute time • Horizontal velocity	[26, 27, 42]

Existing Features	Reference
• Vertical velocity • Tangential velocity • Tangential acceleration • Tangential jerk • Path from the origin in pixels • Slope angel of the tangent • Curvature • Curvature rate of change	
• Movement speed over traveled distance per defined action • Movement speed per direction of movement • Acceleration per direction of movement	[28]
• Down-Up (DU)time • Down-Down (DD)time • Up-Down (UD)time • Up-Up (UU)time • Down-Up2(DU2)time	[29]
• Size of a curve • Length of a Mouse curve • Total time of the Mouse curve • Mouse speed over a pre-defined action • Angle of Mouse Movement • Acceleration • Mouse click duration	[30, 31]
• Speed over 8 directions • Speed over different distance(short, long and very long)	[33]
• Mouse pressure • Electro galvanic skin conductance, • Skin temperature • Speed of mouse pointer's movement • Acceleration speed of mouse pointer's movement • Amplitude of hand trembles • Scroll wheel use • Right- and left-click frequency • Idle time	[35]
• Number of events • For each event category and for the cursor movement data, the mean, standard deviation and third moment of the distance, angle and speed between pairs of points. • For each event category and for the cursor movement data, we compute the mean, standard deviation and third moment for the X and for the Y coordinates.	[36]

Existing Features	Reference
• X and y position value • Distance metrics(Euclidean distance, Manhattan distance & edit distance)	[37
• Number of trajectory points • Amount of time to complete trajectory • Length of trajectory • Velocity from point to point in the trajectory • Acceleration from point to point in the trajectory • Direction angle from point to point in the trajectory • Number of inflection points in the trajectory • Curviness of the trajectory	[38]
• Time elapsed, • Distance traveled, • Direction or range or angle of each movement	[39]
• Movement speed compared to traveled distance(MSD) • Average movement speed per movement direction(MDA) • Movement direction histogram (MDH) • Average movement speed per types of actions(ATA) • Action type histogram(ATH) • Traveled Distance Histogram(TDH) • Movement elapsed time histogram(MTH)	[40]
• Mouse single click action • Mouse double click action • Mouse move action • Mouse drag-drop action	[41]

Features listed above those have been used by different researchers have some weakness. People can work with mouse in free style or users may provide specific interface to detect user. But the above features cannot work for both styles: free style mouse movement and guided style mouse movement.

Sometimes features are captured based on specific environment. But in cloud computing or in a large network it user working environment is unknown.Some features works only angle is created [9]. But it is difficult to get frequently angle value from user mouse gesture.There are some features those depends on specific mouse click event. But in a long session required clicks may not occurred.There are also some mouse movement features those depend on keyboard typing pattern [7] [8]. In these cases users are identified based on combination of user's work using mouse and keyboard. But frequently data of mouse and keyboard cannot be found.

We have planned to capture the mouse movement data based on both freely unguided work with mouse on computer (details in section 4.1.1 & 4.1.2) and guided working particular interface (details in section 4.1.3). The features (section 3.2) that we are proposing can be retrieved from X, Y coordinate and elapsed time. From these data we have generated 12 features. From these features a signature for each user is generated and this signature is a standard key for that user. Based on these features a user is authenticated.

2.3 Limitations of Existing System

Generally existing authentication system uses password/Pin number, finger print, card punch, face detection, voice recognition, retinal scan etc. Finger print, card punch and password is using frequently. Researchers and technicians are working more to make these systems more secure. These systems have some limitations. These limitations are briefly described in the followings.

- Sometimes people are using weak passwords for better remember. Sometimes a single password is used in different application. This is easy to keep in mind but frequently using of such password is secure less. Weak passwords are a problem because they are easy to guess and they are certainly no match for brute force password attacks by criminals using automated password cracking software.
- Passwords can be forgotten, disclosed, lost or stolen. It is easily accessible to colleagues and even occasional visitors and users tend to pass their passwords with their colleagues to make their work easier.

- One way to beef up the security of your authentication process is to force users to create long, complex passwords, but such enforcement comes at the risk of employees writing the passwords down thereby defeating the attempt to increase security.
- It's also important to take into account the environment the biometric authentication system will be used in. For example, fingerprint readers do not work well in environments where users' fingers are likely to be dirty.
- Similarly, voice recognition systems are not a good match for excessively noisy environments. Some effort is done to achieve large and publicly available test sets and protocols like the FERET and the XM2VTS databases for face recognition evaluation.
- Secret is an important factor in biometric authentication system. In password or token the key information is in secret. That it could be possible for a hacker to present a photograph to fool a facial recognition system, to present a wax cast of a fingerprint to a reader, or to play back a recording of a voice to a voice recognition system.
- Surely there is a possibility to lost one time passwords tokens. Based on that full proof approach won't be always 'Something they have'. Instead, there will be a two factor system which will drive information by measuring biological or behavioral characteristics.
- Fingerprints are commonly used for enterprise authentication. There are also palm or figure vein patterns, iris features and voice or face patterns. But these are not used as much as fingerprints. Keystroke dynamics which is a physical appearance measure the way a user types, how fast or slow a user type, how much time they take also can be used to identify a user.

2.4 Challenges of Existing Systems

There are some challenges in existing biometric systems such as accuracy, scale, security, privacy, dependable feature generation and classification[5]. The password protection systems are widely used for authentication. The biometrics sometimes cannot make a match like the password system. There is usually a thread that is used to match the results. The thread should be set sensibly to strengthen the protection. Large dataset is required in biometric system and it is also a challenge to work with a large amount of data. Computing with many users can make the authentication system work slowly [6].There is an alternative way to overcome the mentioned challenge. To get result quickly we need more powerful hardware and processor for server. But this required changes in working environment and cost effective. So minimize the cost is also a challenges issue.

Features those are used and method of calculation is also challenging issues. In a common framework these calculation takes a multi-disciplinary work both in hardware and software. Implementation of existing researches in independent platform and in online is more challenges.

2.5 Expected Outcome

In this proposed system the collection of behavioral characteristics of human beings in mouse movement are used for user authentication and identification. Mouse movement data contains user unique behavior. Characteristics of mouse movement data are important issue to detect a user. Target of this research is to detect and authenticate user based on mouse movement data.

A large amount of mouse movement data can be collected in a certain period of time. These data are needed to be sampled. Sampled can be done in two ways. One is based on specific duration and another is based on specific mouse action.

Our objective is to create new features from collected data blocks. We have to analysis these features to create unique characteristics of user mouse movement behavior.

To observe the accuracy and smartness of this proposed method we will have to train and test the system by benchmark data besides our collected data.

Another important outcome of the proposed method is to increase performance level. There are some researchers working on same field. To compare our research with other related researchers is an important objective of this thesis. The system is compared with related research [4] to see the output result.

Some core target of this research are as the below.

- **To detect valid person:** Online based work or assignments and out sourcing are increasing day by day. Sometimes employer does not know the employee physically but the employee is delivering his/her work through online. In this regards employer needs to check that if the original hired employee is working or evil employee is working. By the development of our proposed system this problem is solved. Employer can check about employee existence though his/her mouse movement behavior.
- **Classification:** User classification is an important role of this research. We have used three popular machine learning classification algorithms to classify and detect user. Performances of these classifications are also vital issue based on mouse movement data.

- **In depth analysis:** To know the characteristics of mouse movement data of users, we need a depth analysis of data and user's mouse movement pattern. Also strong analysis required for feature generation and algorithmic classification. This thesis identifies factors associated with mouse movement data of different users.
- **Generating mouse movement features:** User authentication based on mouse gesture depends on features of mouse movement. Another core target of this thesis to generate new features of user mouse movement.
- **To find reliable process:** To find error free process to identify user, we have to perform many process of experiments. There are many processes or ideas to find user from mouse movement data (details in 4.5.1). Among them we need a reliable a process.Through this proposed methodwe have developed an explanatory theory that associates user authentication based on mouse movement behavior.

Compared with other biometrics such as fingerprint, voice recognition, face detection, or iris, mouse dynamics is less obtrusive and requires no specialized equipment to capture the biometric data. When a user wants to log in to a computer system, the proposed system only requires him/her to provide the user name and perform certain mouse operation tasks. Extracted behavioral features are compared with features of the legitimate user. A matching session authenticates the user, otherwise his/her access is denied. Moreover, the user's mouse behavioral characteristics can be continuously analyzed during the user's subsequent operations to enforce a full session identity monitoring.

Moreover this research plays better role of some systems and solved security problems in ICT sector. The proposed method facilities some issues. Most of them are as below.

- **New concept:** The proposed mouse movement authentication system presents a new concept of user authentication process. We have proposed seven new features.
- **No need extra device:** To authenticate a user or to detect a user, the proposed system needs no extra devices. Though authentication depends on pattern of mouse movement so we can use this method in everywhere. Conventional authentication systems are based on extra hardware and devices. But the proposed system is free from installing new devices, hardware etc.
- **High secured:** It is difficult and quite impossible to hack and steal user access information. There is no use of password/PIN number or card punch for authentication. Even no access information is known to user. The system identify user in such a way that users themself don't guess authentication information. It may even be possible to intercept the biometric data from the reader and replay it later, bypassing the biometric sensor, like voice recognition, face detection etc. But in mouse gesture based recognition system we can solve this problem. Here the key information is in deep secret.
- **Accuracy:** Provide more accurate result than existing works. It is an excellent process to identify user. We have seen that our proposed method shows very low error rate in terms of biometric behavioral authentication.
- **Online based:** Using the proposed system we can identify user from remote location. It can work on client-server environment. Users working in long distance through online can be identified.
- **Periodic and continuous authentication:** This proposed system can work for continuous and periodic authentication. It can identify user in login session and also can identify user whenever user is working.

- **Training free:** if we use the proposed system then we do not need to train up user or customer. Not only that, no need any training manual. So the system helps by making us free from training session.
- **Silence of work:** Whenever the proposed system works in free style of mouse movement (details in section 4.1.1, 4.1.2, 4.2.1 and 4.2.2) then it works background of a user work. User cannot understand how and when he is being identified. It works in deeply and detective way to verify user as valid or imposter.
- **Continuous process:** In free style working environment, the proposed system can monitor user's total working session. So if a user move from a work station and another user starts previous user's rest of task, the proposed system can identify the changes of users. By this way the proposed system can identify user and user's mouse movement behavior continuously.
- **Maintenance free:** Others authentication methods like finger print, card punch, pin number or using password have a large maintenance and backup issues due to its uses of extra devices. But the proposed system required a little maintenance and even no maintenance.
- **Environment free:** Our proposed system works in all environments. Existing authentication process like finger print, scan etc, shows different result based on environments. Sometimes devices response slowly or out of work based on temperate, humidity etc.
- **Cost Effective:** The proposed system needs no extra devices. Moreover there is no cost for training and maintenance. So it is not only cost effective but also hassle free.
- **Tested by best machine learning classification:** We have used three well known machine learning classification algorithms separately to validate the proposed system. The algorithms are tested and in different view and perspectives. From

these three we have found best algorithm for the proposed system. We have used SVM, kNN and Naive Bayes algorithms and kNN shows the best result (details in section 4.7 and 4.8).

2.6 Learning and Classification Algorithm

We used three classification algorithms in this research. These are:

- Support Vector Machine (SVM)
- K Nearest Neighbor (kNN) and
- Naïve Bayes.

These are all supervised learning algorithm.We know supervised learning is useful in cases where anattribute or property is available for a certain dataset and needs to be predicted for other instances. To train, we need a set of instances which are labeled with appropriate class. This is called the training set. The algorithm processes these training data and predicts class label of unknown instances based on these training dataset. For this reason the algorithms are called supervised.

Inthe following section we have briefly described about these three supervised learning algorithms those are used in this research.

Support Vector Machines (SVM)

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does some extremely complex data transformations, and then figures out how to separate data based on the labels or outputs you've defined. SVM is capable of doing both classification and regression. In particular we'll be focusing on non-linear SVM, or SVM using a non-linear kernel. Non-linear SVM means that the boundary that the algorithm calculates doesn't have to be a straight line. The benefit is that we can capture much more complex relationships between my data points without having to perform difficult transformations on my own. The downside is that the training time is much longer as it's much more computationally intensive.

Support vector machines, or SVMs have performed well on traditional text classification tasks, and performed well on ours. The method produces a linear classifier, so its concept description is a vector of weights, $\underset{W}{\rightarrow}$, and an intercept or a threshold, b. However, unlike other linear classifiers, SVMs use a kernel function to map training data into a higher-dimensioned space so that the problem is linearly separable. It then uses quadratic programming to set $\underset{W}{\rightarrow}$ and b such that the hyper plane's margin is optimal, meaning that the distance is maximal from the hyper plane to the closest examples of the positive and negative classes. During performance, the method predicts the positive class if $(\underset{W}{\rightarrow}.\underset{X}{\rightarrow})\text{-b} > 0$ and predicts the negative class otherwise. Quadratic programming can be expensive for large problems, but sequential minimal optimization (SMO) is a fast, efficient algorithm for training SVMs and is the one implemented in WEKA.

For a dataset consisting of features set and labels set, an SVM classifier builds a model to predict classes for new examples. It assigns new example/data points to one of the classes. If there are only 2 classes then it can be called as a Binary SVM Classifier.

There are 2 kinds of SVM classifiers:

1. Linear SVM Classifier
2. Non-Linear SVM Classifier

Linear SVM Classifier:

In the linear classifier model, we assumed that training examples plotted in space. These data points are expected to be separated by an apparent gap. It predicts a straight hyperplane dividing 2 classes (like figure 2.2). The primary focus while drawing the hyperplane is on maximizing the distance from hyperplane to the nearest data point of either class. The drawn hyperplane called as a maximum-margin hyperplane.As a simple example, it is mentioned that hyperplane is, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.

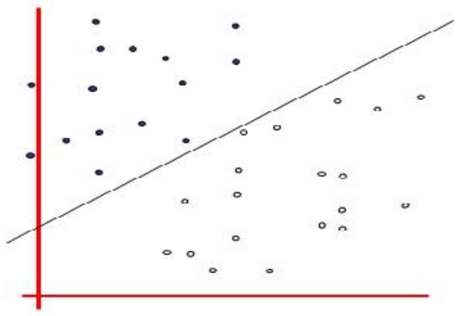


Figure 2.2: A simple diagram of linear SVM classifier.

Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.So when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

Non-Linear SVM Classifier:

In the real world, our dataset is generally dispersed up to some extent. To solve this problem separation of data into different classes on the basis of a straight linear hyperplane can't be considered a good choice. For this Vapnik suggested creating Non-Linear Classifiers by applying the kernel trick to maximum-margin hyperplanes. In Non-Linear SVM Classification, data points plotted in a higher dimensional space.

k Nearest Neighbor (kNN)

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). This algorithm is very simple to understand and used for classifying objects based on closest training examples in the feature space. KNN algorithm measures the distance between a query scenario and a set of scenarios in the data set. Its applications range from vision to proteins to computational geometry to graphs and so on. Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple.

It is also a lazy algorithm. It does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This means the training phase is pretty fast. KNN Algorithm is based on featuresimilarity, how closely out-of-sample features resemble our training set determines how we classify a given data point. The output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It can also be used for regression — output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbors.

This is in contrast to other techniques like SVM where you can discard all non support vectors without any problem. Most of the lazy algorithms – especially KNN – makes decision based on the entire training data set (in the best case a subset of them). There is a nonexistent or minimal training phase but a costly testing phase. The cost is in terms of both time and memory. More time might be needed as in the worst case, all data points might take point in decision. More memory is needed as we need to store all training data.

In the classification setting, the K-nearest neighbor algorithm essentially boils down to forming a majority vote between the K most similar instances to a given “unseen” observation. Similarity is defined according to a distance metric between two data points. A popular choice is the Euclidean distance given by but other measures can be more suitable for a given setting and include the Manhattan, Chebyshev and Hamming distance.

$$d(x,x')=\sqrt{((x_1-x'_1)^2+(x_2-x'_2)^2+\ldots+(x_n-x'_n)^2)}$$

More formally, given a positive integer K, an unseen observation xand a similarity metric d, KNN classifier performs the following two steps:

- It runs through the whole dataset computing dbetween x and each training observation. We’ll call the K points in the training data that are closest to x the set A. Note that K is usually odd to prevent tie situations.
- It then estimates the conditional probability for each class, that is, the fraction of points in Awith that given class label. (Note I(x) is the indicator function which evaluates to 1 when the argument x is true and 0otherwise)

$$P(y=j|X=x) \quad = \frac{1}{K}\sum_{i\in A} I(y^i=j)$$

Finally, our input xgets assigned to the class with the largest probability.

Naïve Bayes

Naive Bayes is a simple probabilistic classifier based on Bayes' theorem with strong independence (naïve) assumption,given a set of objects, each of which belongs to a known class, and each of which has a known vector of variables. This probabilistic method has a long history in information retrieval and text classification.

The characteristic assumption of the naive Bayes classifier is to consider that the value of a particular feature is independent of the value of any other feature, given the class variable.Despite the oversimplified assumptions mentioned previously, naive Bayes classifiers have good results in complex real-world situations. An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification and that the classifier can be trained incrementally.

Bayes theorem named after Rev. Thomas Bayes. It works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. Using the conditional probability, we can calculate the probability of an event using its prior knowledge.
Below is the formula for calculating the conditional probability.

$$\mathrm{P(H|E)}\,\frac{\mathrm{P(E|H)} * \mathrm{P(H)}}{\mathrm{P(E)}}$$

Where
P(H) is the probability of hypothesis H being true. This is known as the prior probability.
P(E) is the probability of the evidence(regardless of the hypothesis).
P(E|H) is the probability of the evidence given that hypothesis is true.

P(H|E) is the probability of the hypothesis given that the evidence is there.

It stores the prior probability of each class, *$P(C_i)$*, and the conditional probability of each attribute value given the class, *$P(v_j I C_i)$.* It estimates these quantities by counting in training data, the frequency of occurrence of the classes and of the attribute values for each class. Then, assuming conditional independence of the attributes, it uses Bayes' rule to compute the posterior probability of each class given an unknown instance, returning as its prediction the class with the highest such value:

$$C = \arg\max_{C_i} P(C_i)\prod_j P(v_j|C_i).$$

In this chapter we have illustrated some authentication researches based on mouse movement. In every research features were generated and based on characteristics of these features users are authenticated. There are many researches where some common features are used [10] [23] [26] [27] [30] [31] [33]. We have shown a list of features of existing research and proposed system.

In next chapter we have explained our proposed system. We have used three algorithms. These algorithms are briefly described in next chapter. The features and how to extract the features are also explained in next chapter.

Chapter – 3
Proposed System

In this chapter we have described our proposed system. In section 3.1 proposed features are listed. A typical structure of proposed system is briefly described in section 3.2. In next section we have provided description of *data acquisition*, *sampling data in to blocks* and *feature generations*.

User authentication occurs within most human-to-computer interactions outside of guest accounts, automatically logged-in accounts and kiosk computer systems. Generally, a user has to choose a username or user ID and provide a valid password to begin using a system. User authentication authorizes human-to-machine interactions in operating systems and applications, as well as both wired and wireless networks to enable access to network and internet-connected systems, applications and resources. We can view a generic authentication system as the figure below.

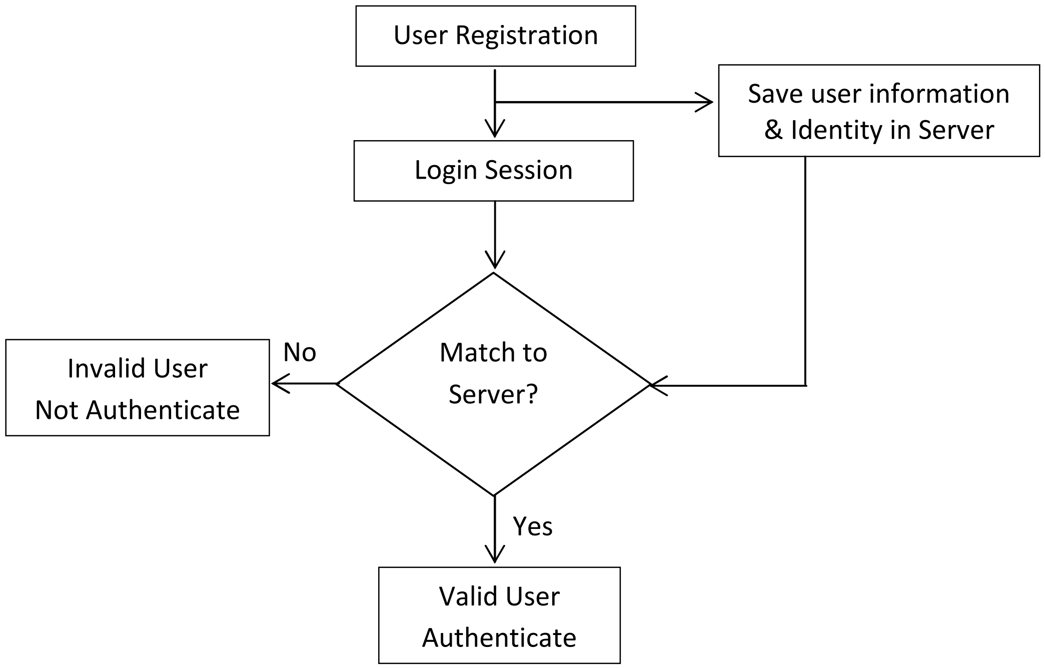


Figure 3.1:A generic flow diagram of authentication.

We have proposed a system which will capture mouse movement data like horizontal and vertical position of mouse, time taken to move mouse for a certain distance, delay time etc. It has two phases, one is *inscription* phase and another is *verification* phase. In inscription phase mouse movement data are captured in several times of particular user and stored in database. From this stored data features are generated. We have proposed twelve features. From these features value a weight is defined for each user. This weight is also stored in database. In verification phase again user mouse movement data are captured, features are generated and weight is calculated. This weight is compared with the stored weight. Finally if the comparison satisfies a particular level of difference then the user is authenticated otherwise not. We are calling this weight as signature in rest of section of this research.

3.1Proposed Features of Mouse Movement Data

In this research we have proposed twelve features as listed below. The features should be calculated using the collection of a user's mouse movement dataset. We have seen that these feature's value differs from user to user.

1) Number of Points in the trajectory
2) Delay Time
3) Number of Delay
4) Number of Action
5) STDEV of the Trajectory Length
6) Total Length of Trajectory
7) STDEV of Slope
8) STDEV of Difference Between Each of Slopes
9) Number of Curvatures
10) Curvature of the Trajectory
11) Number of Changes in Horizontal Position
12) Number of Changes in Vertical Position

** STDEV: Standard Deviation*

3.2 Proposed System

This research recognizes the mouse movement pattern of a user which is carried out by capturing user's mouse movement data. The data are preprocessed, sample into blocks and stored in database. From the saved blocks in database twelve features are generated. In authentication process, user's mouse movement features are generated in the same way. Similarities are compared between stored features during training session with features generated during authentication session. The user is authorized if the similarity found in a certain satisfactory level otherwise not authenticated.

The method that we have proposed is divided into two parts. One is *Inscription Phase* and another is *Verification Phase*. We can consider the inscription phase as training and the verification phase as testing.The proposed system is shown in figure 3.2.

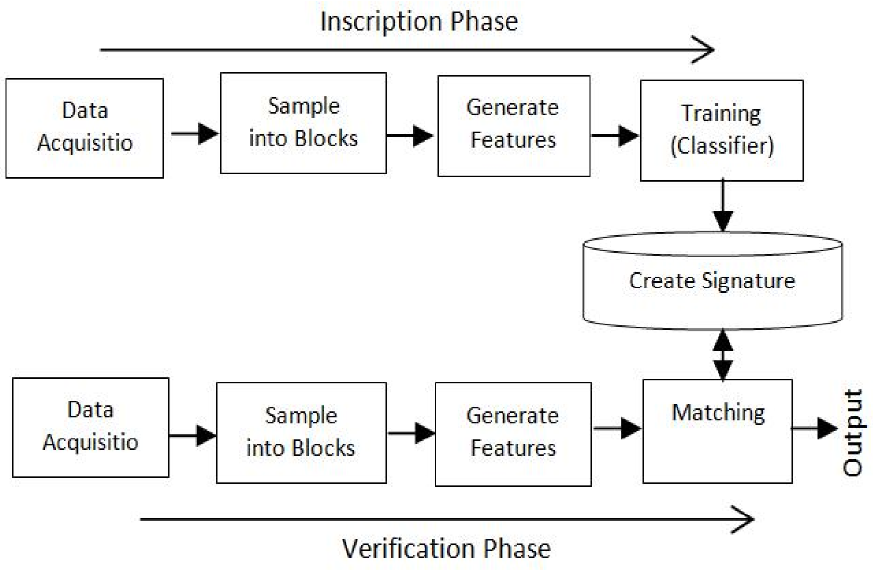


Figure 3.2: Typical structure of the proposed system.

In the sections below we have briefly describe all the sub phases of proposed system.

3.2.1 Data Acquisition

In data acquisition step, mouse movement data of a user is captured in user background whenever user is working with mouse the mouse movement data can be collected into two ways:

a) Free style mouse movement
b) Specific style mouse movement

a) Free style mouse movement

In free style work a user is being not guided for particular task. User can do any work with mouse and in his background mouse movement data are captured. In this style there is no guidance for user to move the mouse. There is no specific user interface for the user. In this style user is independent to move the mouse.

A tool/ software isrun in background when the user works with mouse. This tool keeps track of a user mouse clicks, drag the mouse, idle time of user's work with mouse etc.In this style we need more data than specific style work to identify user. Data are captured for more than an hour or day long or even several days.

In this research we have used two tools those are freely available in Internet. One is Recording User Input (RUI) and another is Jitbit Macro Reader. These tools collect user's mouse movement data in working background in free style mouse movement.(*Details in section 4.1.1 and 4.1.2*)

b) Specific style mouse movement

This is a guided work, where all the users perform similar task with mouse. The users are asked to do a task and in background of this task user's mouse movement data are captured.

For this style we need to develop a common user interface. The interface guides user to do a single job. In this interface users are allowed to perform some mouse movement and click events. In background of this interface user's mouse movement data are captured and stored in database. The user finishes his work after a specific number of clicks or specific time passed. The number of clicks and ending time depends how the interface is designed.

In this research we have also developed customized software to capture data of specific style mouse movement. It provides users a common interface to perform tasks. (*Details in section 4.1.3*)

3.2.2 Sampling into Blocks

After capturing and storing data, the data are preprocessed. Sometimes there may be unnecessary data or more than required data in database. The datasets should be ready for extracting features. For this reason the stored data are needed to be preprocessed and sample into blocks. The data are sampled into several numbers of blocks. This sampling into blocks depends on below criteria:

a) Blocks based on time
b) Blocks based on number of action

a) Blocks based on time

This type of sampling into blocks depends on how the mouse movement data will be captured and user working environment. This blocks size shows the best uses when data are captured for long time and even day long. If we want to create signature of user access permission then we need mouse movement data of general work or task which is no long. On the other hand if we want to identify user as who is working on particular PC then we need mouse movement data of long time working.

The captured data are divided into various sub group/block based on time interval. Sampling of blocks may be based on after a specific time interval of a single user for a single task. The figure below presents a user's mouse movement diagram for 7136 mille seconds. This data is being sampled in to four blocks as shown in figure 3.4, 3.5, 3.6 and 3.7. Each of these four blocks contains data of near about 1784 mille seconds. In these figures X axis defines mouse pointer value in X coordinate and Y axis defined mouse pointer value in Y coordinate.

X, Y coordinates are respectively the horizontal and vertical addresses of any pixel or addressable point on a computer display screen. The X coordinate is a given number of pixels along the horizontal axis of a display starting from the pixel (pixel 0) on the extreme left of the screen. The Y coordinate is a given number of pixels along the vertical axis of a display starting from the pixel (pixel 0) at the top of the screen. Together, the X and Y coordinates locate any specify pixel location on the screen.

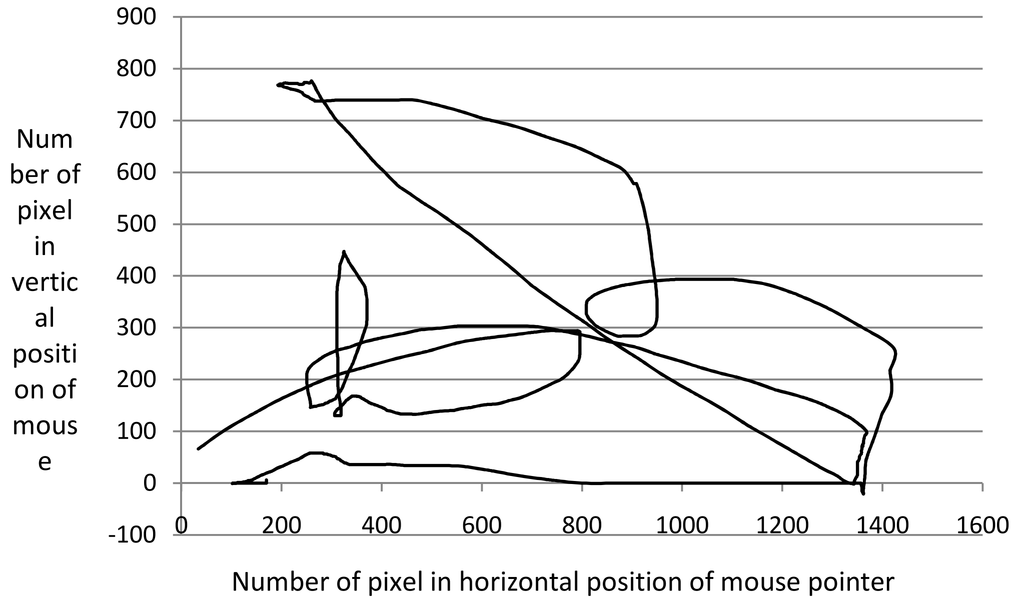


Figure 3.3:A typical mouse movement diagram based on total time of a certain task

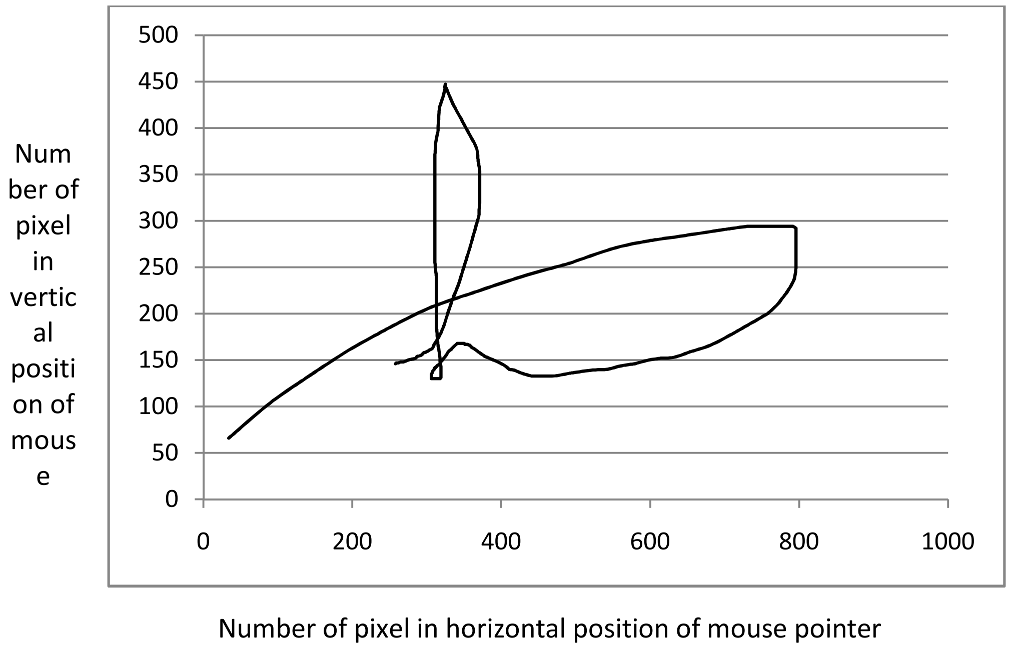


Figure 3.4:A typical mouse movement first block diagram based on time

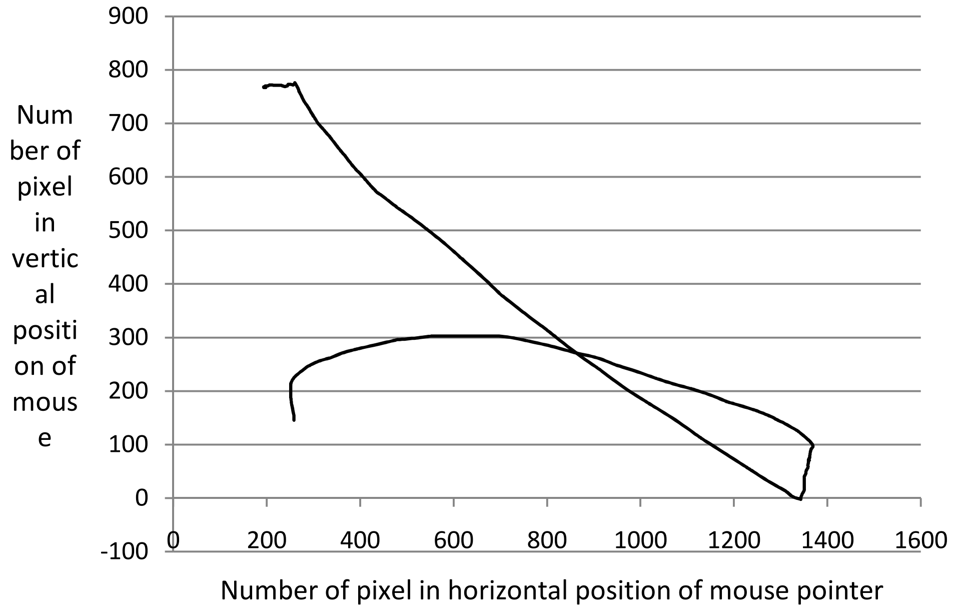


Figure 3.5:A typical mouse movement second block diagram based on time

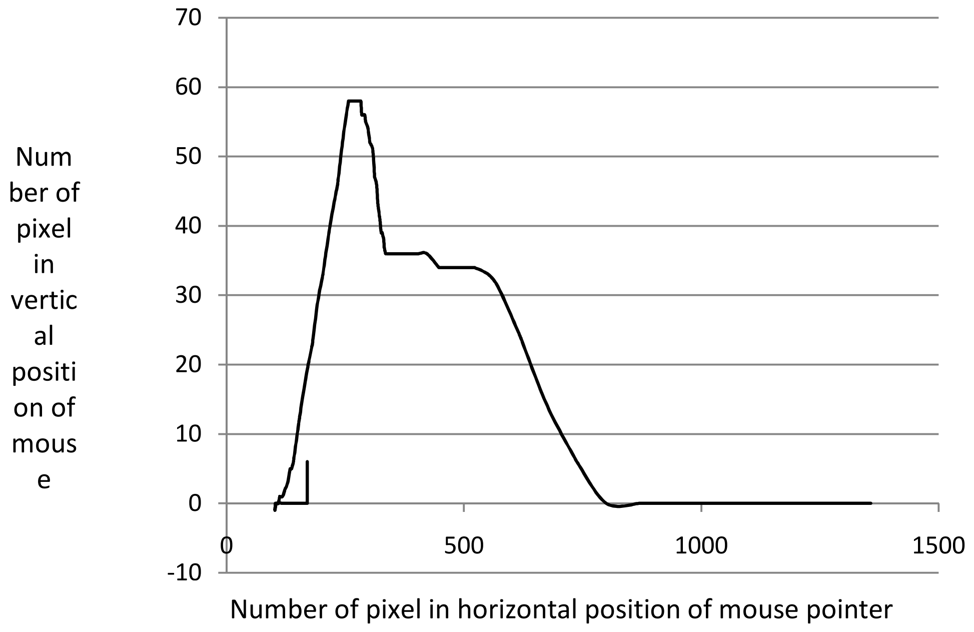


Figure 3.6:A typical mouse movement third block diagram based on time

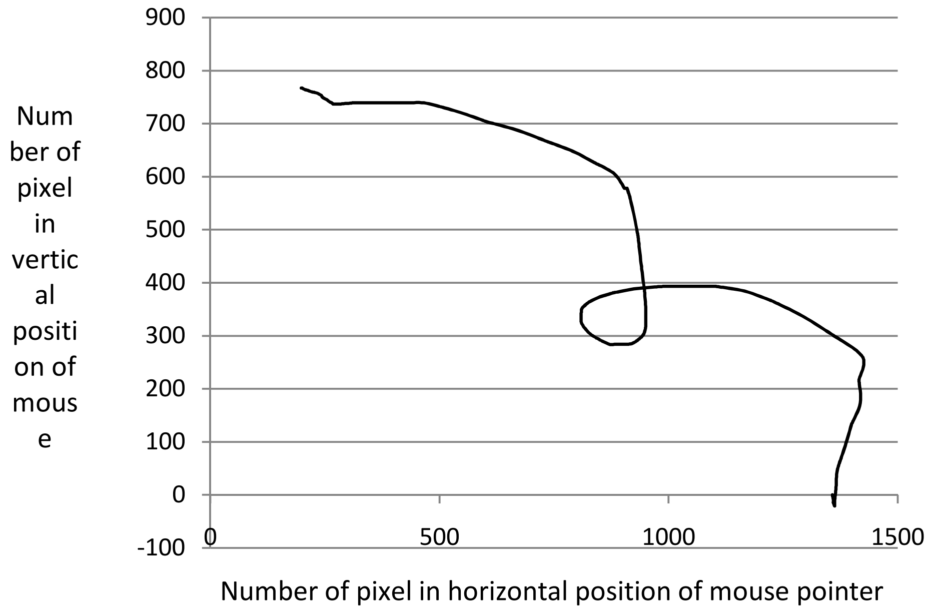


Figure 3.7:A typical mouse movement forth block diagram based on time

b) Blocks Based on Number of Action

In this blockssize the sampling depends on a specific number of action. From a collection of total mouse movement data there are a lot of actions. This single dataset are snacks into many datasetsbased on the number of predefined action. For example in figure 3.8 we have shown typical mouse movement diagram of 200 actions. This dataset is broken into four blocks and each block contains near about 50 actions.

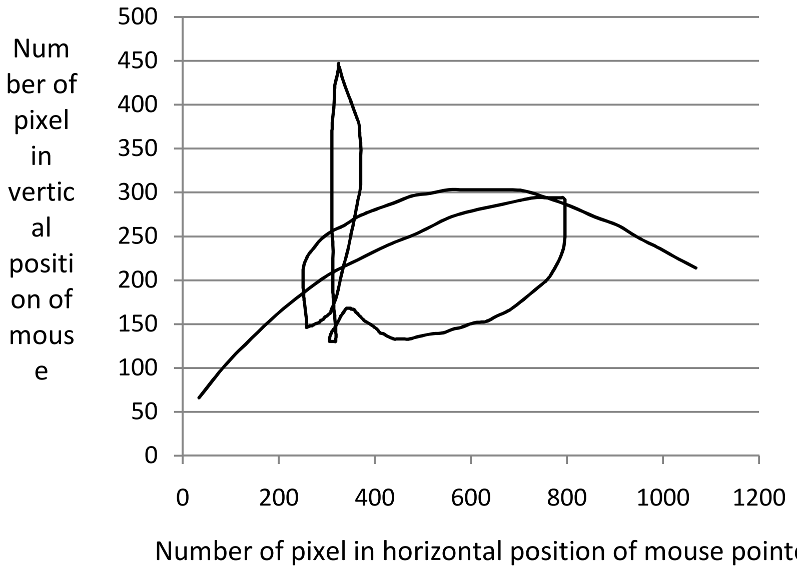


Figure 3.8:A typical mouse movement diagram of a user based on total action of a certain task

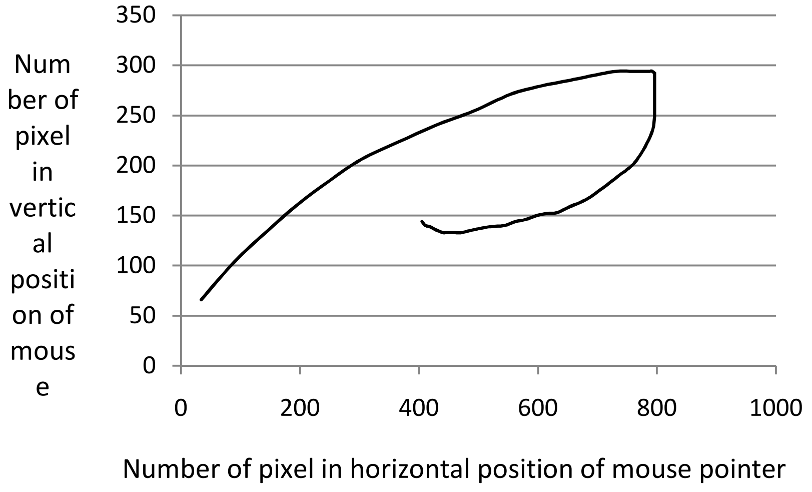


Figure 3.9:A typical mouse movement first block diagram based on action

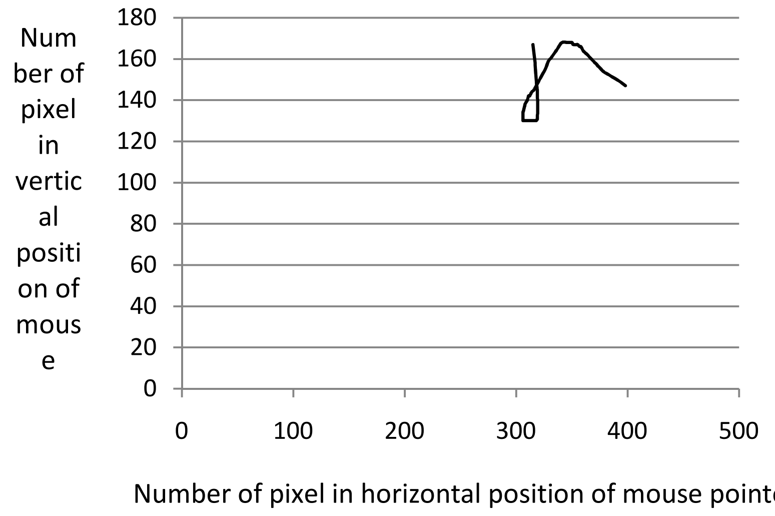


Figure 3.10:A typical mouse movement second block diagram based on action

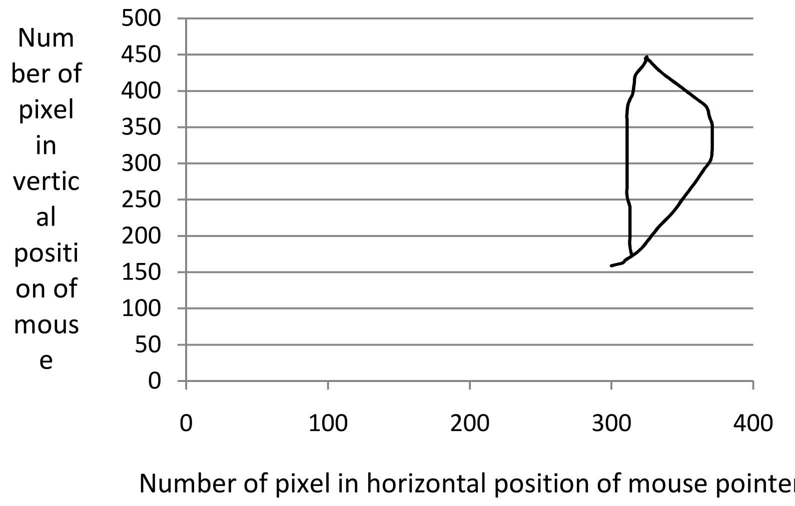


Figure 3.11:A typical mouse movement third block diagram based on action

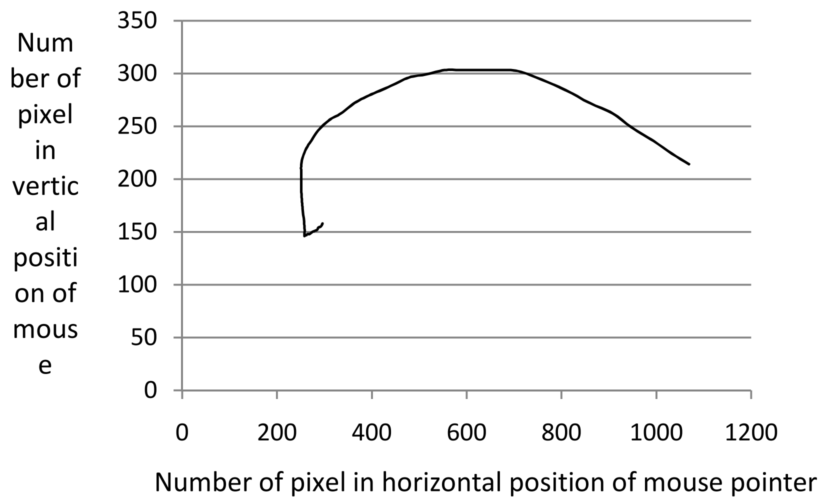


Figure 3.12:A typical mouse movement forth block diagram based on action

Signature

We have proposed signature for every user. This signature is a numeric value for each user. This value is extracted from a calculation of 12 feature's values. This calculation depends on performance of signature value as how it can differentiate. A user authentication depends of this signature. These signatures are stored in a database and work like a key whenever user wants authentication. In this proposed system signature value is unknown to user and no need to remember the signature value for authentication.

Verification

In registration phase signature of a user is defined. It is a standard signature of that user and stored in database. In verification phase again user's mouse movement data are captured, sampled into blocks, features are generated and from this features a temporary signature is captured. The new temporary signature is matched with previously stored standard signature in database. If the difference value of both signature values satisfies a particular level then the user is authenticated.Difference between the temporary signature and stored standard signature is matched by satisfying a certain level of accuracy then the user is valid user otherwise not.

Mouse Movement Data

In this proposed system we have describe some events of mouse action. From these events mouse movement data are captured. We have illustrated these events based on the following criteria.

a) **Mouse Click:** It is the click event for both left or right click

b) **Drag and Drop:** Data are taken from where the drag is started and when it is drop.

Table 3.1: A sample dataset of User A

Mouse Behavior	X Position	Y Position	Delay in Mille Second
DELAY			424
Mouse	463	176	0
Mouse	424	171	0
DELAY			11
Mouse	367	163	0
Mouse	326	156	0
DELAY			14
Mouse	292	152	0
Mouse	259	147	0
DELAY			14
Mouse	235	141	0
Mouse	216	137	0
DELAY			11
Mouse	195	130	0
Mouse	179	126	0
DELAY			14
Mouse	161	119	0
Mouse	142	114	0
DELAY			14
Mouse	125	110	0

Table 3.2: A sample dataset of User B

Mouse Behavior	X Position	Y Position	Delay in Mille Second
DELAY			250
Mouse	34	66	
DELAY			12
Mouse	90	104	
Mouse	144	134	
DELAY			12
Mouse	196	161	
Mouse	250	185	
DELAY			12
Mouse	303	206	
Mouse	364	223	
DELAY			12
Mouse	428	240	
Mouse	491	254	
DELAY			12
Mouse	554	271	
Mouse	603	279	
DELAY			12
Mouse	654	285	
Mouse	703	291	

3.2.3Feature Generation

We have studied how other researchers have generated features. The Mouse trajectory arises from the actions e.g. System wake up, Move and click, Highlight, Drag and drop, Button Event, movement Speed, Direction etc[2] [7][12] [13] [14].

The mouse movement data are captured to uniquely identify auser based on some special features.The proposed mouse movement biometric system calculates twelvefeatures from x-y mouse position and consecutivedelay based on time interval. Mathematical law and calculations are used to generate features.

These features are converted into real number values which differ from user to user.

1. Number of Point in the Trajectory: The total number of point for a certain interval, where there is no idle time or no delay of work.

   $\sum_{t=o}^{n}(p_t)$ Where t = time when there is a mouse action

2. Delay Time: Measures the total amount of time when mouse movement was in delay.

   $\sum_{i=1}^{n}(t_i)$ Where t= delay time

3. Number of Delay: Number of idle time of the Trajectory.

4. Number of Action: Determines the number of action.

5. STDEV of the TrajectoryLength: First adding point to point length then standard deviation of these lengths.

   STDEV of $\sum_{i=2}^{n}\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$,

6. Total Length of Trajectory: Measures the point to point distance of the trajectory.

   $\sum_{i=2}^{n}\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$

7. STDEV of Slope (m): Determines the standard division of direction angle from point to point in the Trajectory.

   STDEV of $m_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$, where i=2 to n.

8. STDEV of Difference Between each of Slopes: Measure the difference between slope and its standard deviation.

   $STDEVof\ (m_i - m_{i-1})$, where i=2 to n.

9. Number of Curvatures: Measures the number of changes between the angles of the Trajectory.

10. Curvature of the Trajectory: Measures the total length of the Trajectory (6) divided by the distance between the starting and ending position of Trajectory.

11. Number of Changes in Horizontal Position: Measures how many times the Trajectory changes in X axis.

12. Number of Changes in Vertical Position: Measures how many times the Trajectory changes in Y axis.

If we calculate these 12 features based on the data of table3.1 and 3.2, then results will look like below:

Table 3.3: Example of features' value of two users

SL	Features	Feature's Value of User A	Feature's Value of User B
1	Number of Points in Trajectory :	13	12
2	Delay Time :	502	72
3	Number of Delay :	7	6
4	Number of Action(no of rows) :	20	18
5	STDEV of Trajectory Length :	12.629556	6.491907
6	Total Length :	345.29009	713.8437
7	STDEV of Slope :	0.0840552	0.181709
8	STDEV of Difference between each of Slopes:	0.0855285	0.052777
9	Number of Curvatures :	12	12
10	Curvature of the Trajectory :	0.6971007	9.614984
11	Number of Changes in Horizontal Position :	12	13
12	Number of Changes in Vertical Position :	12	13

In next chapter we have describe the experiments. Many experiments are done in this research. We have described the main experiments with results. Experimental results are described and by comparing results, better experiments are chose for further use.

To check the validity we have test our proposed system by three classifiers. Output results of these classifiers are discussed. In next chapter we have discussed and compared our proposed system with existing system in similar field using similar benchmark data.

Chapter – 4
Experimental Results and Discussion

In this chapter we have illustrated some related experiments of proposed system and analyzed the results. We have applied the proposed method on different data from different sources. For a sequential discussion of the experiment and result, we have narrated rest of this chapter as the following sections.

Experiment 1: Free Style Data Tracer Using JitbitTool

Experiment 2: Free Style Data Tracer Using RUI Tool

Experiment 3: SpecificGuided User Interface (UI)

Experiment 4: Using Benchmark Data

4.1 Experiment Setup

Many experiments were done during this research to get better result for this proposed system working. The first set of experiments was aimed for finding the parameters, which could relate a user with his mouse movements and could help in authenticating him/her by generating and using signature. These parameters had to be unique for every user to make the variation of mouse movements between different users distinct.

The next set of experiments was done to find a way to make authentication task. Wehavefound a way to record every mouse coordinate in the user's working background. The idea is to filter out the data coordinates to get a limited number of useful data coordinates without the loss of any information.

Another experiment was done to find the dense regions on the screen where the mouse moved the most. The next experiment involved finding a way to use the parameters we found in the first experiment so that we could authenticate users based on their mouse movement's behavior. All the experiments are explained in the next sub sections.

We have performed our experiments using machine learning software WEKA. The following configuration is required for the experiment.

- Operating system Windows, OS X, Linux
- WEKA: 3.6
- Java 1.5 or later

In this thesis we have used:

- Operating system: Windows 7
- WEKA: 3.6
- Java: version 7

In implementing section we need configuration as the below:

- PhP: 5.4
- MySql: 5.5
- Server: Apache 6.0

4.1.1 Experiment 1: Free Style Data Tracer Using JitbitTool

For this experiment the data acquisition is performed by freely available tool. We have captured data using mouse movement capture software named "Jitbit macro reader"[47]. Six students of undergraduate level work as voluntaries in this experiment. They have performed some task using mouse in three times. They have seated together in a computer labsame as [7], [11].JitBit Macro Recorder is installed all the workstation and it records data of mouse movement and mouse clicks. All recorded mouse movementdata can be saved to disk for further use.Mouse movements data are captured from users are given below.

- Points in X axis
- Points in Y axis and
- Delay time between two actions.

We demonstrated the work to volunteers. Another volunteer helped them to setup the software and collecting the raw data. Primarily these raw data are stored in a log file. A sample of such log file is shown in figure 4.1.

4.1.2 Experiment 2: Free Style Data Tracer Using RUI Tool

In previous experiment we have collected data by using a free tool JitBit recorder. This tool collects data in its own style. There are more tools available in Internet. Each tool has its own style to collect data. Methods of collecting data are also differing from one tool to other tool. Another popular tool is Recording User Input (RUI) which is freely available tool to collect user's mouse dynamics [23][48]. RUI is a keystroke and mouse action logger for the Windows and the Mac OS X (10.3 and later) platforms. There are options in RUI's user interface to choose which types of actions to record, including keystrokes and mouse movements. The collected data is stored in a log file as shown in Figure 4.3, as a list of the timestamp, action, and arguments (if any, such as key pressed or move location). To start and stop recording, hot keys are provided (Ctrl+R to Record, and Ctrl+S to Stop). The amount of data recorded with extensive mouse movements can be as high as 22k / min. If the user exclusively moved the mouse it could be as high as 156k / min. So less mouse-intensive activities will have correspondingly much smaller log files. The data collected can be replayed using an included Replay option, and can be played faster or slower than real-time. While replaying the keystrokes, if the observer wishes to see the same results occur in the interface, it is essential that the system be in the same state it was in before the recording started. Otherwise, the mouse moves and the keystrokes are created and performed by RUI but the resulting effects are likely to be different. It is worth noting that very fast-paced interfaces with variable system time lags may lead to difficulties with playback.

4.1.3 Experiment 3: Specific Guided User Interface (UI)

In this experiment we have collected data in a guided way using custom built interface. We have developed an interface to collect data where users perform a particular task for this experiment. It is one type of guided mouse gesture [3][9][11][12][14].This experiment capturesuser's mouse movement behavior fromacommon interface. In this interface, a ball (pointer) is displayed. If a user clicks on that ball (pointer) then it vanishes and displays in a different spot of the screen. If the user clicks on that spot then it vanishes again and displayed in another spot. During the user moves or work with mouse, the mouse movement data are captured in user working background. This task is guided, because every user performs the task as pre specified way by the software we have developed.

4.1.4 Experiment 4: Using Benchmark Data

The proposed system is tested by using benchmark data. We have also collected benchmark data for analyzing and comparing our work with other related researchers. We have collected data from [43]and [44].

[43]Is a research lab named Information security and object technology (ISOT). The Information Security and Object Technology (ISOT) Research Lab was founded in 1999, and since then has been carrying innovative research in computer security and software engineering. The ISOT Lab has collected various dataset from different projects. They are publiclyavailable. Mouse Dynamics Dataset is also available there. The ISOT mouse dynamics dataset consists of mouse dynamics data for 48 users collected over several months. Dr. IssaTraore of University of Victoria is responsible person for the dataset [2][14].

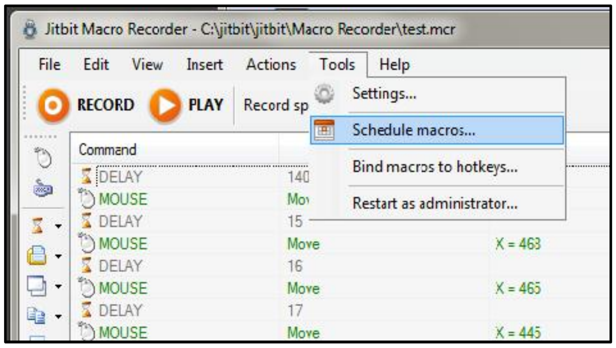
The second data source [44] is a shared data set for mouse dynamics collected for continuous authentication, under a free environment. The Raw Mouse behavior data collected under a free environment from 28 users, for the purpose of continuous authentication. On this webpage, they share the data, scripts, and results of evaluation so that other researchers can use the data, reproduce results, and extend them; or, use the data for investigations of related topics, such as intrusion, masquerader or insider detection. Others researchers have also used this data to test their methods [4].

4.2 Data Acquisition and Description of Data

In this section we have described collecting and storing data for four experiments described in section 4.1.We have collected some raw data from computer lab where students were working. Such data are collected in several days from several people. In this thesiswe are mentioning these data as our own data. Some data are collected from other sources [43][44]which are benchmark data.

4.2.1 Experiment 1: Free Style Data Tracer Using JitbitTool

For this experiment user were working in free style at their PC and in background the JitBit recorder captured mouse movement data. These data are stored in a log file which is needed to be processed for long time storage. So these data are modified and made it suitable for our target criteria. We have made the data suitable to store in database.


Jitbit Macro Recorder - C:\jitbit\jitbit\Macro Recorder\test.mcr
File
Edit
View
Insert
Actions
Tools
Help
RECORD
PLAY
Record sp
Settings...
Schedule macros...
Bind macros to hotkeys...
Restart as administrator...

```
Command
DELAY     140
MOUSE     Mo
DELAY     15
MOUSE     Move              X = 463
DELAY     16
MOUSE     Move              X = 465
DELAY     17
MOUSE     Move              X = 445
```

Figure 4.1: Log file of collecting data using JitBit

Table 4.1: A sample dataset of a user of experiment-1

Mouse Behaviors	X position/ Delay Time	Y position
DELAY	225	
Mouse	40	60
DELAY	325	
Mouse	40	60
DELAY	13	
Mouse	38	67
Mouse	35	74
DELAY	12	
Mouse	32	81
Mouse	27	93
DELAY	12	
Mouse	27	98
Mouse	25	108
DELAY	12	
Mouse	24	119
Mouse	24	129
DELAY	12	
Mouse	21	139
Mouse	20	147
DELAY	12	

After collecting the data we have sampled a total dataset into various numbers of blocks. These sampling is done based on number of action and time (details in section 3.4.2). Blocks are required to analyze the each user mouse movement behavior from very closed.

The Figure 4.2 shows a user's four block diagrams. In these figures mouse movements of X positions are shown in x- axis and Y positions are shown in y-axis.

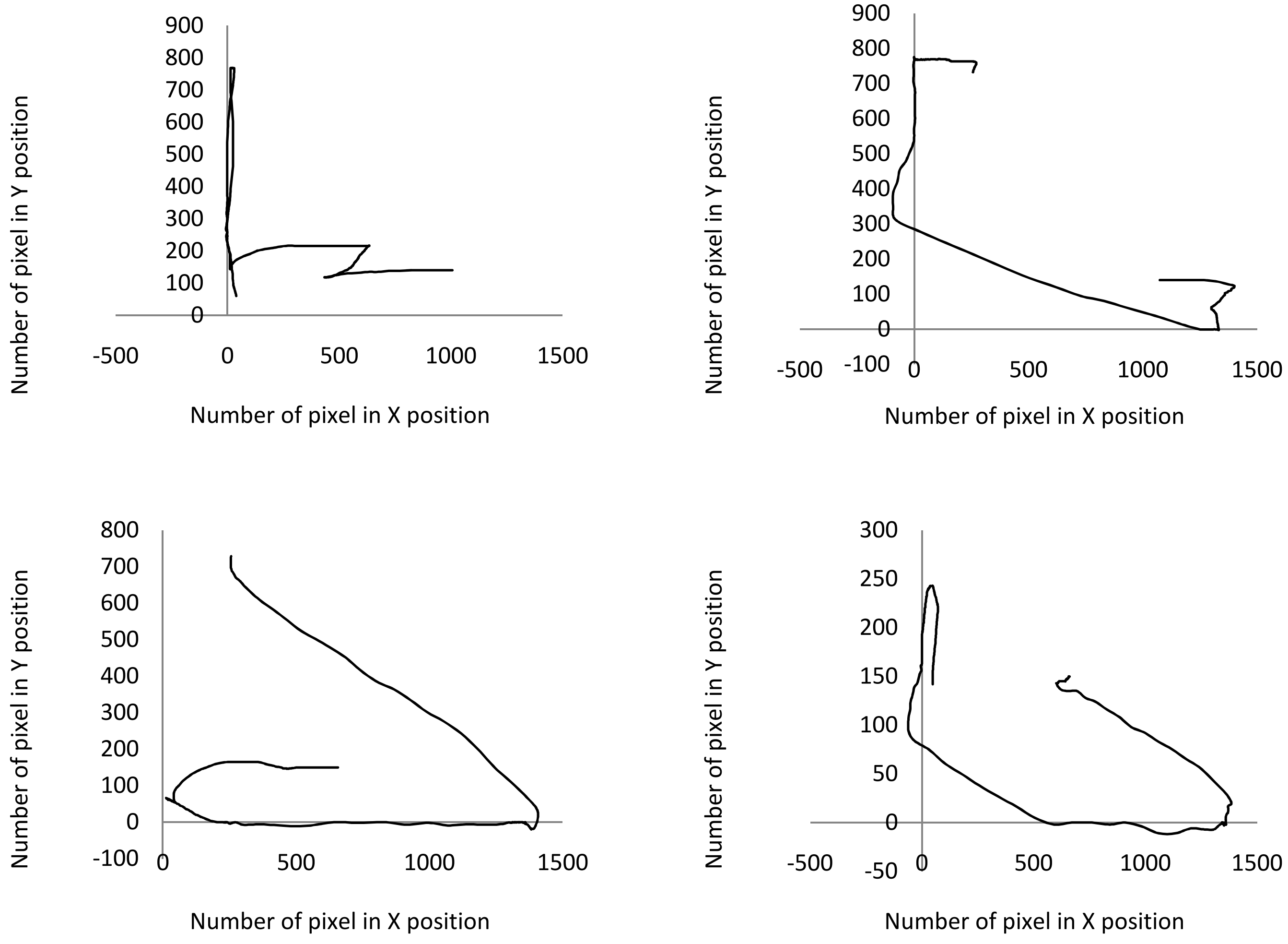
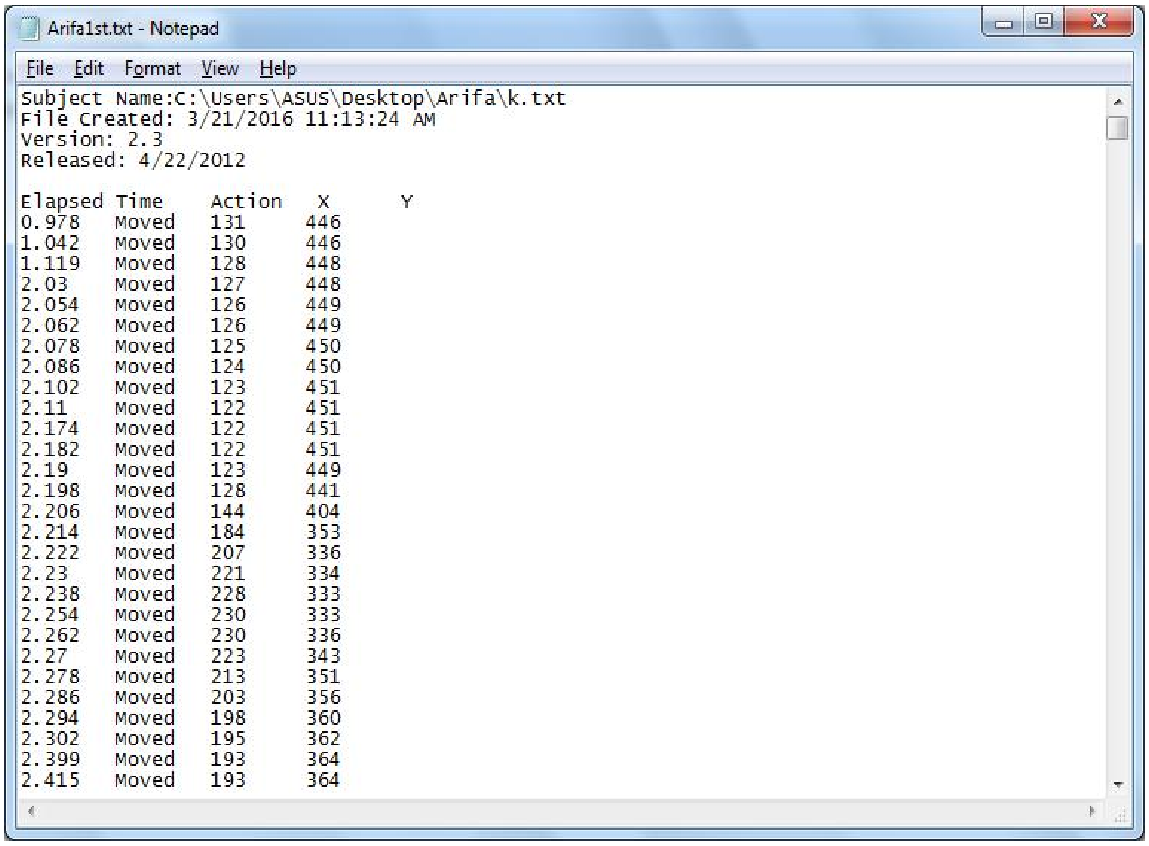


Figure 4.2: Mouse movement visual diagram of a user in four blocks.

4.2.2 Experiment 2: Free Style Data Tracer Using RUI Tool

To test our proposed method with data collected by RUI, RUI is installed in every PC in a computer lab. Students were working there after starting RUI and mouse movement data were being collecting in user work background. Data are collected from several users in two different days.Eight students of computer trainee perform this task as volunteer.

Data are collected in a log file, shown in figure 4.3. We have taken data from each user log file and make the data suitable to store in database.



```
Subject Name:C:\Users\ASUS\Desktop\Arifa\k.txt
File Created: 3/21/2016 11:13:24 AM
Version: 2.3
Released: 4/22/2012

Elapsed Time    Action   X       Y
0.978   Moved   131     446
1.042   Moved   130     446
1.119   Moved   128     448
2.03    Moved   127     448
2.054   Moved   126     449
2.062   Moved   126     449
2.078   Moved   125     450
2.086   Moved   124     450
2.102   Moved   123     451
2.11    Moved   122     451
2.174   Moved   122     451
2.182   Moved   122     451
2.19    Moved   123     449
2.198   Moved   128     441
2.206   Moved   144     404
2.214   Moved   184     353
2.222   Moved   207     336
2.23    Moved   221     334
2.238   Moved   228     333
2.254   Moved   230     333
2.262   Moved   230     336
2.27    Moved   223     343
2.278   Moved   213     351
2.286   Moved   203     356
2.294   Moved   198     360
2.302   Moved   195     362
2.399   Moved   193     364
2.415   Moved   193     364
```

Figure 4.3: Log file of collecting data using RUI

Table 4.2: A sample dataset of a user of experiment-2

Elapsed Time	Action	X Position	Y Position
16.143	Moved	1206	72
16.657	Pressed Left	1206	72
19.142	Moved	215	745
19.158	Moved	214	746
19.27	Pressed Left	214	746
912.152	Moved	690	340
912.424	Key	Space	
1478.545	Key	B	
1479.801	Key	Back	
1479.849	Moved	297	204
1480.337	Moved	294	207
1480.817	Key	LShiftKey	
1481.385	Key	B	Shift
1482.833	Moved	293	207

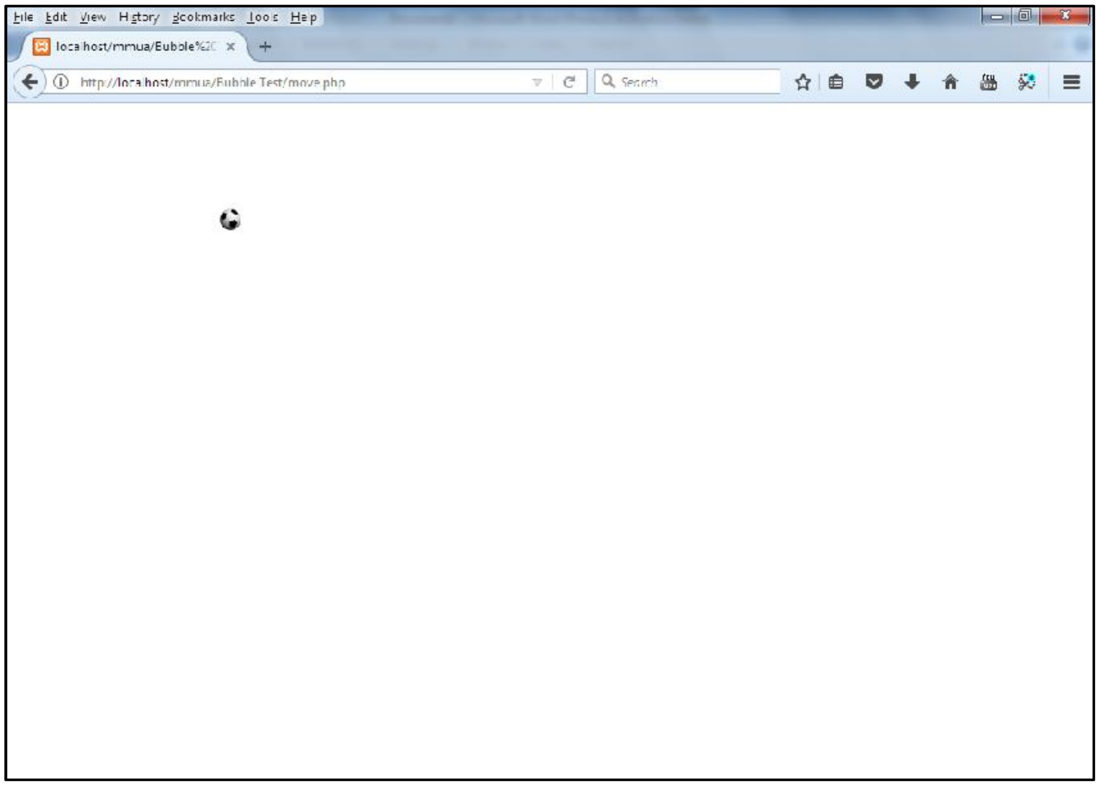
4.2.3 Experiment 3: SpecificGuided User Interface (UI)

For this experiment we have developed a user interface. The interface is stored in a server computer and user can access the interface by using modern browser through client computer. So that, several users can use the interface at a time from different physical location. We have used php and Jquery to develop the interface. The single interface can be used by several users over a computer network. For this experiment we have taken another four volunteers. These four users run the program in same time and mouse movement data are captured in their session background. Data are stored in MySql database. As mentioned in section 4.1.3 when the program loaded a small ball (pointer) displayed in the interface. When the user clicks on the ball (pointer) then it goes to another place. When the user click on that position then the ball (pointer) again goes another place and this procedure runs for 1500 mille second or up to 10 clicks.

We have used Jquery to calculate time as the function below. We have used anonymous function which is similar to regular functions, in that they contain a block of code that is run when they are called. Name of the function here is *setInterval()*.The function can also accept arguments, and return values. By this function we have calculated time as mile second. In every interval of 10 mile second the system records time and X-Y position of mouse. To move the ball (pointer) we have used built-in javascript function, *hide()* and *show()*. Source code of this function is shown below.

```
$(document).ready(function(){
$("#ball1").click(function(){
$("#ball1").hide();
$("#ball2").show();
var timing=0;
asha=setInterval(function(){$("body").one('mousemove',function(e){
if(timing>=15000 || document.getElementById("ball10").style.display=="none"){

document.getElementById("submit").style.display="block";
clearInterval(asha);
return false;
}
else{
    timing=(timing+10)
    var mx=e.pageX;
    var my=e.pageY;
    $("#ashatr:last").after("<tr><td><input type='text' name='x[]' value='"+mx+"'/></td><td><input type='text'
    name='y[]' value='"+my+"'></td><td><input type='text' name='t[]'value='"+timing+"'/></td></tr>");
}
});
},10);
});
$("#ball2").click(function(){
$("#ball2").hide();
$("#ball3").show();
});
$("#ball3").click(function(){
$("#ball3").hide();
$("#ball4").show();
});
$("#ball4").click(function(){
$("#ball4").hide();
$("#ball5").show();
});
$("#ball5").click(function(){
$("#ball5").hide();
$("#ball6").show();
});
$("#ball6").click(function(){
$("#ball6").hide();
$("#ball7").show();
});
$("#ball7").click(function(){
        $("#ball7").hide();
        $("#ball8").show();

});
$("#ball8").click(function(){
        $("#ball8").hide();
        $("#ball9").show();
});
$("#ball9").click(function(){
        $("#ball9").hide();
        $("#ball10").show();
});
$("#ball10").click(function(){
        $("#ball10").hide();
});
});
```

The user interface and stored data are shown in below figures.



File Edit View History Bookmarks Tools Help
localhost/mmua/Bubble%20
http://localhost/mmua/Bubble Test/move.php
Search

Figure 4.4 (1): Loading the interface

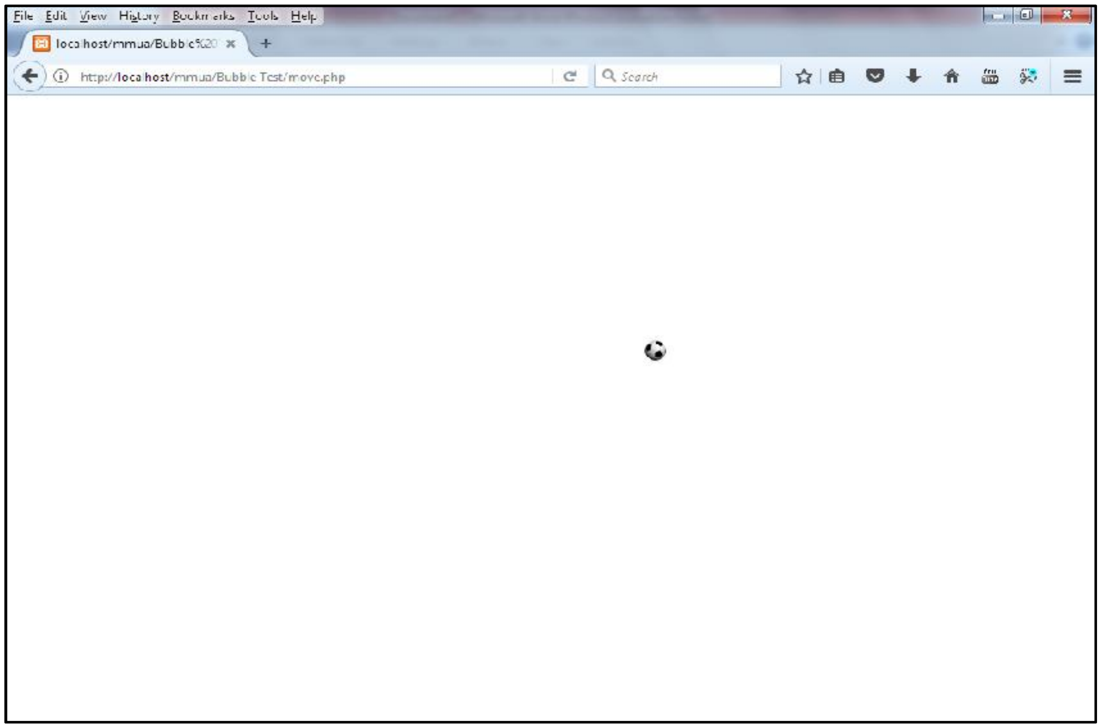

File Edit View History Bookmarks Tools Help
localhost/mmua/Bubble%20
http://localhost/mmua/Bubble Test/move.php
Search

Figure 4.4 (2): After the first click

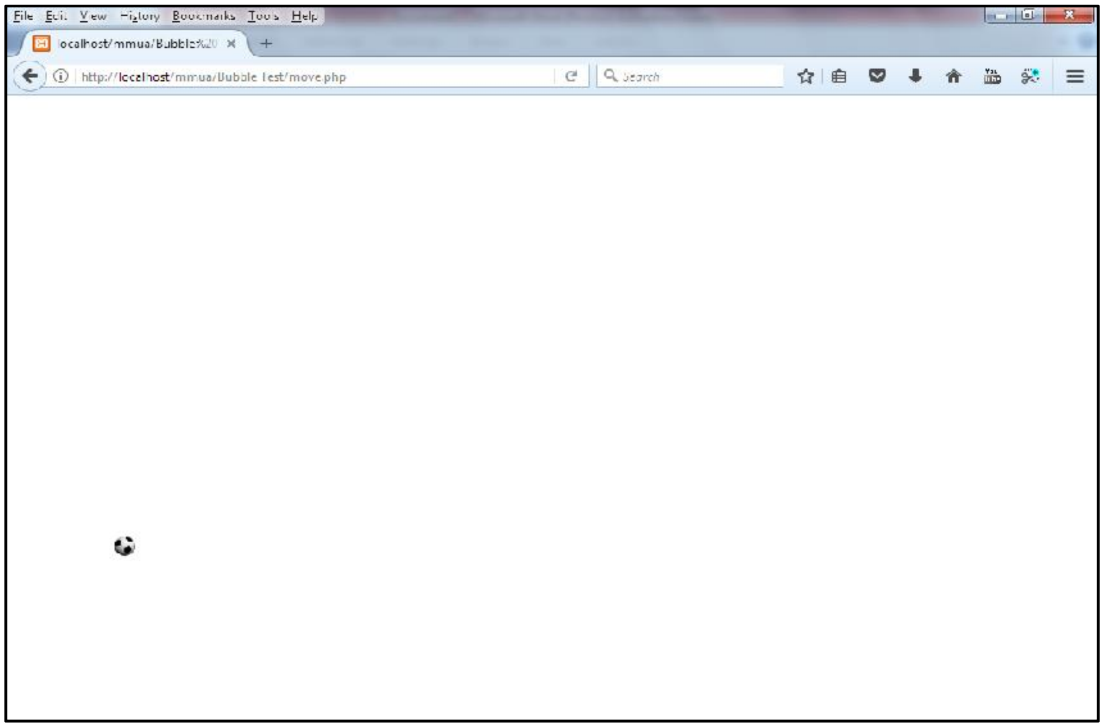

File Edit View History Bookmarks Tools Help
localhost/mmua/Bubble%20
http://localhost/mmua/Bubble Test/move.php
Search

Figure 4.4 (3): After the second click

Figure 4.4(1-3) Particular guided user interface of collecting data

Table 4.3:A sample dataset of a user in experiment-3

X position	Y Position	Time Elapsed in Mile Second
211	113	10
211	113	20
211	113	30
211	113	40
211	113	50
211	113	60
211	113	70
211	113	80
211	113	90
211	113	100
211	113	110
211	113	120
211	113	130
211	113	140
211	113	150
211	113	160
211	113	170
211	113	180
211	113	190
211	113	200

4.2.4 Experiment 4: Using Benchmark Data

The datasets[43] consist of two sets of raw mouse dynamics data for 22 and 26 different human users collected in 2003 and 2007, respectively; this represents in total 48 different users. The datasets consist of several files which can be opened using matlab. Each file consists of a number of matrices, each representing the data collected from one user.

Matrix columns represent the following:

Column 1: Type of Action (1: Mouse Move 2: Silence 3: Point Click 4: Drag-drop)

Column 2: Traveled Distance in pixels

Column 3: Elapsed Time (in seconds)

Column 4: Direction of Movement (1 to 8) (actions performed within 45-degree intervals clockwise).

Table 4.4: A sample dataset of collected benchmark data from [43]

Action Type	Travel Distance	Elapsed Time	Movement Direction
1	351	0.75	4
3	277	1.75	4
1	408	2	8
2	0	11	0
3	369	1	2
3	551	1.25	2
3	475	1.25	6
4	1	0.25	6
3	1252	1.75	4
3	1	0.25	4
1	627	1.5	7
1	345	2.75	1
2	0	4	0
1	511	2.25	2
1	634	1.75	5
1	451	2	2
3	1	0.5	2

The dataset [44]consist of mouse dynamics information from 28 subjects, each of who accomplish at least 30 data sessions. Each session consists of about thirty minutes of a user's mouse activity data (around 3000 mouse operations).

The following data is for map relationship between the code and the mouse event.

512 => "MouseEvent(WM_MOUSEMOVE)", # WM_MOUSEMOVE

513 => "Click()", # WM_LBUTTONDOWN

514 => "MouseEvent(WM_LBUTTONUP)", # WM_LBUTTONUP

515 => "DblClick()", # WM_LBUTTONDBLCLICK

516 => "RightClick()", # WM_RBUTTONDOWN

517 => "MouseEvent(WM_RBUTTONUP)", # WM_RBUTTONUP

518 => "RightDblClick()", # WM_RBUTTONDBLCLICK

519 => “MiddleClick()”, # WM_MBUTTONDOWN

520 => “MouseEvent(WM_MBUTTONUP)”, # WM_MBUTTONUP

521 => “MiddleDlbClick()”, # WM_MBUTTONDBLCLICK

We collect the details from “Mouse Input Notifications” on MSDN, from [45].

Table 4.5: A sample dataset of collected benchmark data from [44]

Event	X position	Y Position	Time
512	165	98	1252718125
512	165	97	1252718125
512	165	97	1252718156
513	165	97	1252718218
514	165	97	1252718296
513	165	97	1252718390
514	165	97	1252718468
512	165	97	1252718484
512	166	97	1252718500
512	166	97	1252718500
512	167	97	1252718515
512	168	97	1252718515
512	169	97	1252718531
512	171	97	1252718531
512	174	97	1252718546
512	177	97	1252718546
512	185	98	1252718562

These data are collected using a free experimental environment was established to collect the mouse behavior data in this study. They developed data collection software that runs as a background job, starts monitoring the user’s actions when the user’s login occurs and stops running when the user’s logout occurs; the software is totally transparent and does not affect other applications. Mouse behavior data were collected during users’ routine computing activities, which mainly cover the mouse actions under the application of Internet surfing, word processing, online chatting, programming, and online games. This setting reflects a real computing environment. During the course of data collection, all users were asked to use mouse to do their routine work for about thirty minutes long, which represents one data collection session. Whenever the user moves or clicks the mouse, the data collection software records the event type (i.e., mouse move or mouse click), the position of the event occurred, the timestamp when the event occurred, and the application information in which the event occurred. In this way, mouse activity data are collected in terms of sessions, and every session consists of about thirty minutes of a user’s mouse activity data.

They [44]set up several desktops to collect the data, and all of them were connected to a central server via the Internet. The server stores the collected data in an internal database, along with the user ID. The desktops are HP workstations with a Core 2 Duo 3.0 GHz processor and 2.0GB of RAM; they are equipped with identical 17” HP LCD monitors (set at 1280×1024 resolution). They equipped the computers with a USB HP optical mouse, running the Windows XP operating system. The server configuration is a Dell PowerEdge server with an Intel Xeon X5677 3.46 GHz Quad Core Processor and 12.0 GB of RAM, running the Windows Server 2003 operating system.

4.3 Feature Description

In this section we have briefly described twelve features those we have proposed in this research. We have collected data of six users in experiment 1 (section 4.1.1). These data are stored in a log file which contains a raw data. From these raw data the 12 features are extracted.

After collecting the data we have preprocessed and filtered those and generated features. From these features unique signature is generated for each user. There are some parameters which can be used for finding unique characteristics of a user. Nature of parameters and features depend on behavior of using the mouse. We have emphasize on mouse action, vertical position, horizontal position, elapsedtime in milliseconds, curvature etc. These data are captured inuser's working background.

Feature Vector Details

The captured mouse movementdata were used to uniquely identify a user based on the profile of the user, developed while using the mouse to complete their task using mouse. The proposed mouse movement biometric system divided the featured measurements into categories: Mouse Trajectory, Mouse Click, Mouse Wheel or Spin Scroll and Mouse Activity time. The Mouse Trajectory arises from the following actions [19]:

1. System wake up: The mouse is moved or interacted to wake up the operating system, no mouse clicks at either end of the trajectory.

2. Move and click: The mouse is moved to a location on the screen to perform an action such as clicking on an object, etc. The trajectory begins without a mouse event and ends in a mouse click

3. Highlight: A section of text or an object is highlighted. This action begins with a left mouse click/hold to begin the highlighting and ends with the mouse release.

4. Drag and drop: An object is dragged and dropped. This action begins with a left mouse click/hold and ends with the mouse release

Each of the actions was compromised by the basic mouse trajectoryfeatures thatwere used to characterize all mouse movement and those features are described below using data captured in experiment -1(section 4.1.1).

Features are described below separately for each six users.

1. **Number of Point in the Trajectory:**The total number of point for a certain interval, where there is no idle time or delay of work.

$$\sum_{t=o}^{n}(p_t)$$

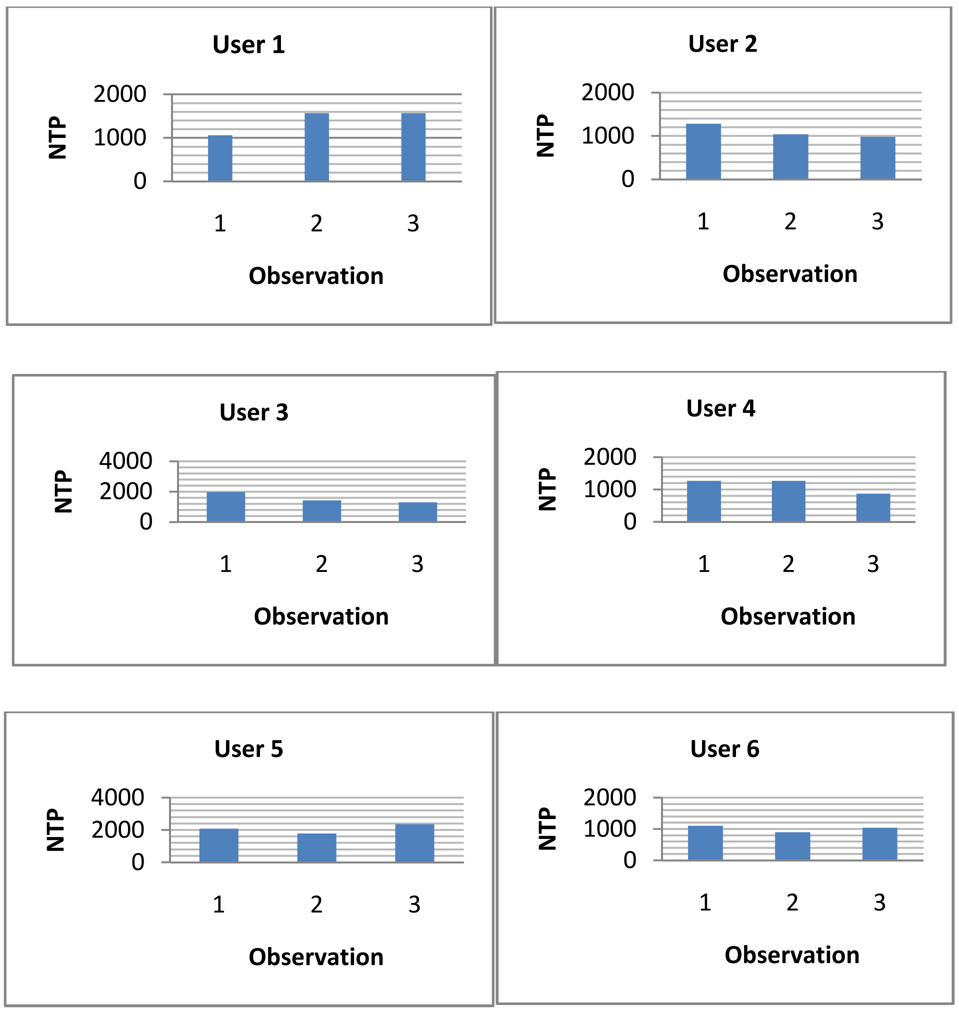


Figure4.5: Number of Trajectory Points (NTP) of Different user.

2. **Delay Time:**Measures the total amount of time when mouse movement was in delay.

$$\sum_{i=1}^{n}(t_i) \text{ Where t= delay time}$$

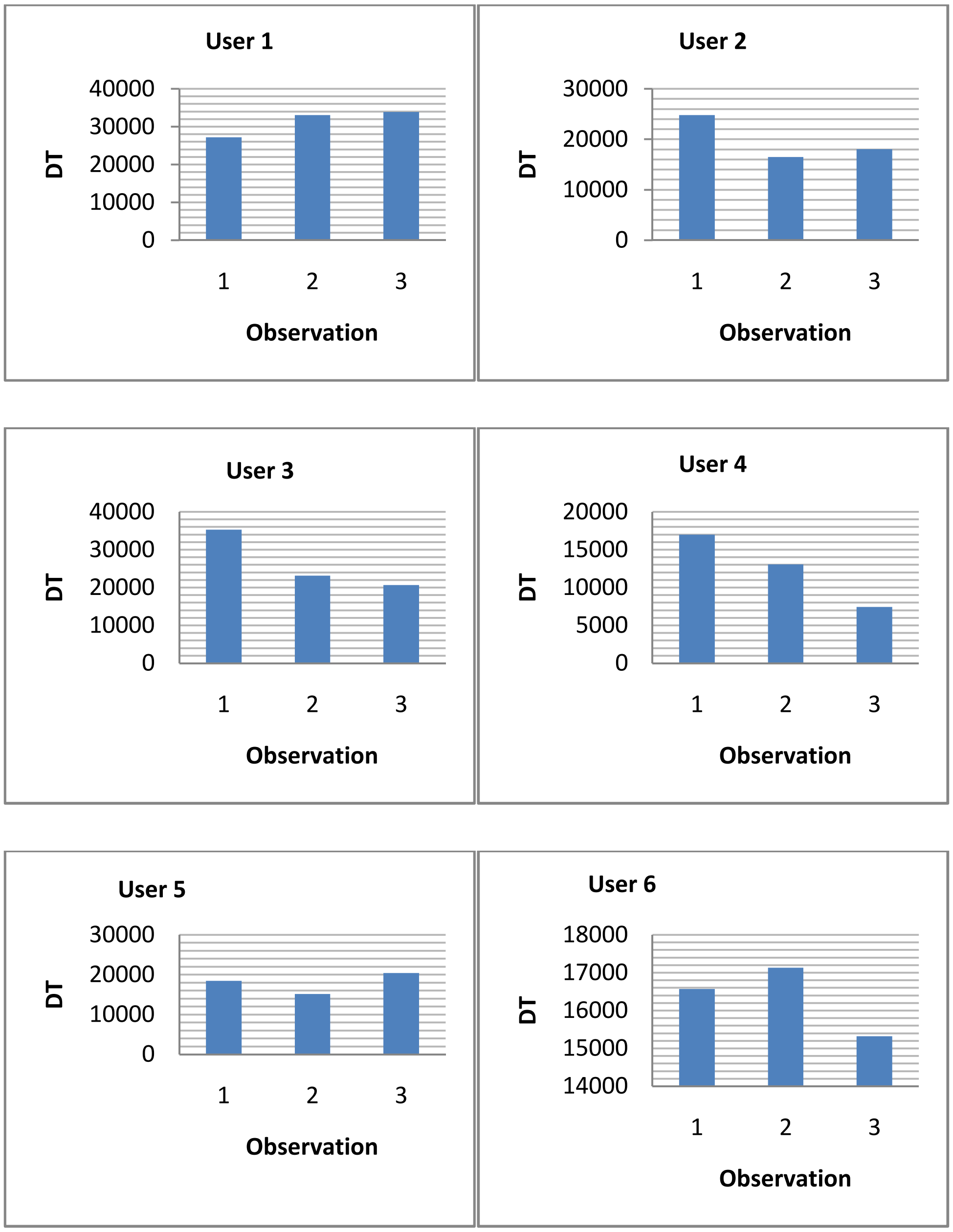


Figure 4.6: Delay Time (DT) of Different user.

3. **Number of Delay:**Number of time when the Trajectory is in delay.

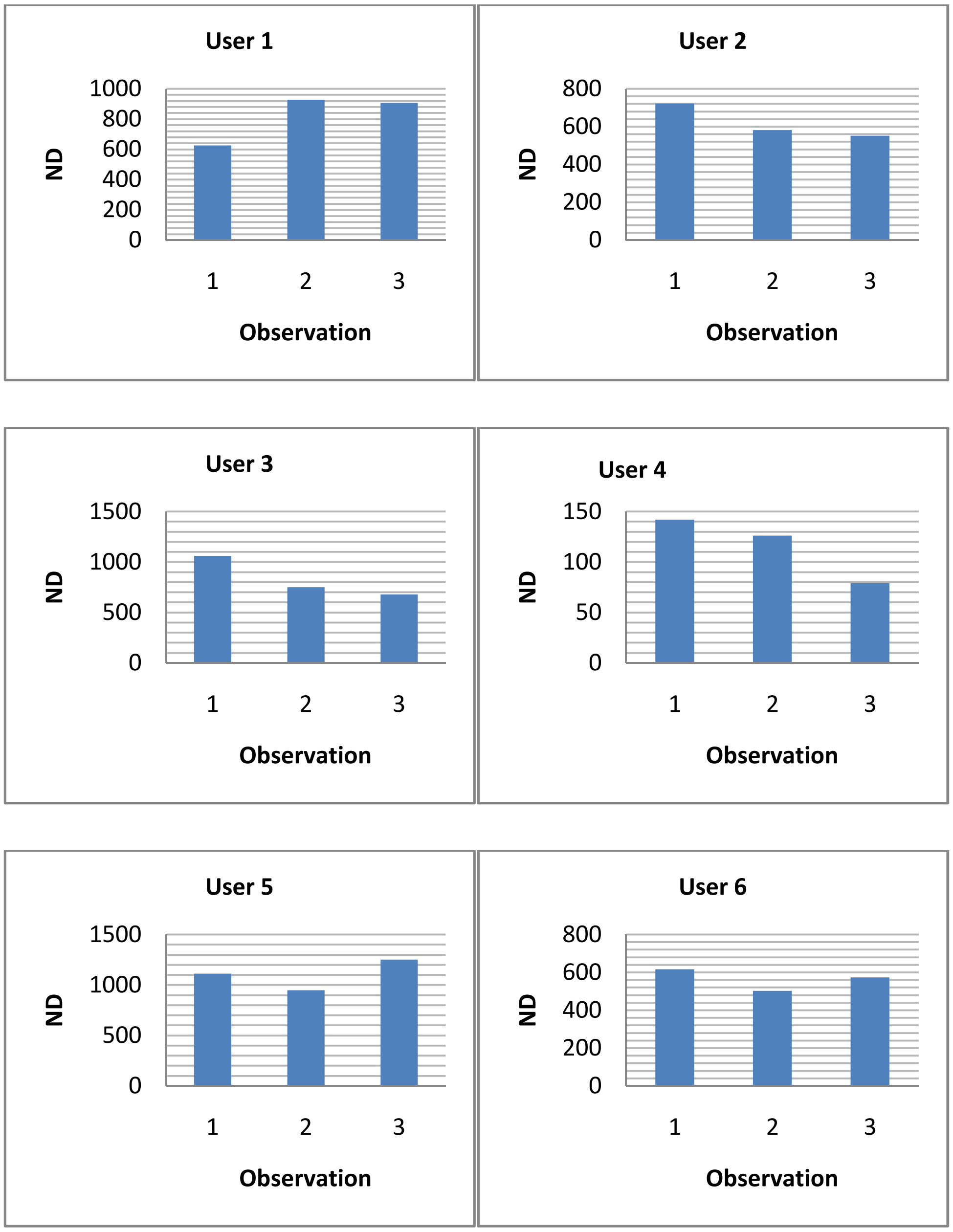


Figure 4.7: Number of Delay (ND) of Different user.

4. **Number of Action:** Determines the number of action.

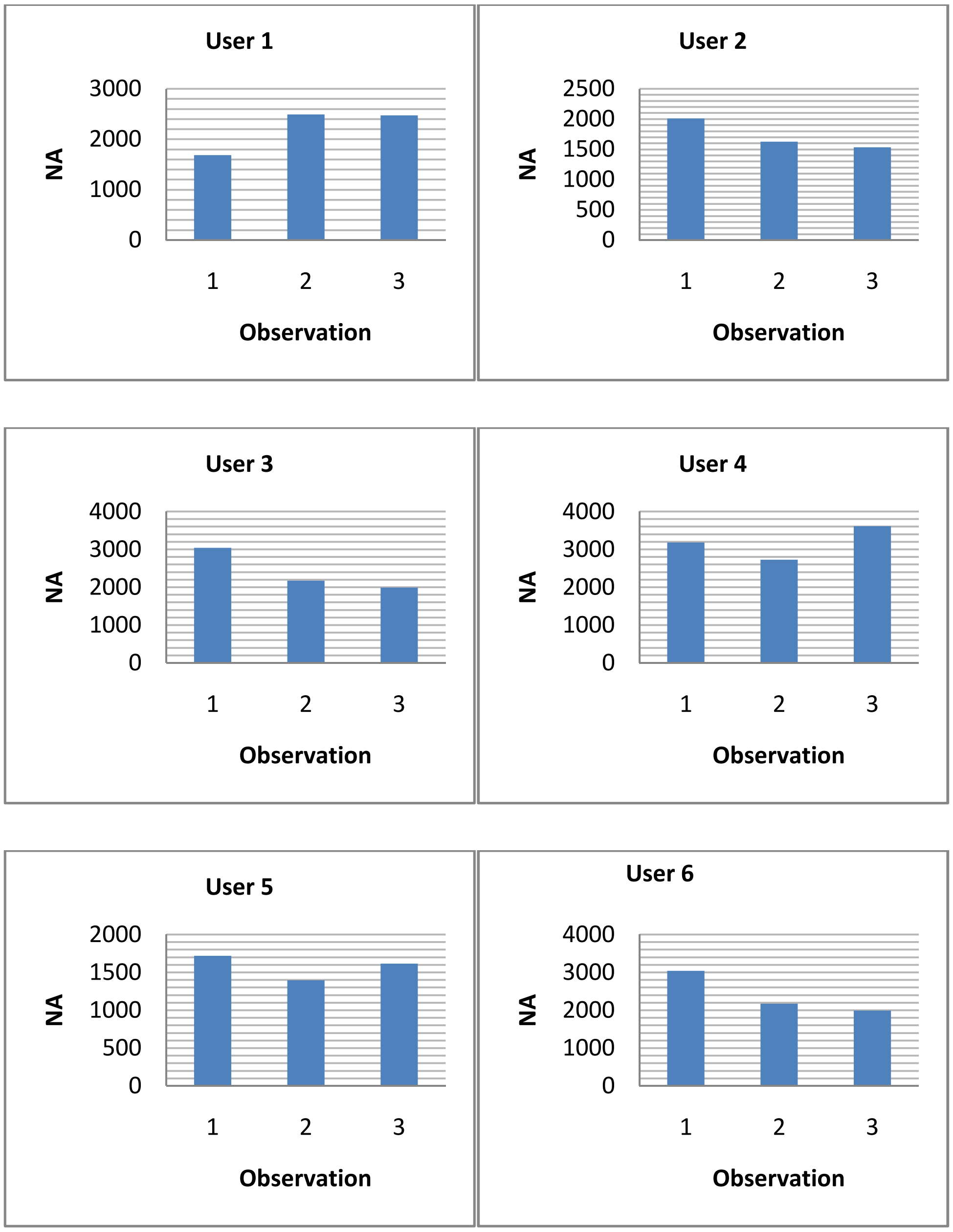


Figure 4.8: Number of Action (NA) of Different user.

5. **STDEV of the Trajectory Length:** First adding point to point length then standard deviation of these lengths.

$$\text{STDEV of } \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

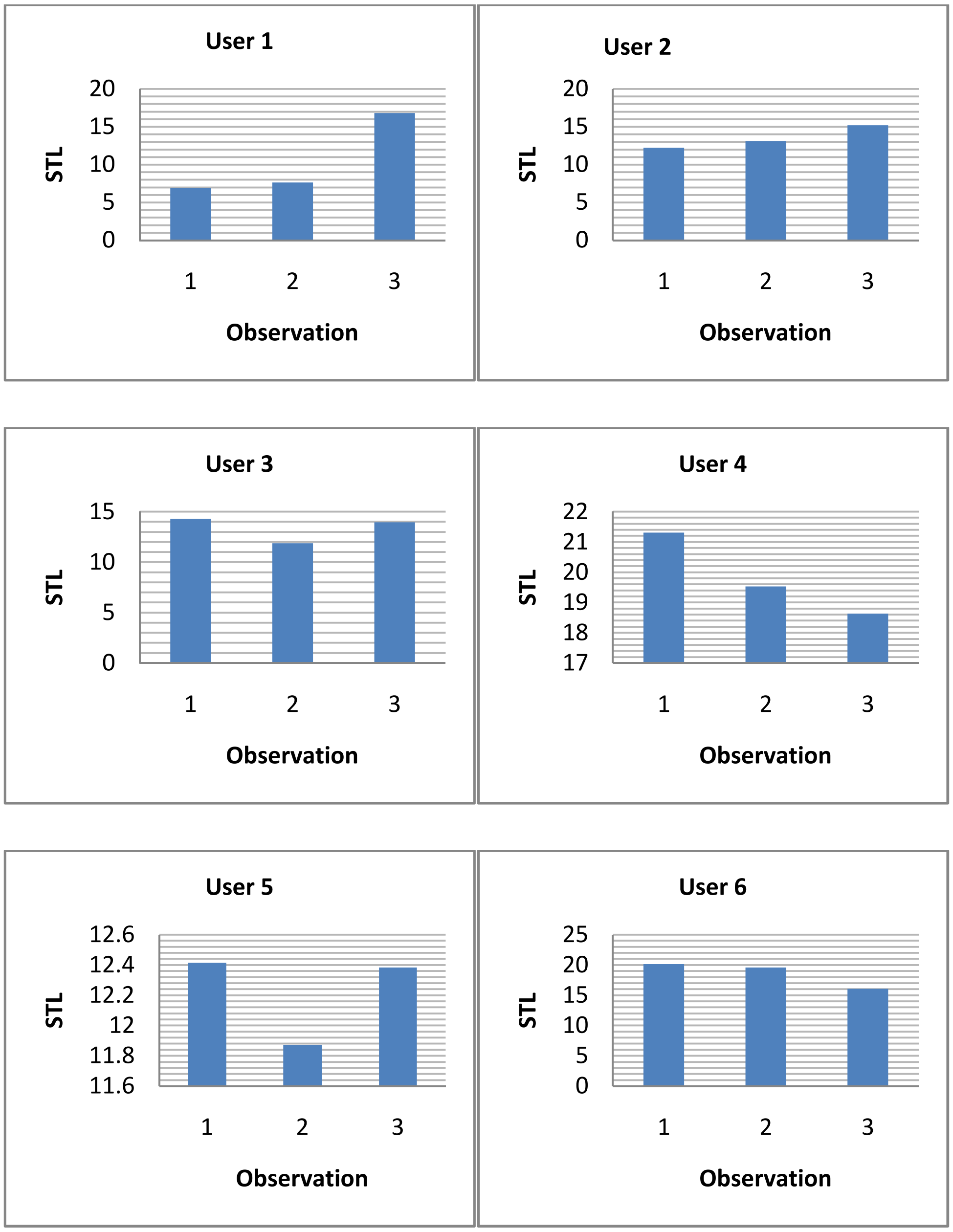


Figure 4.9: STDEV of Trajectory Length (STL) of Different user.

6. **Total Length of Trajectory:**Measures the point to point distance of the trajectory.

$$\sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

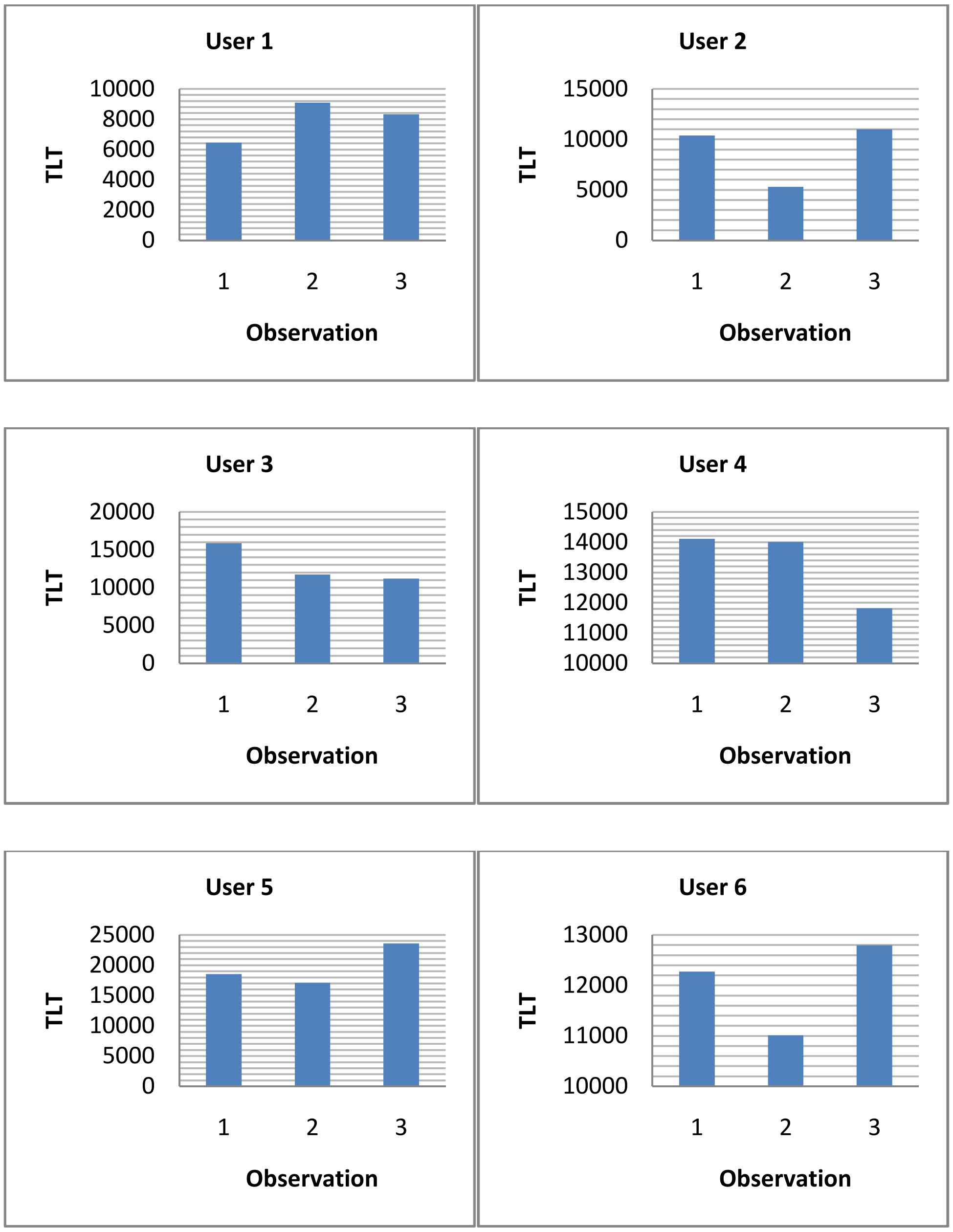


Figure 4.10:Total Length of Trajectory(TLT) of Different user.

7. **STDEV of Slope:** Determines the standard deviation of direction angle from one point to next point in the Trajectory.

$$\text{STDEVof } m_i = \frac{y_{i-}y_{i-1}}{x_i - x_{i-1}} \text{, where i=2 to n.}$$

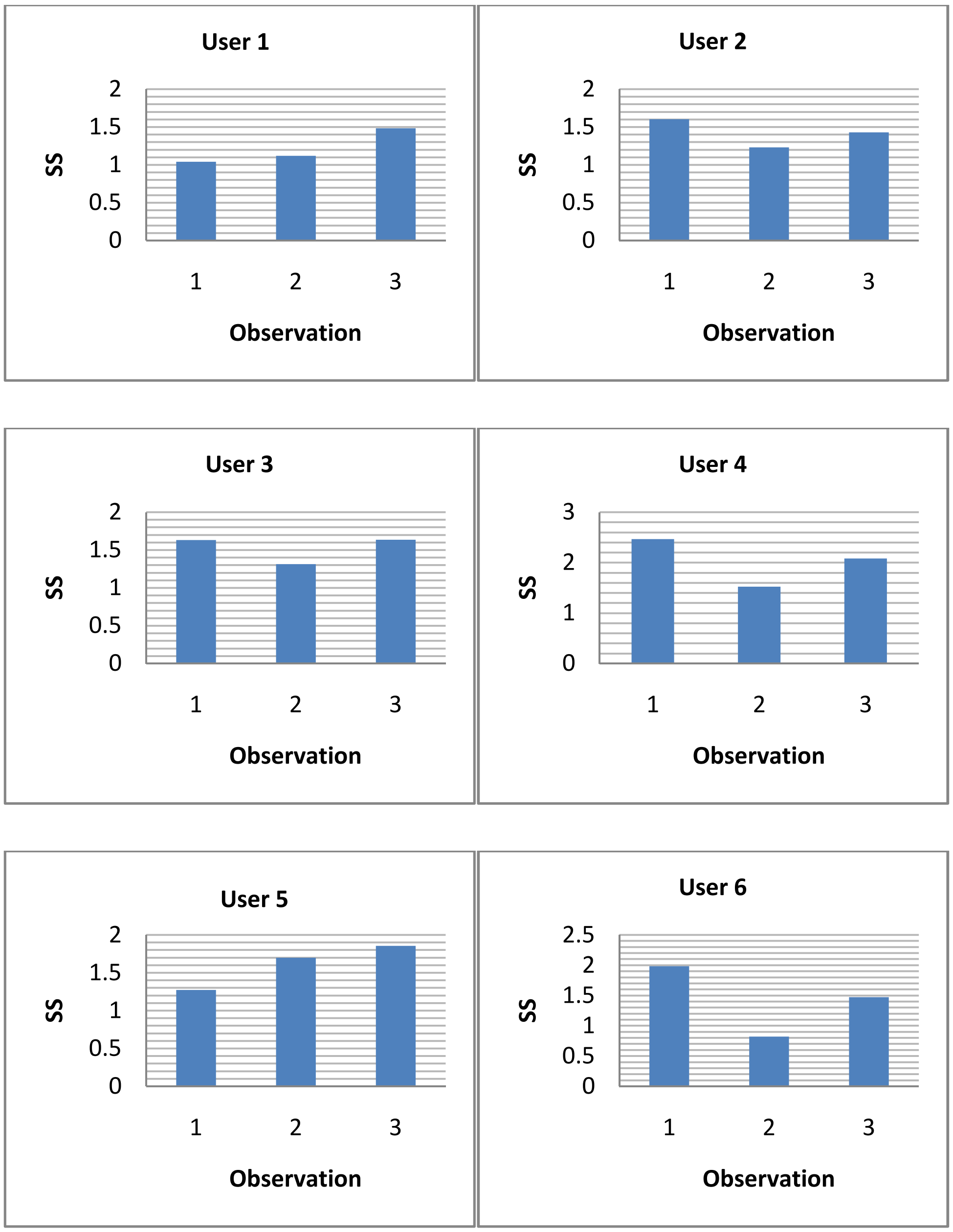


Figure 4.11:STDEV of Slope (SS) of Different user.

8. **STDEV of Difference betweenEach of Slopes:**Measure the difference between each consecutive slope and its standard deviation.

$$STDEVof\ (m_i - m_{i-1}) \text{, where i=2 to n.}$$

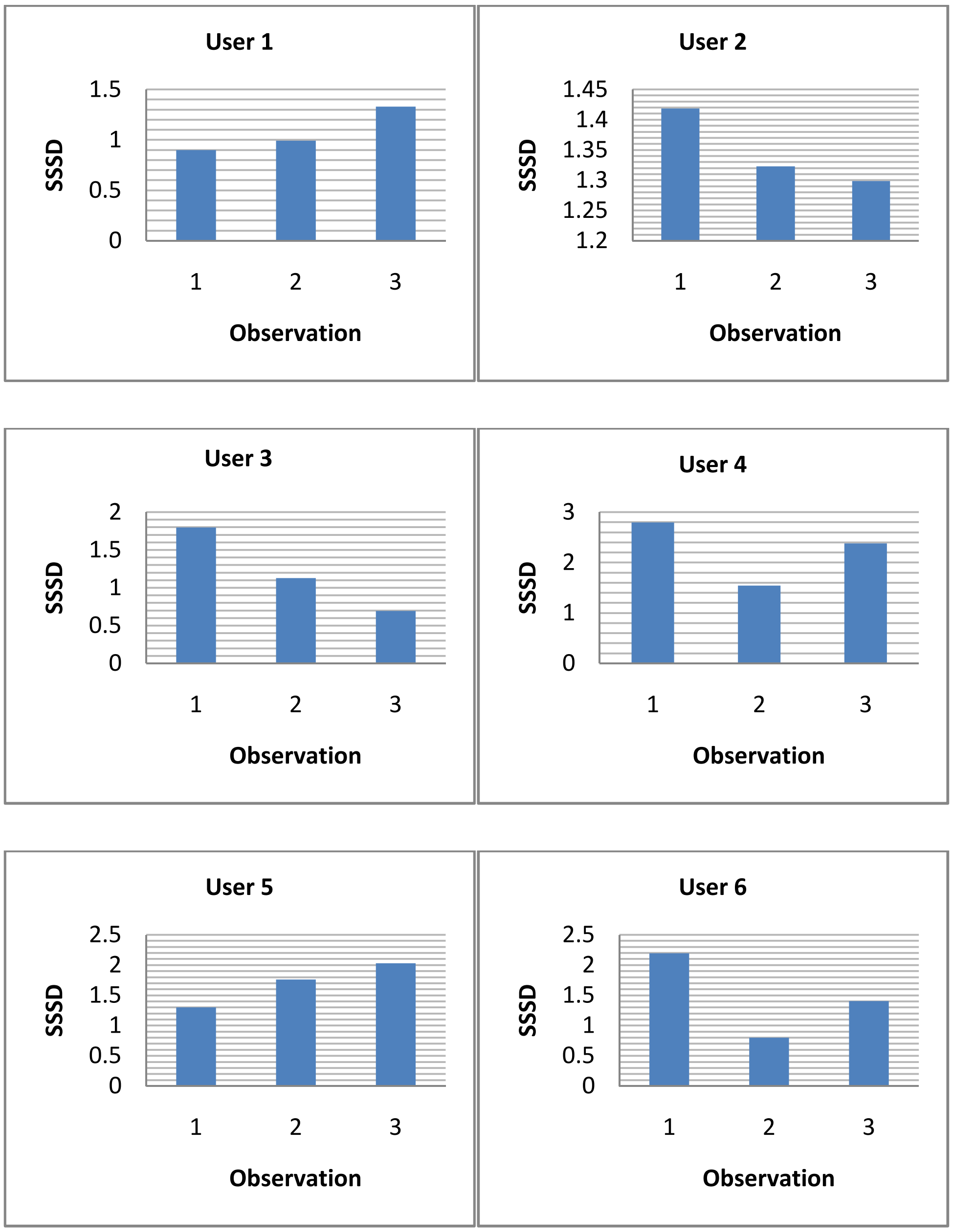


Figure 4.12:STDEV of Difference between each of Slopes (SDBES)of Different user.

9. **Number of Curvatures:**Measures the number of changes between the angles of the Trajectory.

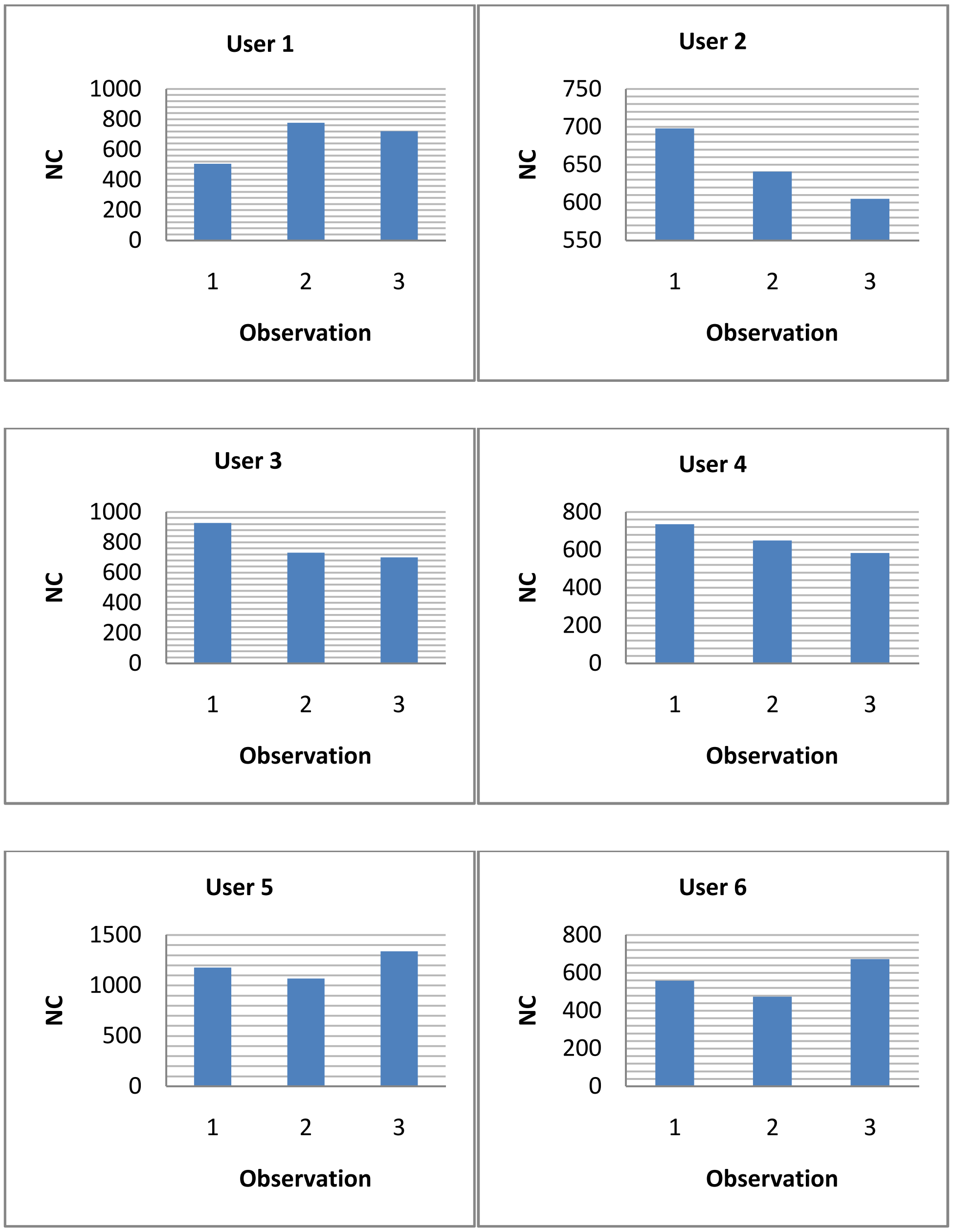


Figure 4.13:Number of Curvatures (NC) of Different user.

10. **Curvature of the Trajectory:**Measures the total length of the Trajectory (6) divided by the distance between the starting and ending position of Trajectory.

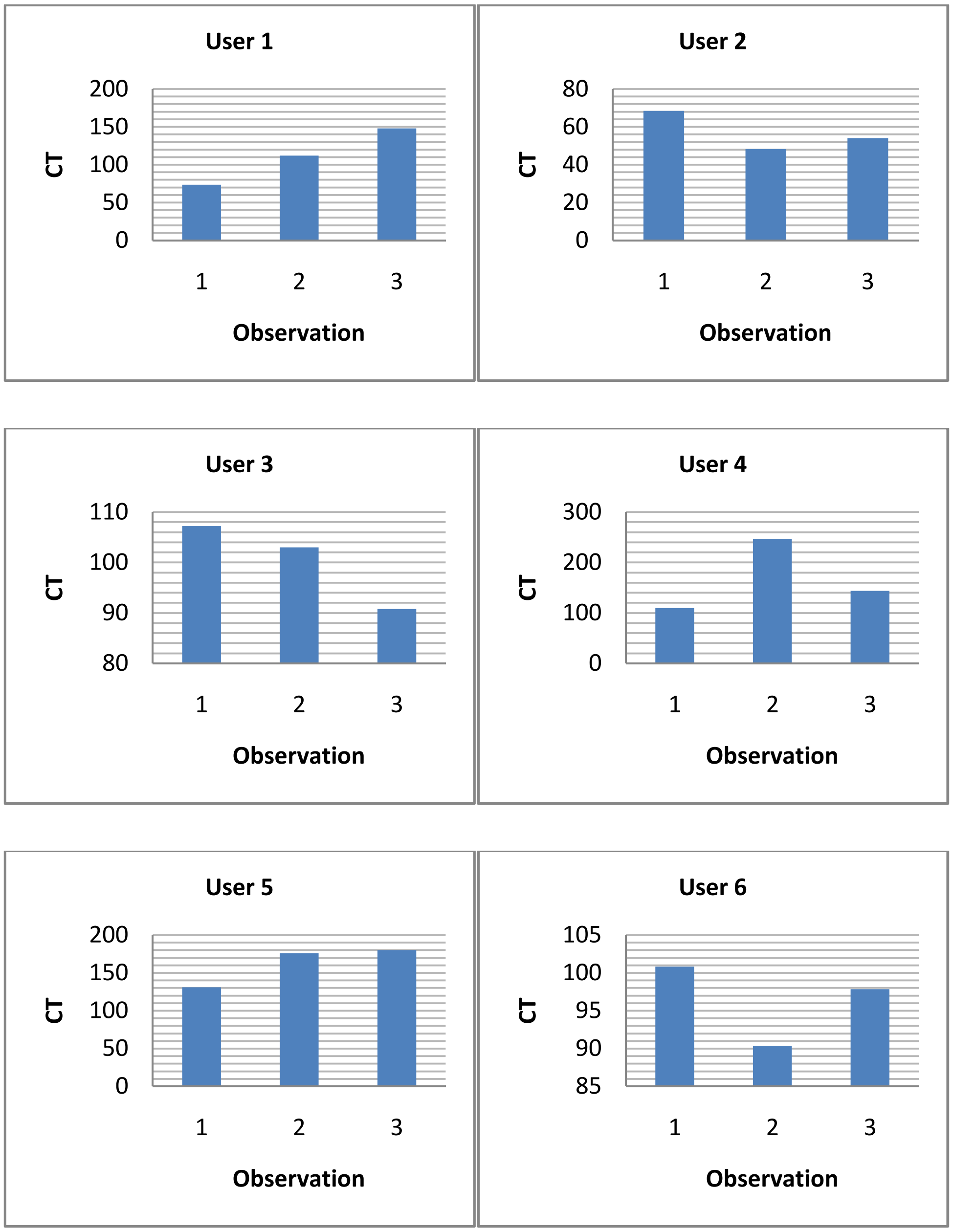


Figure 4.14:Curvature of the Trajectory(CT) of Different user.

11. **Number of Changes in Horizontal Position:** Measures how many times the Trajectory changes in X axis.

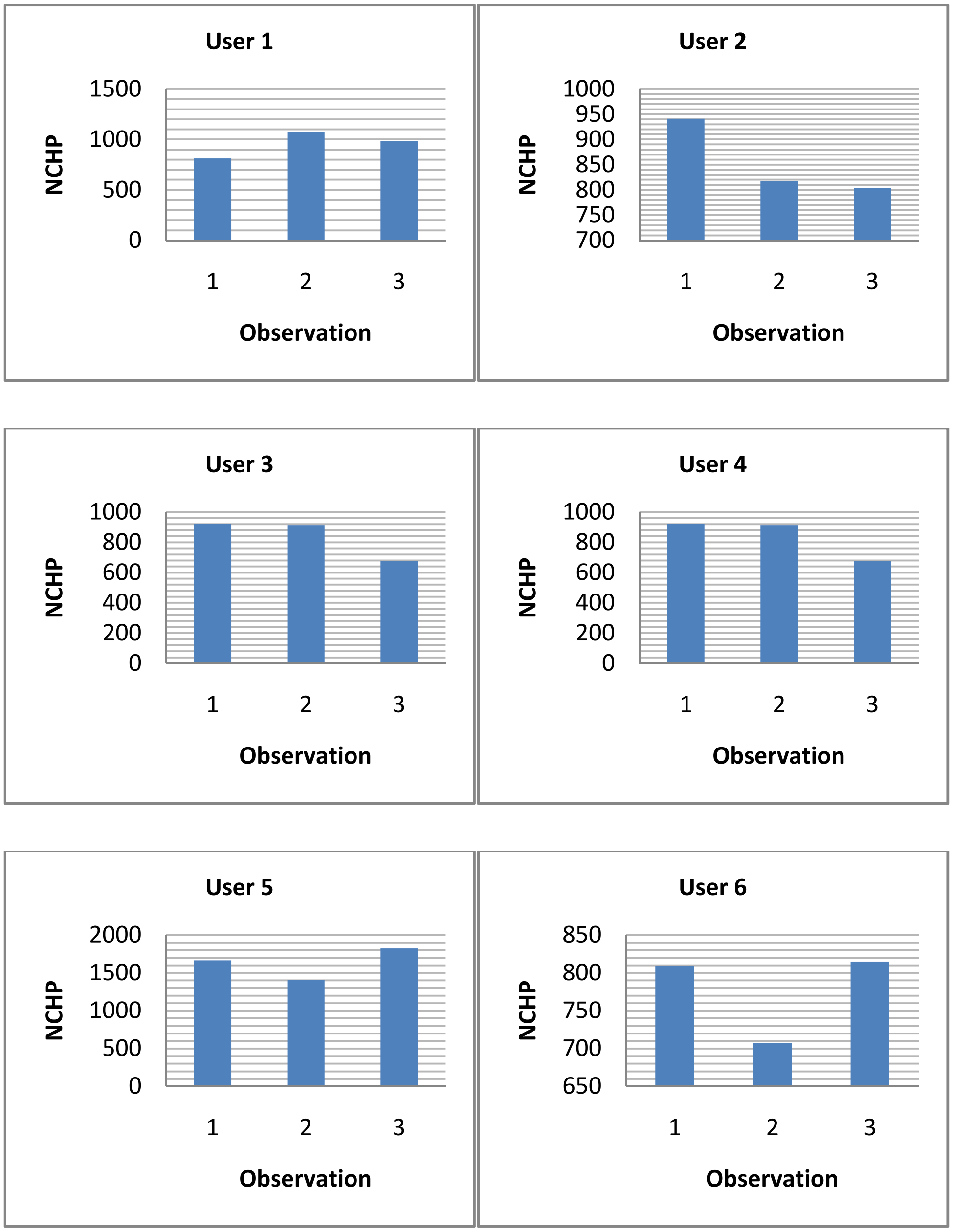


Figure 4.15:Number of Changes in Horizontal Position (NCHP) of Different user.

12. **Number of Changes in Vertical Position:**Measures how many times the Trajectory changes in Y axis.

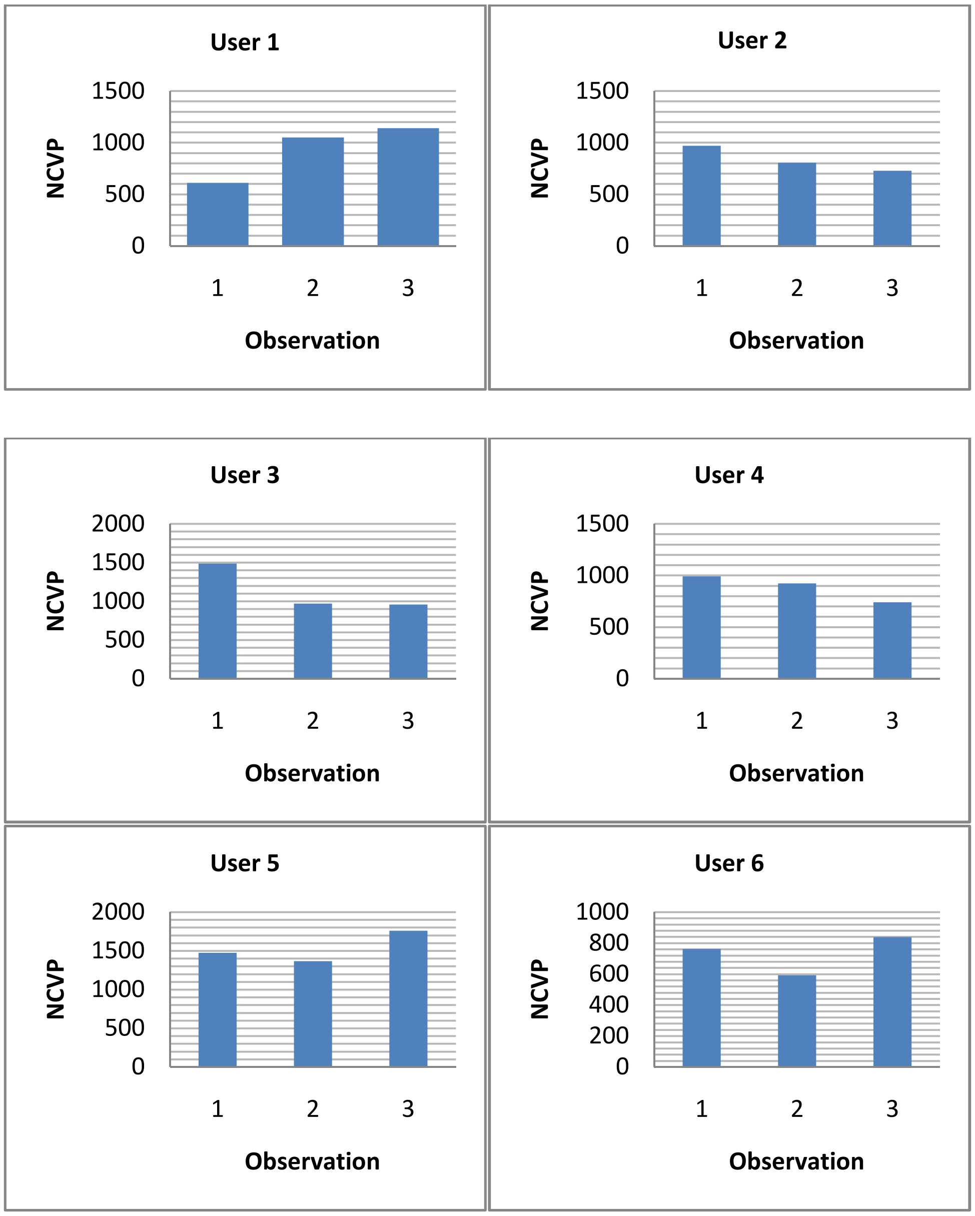


Figure 4.16:Number of Changes in Vertical Position(NCVP) of Different user.

After collecting data we have analyzed to find the feature vectors. The table 4.6 below shows 12features of a training data of six different users. Each user was trained three times. From the training mouse dynamic points and delay times are taken using the tools described in section 4.1.1. We have collected the x-y points and delay times. From these data we can calculate the 12 feature vectors for each user of every training.

Table 4.6: Training data from 6 users

SL	User	Test ID	(1) Number of Points in Trajectory	(2) Delay Time	(3) Number of Delay	(4) Number of Action	(5) STEV of Trajectory Length	(6) Total Length of Trajectory
1	User 1	1	1379	31714	766	2145	18.46143862	16519.3819
		2	1241	430449	703	1944	10.28796263	9794.616674
		3	553	115437	304	857	10.28796263	9794.616674
2	User 2	1	1287	24772	723	2010	12.24026058	10378.86046
		2	1044	16510	582	1626	13.09001092	5311.833343
		3	982	18039	552	1534	15.18734426	10989.47847
3	User 3	1	1983	35266	1059	3042	14.27190416	15876.83388
		2	1426	23112	750	2176	11.85478026	11749.21378
		3	1311	20657	679	1990	13.94533876	11211.91277
4	User 4	1	1267	16968	142	1409	21.30618986	14115.28344
		2	1264	13053	126	1390	19.52921617	14012.73231
		3	875	7424	79	954	18.62812618	11814.40901
5	User 5	1	2072	18431	1113	3185	12.41355638	18542.61129
		2	1783	15180	948	2731	11.87195961	17093.47783
		3	2357	20431	1253	3610	12.38310577	23614.79719
6	User 6	1	1103	16568	617	1720	20.1225844	12275.78982
		2	894	17132	503	1397	19.55526166	11013.08332
		3	1042	15324	573	1615	16.05468317	12794.424

Table 4.6: Training data from 6 users

SL	User	Test ID	(7) STDEV of Slope	(8) STDEV of Difference between Each Slope	(9) Number of Curvatures	(10) Curvature of Trajectory	(11) Number of Changes in Horizontal	(12) Number of Changes in Vertical
1	User 1	1	2.195551	1.803605	846	140.9854649	1095	1046
		2	2.718169	3.15087	672	85.82198949	881	887
		3	2.718169	3.15087	672	85.82198949	881	887
2	User 2	1	1.603324	1.418507	698	68.36191315	941	971
		2	1.2329	1.323176	641	48.25986931	817	807
		3	1.430432	1.298839	605	54.0547511	804	727
3	User 3	1	1.630612	1.797052	928	107.2318538	1279	1487
		2	1.315328	1.128043	731	103.0078109	1081	972
		3	1.637094	0.694643	701	90.79141575	941	957
4	User 4	1	2.465227	2.796621	735	109.8340925	921	991
		2	1.525168	1.541963	649	245.7995848	913	924
		3	2.080471	2.378732	584	143.5030791	676	740
5	User 5	1	1.273509	1.301606	1177	130.8937314	1665	1475
		2	1.700377	1.759042	1070	176.2120572	1407	1366
		3	1.855734	2.030171	1339	180.1239854	1821	1762
6	User 6	1	1.98373	2.19635	559	100.8212536	809	763
		2	0.820811	0.800653	475	90.3896748	707	594
		3	1.472566	1.403081	672	97.86143792	815	839

Now all the related data and result of features values are ready to us. For training we have total 216 parameters of 6 users, 3 times and 12 vectors. From these data of parameter of each user we will find the unique characteristics of each user based on mouse movement pattern. We will apply data mining classification technique and need to evaluate the performance. For this reason in next section we will briefly describe the performance evaluation in context of False Acceptance Rate (FAR) and False Rejection Rate (FRR). Thenwewill find out a process that performs best result to verify a user.

4.4 Performance Evaluation Criteria

To evaluate how accurate a biometric system works to measure its biometric performance, many genuine and impostor attempts are made with the system and all similarity scores are saved. By applying a varying score threshold to the similarity scores, we have calculated pairs of FRR and FAR. We have also observed the receiver operating characteristic ROC curve to evaluate performance of classifiers. In this section we have describe some of these criteria to compare classifications results.

FAR and FRR

The performance of a biometrical system is usually measured in terms of false accept rate (FAR),false reject rate (FRR) and equal error rate (EER) [32], [33], 36]. The false accept rate is the percentage of invalidinputs that are incorrectly accepted (match between input and a non-matching template). The falsereject rate is the percentage of valid inputs that are incorrectly rejected (fails to detect a matchbetween input and matching template).The equal error rate (ERR) indicates the accuracy of the system (Figure 4.19). The false accept rate and false rejectrate intersect at a certain point which is called the equal error rate (the point in which the FAR andFRR have the same value) [34].

Results are presented either as such pairs of FRR at a certain level of FAR, or in plots (describe below). Rates can be expressed in many ways like in percent (1%), as fractions (1/100), in decimal format (0,01) or by using powers of ten (10-2). When comparing two systems, the more accurate one would show lower FRR at the same level of FAR. We have rate the value in percent. In section 4.5 we have find the best process to calculate features vector in different way. The performances of the processes are evaluated by considering lower FAR and FRR.

Some systems don't report a similarity score, only the match/non-match decision. In that case it is only possible to gain a single FRR/FAR pair (and not a continuous series) as result of a performance evaluation. If the mode of calculation is adjustable that we have a means of controlling the internally used score threshold, the performance evaluation can be run again and again in different modes to obtain further FRR/FAR pairs. Evaluation is measured based on threshold that we have fixed.

Identification and verification

A biometric recognition system can run in two different modes: identification or verification [37]. Identification is the process of trying to find out a person's identity by examining a biometric pattern calculated from the person's biometric features.

In the identification case, the system is trained with the patterns of several persons. For each of the persons, a biometric template is calculated in this training stage. A pattern that is going to be identified is matched against every known template, yielding either a score or a distance describing the similarity between the pattern and the template. The system assigns the pattern to the person with the most similar biometric template. To prevent impostor patterns (in this case all patterns of persons not known by the system) from being correctly identified, the similarity has to exceed a certain level. If this level is not reached, the pattern is rejected. By this concept we have complete some processes in section 4.5 and have choose best one after evaluation.

In the verification case, a person's identity is claimed a priori. The pattern that is verified only is compared with the person's individual template. Similar to identification, it is checked whether the similarity between pattern and template is sufficient to provide access to the secured system or area.

Threshold

We have used scores to express the similarity between a pattern and a biometric template. In some cases we have mentioned this score as *weight* like in describing *process* in section 4.5. The higher the score is, the higher is the similarity between them. As described in the preceding section, access to the system is granted only, if the score for a trained person (identification) or the person that the pattern is verified against (verification) is higher than a certain threshold.

In theory, user scores of patterns from persons known by the system should always be higher than the scores of impostors. If this would be true, a single threshold, that separates the two groups of scores, could be used to differ between users and impostors.

Due to several reasons, this assumption isn't true for real world biometric systems. In some cases impostor patterns generate scores that are higher than the scores of some user patterns. For that reason it is a fact, that however the classification threshold is chosen, some classification errors occur.

To set our threshold value we can choose the threshold such high, that really no impostor scores will exceed this limit. As a result, no patterns are falsely accepted by the system. On the other hand the user patterns with scores lower than the highest impostor scores are falsely rejected. In opposition to this, we can choose the threshold such low, that no user patterns are falsely rejected. Then, on the other hand, some impostor patterns are falsely accepted. If we choose the threshold somewhere between those two points, both false recognitions and false acceptances occur.

Researchers in biometric authentication sector are using standard figure (figure 4.17-4.19) to identify best value of threshold. We have used these figures to achieve a suitable value for threshold. Our proposed method of biometric verification system is tested with a large amount of test data. The test data consists of both impostor and valid user's patterns. If we think for the impostor patterns, the belonging threshold scores would be somehow distributed around a certain mean score. This is depicted in figure 4.17 on the left side. A Gaussian normal distribution is chosen in this example.

Depending on the choice of the classification threshold, between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR). Its value is one, if all impostor patterns are falsely accepted and zero, if none of the impostor patterns is accepted. We see in figure 4.17 on the right, the values of the FAR for the score distribution of the left image for varying threshold.

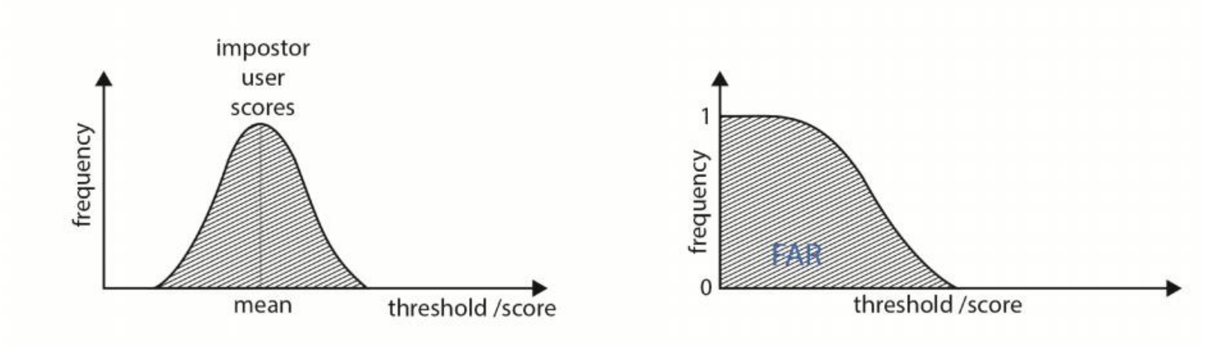


Figure 4.17: False Acceptance Rate

Now let's change to the user patterns. Similar to the impostor scores, the user pattern's scores vary around a certain mean value. The mean score of the user patterns is higher than the mean value of the impostor patterns, as shown in the left of the following two images. If a classification threshold that is too high is applied to the classification scores, some of the user patterns are falsely rejected. Depending on the value of the threshold, between none and all of the user patterns will be falsely rejected. The fraction of the number of rejected user patterns divided by the total number of user patterns is called false recognition Rate (FRR). According to the FAR, its value lies in between zero and one. The figure 4.17shows the FAR for a varying threshold for the score distribution shown on the left of figure 4.17.

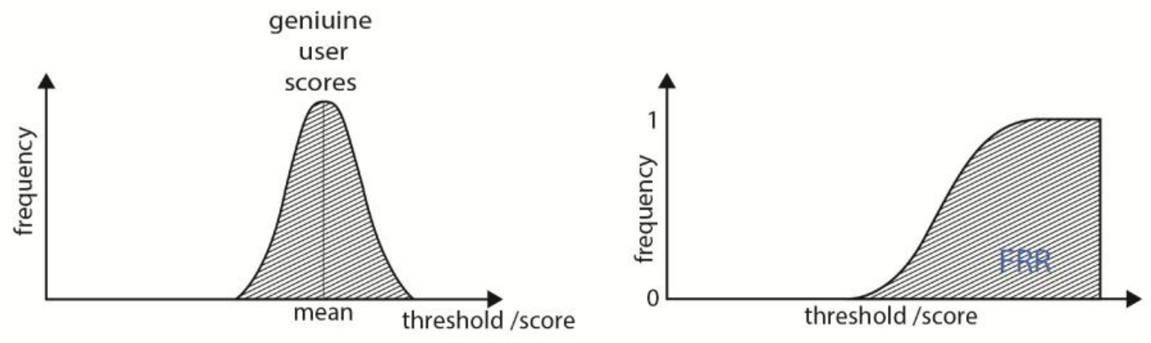


Figure 4.18: False Rejection Rate

The choice of the threshold value becomes a problem if the distributions of the user and the impostor scores overlap, as shown in the figure 4.19. In this figure the graph of corresponding false acceptance and false recognition rates are shown.

It is mentioned that in this research we have used WEKA data mining tool. We have generated ROC curve and threshold by WEKA. WEKA creates such curve by *false positive rate* in X axis and *true positive rate* in Y axis. Figures from 4.50 to 4.73 shows such curve.

ROC Curve

The false acceptance rate, or FAR, is the measure of the likelihood that the biometric security system will incorrectly accept an access attempt by an unauthorized or impost user. A system’s FAR typically is stated as the ratio of the number of false acceptances divided by the number of identification attempts.

The false rejection rate, or FRR, is the measure of the likelihood that the biometric security system will incorrectly reject an access attempt by an authorized or genuine user. A system's FRR typically is stated as the ratio of the number of false rejection divided by the number of identification attempts.

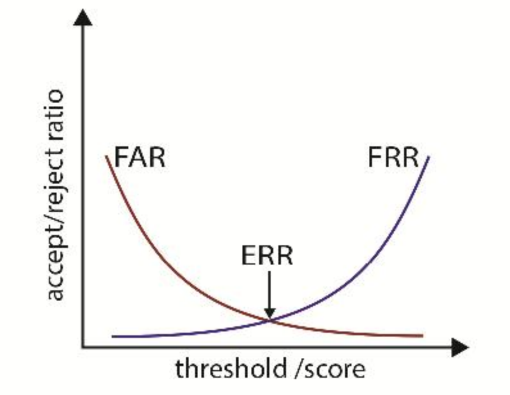


Figure 4.19: ROC Curve

In our experiment we have used WEKA and observe average ROC area by plotting*True Positives* on Y-axis and *True Negatives* on X-axis. As the area under an ROC curve is a measure of the usefulness of a test in general, where a greater area means a more useful test, the areas under ROC curves are used to compare the usefulness of tests (table 4.20, section 4.8).

Note that if the score distributions overlap, the FAR and FRR intersect at a certain point. The value of the FAR and the FRR at this point, which is of course the same for both of them, is called the Equal Error Rate (EER) [34].In theory, the correct users should always score higher than the impostors. A single threshold could then be used to separate the correct user from the impostors. In general, the matching algorithm performs a decision based on a threshold which determines how close to a template the input needs to be for it to be considered a match. If the threshold is reduced, there will be less false non-matches but more false accepts. Correspondingly, a higher threshold will reduce the false accept rating but increase the false reject rating [2] [14] [25].

Comparing biometric systems

Imagine the comparison of two biometric systems. The manufacturers of the systems just specify a single value for the FARs of them. But this is not sufficient if the manufacturers do not provide the corresponding FRRs. In this case, it is possible that the system with the lower FAR has got an unacceptable high FRR.

But also when the values for FAR and FRR are given, there still exists the problem, that those values are threshold depending. Assuming that the threshold of the systems is adjustable, there is no reasonable way to decide if a system with a higher FAR and a lower FRR performs better than a system with a lower FAR and a higher FRR value.

These days there is a lot of emphasize on FAR (False Acceptance Rate). It is good that in market users are getting aware of these terms and asking the right questions before implementing a solution.However, FAR only provides half the information. When selecting a biometric solution, we need to find out what the false rejection Rate (FRR) is at the said FAR.

So when a biometric solution provider claims to have a very low FAR, it is very important to find out what is the FRR at this 'low' FAR. Then depending upon the application one needs to evaluate whether the FAR & FRR ratio is acceptable for the application.

In a practical scenario a low FAR & a high FRR would ensure that any unauthorized person will not be allowed access. It would also mean that the authorized people will have to repeat log in session several times before they are allowed access.Therefore, it is good to have a very low FAR as well as very low FRR.

4.5 Classification Process

We have used three classifiersto evaluate the proposed method. Three classifier, SVM, kNN and Naïve Bayes are used in this research. According to proposed method data are captured and then processed as table 4.6. We will pass these data through the above multi classifiers. Before that we need to find the pattern of user mouse movement. We find the pattern by calculating feature vectors of user's mouse movement data from table 4.6. We have tested many processes of calculations to bring most suitable mouse movement pattern in different calculations and by using different algorithms. By comparing different process of calculation we havechoose the best process. Using the best process we haveset signature of each user to identify user.

4.5.1 Finding the Best Processto Authenticate User

To find out a unique signature we have done different experiment. The raw data and features are analyzed in different thinking, measurement, calculation and ways [14]-[19]. After training and testing some processesof calculating signature are founded good to user authentication and some are very poor. Comparing among them four processesaredescribed in the following sections. From these four process we have select the best one.

Process-1: Root of Average Then Square of Difference is Below ofThreshold

First we have calculated the root of average (*ar*) of all vectors then calculated the square of the difference of the ar vectors between each consecutiveuser. We chose the smallest of difference that existed between any two registered users. It gave us the lowest difference of vectors between any two users i.e., the lowest difference by which two users could differ in that particular set of users. This lowest difference can be taken as threshold value (ThV). A simple algorithm of this calculation and values are shown in below.

```
u1=$\sqrt{average(f1, f2, \ldots..f12)}$    where $f$ presents the features
u2=$\sqrt{average(f1, f2, \ldots..f12)}$
Difference=$(u1-u2)^2$
IF (the Difference<ThV) THEN
        Valid user and Authenticate
ELSE
        Invalid User
END IF
```

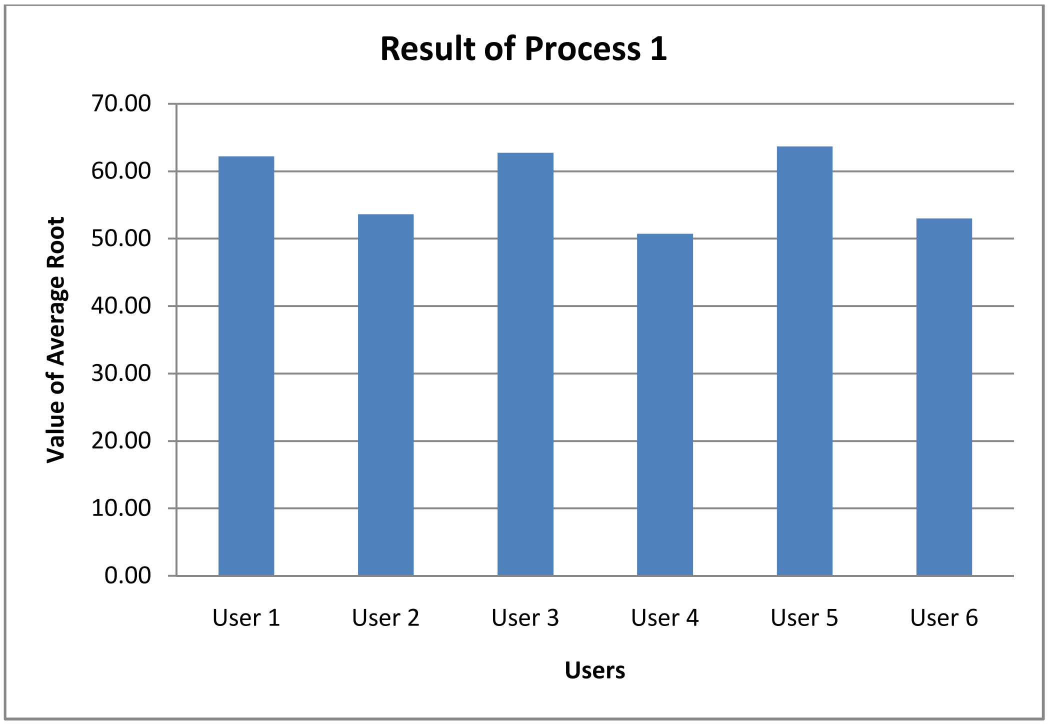


Figure 4.20:Average root histogram of different user.

Table 4.7: Square Difference of average root of 6 users

SL	User	Avg Root	SQR Difference					
			User 1	User 2	User 3	User 4	User 5	User 6
1	User 1	62.20		73.37	0.29	131.94	2.17	84.53
2	User 2	53.64	73.37		82.89	8.53	100.74	0.40
3	User 3	62.74	0.29	82.89		144.61	0.87	94.73
4	User 4	50.72	131.94	8.53	144.61		167.91	5.25
5	User 5	63.67	2.17	100.74	0.87	167.91		113.76
6	User 6	53.01	84.53	0.40	94.73	5.25	113.76	

Though the lowest value is 0.4 so the threshold value is 0.4.

When any user tried to login, then square difference of the *ar* between that user and all registered users are calculated. For any registered user if the differences was less than the threshold, and that user was the one who tried to login, we would conclude that the user was verified else he would be rejected.

Table 4.8: Confusion Matrix of Process-1

	User 1	User 2	User 3	User 4	User 5	User 6
User 1	1	2			1	
User 2			1			
User 3						1
User 4						
User 5			2			
User 6						

After a testing FAR and FRR are found shown as the below.

FAR	19.44
FRR	77.78

This Processwill not work as the target. Though the logic seemed to be correct but not enough data points were below the threshold for user to make any conclusion but most of the time it gave the lowest difference with the actual user. So using this process there is high risk of accessing imposter user and deny actual user.

Process-2: Calculating the lowest square difference of average-root among all users

The results from *Process-1* showed that threshold is not significance. So we decided to skip threshold and just check the difference of average-root and find with which user it give the lowest difference. So we just computed the differences of average-root of the current user with all other registered users. If it showed the lowest number with himself, he would be authenticated. We can see the process by a simple algorithm as the below.

u1=$\sqrt{average(f1, f2, \ldots..f12)}$ where *f* presents the features

u2=$\sqrt{average(f1, f2, \ldots..f12)}$

Difference=$(u1-u2)^2$

```
IF (the Difference=lowest) THEN
        Valid user and Authenticate
ELSE
        Invalid User
END IF
```

Table 4.9: Confusion Matrix of Process 2

	User 1	User 2	User 3	User 4	User 5	User 6
User 1		2		2	2	
User 2			6			
User 3			6			
User 4			6			
User 5			6			
User 6			5	1		

After a testing FAR and FRR are found shown as the below.

FAR	83.33
FRR	0

This process showed some unexpected results. It had a false acceptance rate (FAR) of about 83% and false rejection rate (FRR) of 0%. The results gave some encouragement to proceed with the strategy of checking the lowest of difference, but we still need more improvement due to its high possibility of deny actual user.

Process-3: Using the Threshold in a Rang of the Average-Root

To take the FAR and FRR in a suitable positionwe have decided to select threshold value within a range. First we have calculated the root of average of each parameter. Then create a lower range which is 10% less than the root of average and an upper range which is 10% more than the root of average. We can mention the 10% as acceptance of weight for error. Though it is very critical to find the exact value for a certain threshold, weare trying with several weights in several times to find the bestprocess.

In Inscription Phase

$U_x=\sqrt{average(f1, f2, \ldots.. f12)}$ where *f* presents the features of User X.

$UV_x=U_x+U_x*10\%$ where UV_x presents Upper Value of User X.

$LV_x=U_x-U_x*10\%$ where LV_x presents Lower Value of User X.

When a user tried to login then average root of all 12 parameters are calculated. The average root is compared considering the 10% error rate with all stored average root those are registered users. If the average root is in range for a registered user then this is the valid user.

The upper level and lower level of six users are as the below:

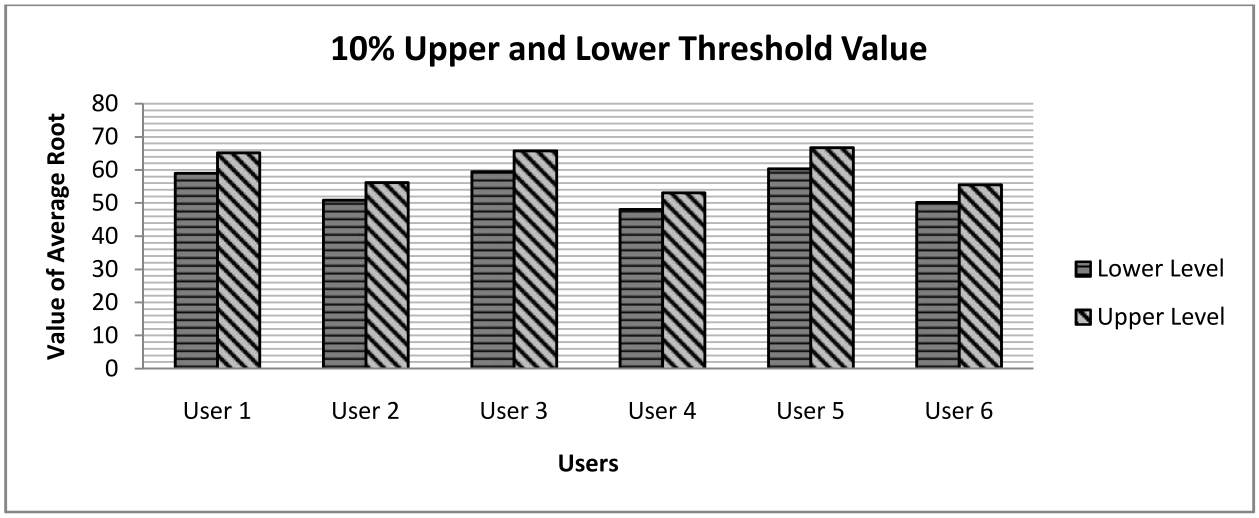


Figure 4.21: Histogram of threshold in a rangeof average-root

In verification phase

```
$U_{xv}=\sqrt{average(f1, f2, \ldots\ldots f12)}$   where f presents the features of User X.
IF ($U_{xv}$<= $UV_x$&&$U_{av}$>= $LV_x$) THEN
    X is valid user and Authenticate
ELSE
    X is invalid user
END IF
```

For these six users we have taken different threshold value other than 10 but similar results are found. Based on the result after testing we have found number of identified used as the below table.

Table 4.10: Confusion Matrix of Process 3

	User 1	User 2	User 3	User 4	User 5	User 6
User 1	2	2	0	2	2	2
User 2	0	0	2	0	0	1
User 3	0	1	1	0	0	1
User 4	0	0	1	0	0	0
User 5	0	0	2	0	0	1
User 6	0	1	2	0	0	1

After testing FAR and FRR are found shown as the below.

FAR	55.56
FRR	33.33

This Process shows a more satisfactory result than the above Process1 and 2. But we can't take this process due to high of FAR and FRR. There are many possibilities in this process to accept fraud user and also more possibilities to deny actual user.

This Process has also a drawback. Sometimes it may satisfy requirements of more than one user.

Process-4:Using the Threshold within a Range of all Feature Vector

Though results from processes above are not showing satisfactory FAR and FRR so we have planned to identify a user depending on characteristics of all features vectors separately. Here we have observed the behaviors of all feature vectors for every user. This process works as the below algorithm.

In Inscription Phase

Finding each feature's value of every user

$U_r f_1$=*average* ($f_{1.1}, f_{1.2}, \ldots\ldots., f_{1.n}$)where f_1 presents the 1st feature of User.

$U_r f_2$=*average* ($f_{1.1}, f_{1.2}, \ldots\ldots., f_{1.n}$) where f_2 presents the 2nd feature of User.

....

....

$U_r f_{12}$=*average* ($f_{1.1}, f_{1.2}, \ldots\ldots., f_{1.n}$) where f_{12} presents the 12th feature of User.

Finding Rang ofeach feature's value of every user

$UVU_rf_1= U_rf_1+ U_rf_1*10\%$ where UVU_rf_1 presents Upper Value of User of 1st feature.

$LVU_rf_1= U_rf_1- U_rf_1*10\%$ where LVU_rf_1 presents Lower Value of User of 1stfeature.

$UVU_rf_2= U_rf_2+ U_rf_2*10\%$ where UVU_rf_2 presents Upper Value of User of 2ndfeature.

$LVU_rf_2= U_rf_2- U_rf_2*10\%$ where LVU_rf_2 presents Lower Value of User of 2nd feature.

....

....

$UVU_rf_{12}= U_rf_{12}+ U_rf_{12}*10\%$ where UVU_rf_{12} presents Upper Value of User of 12thfeature.

$LVU_rf_{12}= U_rf_{12}- U_rf_{12}*10\%$ where LVU_rf_{12} presents Lower Value of User of 12thfeature.

In verification phase

```
NoSF= 0                          //NoSF= Number of satisfied feature
LOOP UNTIL availableregister user
    LOOP UNTIL n=12
    IF ($FV_n$<=$UVU_rf_n$&&$FV_n$>=$LVU_rf_x$) THEN
        NoSF= NoSF+1
    ELSE
        NoSF= NoSF
    NEXT LOOP
NEXT LOOP
IF (NoSF= MAX for User X)
    X is valid user and Authenticate
ELSE
    X is invalid user
 END IF
```

In inscription phase mean value of each feature vector is calculated for several times of training sessions depending on blocks. From these mean values of features a range is defined for each feature of each user. In verification phase values of each feature vector is compared with the corresponding feature range value and difference of various percentage of range value is allowed. We are calling this percentage of range value as weight. We are taking this weight as threshold value (ThV).

In authentication phase, for a valid user this Process may satisfy features of others user but the number of satisfied features (NoSF) of valid user will be maximum. We are calling this number of invalid user's features those are satisfied as "Duplicate Identification" (DI). For all the 12 features of each user, we have calculated range of features by using weight value from 1 to 20 and similar results are found from this range of values (Table 4.12). In section below we have briefly describe how we can find a best threshold value.

Table 4.11: Confusion matrix for Process -4

	User 1	User 2	User 3	User 4	User 5	User 6
User 1	6	0	0	0	0	0
User 2	0	5	0	0	0	1
User 3	0	0	6	0	0	0
User 4	0	0	0	6	0	0
User 5	0	0	0	0	6	0
User 6	0	0	0	0	0	6

After a testing FAR and FRR are found shown as the below.

FAR	2.78
FRR	0

This Process shows a satisfactory result. We have tested this process by using several threshold values and we have got similar result of FAR and FRR. Though for several threshold value FAR and FRR are similar so measuring the number of duplication is also a key performance indicator of this process (Process-4). The less number of duplicate identification (DI) performs the more accurate the Process. As the table 4.12 below, we got the minimum number of duplicate identificationfor threshold value of 15, 17 and 18.

Finding Threshold Value

Though for a range of threshold value the results are similar so we can find best threshold value by measuring the number of duplicate identification. According to the Table4.12 we can set threshold value 18 for the users tested in this study.

Table 4.12: Threshold value against the number of duplicate identification

Threshold Value	FAR	FRR	Number Duplicate Identification
1	2.78	2.78	82
2	2.78	2.78	18
3	2.78	2.78	46
4	2.78	2.78	39
5	2.78	2.78	23
6	2.78	2.78	23
7	2.78	2.78	23
8	2.78	2.78	17
9	2.78	2.78	23
10	2.78	2.78	28
11	2.78	2.78	24
12	2.78	2.78	16
13	2.78	2.78	15
14	2.78	2.78	13
15	2.78	2.78	8
16	2.78	2.78	14
17	2.78	2.78	9
18	2.78	2.78	7
19	2.78	2.78	10
20	2.78	2.78	10

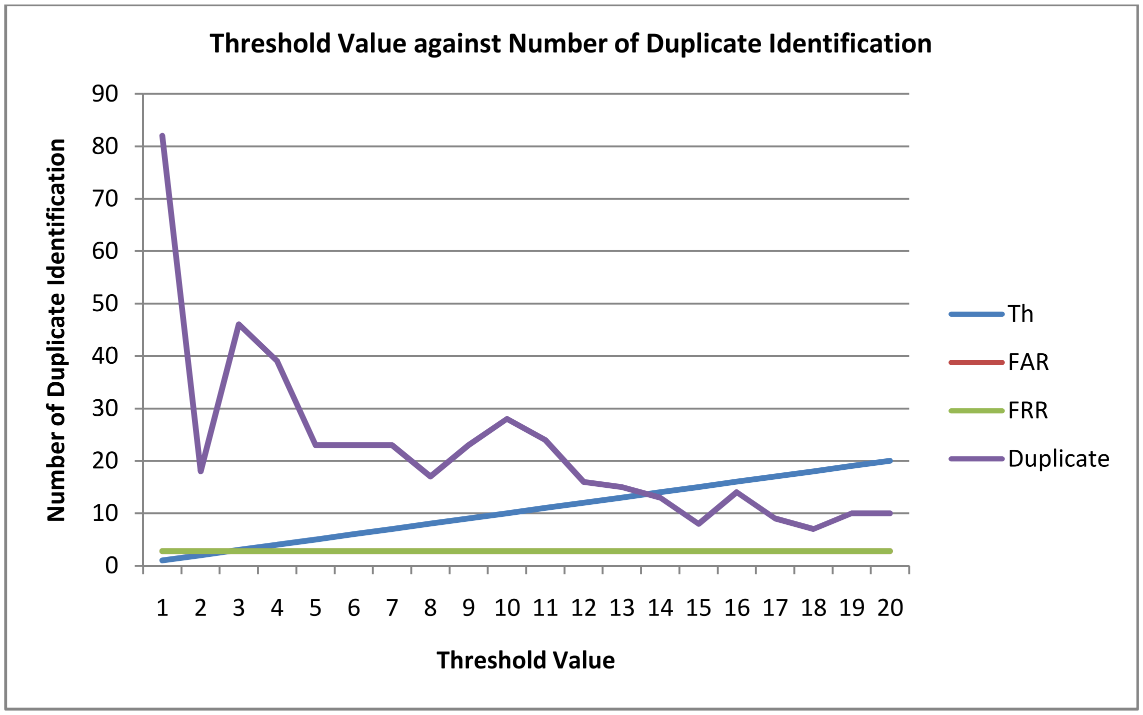


Figure 4.22:Comparing TH value with number of duplicate identified user.

During this research we have done many calculations to find best process. Among these processes we have discussed four processes in above section. These processes shows better result and process 4 is the best. As we have describe the processes and output result in figure below (Figure 4.23) we have choose process 4 for next of work. One of major work is to apply three classifiers.

In section below we have taken more mouse movement data from more users. From these data features are generated and classifiers algorithms are used to test the validity of the proposed system. We have predicted user based on mouse movementpatterns (features) by applying mentioned classifier algorithms. We have also compared performances of the classifiers.

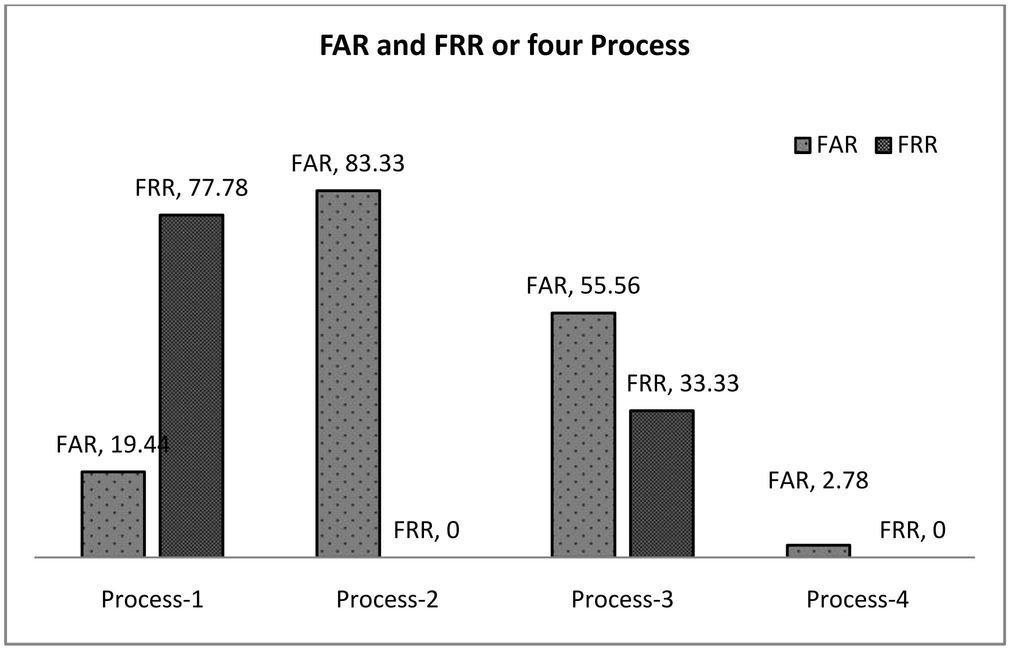
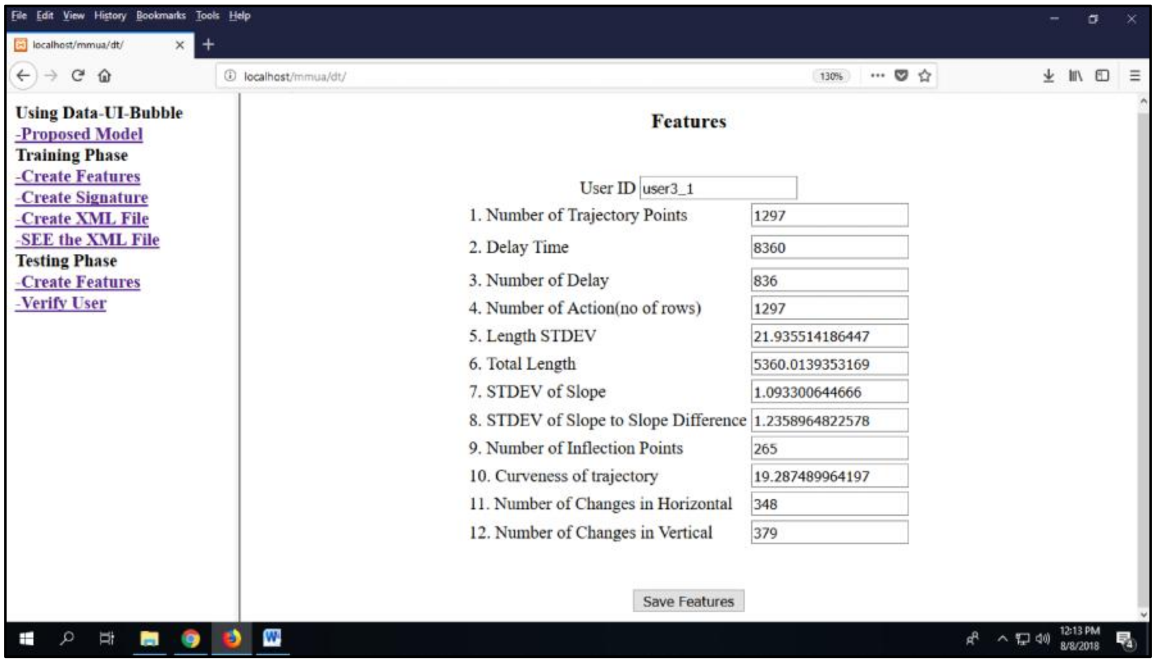


Figure 4.23:FAR and FRR of four Processes.

4.5.2 Implementation of Best Process

From the above discussion we have found that process-4 shows the best result. So using this process we will identify user from in his/her mouse movement data. Mouse movement data is ready to us as discussion in section 4.2. Now we will implement process-4 using these data. For implementation we have developed a system by using PhP and MySql. This system shows how many features are matched with all users. This number of matching will be largest for valid user. Requirements to run this system are described in previous section 4.1. Some important output screens are shown below.

Figure 4.24 shows output of feature generation. After selecting a user we get such features and these features values are needed to be saved into database by clicking on 'Save Features' button. By this way we have created features three times of a user by using mouse movement data of three sessions. Features of all users are created in same way.

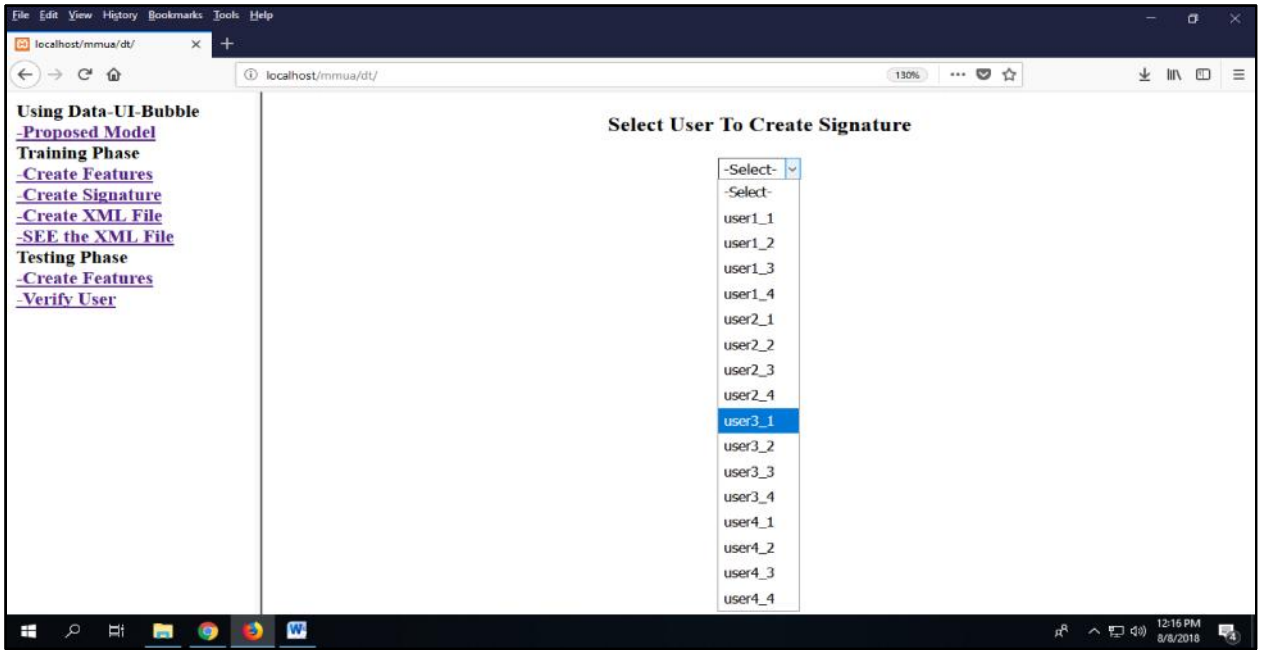


Features

User ID	user3_1
1. Number of Trajectory Points	1297
2. Delay Time	8360
3. Number of Delay	836
4. Number of Action(no of rows)	1297
5. Length STDEV	21.935514186447
6. Total Length	5360.0139353169
7. STDEV of Slope	1.093300644666
8. STDEV of Slope to Slope Difference	1.2358964822578
9. Number of Inflection Points	265
10. Curveness of trajectory	19.287489964197
11. Number of Changes in Horizontal	348
12. Number of Changes in Vertical	379

Figure 4.24: Output screen of feature generation using Specific Guided User Interface.

After creating features we have created signature of each user. As the figure 4.25 we will select a user from drop down menu then signature value will be created in user's background ofwork. The signature value is stored in database for each particular user.

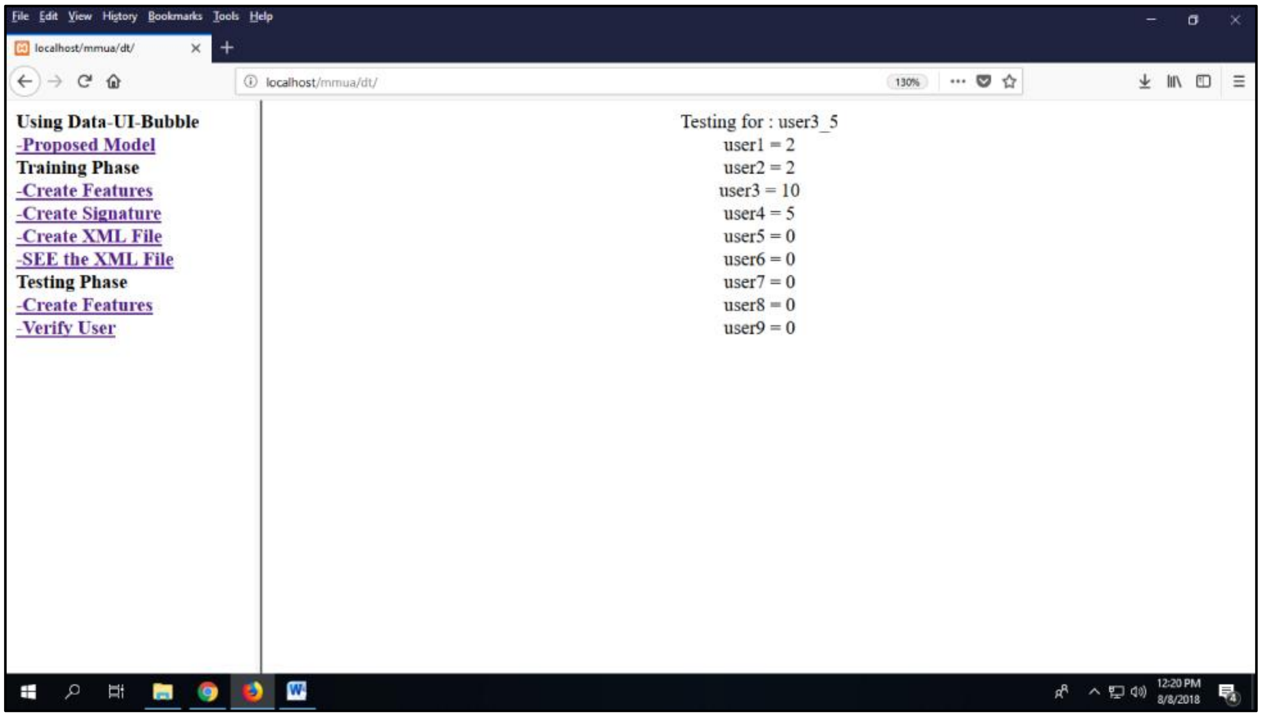


Using Data-UI-Bubble
-Proposed Model
Training Phase
-Create Features
-Create Signature
-Create XML File
-SEE the XML File
Testing Phase
-Create Features
-Verify User

Select User To Create Signature

-Select-
-Select-
user1_1
user1_2
user1_3
user1_4
user2_1
user2_2
user2_3
user2_4
user3_1
user3_2
user3_3
user3_4
user4_1
user4_2
user4_3
user4_4

Figure 4.25: Output screen of creating signature using Specific Guided User Interface.



Using Data-UI-Bubble
-Proposed Model
Training Phase
-Create Features
-Create Signature
-Create XML File
-SEE the XML File
Testing Phase
-Create Features
-Verify User

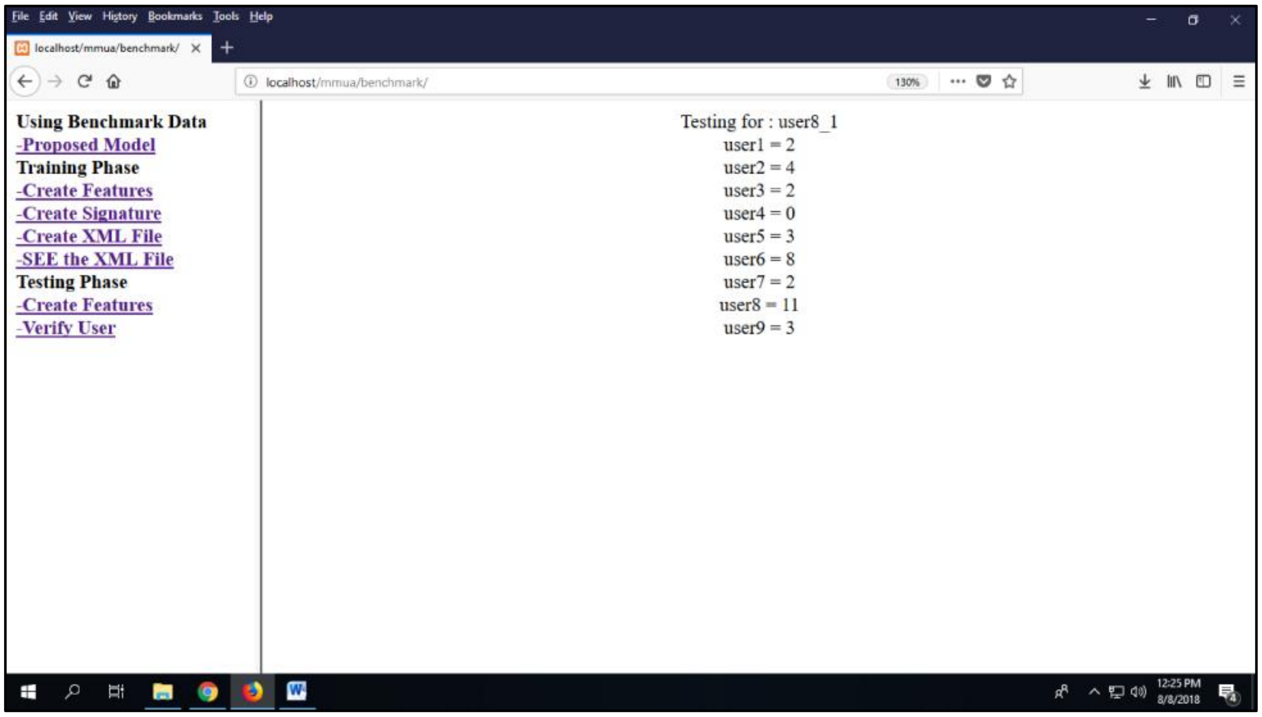
```
Testing for : user3_5
user1 = 2
user2 = 2
user3 = 10
user4 = 5
user5 = 0
user6 = 0
user7 = 0
user8 = 0
user9 = 0
```

Figure 4.26: Output screen of user verification using Specific Guided User Interface.

In testing phase or whenever user wants to login then again we have collected his/her mousemovement data and will create features as above. From these features a signature will be generated and will be compared with stored signature in database whenever we will verify the user.

In verification session, we will select a user and will get number of matching features with all corresponding users. In figure 4.26 we have seen that features of user-3 matches with 2 features of user-1, 2 features of user-2, 10 features of user-3 and 5 features of user-4. But for valid user the number (10) of matching feature is the largest.

In below figure 4.27, we have shownoutput of user verification for benchmark data. Here we have found that number of matching features of valid user (user-8) is the largest.


File Edit View History Bookmarks Tools Help
localhost/mmua/benchmark/
localhost/mmua/benchmark/
130%

Using Benchmark Data
-Proposed Model
Training Phase
-Create Features
-Create Signature
-Create XML File
-SEE the XML File
Testing Phase
-Create Features
-Verify User

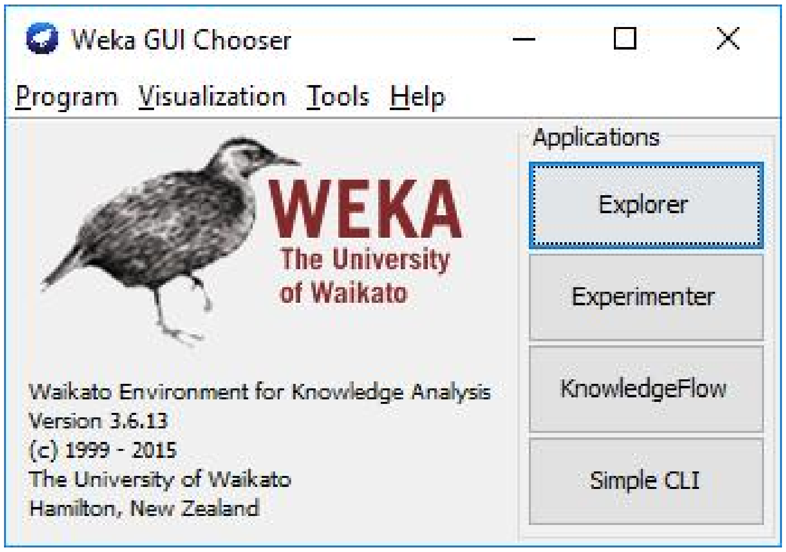
```
Testing for : user8_1
user1 = 2
user2 = 4
user3 = 2
user4 = 0
user5 = 3
user6 = 8
user7 = 2
user8 = 11
user9 = 3
```

Figure 4.27: Output screen of user verification using Benchmark Data.

By this way we can identify user by calculating number of matched features. If this number is the largest for a user then he/she is valid user and permitted to access otherwise not.

4.5.3 Multi Classifier Experiment

In this experiment we have use three classifiers, SVM, kNN and Naïve Bayes. We have used WEKA data mining tool to apply these classification[46].WEKA is an acronym which stands for Waikato Environment for Knowledge Analysis.A detail output of these three classifiers in WEKA is described in section 4.7. The WEKA is also the name of a bird, WEKA in New Zealand. WEKA is open source data mining software that uses a collection of machine learning algorithms.It is developed in The University of Waikato, New Zealand. Itis a modern platform for applied machine learning.



Waikato Environment for Knowledge Analysis
Version 3.6.13
(c) 1999 - 2015
The University of Waikato
Hamilton, New Zealand

Figure 4.28: WEKA Graphical User Interface.

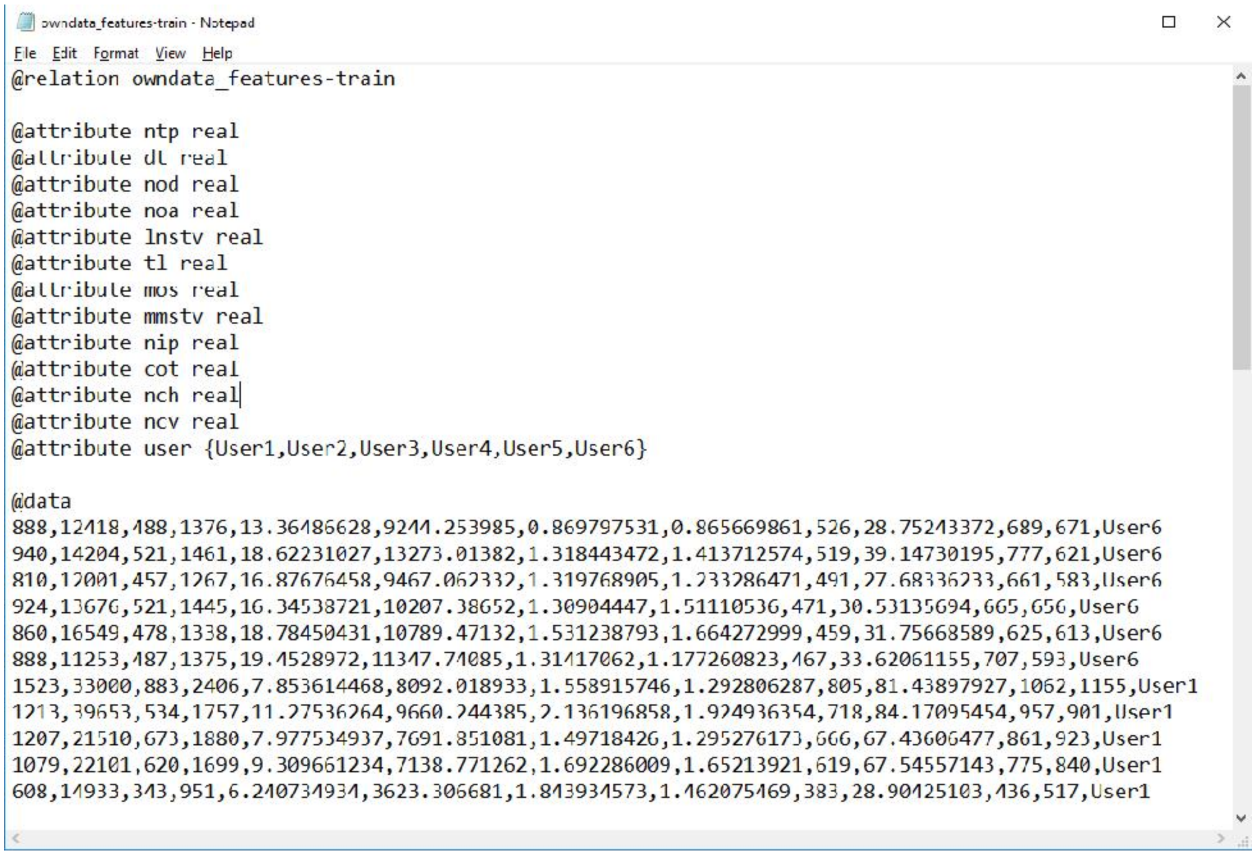
For this experiment we have train the classifier with a training set. Then we have tested the data with a test data set. From this training and testing we have predict the user based on mouse movement feature’s data. Multi classifiers experiment examples have been used on mouse movement features data as described section 4.3 and data in table 4.6. For this experiment we have taken data in arff format. Some definitions are briefly illustrated in below related to these experiments.

ARFF File Format

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files have two distinct sections. The first section is the Header information, which is followed the Data information. The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The figure 4.29 and 4.30 are shown arff format file.

Training

The concept of using a "training set" is to produce the model. This takes a data set with known output values and uses this data set to build our model. Then, whenever we have a new data set, with an unknown output value, we put it through the model and produce our expected output.



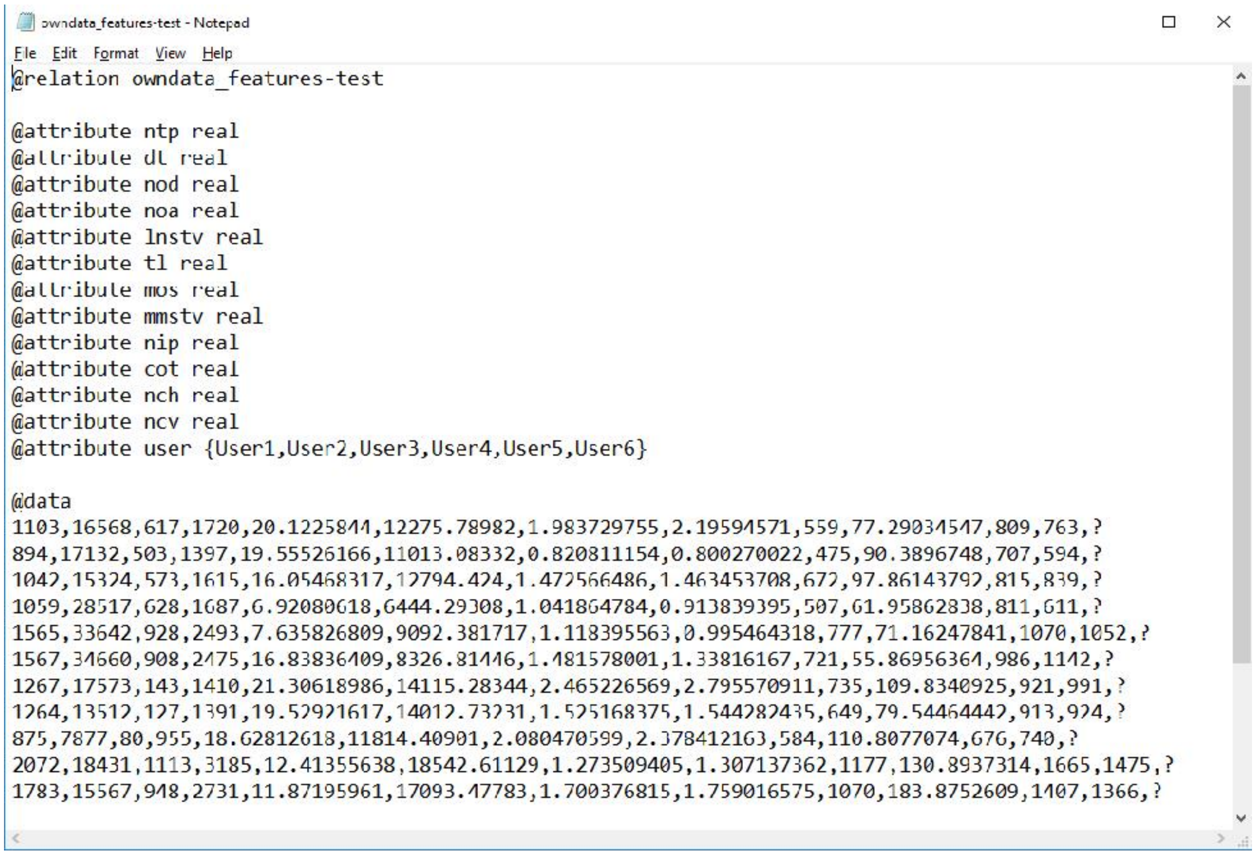
```
@relation owndata_features-train

@attribute ntp real
@attribute dl real
@attribute nod real
@attribute noa real
@attribute lnstv real
@attribute tl real
@attribute mos real
@attribute mmstv real
@attribute nip real
@attribute cot real
@attribute nch real
@attribute ncv real
@attribute user {User1,User2,User3,User4,User5,User6}

@data
888,12418,488,1376,13.36486628,9244.253985,0.869797531,0.865669861,526,28.75243372,689,671,User6
940,14204,521,1461,18.62231027,13273.01382,1.318443472,1.413712574,519,39.14730195,777,621,User6
810,12001,457,1267,16.87676458,9467.062332,1.319768905,1.233286471,491,27.68336233,661,583,User6
924,13676,521,1445,16.34538721,10207.38652,1.30904447,1.51110536,471,30.53135694,665,656,User6
860,16549,478,1338,18.78450431,10789.47132,1.531238793,1.664272999,459,31.75668589,625,613,User6
888,11253,487,1375,19.4528972,11347.74085,1.31417062,1.177260823,467,33.62061155,707,593,User6
1523,33000,883,2406,7.853614468,8092.018933,1.558915746,1.292806287,805,81.43897927,1062,1155,User1
1213,39653,534,1757,11.27536264,9660.244385,2.136196858,1.924936354,718,84.17095454,957,901,User1
1207,21510,673,1880,7.977534937,7691.851081,1.49718426,1.295276173,666,67.43606477,861,923,User1
1079,22101,620,1699,9.309661234,7138.771262,1.692286009,1.65213921,619,67.54557143,775,840,User1
608,14933,343,951,6.240734934,3623.306681,1.843934573,1.462075469,383,28.90425103,436,517,User1
```

Figure 4.29: A sample of training dataset.

However, this type of model takes it one step further, and it is common practice to take an entire training set and divide it into two parts: take about 60-80 percent of the data and put it into our training set, which we will use to create the model; then take the remaining data and put it into a test set, which we'll use immediately after creating the model to test the accuracy of our model. But in experiment below we have use different data set for training and testing. In training session we have used data set of known registered user's mouse movement features value (Figure 4.29). After the training the model is built and we have preserved the model for further use.

Testing

The classifier is evaluated on how well it predicts the class (user) of a set of instance load from a file. For each instance of the test set the model is used to predict a class value, although this instance already have a class value. Summing up the cases where the predicted value was the same as the real value gives the accuracy measurement. In figure 4.30 a sample testing dataset is shown where we want to predict user. The question mark (?) shows unknown user. After the testing based on trained model we will find the user. The classifiers will predict the user in ?sign by following model that is preserved in training session.

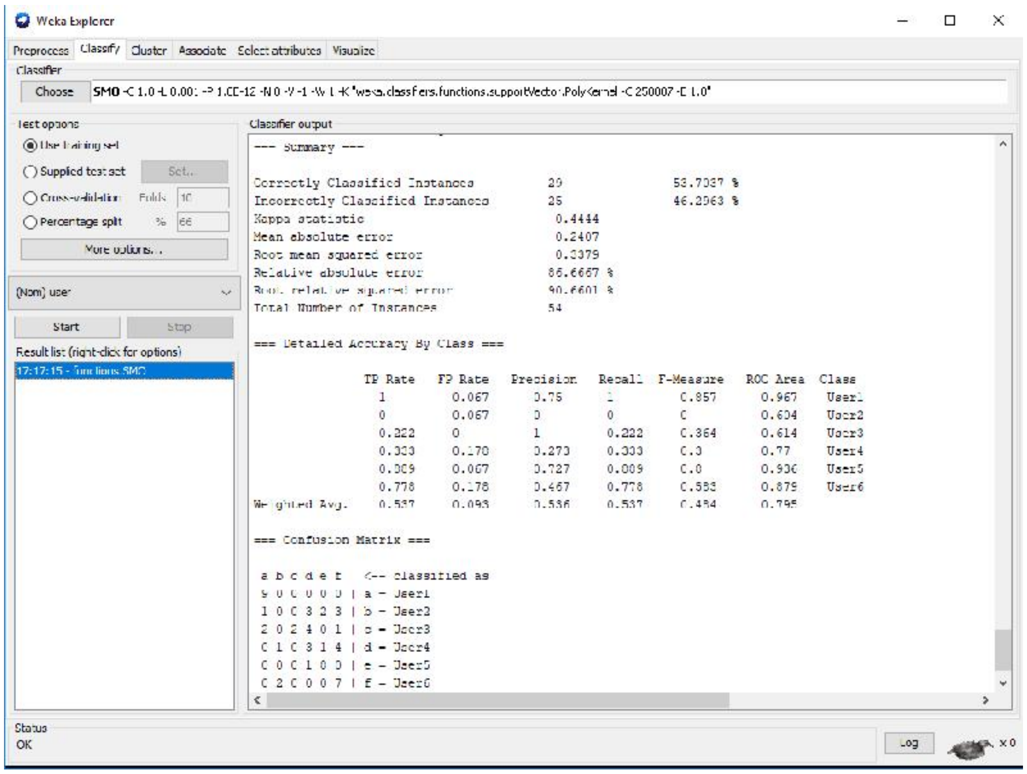


```
@relation owndata_features-test

@attribute ntp real
@attribute dt real
@attribute nod real
@attribute noa real
@attribute lnstv real
@attribute tl real
@attribute mos real
@attribute mmstv real
@attribute nip real
@attribute cot real
@attribute nch real
@attribute ncv real
@attribute user {User1,User2,User3,User4,User5,User6}

@data
1103,16568,617,1720,20.1225844,12275.78982,1.983729755,2.19594571,559,77.29034547,809,763,?
894,17132,503,1397,19.55526166,11013.08332,0.820811154,0.800270022,475,90.3896748,707,594,?
1042,15324,573,1615,16.05468317,12794.424,1.472566486,1.463453708,672,97.86143792,815,839,?
1059,28517,628,1687,6.92080618,6444.29308,1.041864784,0.913839395,507,61.95862838,811,611,?
1565,33642,928,2493,7.635826809,9092.381717,1.118395563,0.995464318,777,71.16247841,1070,1052,?
1567,34660,908,2475,16.83836409,8326.81446,1.481578001,1.33816167,721,55.86956364,986,1142,?
1267,17573,143,1410,21.30618986,14115.28344,2.465226569,2.795570911,735,109.8340925,921,991,?
1264,13512,127,1391,19.52921617,14012.73231,1.525168375,1.544282435,649,79.54464442,913,924,?
875,7877,80,955,18.62812618,11814.40901,2.080470599,2.378412163,584,110.8077074,676,740,?
2072,18431,1113,3185,12.41355638,18542.61129,1.273509405,1.307137362,1177,130.8937314,1665,1475,?
1783,15567,948,2731,11.87195961,17093.47783,1.700376815,1.759016575,1070,183.8752609,1407,1366,?
```

Figure 4.30: A sample of testing dataset.

Now we will complete the following task for each classifier, SVM, kNN and Naïve Bayes.

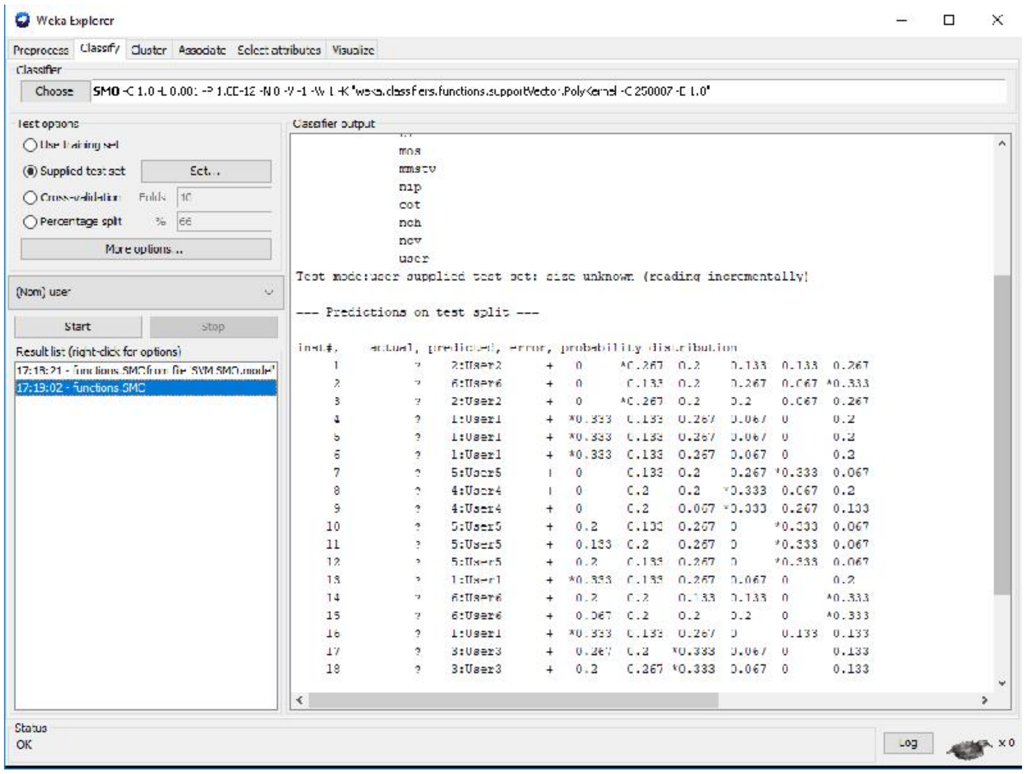
- Build a model by training
- Test by the model
- Collect the predicted data
- Analysis and compare performance of prediction



SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

```
=== Summary ===

Correctly Classified Instances          29               53.7037 %
Incorrectly Classified Instances        25               46.2963 %
Kappa statistic                          0.4444
Mean absolute error                      0.2407
Root mean squared error                  0.3379
Relative absolute error                 86.6667 %
Root relative squared error             90.6601 %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.067     0.75       1         0.857      0.967    User1
                 0         0.067     0          0         0          0.604    User2
                 0.222     0         1          0.222     0.364      0.614    User3
                 0.333     0.178     0.273      0.333     0.3        0.77     User4
                 0.889     0.067     0.727      0.889     0.8        0.936    User5
                 0.778     0.178     0.467      0.778     0.583      0.879    User6
Weighted Avg.    0.537     0.093     0.536      0.537     0.484      0.795

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 9 0 0 0 0 0 | a = User1
 1 0 0 3 2 3 | b = User2
 2 0 2 4 0 1 | c = User3
 0 1 0 3 1 4 | d = User4
 0 0 0 1 8 0 | e = User5
 0 2 0 0 0 7 | f = User6
```

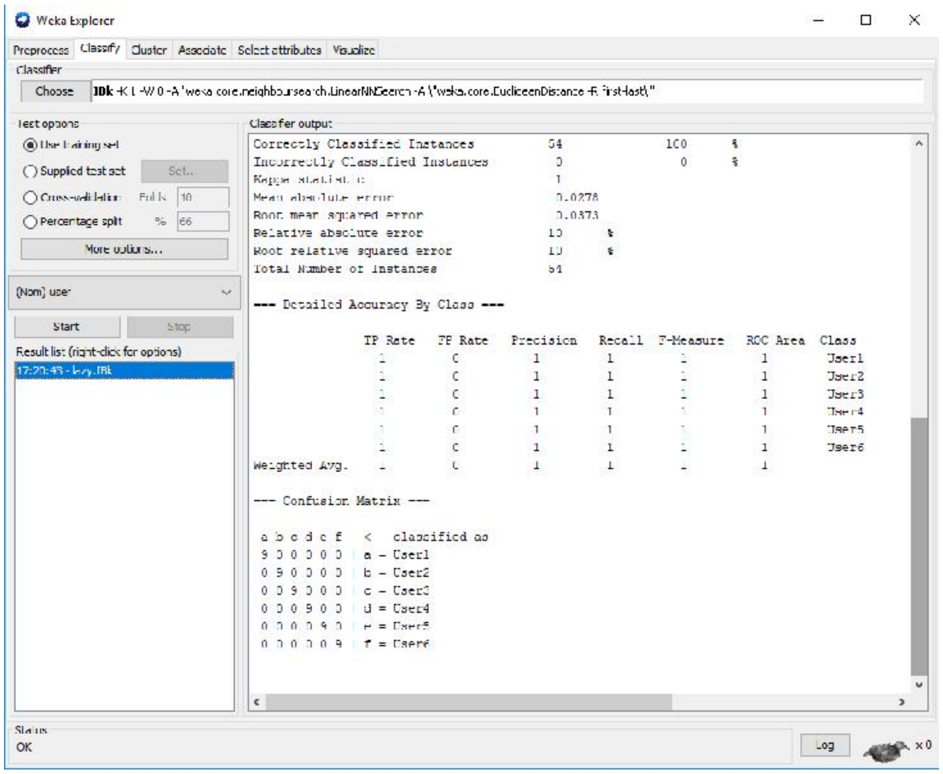
Figure 4.31: Screen short of SVM Training with WEKA.



```
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
```

```
              mos
              mmstv
              nip
              cot
              nch
              nev
              user
Test mode:user supplied test set: size unknown (reading incrementally)

=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1          ?      2:User2       +   0      *0.267  0.2    0.133  0.133  0.267
     2          ?      6:User6       +   0       0.133  0.2    0.267  0.067 *0.333
     3          ?      2:User2       +   0      *0.267  0.2    0.2    0.067  0.267
     4          ?      1:User1       +  *0.333   0.133  0.267  0.067  0      0.2
     5          ?      1:User1       +  *0.333   0.133  0.267  0.067  0      0.2
     6          ?      1:User1       +  *0.333   0.133  0.267  0.067  0      0.2
     7          ?      5:User5       +   0       0.133  0.2    0.267 *0.333  0.067
     8          ?      4:User4       +   0       0.2    0.2   *0.333  0.067  0.2
     9          ?      4:User4       +   0       0.2    0.067 *0.333  0.267  0.133
    10          ?      5:User5       +   0.2     0.133  0.267  0     *0.333  0.067
    11          ?      5:User5       +   0.133   0.2    0.267  0     *0.333  0.067
    12          ?      5:User5       +   0.2     0.133  0.267  0     *0.333  0.067
    13          ?      1:User1       +  *0.333   0.133  0.267  0.067  0      0.2
    14          ?      6:User6       +   0.2     0.2    0.133  0.133  0     *0.333
    15          ?      6:User6       +   0.067   0.2    0.2    0.2    0     *0.333
    16          ?      1:User1       +  *0.333   0.133  0.267  0      0.133  0.133
    17          ?      3:User3       +   0.267   0.2   *0.333  0.067  0      0.133
    18          ?      3:User3       +   0.2     0.267 *0.333  0.067  0      0.133
```

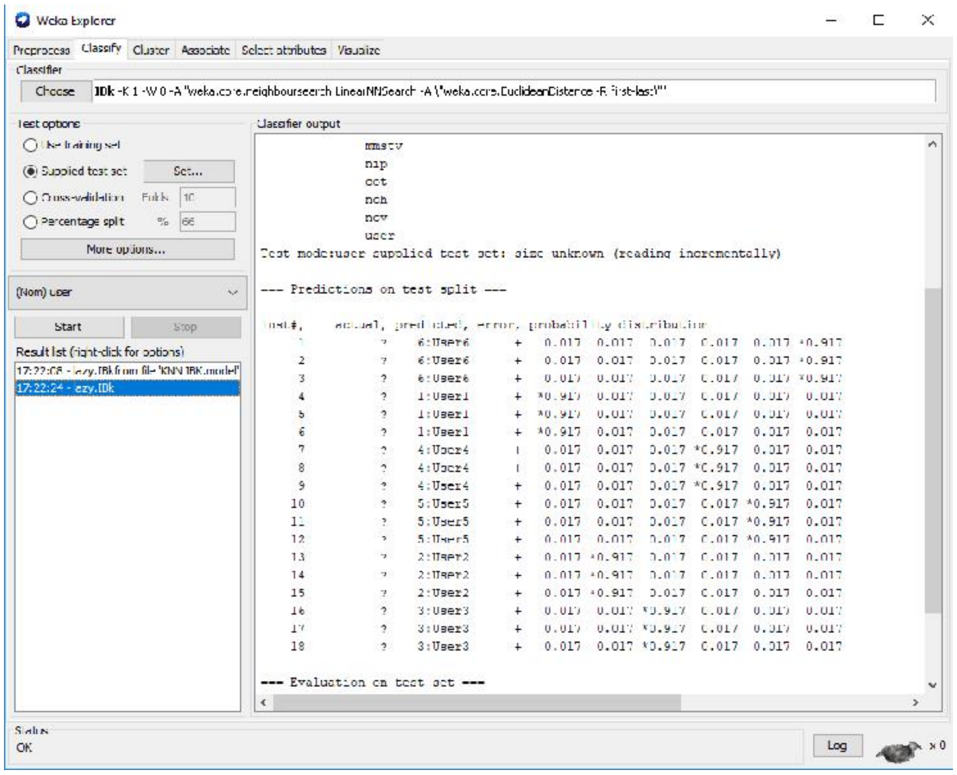
Figure 4.32: Screen short of SVM testing & user prediction with WEKA.



```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
```

```
Correctly Classified Instances          54              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0278
Root mean squared error                  0.0373
Relative absolute error                 10      %
Root relative squared error             10      %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                   1         0         1          1        1          1        User1
                   1         0         1          1        1          1        User2
                   1         0         1          1        1          1        User3
                   1         0         1          1        1          1        User4
                   1         0         1          1        1          1        User5
                   1         0         1          1        1          1        User6
Weighted Avg.      1         0         1          1        1          1

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 9 0 0 0 0 0 | a = User1
 0 9 0 0 0 0 | b = User2
 0 0 9 0 0 0 | c = User3
 0 0 0 9 0 0 | d = User4
 0 0 0 0 9 0 | e = User5
 0 0 0 0 0 9 | f = User6
```

Figure 4.33: Screen short of kNNtraining with WEKA.



Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: Supplied test set

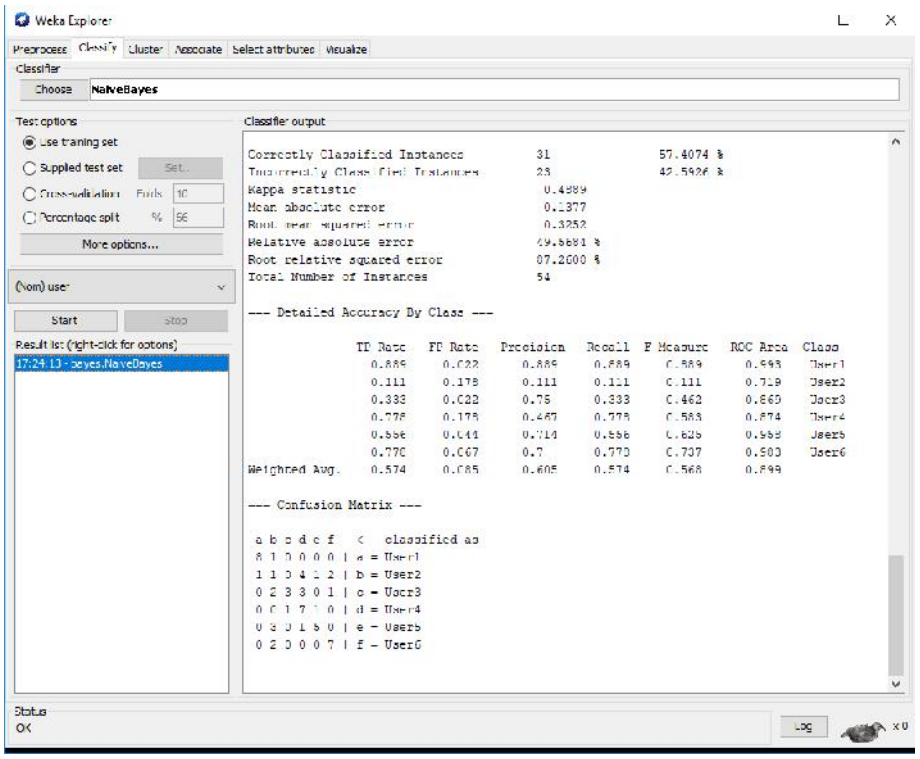
(Nom) user

Result list (right-click for options)
17:22:08 - lazy.IBk from file 'KNN IBK.model'
17:22:24 - lazy.IBk

```
               mmstv
               nip
               oct
               nch
               nov
               user
Test mode:user supplied test set: size unknown (reading incrementally)

=== Predictions on test split ===

inst#,     actual, predicted, error, probability distribution
     1          ?      6:User6      +   0.017  0.017  0.017  0.017  0.017 *0.917
     2          ?      6:User6      +   0.017  0.017  0.017  0.017  0.017 *0.917
     3          ?      6:User6      +   0.017  0.017  0.017  0.017  0.017 *0.917
     4          ?      1:User1      +  *0.917  0.017  0.017  0.017  0.017  0.017
     5          ?      1:User1      +  *0.917  0.017  0.017  0.017  0.017  0.017
     6          ?      1:User1      +  *0.917  0.017  0.017  0.017  0.017  0.017
     7          ?      4:User4      +   0.017  0.017  0.017 *0.917  0.017  0.017
     8          ?      4:User4      +   0.017  0.017  0.017 *0.917  0.017  0.017
     9          ?      4:User4      +   0.017  0.017  0.017 *0.917  0.017  0.017
    10          ?      5:User5      +   0.017  0.017  0.017  0.017 *0.917  0.017
    11          ?      5:User5      +   0.017  0.017  0.017  0.017 *0.917  0.017
    12          ?      5:User5      +   0.017  0.017  0.017  0.017 *0.917  0.017
    13          ?      2:User2      +   0.017 *0.917  0.017  0.017  0.017  0.017
    14          ?      2:User2      +   0.017 *0.917  0.017  0.017  0.017  0.017
    15          ?      2:User2      +   0.017 *0.917  0.017  0.017  0.017  0.017
    16          ?      3:User3      +   0.017  0.017 *0.917  0.017  0.017  0.017
    17          ?      3:User3      +   0.017  0.017 *0.917  0.017  0.017  0.017
    18          ?      3:User3      +   0.017  0.017 *0.917  0.017  0.017  0.017

=== Evaluation on test set ===
```

Status: OK

Figure 4.34: Screen short of kNNtesting & user prediction with WEKA.



Classifier: NaiveBayes

Test options: Use training set

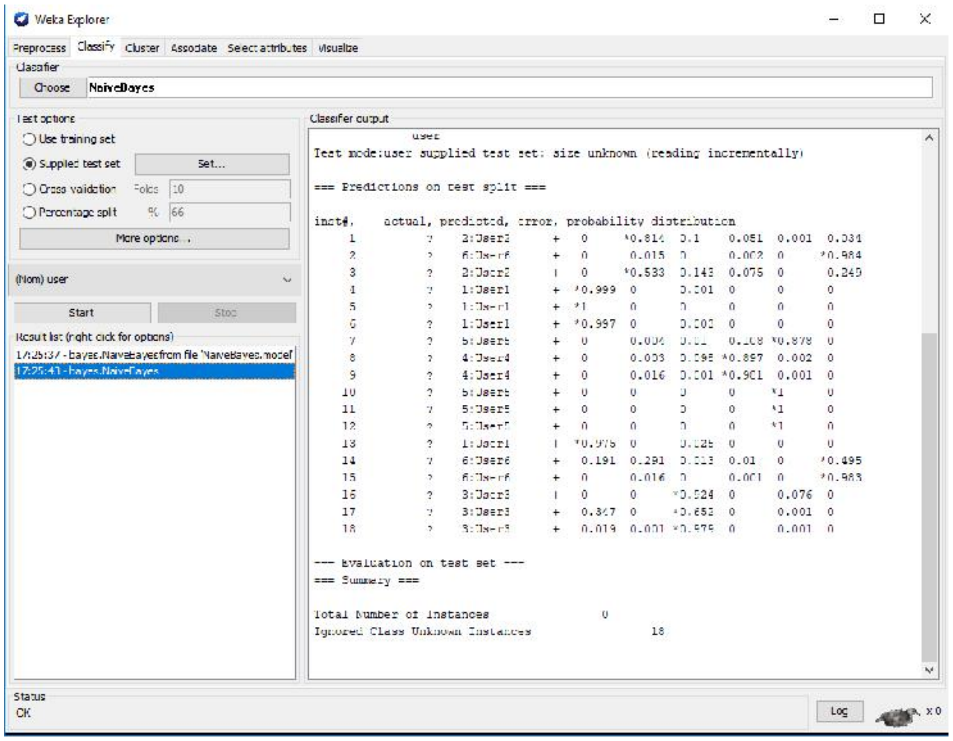
(Nom) user

Result list (right-click for options)
17:24:13 - bayes.NaiveBayes

```
Correctly Classified Instances          31               57.4074 %
Incorrectly Classified Instances        23               42.5926 %
Kappa statistic                          0.4889
Mean absolute error                      0.1377
Root mean squared error                  0.3252
Relative absolute error                 49.5681 %
Root relative squared error             87.2600 %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.889     0.022     0.889     0.889     0.889      0.993    User1
                 0.111     0.178     0.111     0.111     0.111      0.719    User2
                 0.333     0.022     0.75      0.333     0.462      0.869    User3
                 0.778     0.178     0.467     0.778     0.583      0.874    User4
                 0.556     0.044     0.714     0.556     0.625      0.958    User5
                 0.778     0.067     0.7       0.778     0.737      0.983    User6
Weighted Avg.    0.574     0.085     0.605     0.574     0.568      0.899

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 8 1 0 0 0 0 | a = User1
 1 1 0 4 1 2 | b = User2
 0 2 3 3 0 1 | c = User3
 0 0 1 7 1 0 | d = User4
 0 3 0 1 5 0 | e = User5
 0 2 0 0 0 7 | f = User6
```

Status: OK

Figure 4.35: Screen short of Naïve Bayes training with WEKA.



NaiveBayes

Test options: Use training set [✓ Supplied test set] Cross validation (Folds 10) Percentage split (% 66)

(Nom) user

Result list (right click for options)
17:25:37 - bayes.NaiveBayes from file 'NaiveBayes.model'
17:25:43 - bayes.NaiveBayes

```
                user
Test mode:user supplied test set: size unknown (reading incrementally)

=== Predictions on test split ===

inst#,    actual, predicted, error, probability distribution
     1          ?    2:User2      +   0      *0.814  0.1     0.051  0.001  0.034
     2          ?    6:User6      +   0       0.015  0       0.002  0      *0.984
     3          ?    2:User2      +   0      *0.533  0.143   0.075  0       0.249
     4          ?    1:User1      +  *0.999   0      0.001   0      0      0
     5          ?    1:User1      +  *1       0      0       0      0      0
     6          ?    1:User1      +  *0.997   0      0.003   0      0      0
     7          ?    5:User5      +   0       0.004  0.01    0.108 *0.878  0
     8          ?    4:User4      +   0       0.003  0.098  *0.897  0.002  0
     9          ?    4:User4      +   0       0.016  0.001  *0.901  0.001  0
    10          ?    5:User5      +   0       0      0       0     *1      0
    11          ?    5:User5      +   0       0      0       0     *1      0
    12          ?    5:User5      +   0       0      0       0     *1      0
    13          ?    1:User1      +  *0.975   0      0.025   0      0      0
    14          ?    6:User6      +   0.191   0.291  0.013   0.01   0     *0.495
    15          ?    6:User6      +   0       0.016  0       0.001  0     *0.983
    16          ?    3:User3      +   0       0     *0.924   0      0.076  0
    17          ?    3:User3      +   0.347   0     *0.652   0      0.001  0
    18          ?    3:User3      +   0.019   0.001 *0.979   0      0.001  0

=== Evaluation on test set ===
=== Summary ===

Total Number of Instances              0
Ignored Class Unknown Instances                18
```

Status
OK

Figure 4.36: Screen short of Naïve Bayes testing & user prediction with WEKA.

Collection of Predicted Data:

Table 4.13: Predicted user using SVM Classifier

Instances	Predicted	Actual
1	User6	User6
2	User6	User6
3	User2	User6
4	User1	User1
5	User1	User1
6	User1	User1
7	User5	User4
8	User4	User4
9	User4	User4
10	User5	User5
11	User5	User5
12	User5	User5
13	User1	User2
14	User6	User2
15	User6	User2
16	User1	User3
17	User3	User3
18	User3	User3

Table 4.14: Predicted user using kNN Classifier

Instances	Predicted	Actual
1	User6	User6
2	User6	User6
3	User6	User6
4	User1	User1
5	User1	User1
6	User1	User1
7	User4	User4
8	User4	User4
9	User4	User4
10	User5	User5
11	User5	User5
12	User5	User5
13	User2	User2
14	User2	User2
15	User2	User2
16	User3	User3
17	User3	User3
18	User3	User3

Table 4.15: Predicted user using Naïve Bayes Classifier

Instances	Predicted	Actual
1	User2	User6
2	User6	User6
3	User2	User6
4	User1	User1
5	User1	User1
6	User1	User1
7	User5	User4
8	User4	User4
9	User4	User4
10	User5	User5
11	User5	User5
12	User5	User5
13	User1	User2
14	User6	User2
15	User6	User2
16	User3	User3
17	User3	User3
18	User3	User3

Comparing Results of Classifiers to Predict User

We have used WEKA data mining tool to predict user. As describe above, first we have train a classifier by known data set to create a model. This model is stored for further using. In testing data set we have used “?” sign instead of actual data (figure 4.30). The testing phase is performed based on previously saved model. In testing output we have found predicted data in correspond of “?” sign (figure 4.32, 4.34 and 4.36). In same way we have perform this experiment by three algorithms, SVM, kNN and Naive Bayes.

After the user prediction we have compared performance of each classifier. As the table below the kNN classifier predicts all users accurately.

Table 4.16: Comparing Results of Predicted User by Classifier

Classifiers	Total Tested Instances	Number of valid User Prediction	Number of invalid User Prediction
SVM	18	12	6
kNN		18	0
Naïve Bayes		12	6

We have used same data set for testing all the classifiers. User prediction depends on model created by the classifiers. The corresponding persevered models are used for corresponding classifiers. From the table 4.16 above it is meant that kNN classifier performs the best to predict user.

4.6 Performance Analysis

In this section we have illustrate the performance of proposed Process. These are described for four experiments. It is mentioned that we have applied same process(Process-4 from section 4.5.1) over our four experiments to check its accuracy and performance.

4.6.1 Experiment 1: Free Style Data Tracer Using JitbitTool

In section 4.5 we have find out a best process of calculating features to identify valid user. These calculations were done based on data collected by Jitbit tool[47]. We have chosen calculation of Process-4 among the mentioned Process for the best result. We have used data for finding best process by capturing data with Jitbit tool in free style user mouse movement. So this section is described in the previous section 4.5.

4.6.2 Experiment 2: Free Style Data Tracer Using RUI Tool

To evaluate our method we have again collected data from 8 users using RUI tool [48]. This section is related with section 4.1.2 and section 4.2.2 where experiments are setup and data are collected. The users have applied our proposed method. In this result we have found satisfactory result. That means the proposed method is independent over various mouse movement data collecting tools. For this experiment users were trainees of an institute. They were taking training with computer in a lab. We have installed the RUI in every computer of the lab. Mouse movement data are collected in user background work whenever users were working with mouse.

Data collected in first day are used for registration session and data collected in second day are used for authentication session. After training and testing with our proposed method, feature matching confusion matrixare found as below.

Table 4.17:Feature Matching Confusion Matrix of Data Collected by RUI

User		User Predicted							
		User 1	User 1	User 3	User 4	User 5	User 6	User 7	User 8
User Tested	User 1	6	2	4	0	3	3	4	3
	User 2	4	9	5	2	7	6	3	8
	User 3	2	7	9	0	8	5	5	4
	User 4	1	1	1	8	2	2	0	3
	User 5	4	3	6	1	7	3	2	5
	User 6	3	3	8	2	6	6	7	5
	User 7	4	7	2	1	5	8	4	5
	User 8	2	4	2	0	3	2	3	5

We have total twelve features. The table 4.17 shows number features matched with each user. We have found that maximum number of features will be matched for valid user. We get the accurate result for user all eight users except user 6 and user7. From the above table it is meant that the proposed method shows good result to identify valid users.

4.6.3 Experiment 3: Specific Guided User Interface (UI)

This section is related with section 4.1.3 and section 4.2.3. For this experiment we have selected four voluntary for collecting mouse movement data. Data are collected as described in section 4.1.3 after the experiment set up described in section 4.1.3 and Its Confusion Matrix is shown below.

Table 4.18: Confusion Matrix of Data Collected bySpecific GuidedUser Interface

User		User Predicted			
		User 1	User 1	User 3	User 4
User Tested	User 1	4	4	2	1
	User 2	1	3	2	2
	User 3	3	2	4	2
	User 4	1	3	2	3

As the previous experiment, we have again search for maximum number of features those are matched for valid user. We get the accurate result for user all four users. From the above table it is meant that the proposed method shows good result to identify valid users. This experiment also proved that proposed method performs good to identify users.

4.6.4 Experiment 4: Using Benchmark Data

After collecting the benchmark data (described in section 4.1.4 and 4.2.4)we have analyzed these data to find the feature vectors. Then we have applied our proposed method on this benchmark data. The Table4.4 and 4.5 shows a sample of training data. From the mouse trajectory,mouse positions and delay times are filtered outfrom the dataset. We have collected the x-y mouse positions and delay times to generate features. From these data we have calculated the 12 feature vectors for each user [2] [4] [10]. After the training and testing we have found confusion matrix as the below.

Table 4.19: Confusion matrix using benchmark data

User		**User Predicted**								
		User1	User2	User3	User4	User5	User6	User7	User8	User9
User Tested	User1	6	2	4	0	3	4	3	4	3
	User2	4	10	5	2	7	4	9	3	8
	User3	2	7	5	0	8	5	9	5	10
	User4	1	1	1	10	2	2	2	0	3
	User5	4	3	6	1	10	5	3	8	5
	User6	3	3	8	2	6	8	6	7	5
	User7	4	7	10	1	10	10	6	9	5
	User8	2	4	2	0	3	8	2	11	3
	User9	0	7	3	3	6	2	8	1	8

From this experiment we have also find that our proposed method shows good result on benchmark data. Among nine users seven users are identified correctly and two users are identified wrongly by searching maximum number of matched feature.

4.7Output of Learning and Classification

In this section we have illustrated output status of classification algorithms. Performance of three classifiers, SVM, kNN, Naive Bayes is shown here in terms of *visual classifiers error* curve and *margin curve*. We have shown the performance of output result in comparatively for both our own data and benchmark data.

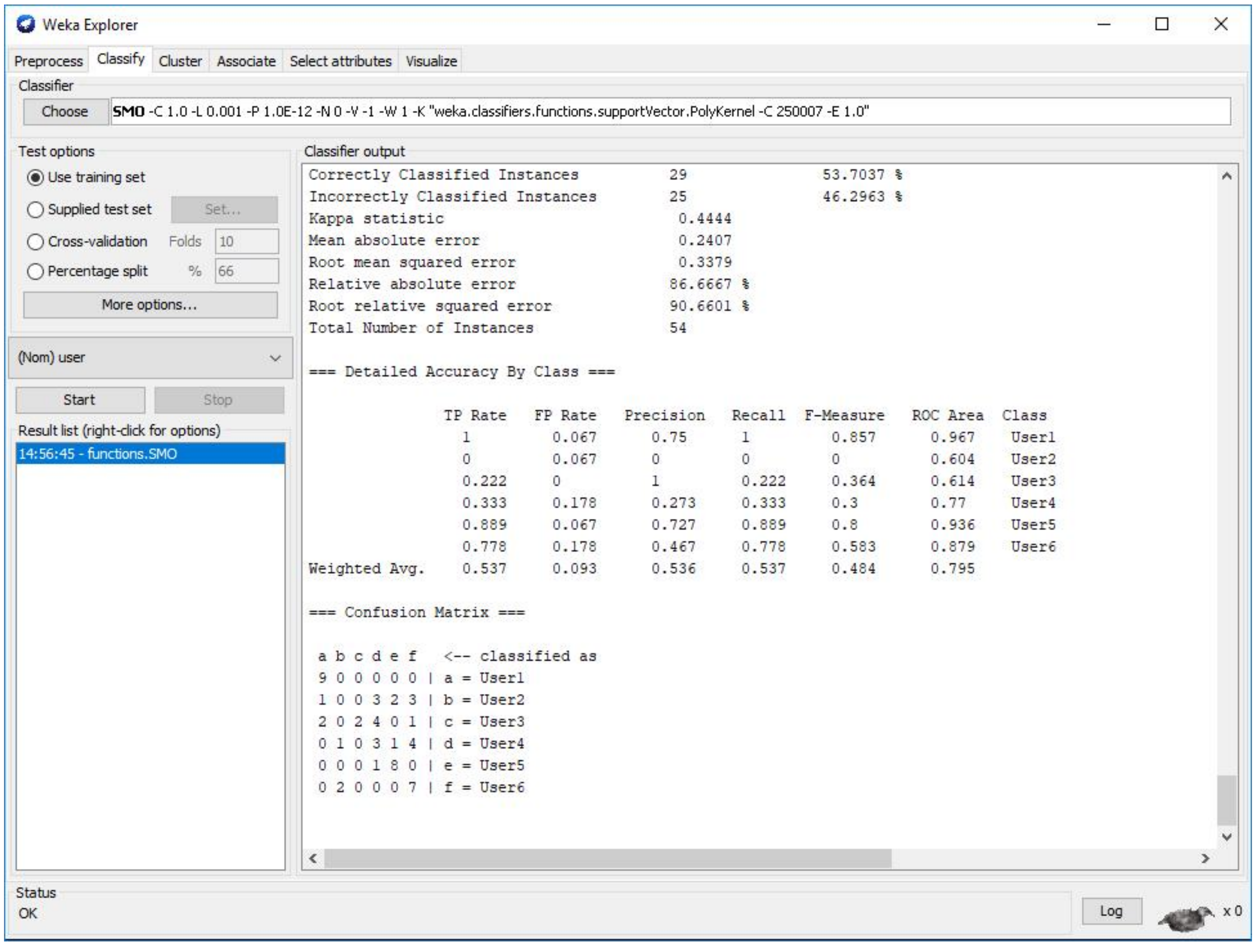
The v*isualize classifier error curve* brings up a visualization window that plotsthe results of classification. Correctly classified instances are representedby crosses, whereas incorrectly classified ones show up as squares.

The *visual margin curve* print the cumulative frequency by generates a plot illustrating the prediction margin. The margin is defined as the difference between the probability predicted for the actual class and the highest probability predicted for the other classes. The negative values denote classification errors and the dominant class is not the correct one. On the other side the positive values denote good performance of classification.

To test the classification result we have used classification algorithms over the features generated. As the research target we have used SVM, K Nearest Neighbor and Naïve Bayes classifier. WEKA data mining tool is used for these classifiers. Support vector machines (SVMs) [15][16] have good performance for classification. SVMs use a kernel function to map training data into a higher-dimensioned space so that the problem is linearly separable. Quadratic programming can be expensive for large problems, but sequential minimal optimization (SMO) is a fast, efficient algorithm for training SVMs and is the one implemented in WEKA. Naive Bayes is a simple probabilistic classifier based on applying Bayes' theorem (or Bayes's rule) with strong independence (naive) assumptions. Given a set of objects, each of which belongs to a known class, and each of which has a known vector of variables. In WEKA we have used NaïveBayes methods. k Nearest Neighbor algorithm is very simple to understand and used for classifying objects based on closest training examples in the feature space.

kNN algorithm measures the distance between a query scenario and a set of scenarios in the data set. Its applications range from vision to proteins to computational geometry to graphs and so on. Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple. It is also a lazy algorithm. It does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This means the training phase is pretty fast. kNN is also provided by Weka as a class "IBk". IBk implements kNN. It uses normalized distances for all attributes so that attributes on different scales have the same impact on the distance function. It may return more than k neighbors if there are ties in the distance. Neighbors are voted to form the final classification. The proposed method is tested using our collected data and then using bench mark data from other source. The testing output shows that KNN has the lowest error rate.

We have shown output of learning and classification from figure 4.37 to figure 4.48. A total output data can be taken from these figures. We have observed the performance of classifiers in terms ofvisual classifier error (4.39, 4.42 and4.45) and visual margin curve (Figure 4.46 to 4.48) of SVM, kNN and Naïve Bayes classifiers. These outputs are shown for both experiments, with using data of our own and benchmark data. In both cases we have found that kNN classifier shows more accuracy than other classifiers.



SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

Use training set [✓]
Supplied test set
Cross-validation Folds 10
Percentage split % 66
More options...

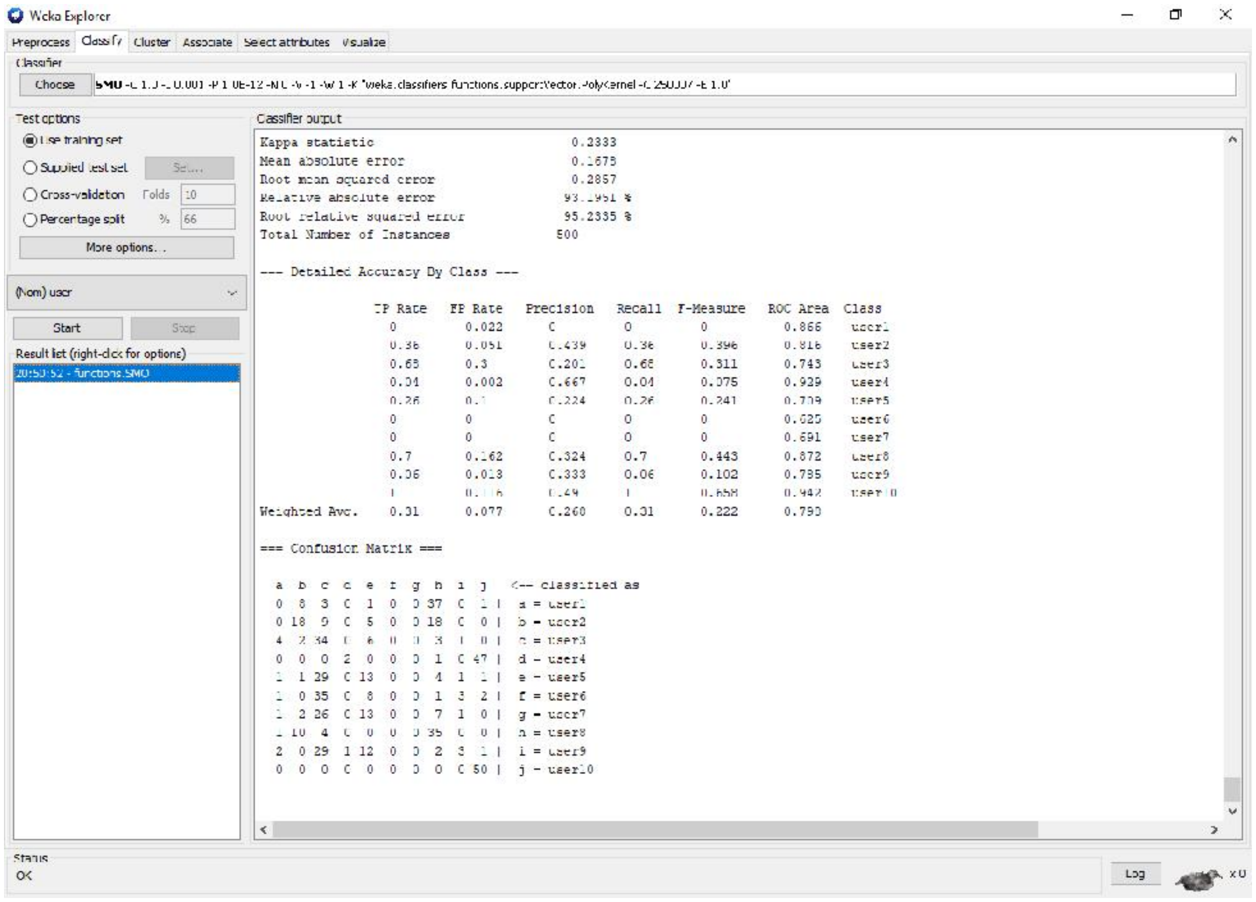
(Nom) user

14:56:45 - functions.SMO

```
Correctly Classified Instances          29               53.7037 %
Incorrectly Classified Instances        25               46.2963 %
Kappa statistic                          0.4444
Mean absolute error                      0.2407
Root mean squared error                  0.3379
Relative absolute error                 86.6667 %
Root relative squared error             90.6601 %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.067     0.75      1         0.857      0.967    User1
                 0         0.067     0         0         0          0.604    User2
                 0.222     0         1         0.222     0.364      0.614    User3
                 0.333     0.178     0.273     0.333     0.3        0.77     User4
                 0.889     0.067     0.727     0.889     0.8        0.936    User5
                 0.778     0.178     0.467     0.778     0.583      0.879    User6
Weighted Avg.    0.537     0.093     0.536     0.537     0.484      0.795

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 9 0 0 0 0 0 | a = User1
 1 0 0 3 2 3 | b = User2
 2 0 2 4 0 1 | c = User3
 0 1 0 3 1 4 | d = User4
 0 0 0 1 8 0 | e = User5
 0 2 0 0 0 7 | f = User6
```

OK

Figure4.37: SVM classifier output using Own Data



```
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"
```

```
Kappa statistic                          0.2333
Mean absolute error                      0.1675
Root mean squared error                  0.2857
Relative absolute error                 93.1951 %
Root relative squared error             95.2335 %
Total Number of Instances              500

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0.022      0          0         0          0.866    user1
                 0.36      0.051      0.439      0.36      0.396      0.816    user2
                 0.68      0.3        0.201      0.68      0.311      0.743    user3
                 0.04      0.002      0.667      0.04      0.075      0.929    user4
                 0.26      0.1        0.224      0.26      0.241      0.709    user5
                 0         0          0          0         0          0.625    user6
                 0         0          0          0         0          0.691    user7
                 0.7       0.162      0.324      0.7       0.443      0.872    user8
                 0.06      0.013      0.333      0.06      0.102      0.785    user9
                 1         0.116      0.49       1         0.658      0.942    user10
Weighted Avg.    0.31      0.077      0.260      0.31      0.222      0.790

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
  0  8  3  0  1  0  0 37  0  1 |  a = user1
  0 18  9  0  5  0  0 18  0  0 |  b = user2
  4  2 34  0  6  0  0  3  1  0 |  c = user3
  0  0  0  2  0  0  0  1  0 47 |  d = user4
  1  1 29  0 13  0  0  4  1  1 |  e = user5
  1  0 35  0  8  0  0  1  3  2 |  f = user6
  1  2 26  0 13  0  0  7  1  0 |  g = user7
  1 10  4  0  0  0  0 35  0  0 |  h = user8
  2  0 29  1 12  0  0  2  3  1 |  i = user9
  0  0  0  0  0  0  0  0  0 50 |  j = user10
```

Figure4.38: SVM classifieroutput using Benchmark Data

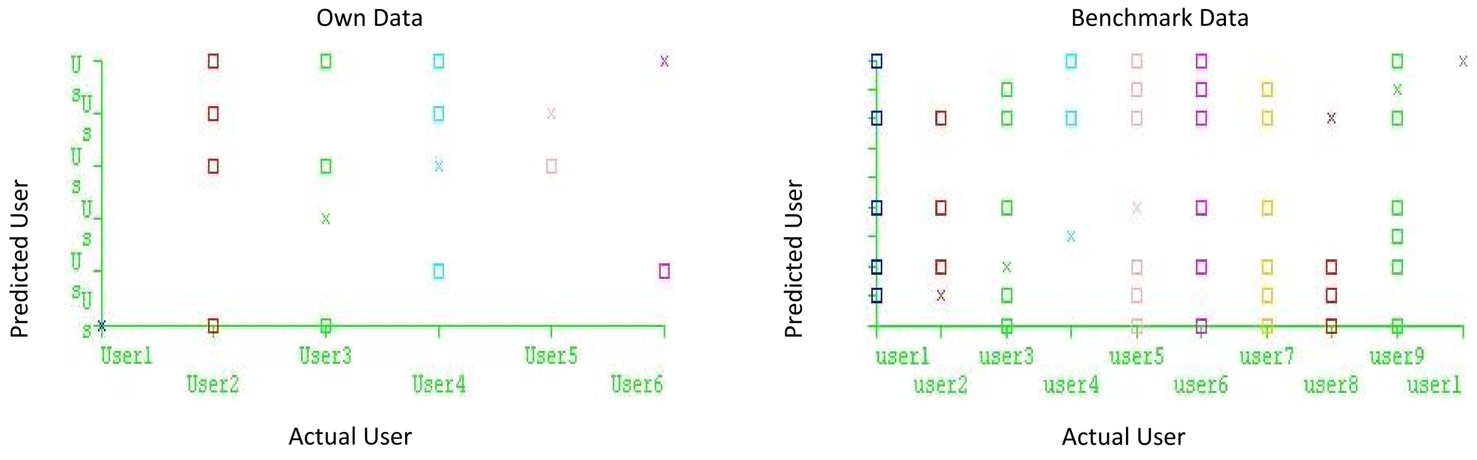
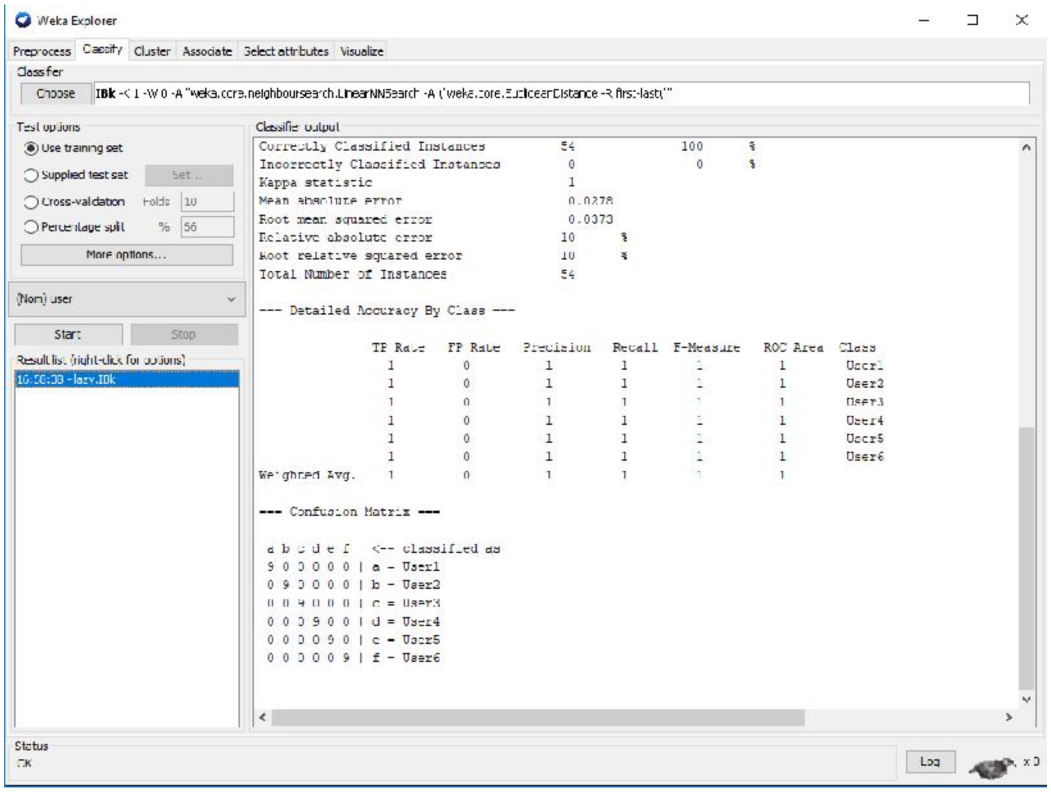


Figure 4.39: Visual classifier error of SVM

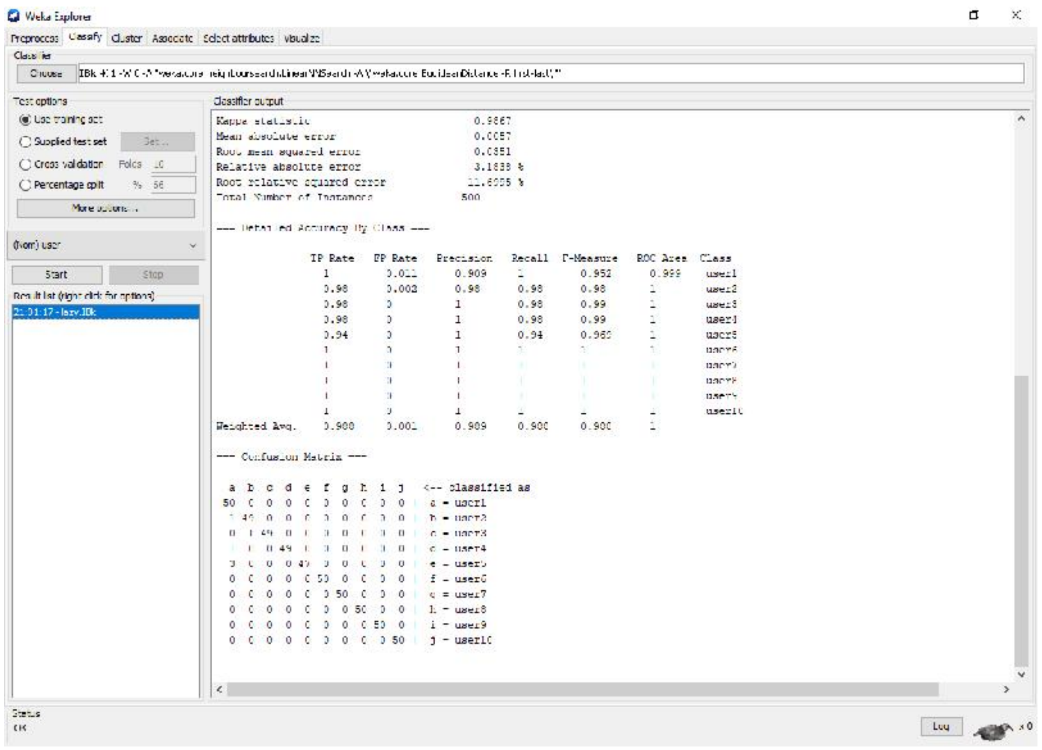


```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
```

Test options: (●) Use training set; Folds 10; % 56

```
Correctly Classified Instances          54              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0278
Root mean squared error                  0.0373
Relative absolute error                 10      %
Root relative squared error             10      %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1       User1
                 1         0          1         1         1          1       User2
                 1         0          1         1         1          1       User3
                 1         0          1         1         1          1       User4
                 1         0          1         1         1          1       User5
                 1         0          1         1         1          1       User6
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 9 0 0 0 0 0 | a = User1
 0 9 0 0 0 0 | b = User2
 0 0 9 0 0 0 | c = User3
 0 0 0 9 0 0 | d = User4
 0 0 0 0 9 0 | e = User5
 0 0 0 0 0 9 | f = User6
```

Figure4.40: kNNclassifieroutput using Own Data



```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
```

Test options: (●) Use training set; Folds 10; % 56

```
Kappa statistic                          0.9867
Mean absolute error                      0.0057
Root mean squared error                  0.0351
Relative absolute error                  3.1833 %
Root relative squared error             11.6955 %
Total Number of Instances              500

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.011      0.909     1         0.952      0.999    user1
                 0.98      0.002      0.98      0.98      0.98       1        user2
                 0.98      0          1         0.98      0.99       1        user3
                 0.98      0          1         0.98      0.99       1        user4
                 0.94      0          1         0.94      0.969      1        user5
                 1         0          1         1         1          1        user6
                 1         0          1         1         1          1        user7
                 1         0          1         1         1          1        user8
                 1         0          1         1         1          1        user9
                 1         0          1         1         1          1        user10
Weighted Avg.    0.988     0.001      0.989     0.988     0.988      1

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 50  0  0  0  0  0  0  0  0  0 |  a = user1
  1 49  0  0  0  0  0  0  0  0 |  b = user2
  0  1 49  0  0  0  0  0  0  0 |  c = user3
  1  0  0 49  0  0  0  0  0  0 |  d = user4
  3  0  0  0 47  0  0  0  0  0 |  e = user5
  0  0  0  0  0 50  0  0  0  0 |  f = user6
  0  0  0  0  0  0 50  0  0  0 |  g = user7
  0  0  0  0  0  0  0 50  0  0 |  h = user8
  0  0  0  0  0  0  0  0 50  0 |  i = user9
  0  0  0  0  0  0  0  0  0 50 |  j = user10
```

Figure4.41: kNNclassifieroutput using Benchmark Data

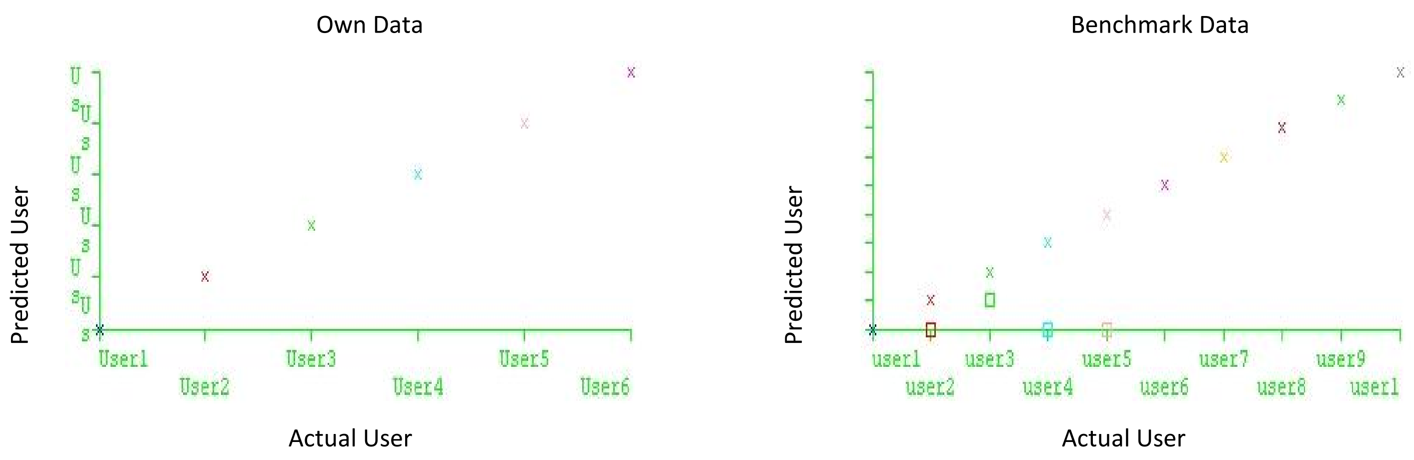
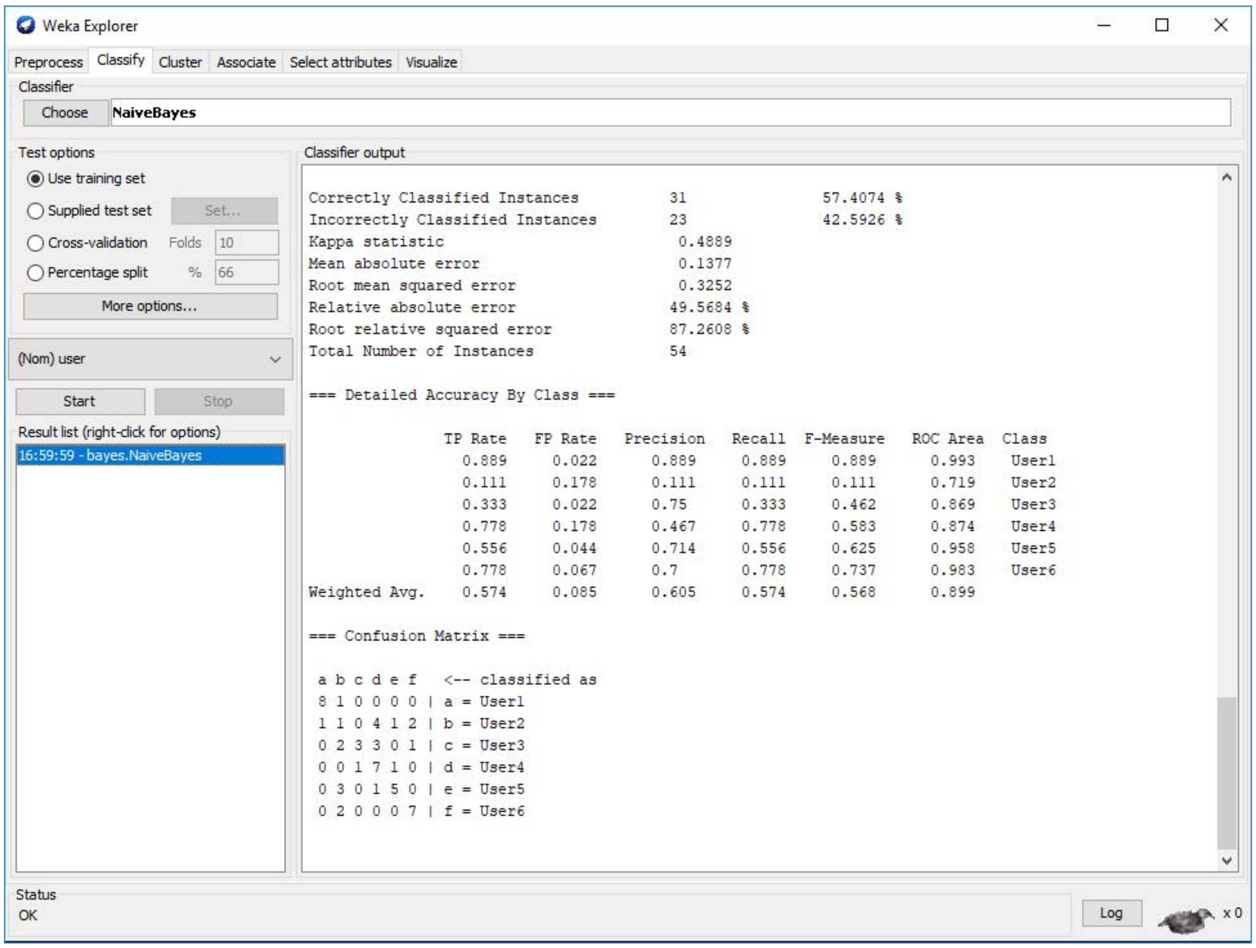


Figure 4.42: Visual classifier error of k nearest neighbor



Classifier: NaiveBayes

Test options: [✓ Use training set]

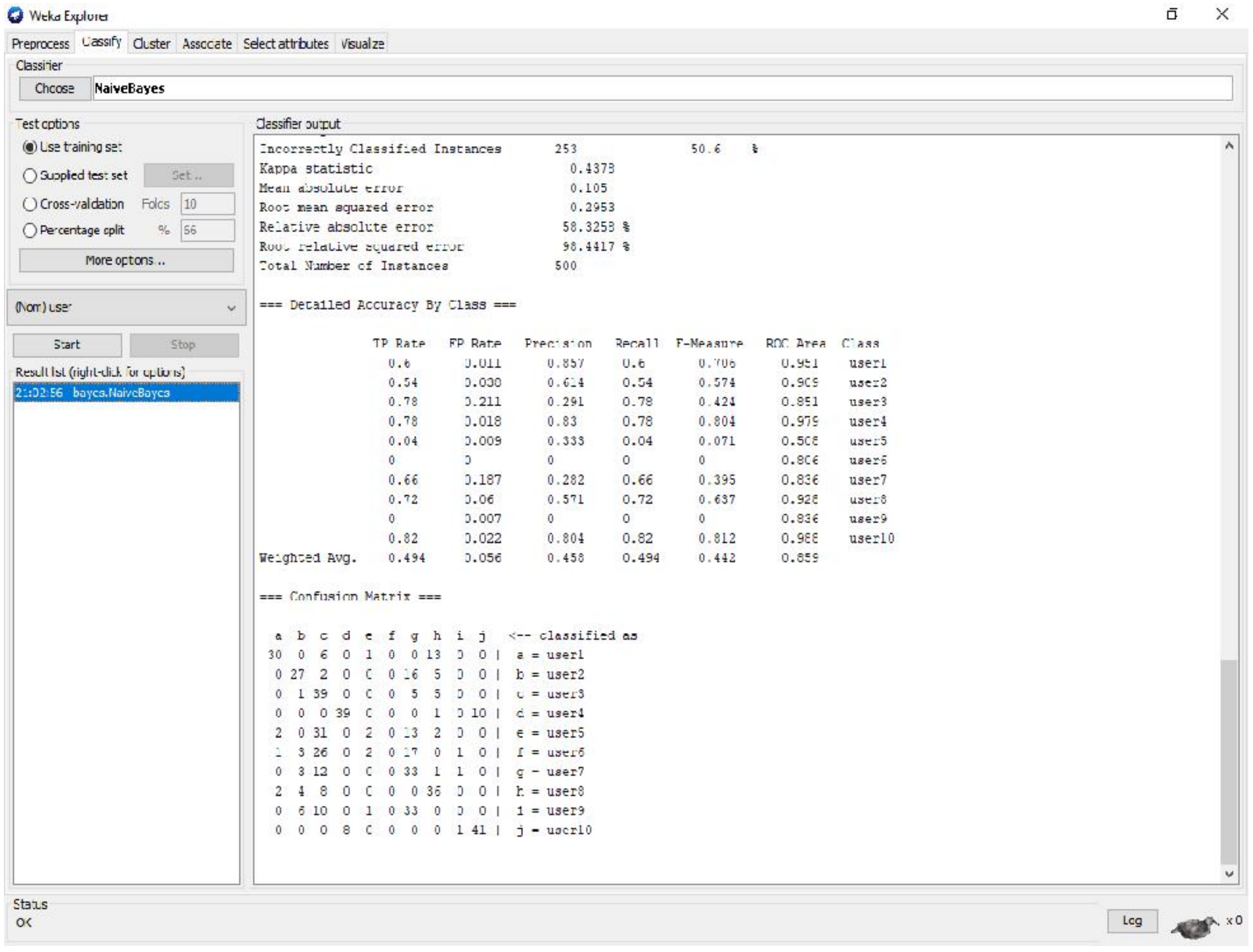
(Nom) user

16:59:59 - bayes.NaiveBayes

Status: OK

```
Correctly Classified Instances          31               57.4074 %
Incorrectly Classified Instances        23               42.5926 %
Kappa statistic                          0.4889
Mean absolute error                      0.1377
Root mean squared error                  0.3252
Relative absolute error                 49.5684 %
Root relative squared error             87.2608 %
Total Number of Instances               54

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
                 0.889     0.022     0.889      0.889     0.889       0.993     User1
                 0.111     0.178     0.111      0.111     0.111       0.719     User2
                 0.333     0.022     0.75       0.333     0.462       0.869     User3
                 0.778     0.178     0.467      0.778     0.583       0.874     User4
                 0.556     0.044     0.714      0.556     0.625       0.958     User5
                 0.778     0.067     0.7        0.778     0.737       0.983     User6
Weighted Avg.    0.574     0.085     0.605      0.574     0.568       0.899

=== Confusion Matrix ===

 a b c d e f   <-- classified as
 8 1 0 0 0 0 | a = User1
 1 1 0 4 1 2 | b = User2
 0 2 3 3 0 1 | c = User3
 0 0 1 7 1 0 | d = User4
 0 3 0 1 5 0 | e = User5
 0 2 0 0 0 7 | f = User6
```

Figure4.43: Naive Bayesclassifieroutput using Own Data



Classifier: NaiveBayes

(Nom) user

21:02:56 - bayes.NaiveBayes

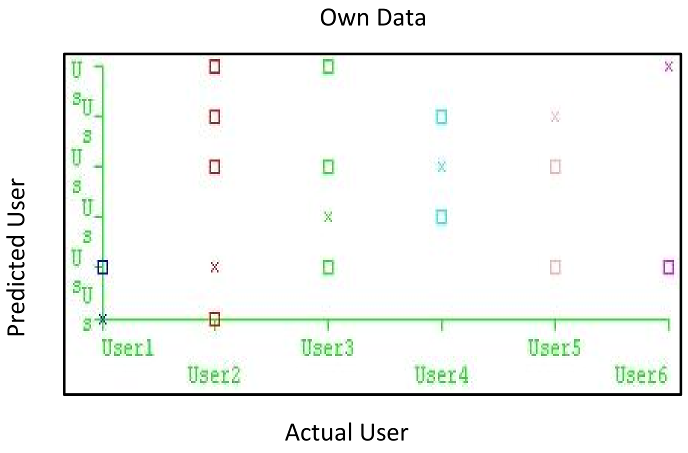
Folds 10

% 66

Status: OK

```
Incorrectly Classified Instances       253               50.6    %
Kappa statistic                          0.4378
Mean absolute error                      0.105
Root mean squared error                  0.2953
Relative absolute error                 58.3258 %
Root relative squared error             98.4417 %
Total Number of Instances              500

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.6       0.011     0.857     0.6       0.706      0.951    user1
                 0.54      0.030     0.614     0.54      0.574      0.909    user2
                 0.78      0.211     0.291     0.78      0.424      0.851    user3
                 0.78      0.018     0.83      0.78      0.804      0.979    user4
                 0.04      0.009     0.333     0.04      0.071      0.508    user5
                 0         0         0         0         0          0.806    user6
                 0.66      0.187     0.282     0.66      0.395      0.836    user7
                 0.72      0.06      0.571     0.72      0.637      0.928    user8
                 0         0.007     0         0         0          0.836    user9
                 0.82      0.022     0.804     0.82      0.812      0.988    user10
Weighted Avg.    0.494     0.056     0.458     0.494     0.442      0.859

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 30  0  6  0  1  0  0 13  0  0 |  a = user1
  0 27  2  0  0  0 16  5  0  0 |  b = user2
  0  1 39  0  0  0  5  5  0  0 |  c = user3
  0  0  0 39  0  0  0  1  0 10 |  d = user4
  2  0 31  0  2  0 13  2  0  0 |  e = user5
  1  3 26  0  2  0 17  0  1  0 |  f = user6
  0  3 12  0  0  0 33  1  1  0 |  g = user7
  2  4  8  0  0  0  0 36  0  0 |  h = user8
  0  6 10  0  1  0 33  0  0  0 |  i = user9
  0  0  0  8  0  0  0  0  1 41 |  j = user10
```

Figure4.44: Naive Bayes classifieroutput using Benchmark Data



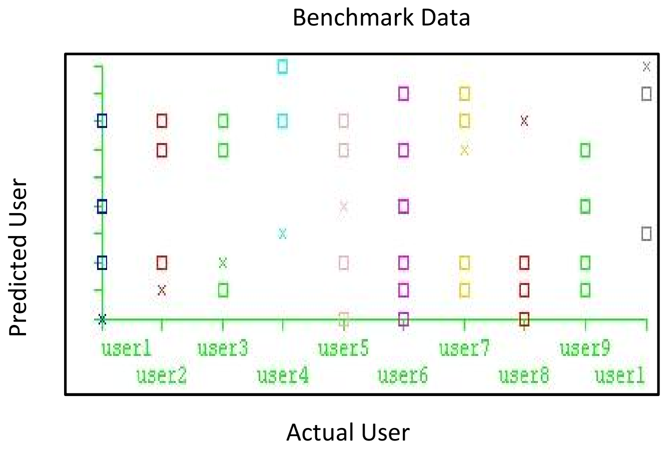
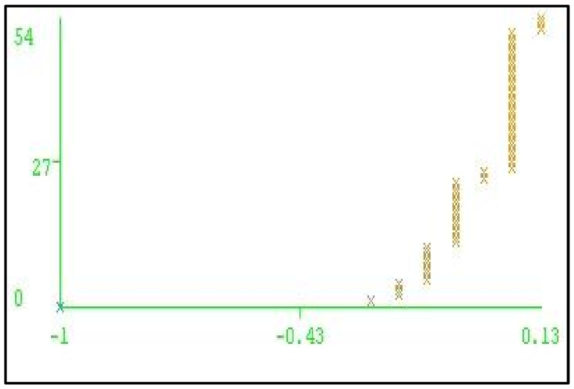


Figure 4.45: Visual classifier error of Naive Bayes

We have also compared the performance among algorithms using visual margin curve (Figure 4.46-4.48). This curve generates a plot by showing a prediction of difference for actual class and the highest probability predicted for the other class. From these curve we find points illustrating the prediction margin. The margin isdefined as the difference between the probability predicted for theactual class and the highest probability predicted for the otherclasses.

For kNN the margin is much close to 1. Here we see that prediction performance of kNN is the best.

Own Data



Benchmark Data

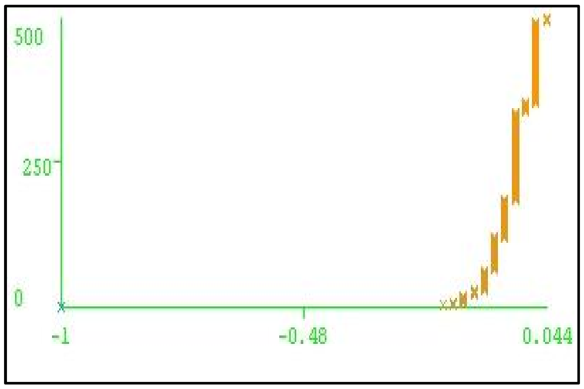
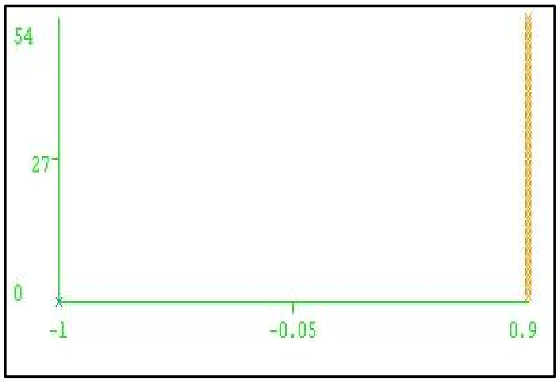


Figure 4.46: Margin curve of SVM

Own Data



Benchmark Data

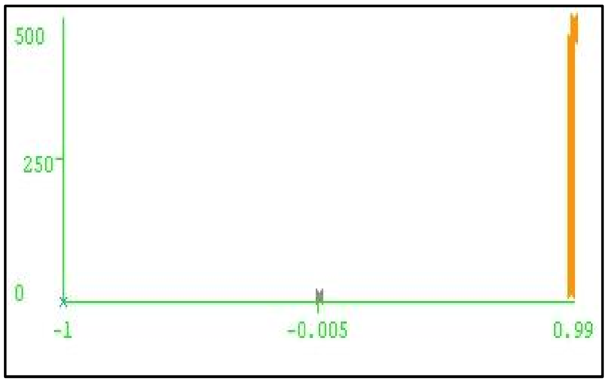


Figure 4.47: Margin curve of k nearest neighbor

Own Data

Benchmark Data

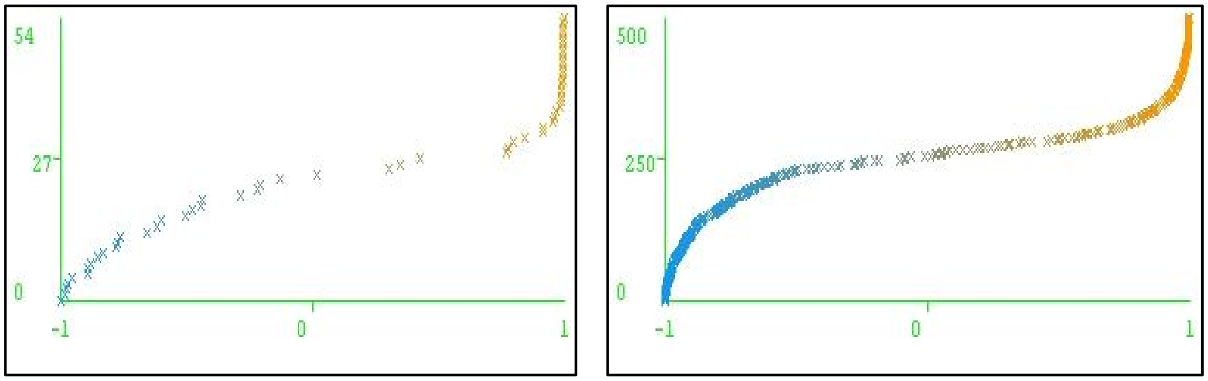


Figure 4.48: Margin curve of Naive Bayes

4.8 Comparative Analysis of Performance

Many methods and procedures are proposed since last of years for mouse movement user authentication. We have applied our proposed method on benchmark data. The data are collected from [45], which is freely available (see section 4.2.4). Wehave generated 12 features those we have proposed from the benchmark data and have tested our method with benchmark data. After the training and testing the following results are found. We have shown a comparative result with our own collected data and benchmark data in the table below.

Table 4.20: Performance of Proposed system

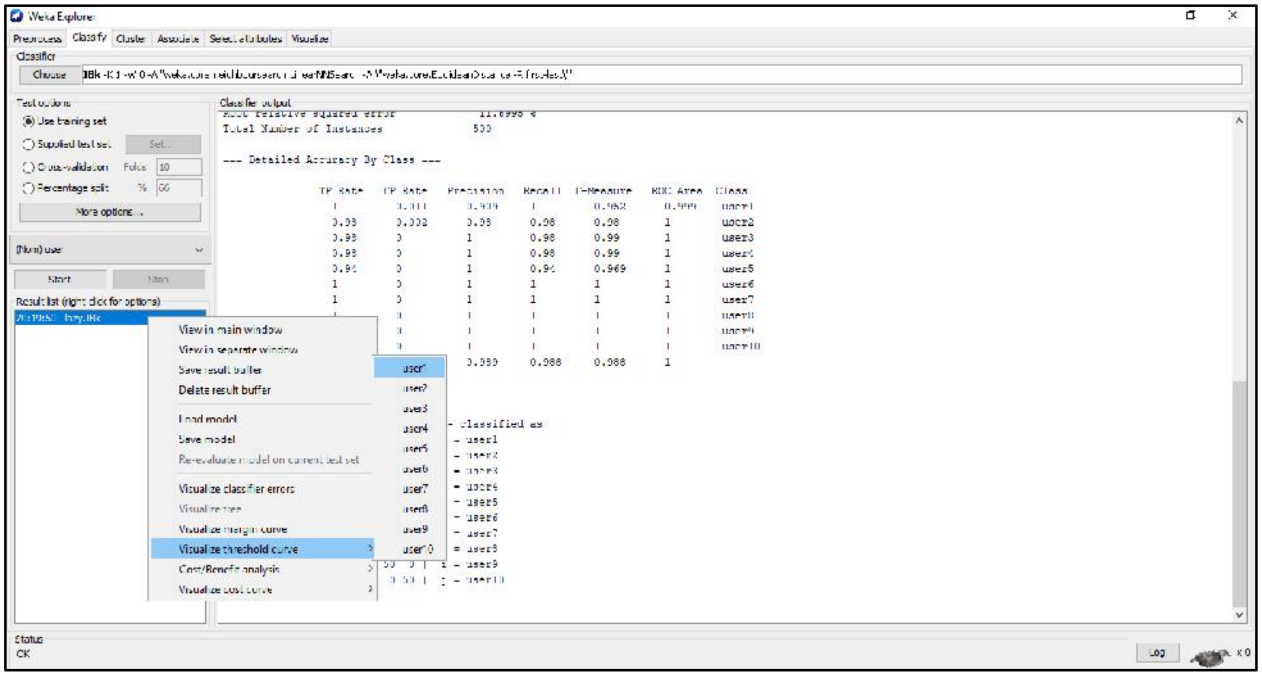
DataSource	Performance Criteria	SVM	k Nearest Neighbor	Naive Bayes
Own Data	Average ROC Area	0.795	1	0.899
	FAR	0.463	0	0.085
	FRR	0.093	0	0.426
Benchmark Data [44]	Average ROC Area	0.798	1	0.859
	FAR	0.077	0.001	0.056
	FRR	0.69	0.012	0.505

From the performance of the above table, it is assumed that our proposed method works well over the bench mark data. Result of benchmark data shows lower error rate than result of our own data and kNN classifier has the best performance. Average ROC area also represents the performance as describe in section 4.4. The accuracy of the test depends on how well the test separates the group being tested into those with and without the disease in question. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test on the other hand an area of .5 represents a worthless test. From the table 4.20 it is meant the in terms of ROC are the evaluation of classifiers shows best performance of kNN classifier.

To understand the performance of proposed system we have analyzed the ROC curve. Values of average ROC area are mentioned in table 4.20 which shows the comparative result of three classifier algorithms. Some ROC curves are shown below of these three algorithms. Though, we have used WEKA data mining tool in this research, so we have generated ROC curve from threshold curve of WEKA. For our own generated mouse movement we have taken six users to test the proposed system. So for three algorithms we have found18 ROC curves for every user of each algorithm. On the other hand whenever testing the system with Benchmark data, there were ten user. So we have found 30 ROC curves for every user of each algorithm.

The accuracy of the test depends on how well the test separates the group being tested into those with and without the disease in question. In this research this group is user. Its accuracy depends to make differentiate among the users. In respect to ROC curve the accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of 0.5 represents a worthless test. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system:

- 0.90-1 = excellent
- 0.80-.90 = good
- 0.70-.80 = fair
- 0.60-.70 = poor
- 0.50-.60 = fail



```
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
```

```
Root relative squared error             11.6995 %
Total Number of Instances              500

--- Detailed Accuracy By Class ---

                 TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                 1        0.011    0.909      1       0.952      0.999     user1
                 0.98     0.002    0.98       0.98    0.98       1         user2
                 0.98     0        1          0.98    0.99       1         user3
                 0.98     0        1          0.98    0.99       1         user4
                 0.94     0        1          0.94    0.969      1         user5
                 1        0        1          1       1          1         user6
                 1        0        1          1       1          1         user7
                 1        0        1          1       1          1         user8
                 1        0        1          1       1          1         user9
                 1        0        1          1       1          1         user10
                                   0.989      0.988   0.988      1

                                  <-- classified as
                                    - user1
                                    - user2
                                    - user3
                                    - user4
                                    - user5
                                    - user6
                                    - user7
                                    = user8
          50  0 |  1 - user9
           0 50 |  j - user10
```

Figure 4.49: Creating ROC curve using WEKA

We have seen that the ROC curves of Benchmark data shows better result. Area under these cases is near to1. So it is easily understood that the more data will show more accurate result. On the other hand K Nearest Neighbor algorithm also shows more accurate result in terms of ROC curve.

The figures below (4.50 to 4.73) present ROC curve of three classifiers. The curves are created for both our own generated data (own data) and Benchmark data. Though we have used WEKA data mining tool, so these below curves are also created using WEKA. WEKA creates the ROC curve in *false positive rate* against *true positive rate.*

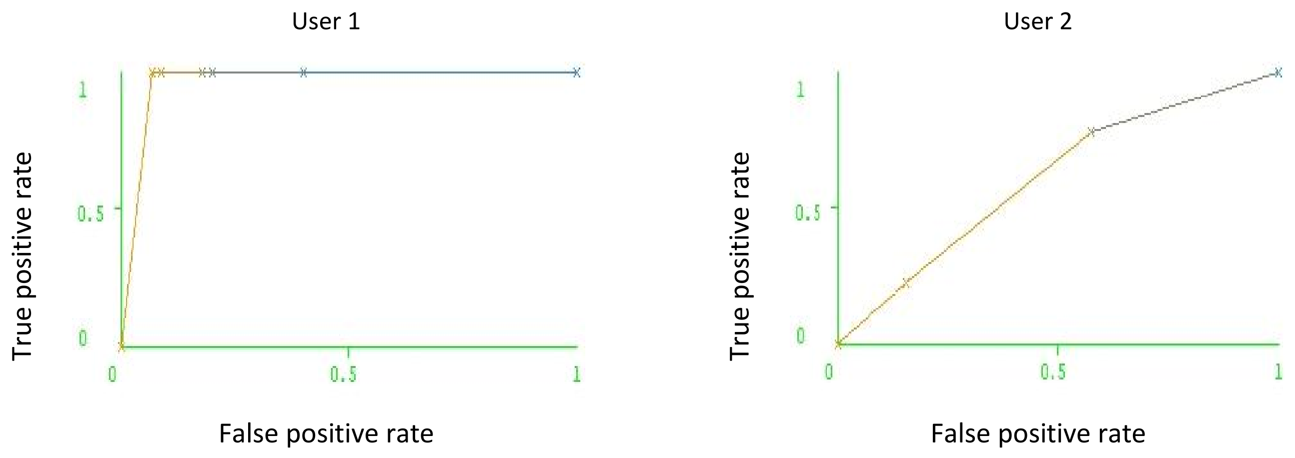


Figure 4.50: ROC curve of SVM using Own Data for User 1 & 2

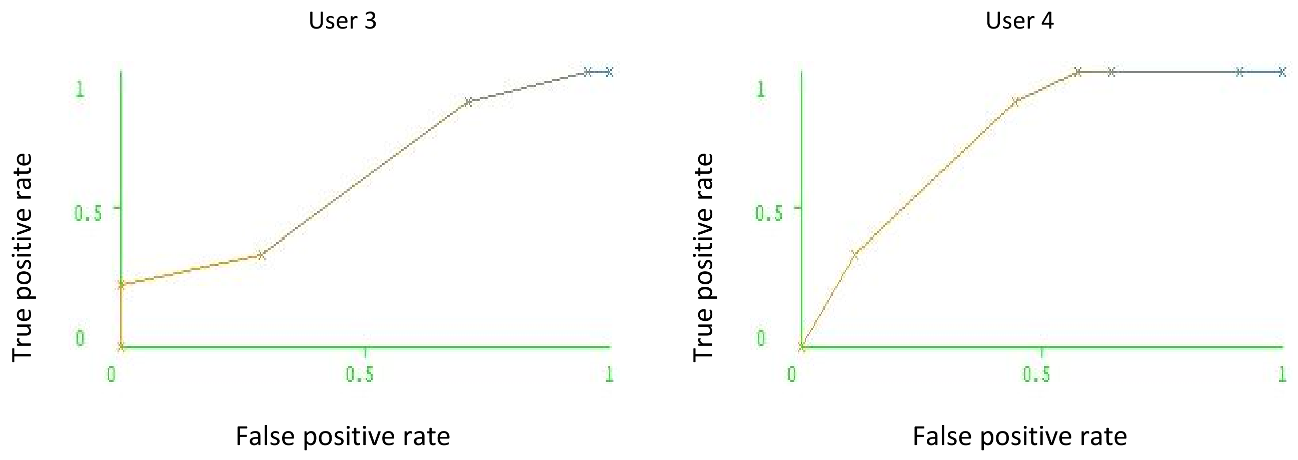
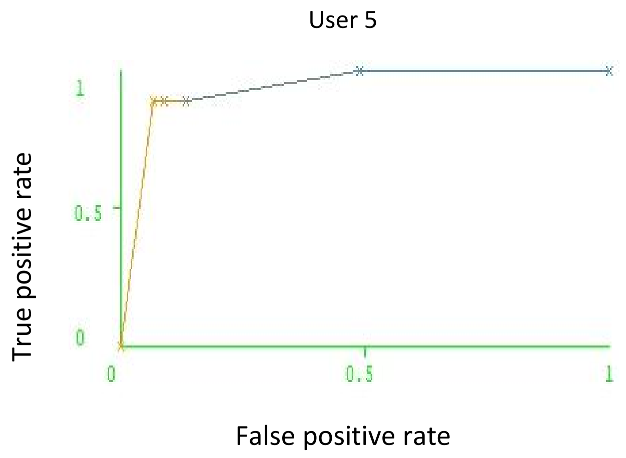


Figure 4.51: ROC curve of SVM using Own Data for User 3&4



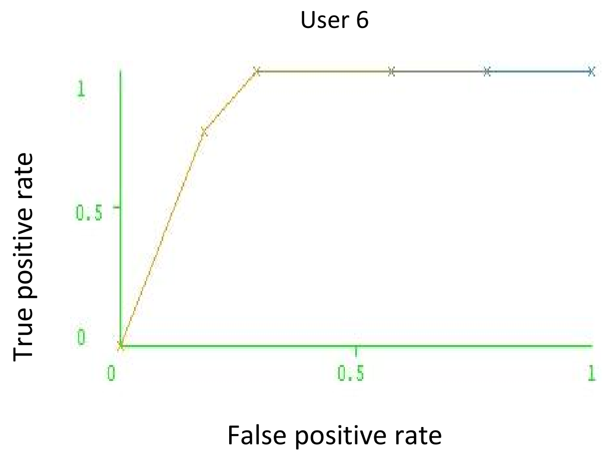
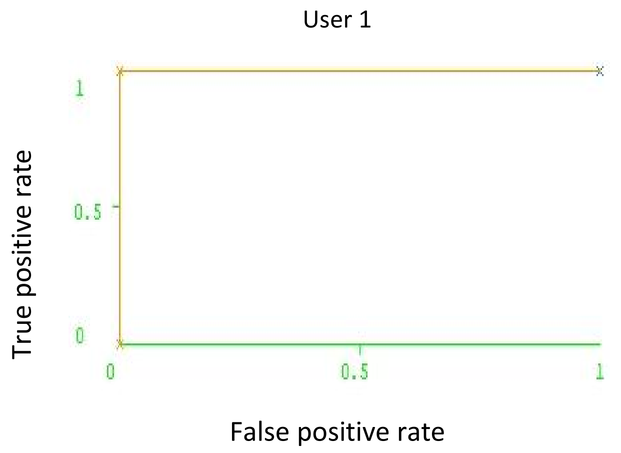


Figure 4.52: ROC curve of SVM using Own Data for User 5 & 6



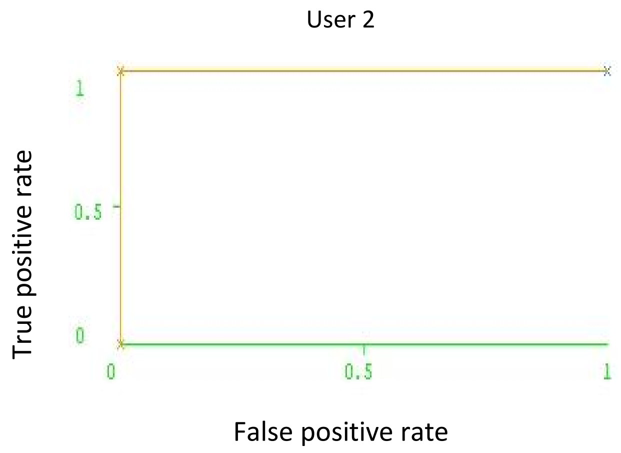
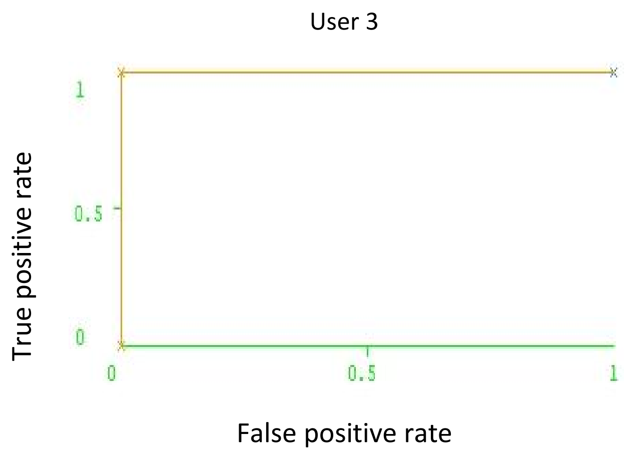


Figure 4.53: ROC curve of kNN using Own Data for User 1 & 2



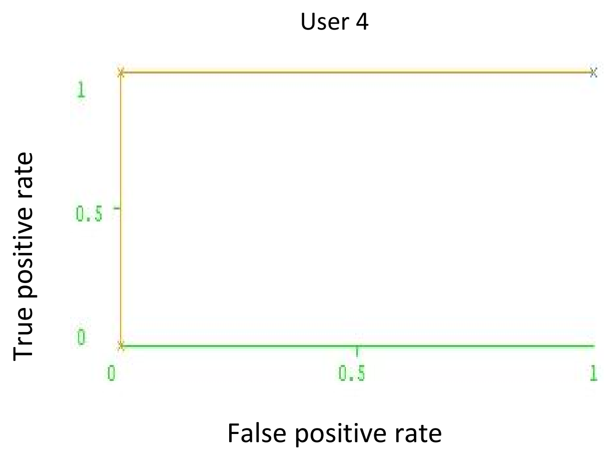


Figure 4.54: ROC curve of kNN using Own Data for User 3 & 4

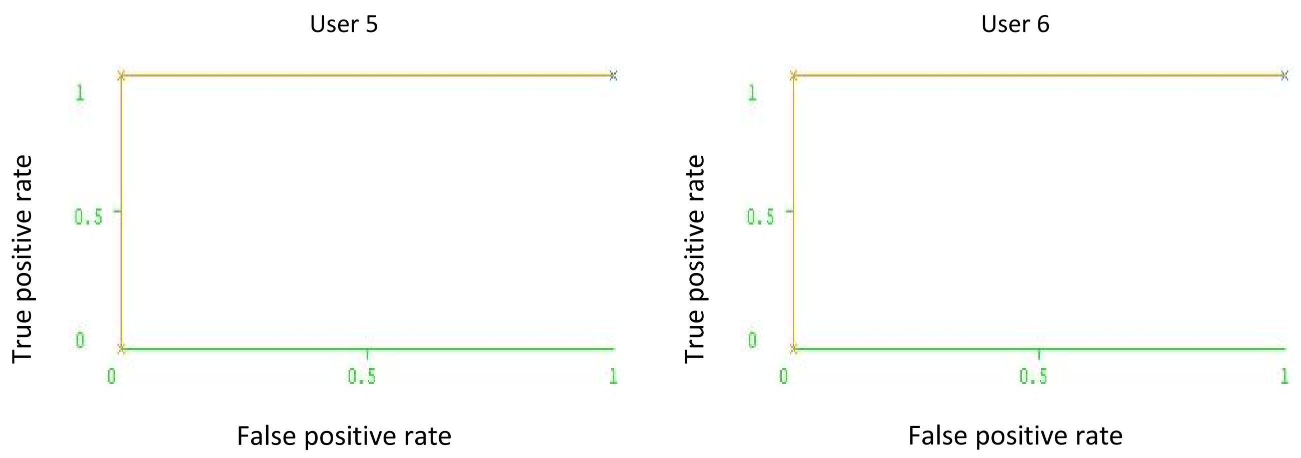


Figure 4.55: ROC curve of kNN using Own Data for User 5 & 6

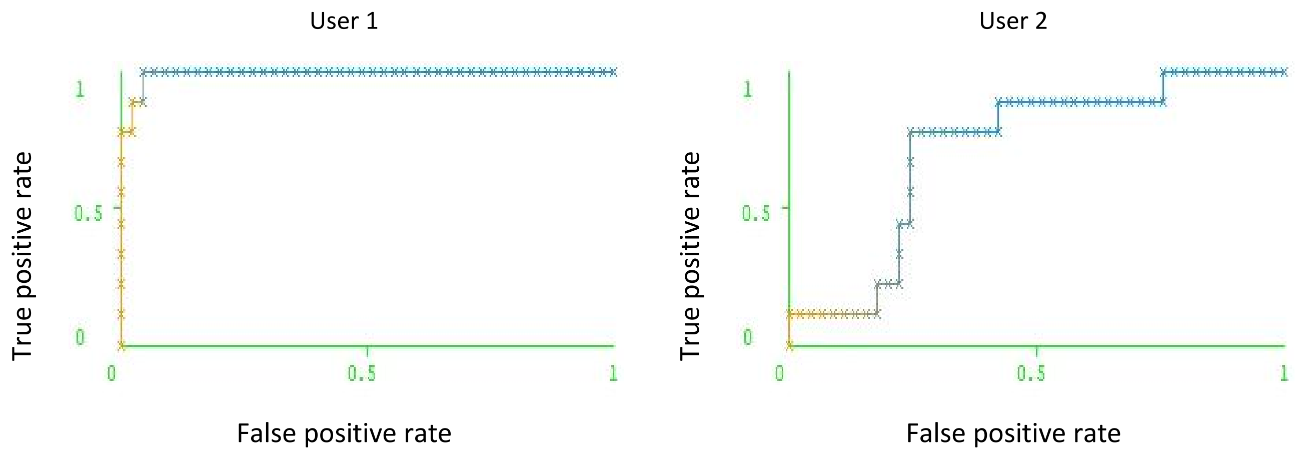


Figure 4.56: ROC curve of Naive Bayes using Own Data for User 1 & 2

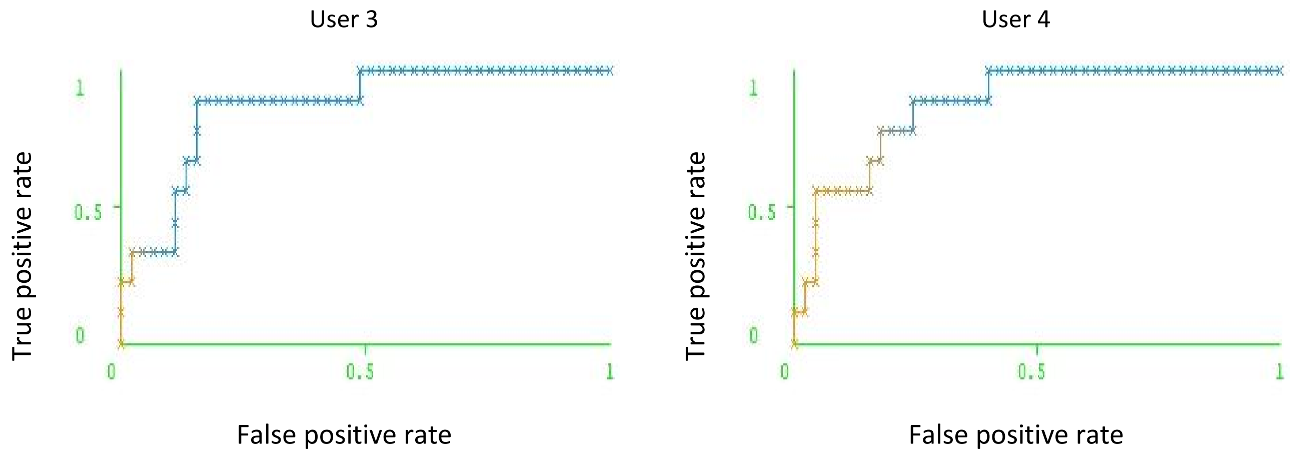


Figure 4.57: ROC curve of Naive Bayes using Own Data for User 3 & 4

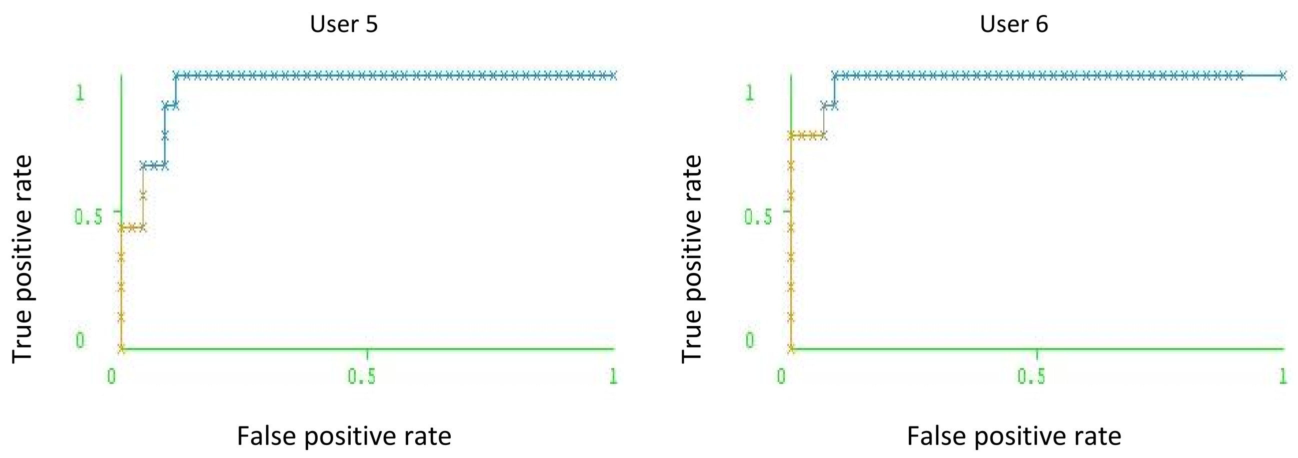


Figure 4.58: ROC curve of Naive Bayes using Own Data for User 5 & 6

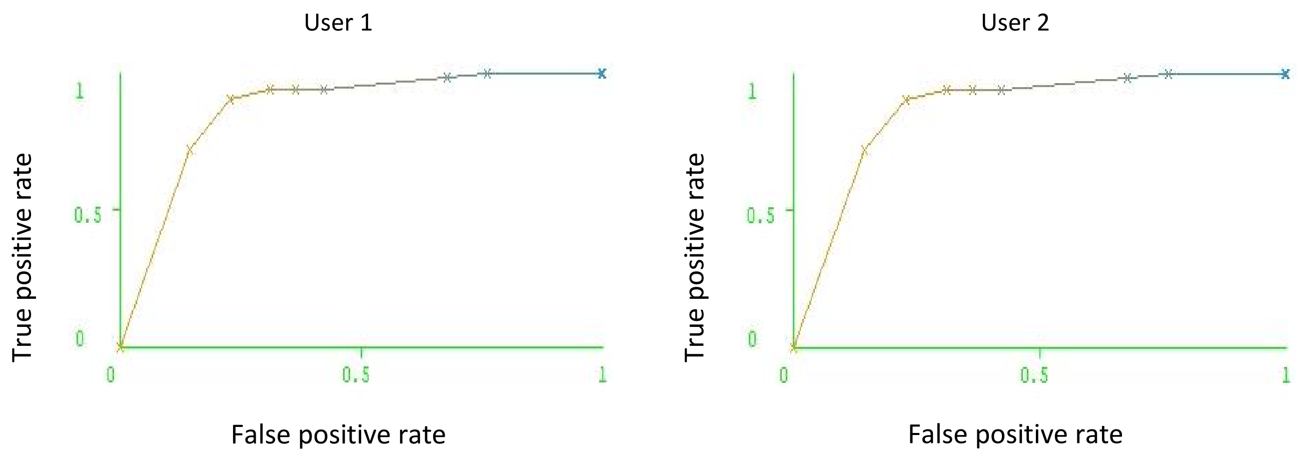


Figure 4.59: ROC curve of SVM using Benchmark Data for User 1&2

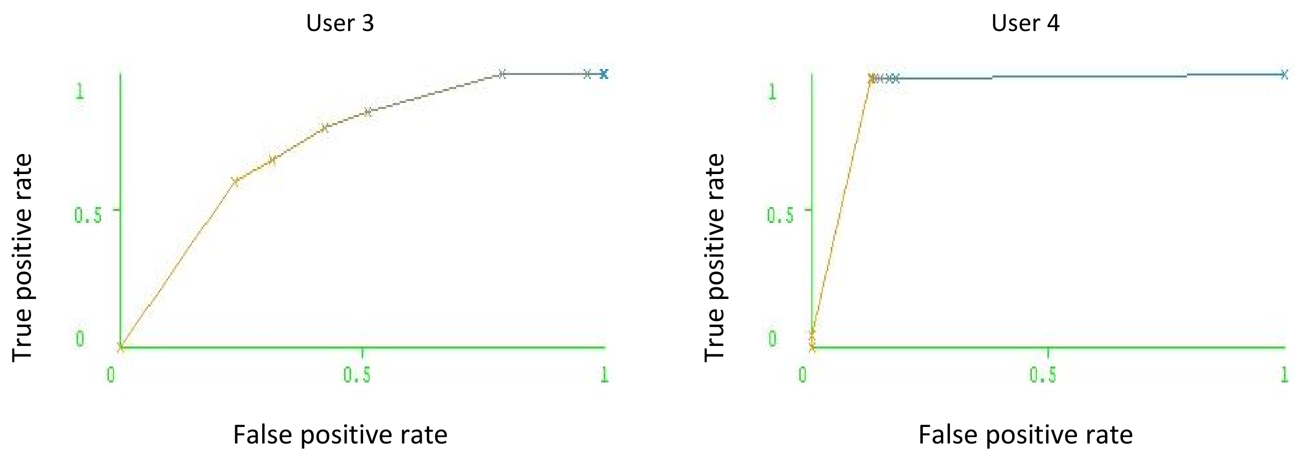


Figure 4.60: ROC curve of SVM using Benchmark Data for User 3&4

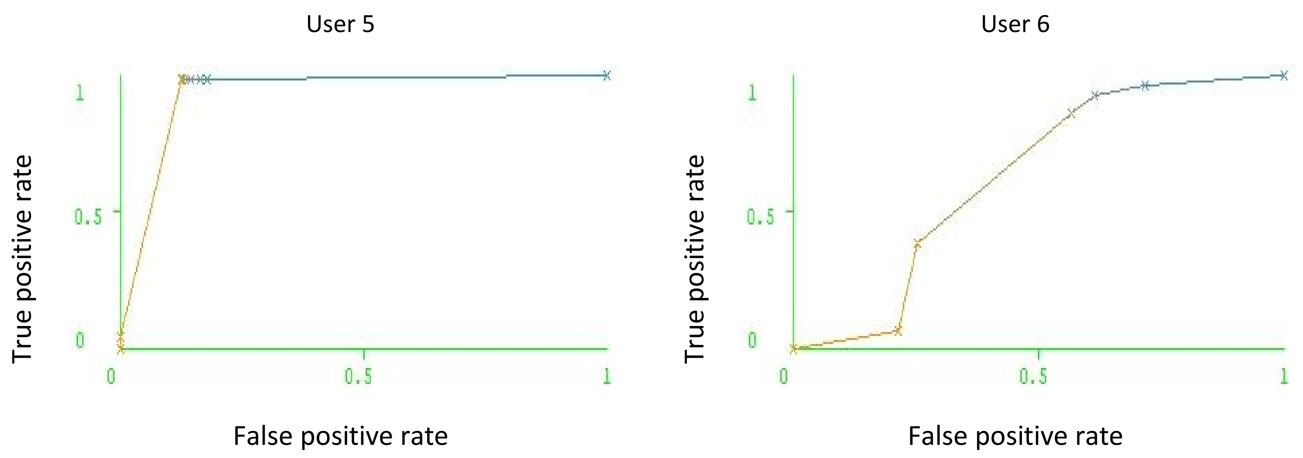


Figure 4.61: ROC curve of SVM using Benchmark Data for User 5&6

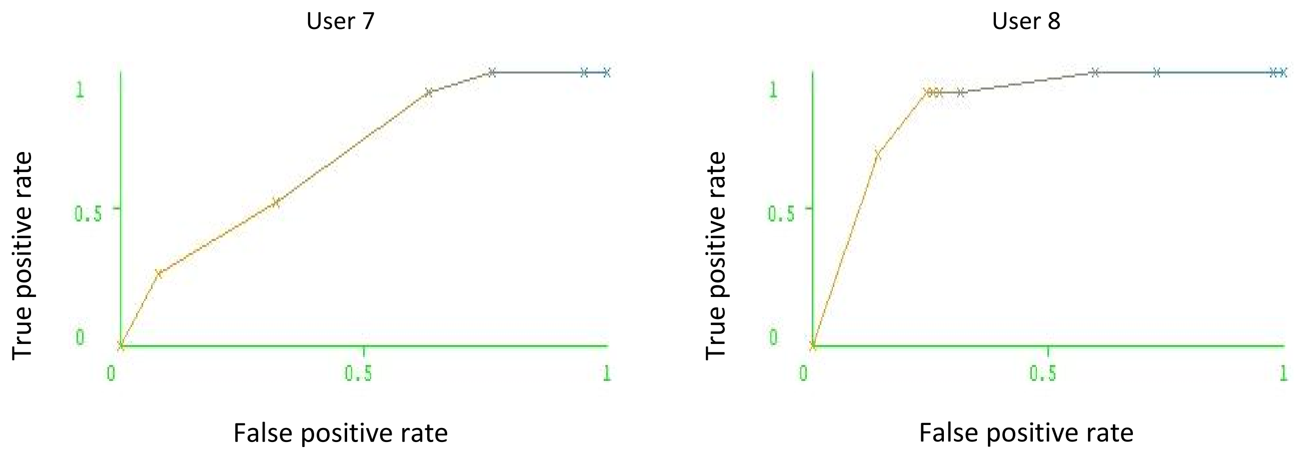


Figure 4.62: ROC curve of SVM using Benchmark Data for User 7&8

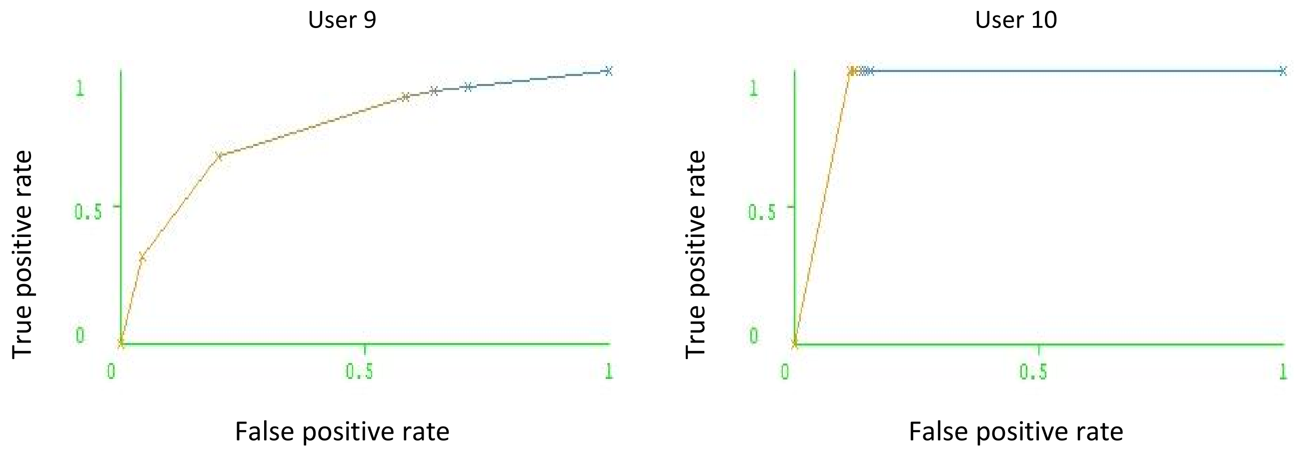


Figure 4.63: ROC curve of SVM using Benchmark Data for User 9&10

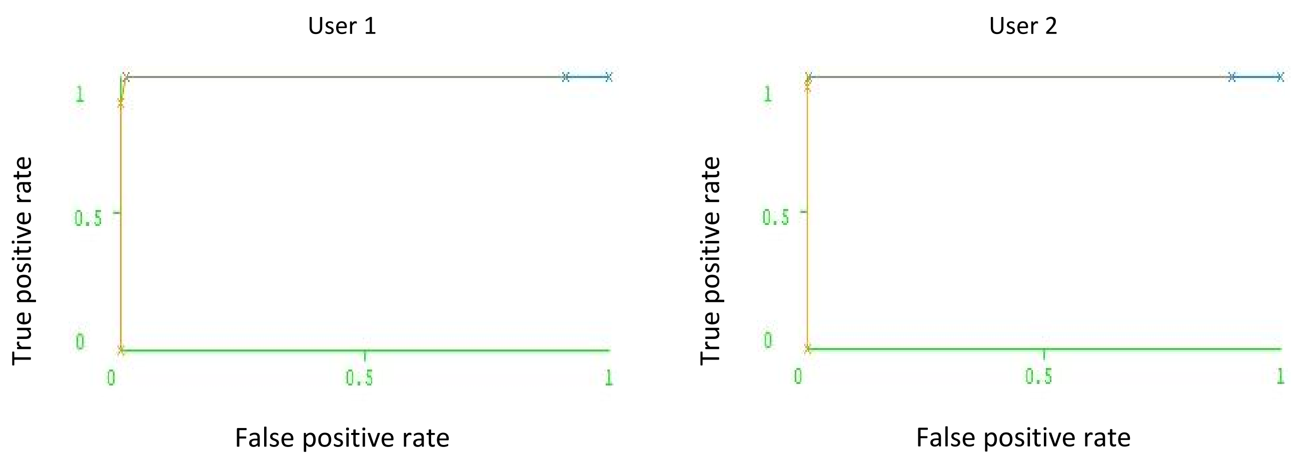


Figure 4.64: ROC curve of kNN using Benchmark Data for User 1&2

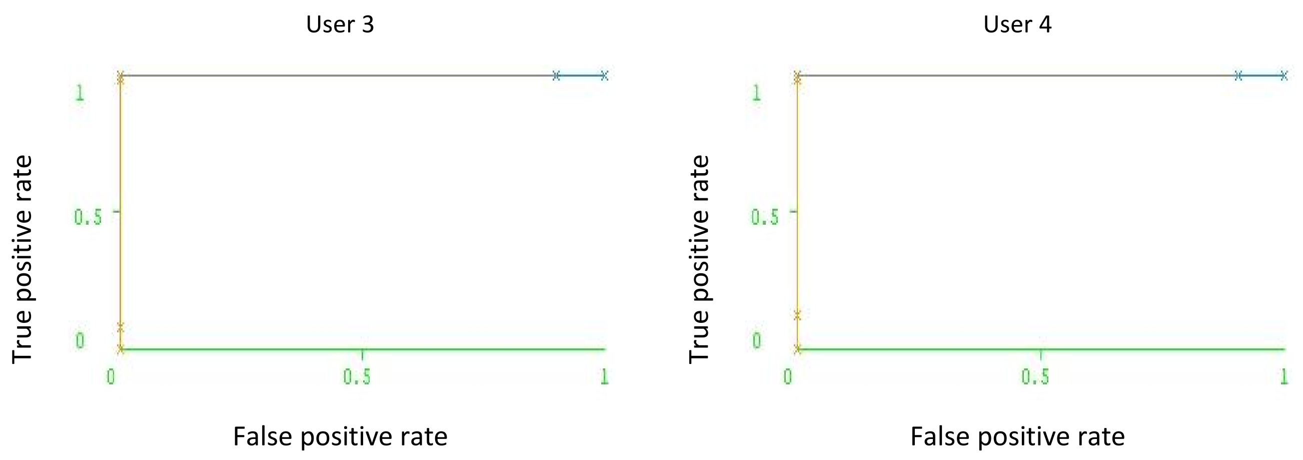


Figure 4.65: ROC curve of kNNusing Benchmark Data for User 3&4

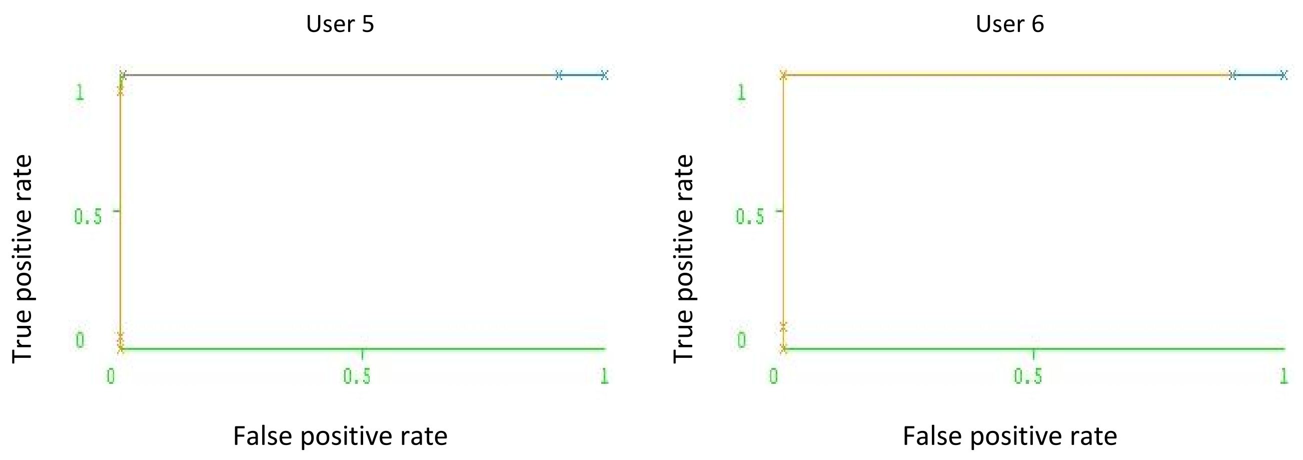


Figure 4.66: ROC curve of kNNusing Benchmark Data for User 5&6

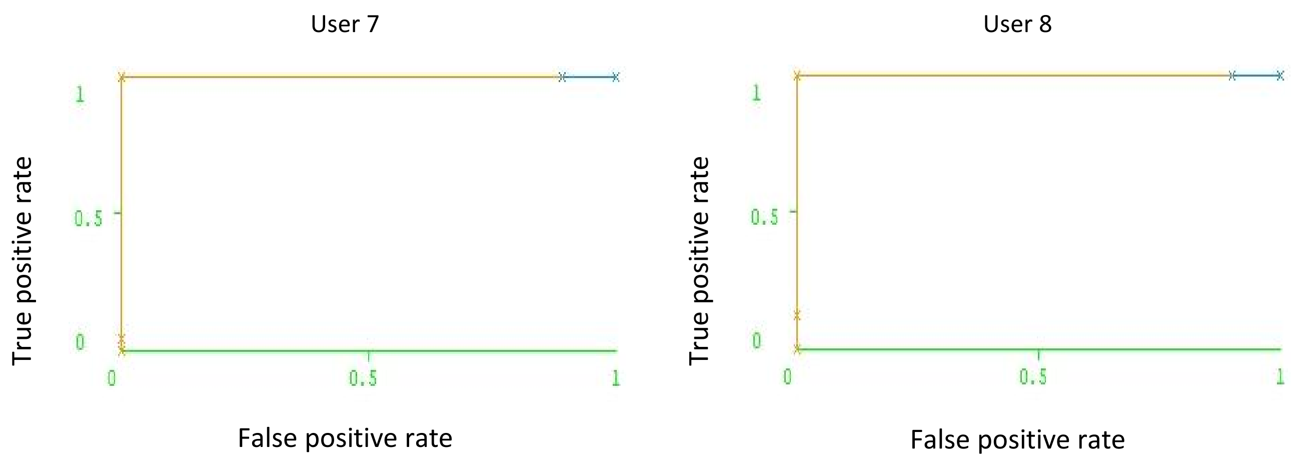


Figure 4.67: ROC curve of kNNusing Benchmark Data for User 7&8

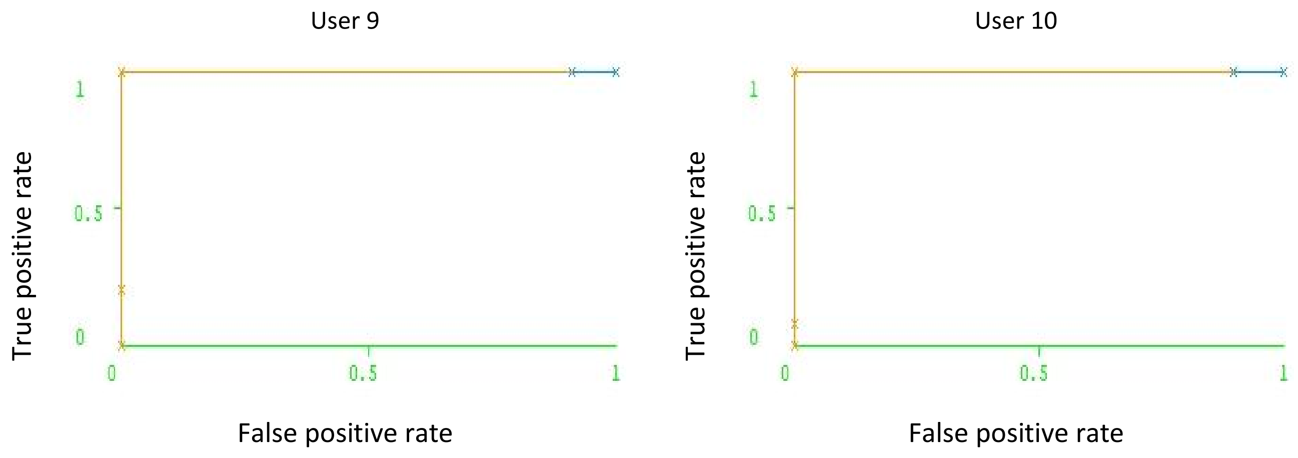


Figure 4.68: ROC curve of kNNusing Benchmark Data for User 9&10

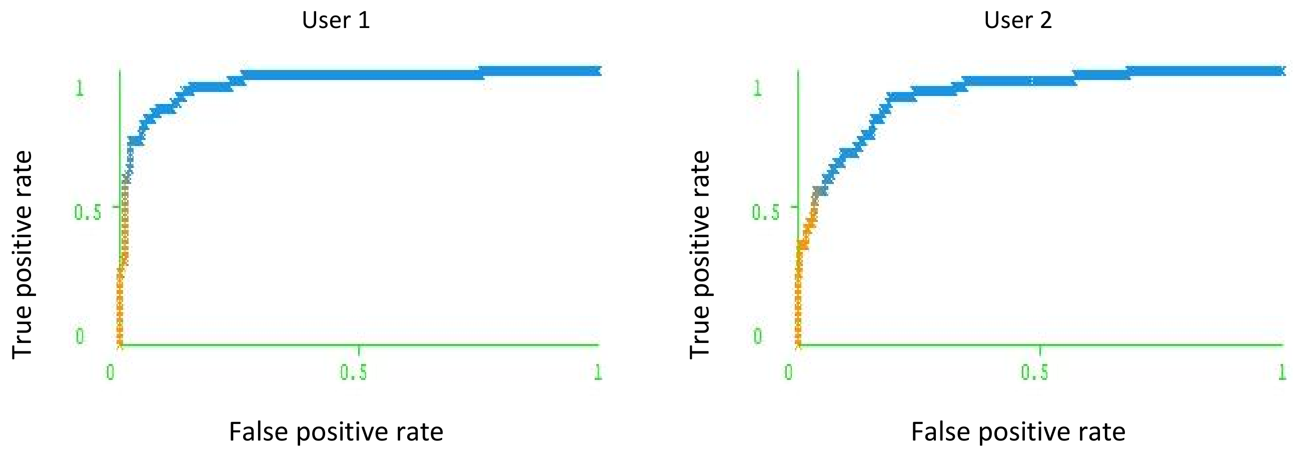
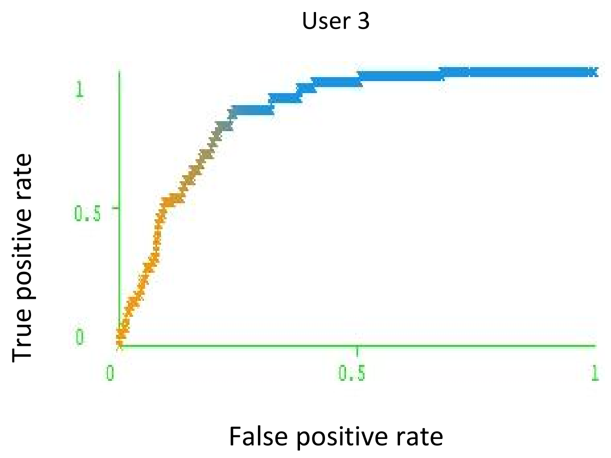


Figure 4.69: ROC curve of Naive Bayes using Benchmark Data for User 1&2



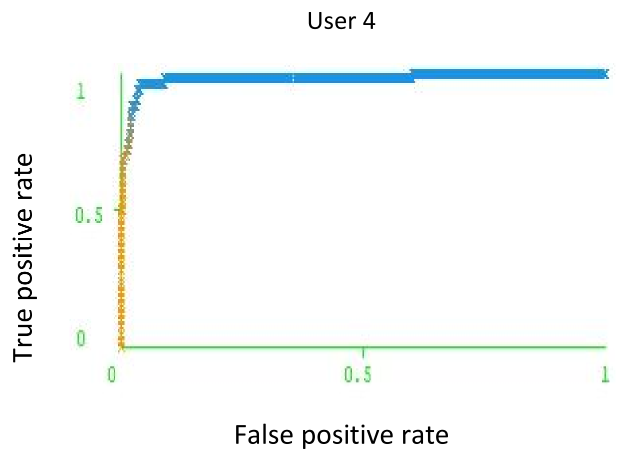
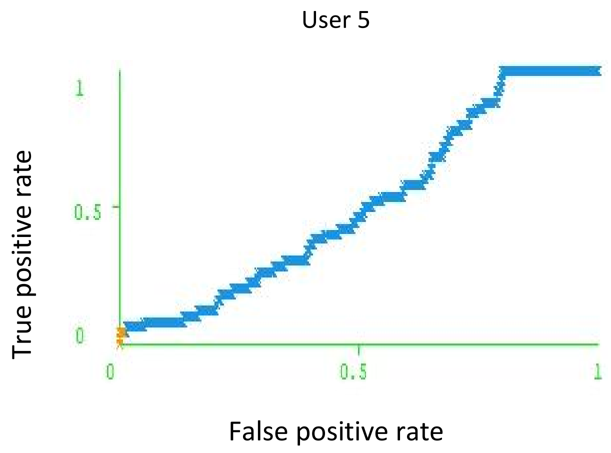


Figure 4.70: ROC curve of Naive Bayes using Benchmark Data for User 3&4



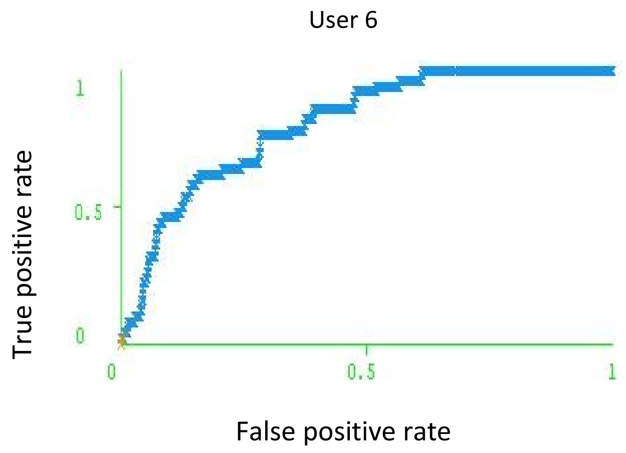
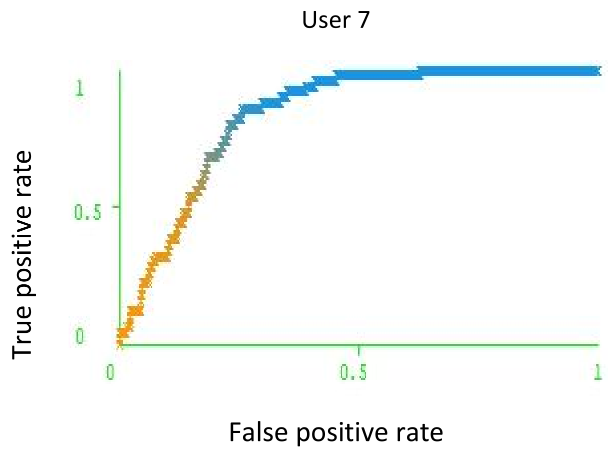


Figure 4.71: ROC curve of Naive Bayes using Benchmark Data for User 5&6



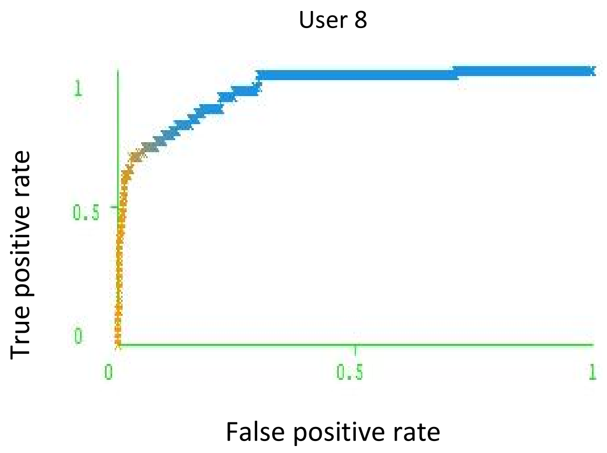
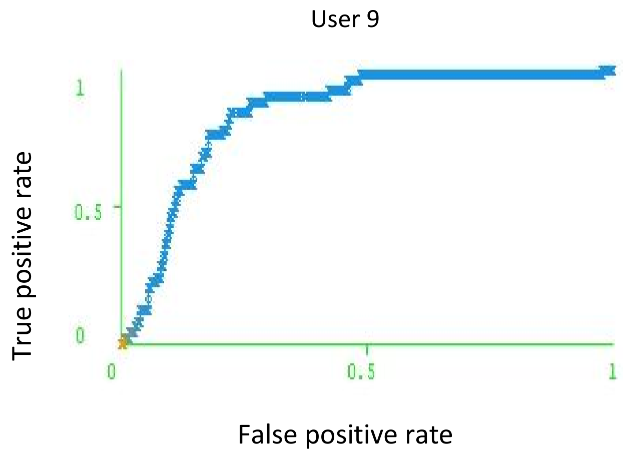


Figure 4.72: ROC curve of Naive Bayes using Benchmark Data for User 7&8



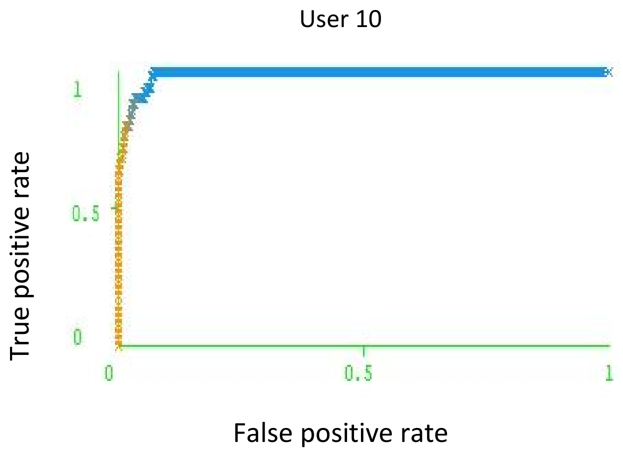


Figure 4.73: ROC curve of Naive Bayes using Benchmark Data for User 9&10

4.9 Comparison with Existing Work

Other researchers [4] also work with this benchmark data. Like S. Suganyaand others[4] have proposed Hopfield networkwith diffusion map method to identify user based on mouse movement. We have used same data set that they used for testing. From the data set they have extracted features: schematic features, holistic features and motor-skill features like average speed, average distance, mean standard deviation and mouse silence ratio, velocity, slope angle and curvature. They have also provided performance comparison of their proposed Hopfield network with one-class SVM and Original feature space. Among those their proposed system has showed the best performance. We have also compared result of their [4] method with our proposed method. Comparative results between [4] and our proposed system in this research are shown below.

Table 4.21:Comparing performance with related work using same data

Source	FAR (%)	FRR (%)
[4]	5.05	4.15
Proposed	1	1.2

By comparing our proposed system with [4] we have seen that our proposed system has better performance. Performance also increased by adding more features likes number of points in trajectory, delay time, number of delay, number of action, standard division of trajectory length, standard division of slope, slope to slope differences, number of curvature, curvature of trajectory, number of changes in vertical position and number of changes in horizontal position those are proposed in this research.

Chapter – 5
Conclusions

User authentication is a prime challenge of information system. This thesis presents a user authentication system using mouse movement data which is simple, robust and independent of user and avoids any additional device in this regard.

5.1 Contribution

This system collected real time mouse movement data as well as benchmark data set [44] to test the proposed user authentication system.

The collected data are sampled into blocks which depend on number of action or time intervals. To find and analyze these mouse movement behavior and pattern, we have generated twelve features from sampled blocks. Among these twelve features we have introduce seven new features with five existing features. Proposed new features are: Number of Delay, Standard Deviation of the Trajectory Length, Standard Deviation of Slope, Standard Deviation of Difference Between Each of Slopes, Number of Curvatures, Number of Changes in Horizontal Position and Number of Changes in Vertical Position; and the existing features are: Number of Points in the Trajectory, Delay Time, Number of Action, Total Length of Trajectory and Curvature of Trajectory.

The system is trained and tested using collected real time data and benchmark data. To validatethe proposed system the system uses three different classifiers: SVM, k Nearest Neighbor and Naive Bayes. From each classifier percentage of user prediction, ROC curve, FAR, FRR, margin curve and visual classifier error are generated to check the performance of classifiers. It is found that k Nearest Neighbor classifier shows the best performance. Output result of this thesis is compared with similar research as S. Suganya and others has done [4] and the proposed method shows better result.

The proposed user authentication system identifies user based on user mouse movement data which is collected from mouse movement patterns. So it cannot be stolen or hacked like conventional authentication system. One cannot share the hidden mouse movement characteristics to others so the proposed approach ensures the actual user authentication.

The major advantage of using this proposed authentication system is to use it in online environment as an alternative of usingpassword/PIN code, fingerprint, card punch etc. The proposed mouse movement behavioral authentication is so strong to handle in any public engagement occurrences. Unethical employees cannot deny their responsibilities saying that any other used their action in their absence because in the proposed mouse movement authentication system, users must need to be present to log in.

5.1Limitation and Future Improvements

The one of major drawback of biometric behavioral authentication system is this take more time to register a user. First of all user need to provide some example of biometrical character to register the system. When the user will try to log in, the system will find the similaritybetween biometrical characteristics of login session and biometrical characters which were given before in registration session.

Another drawback of the system is that, it needs to be trained strongly before user identification and authentication. So the system needs more mouse movement data to detect a user. Otherwise, it may falsely identify a wrong user or deny access to a valid user. It can be used effectively to track suspicious user actions, completing online jobs, online exam, e-commerce signature transactions etc.

There are some extra works to research with mouse movement authentication system. It needs a large number of human users for testing. Arranging the large numbers of interested volunteers is like a challenge. User interface is also a challenging issue. The test process may take a long session like days or many hours. The effectiveness of biometric system can be tested for different environments also. To be the more accuracy of the method, more features are required.Future experiments or further development of this method would build depending on raw data and features. A common framework and user interface is needed to be familiar of mouse dynamic authentication.

In future this system is needed to be tested for different IT environments.The model can be further improved if we add some more suitable features. Increasing the number of features or parameters will help to reduce false acceptance rate as well as false rejection rate.

Some researchers have focused on mouse speed, some focused on mouse movement angle some focused on mouse action, some focused on pressure on mouse button etc. It hasn’t been analyzed at whole.For further development and more accurate result we need to combine all related features.

5.2ConcludingandRemarks

A systemof user verification and authentication based on mouse activity is introduced in this research.This work can improve computer security. Day by day numbers of online users are increasing. Computer networks are spreading more and more. Most of our regular tasks are online based. So the security and user identity are becoming issues of importance discussion and solution. In this regard our interests lie in the development of integrated approaches that provide simple,robust and resilient security solution.This thesis provides such solution to authenticate user based on mouse movement data. The proposed method removes using extra devices in order to authenticate or detect a user. As well as it reduces cost. This system is capable to detect user continuously.

Many authentication technologies has been created, synthesized and replicated over a last of years. They have been improved and made available in the market now. But, mouse dynamics is a quite a new concept.

References

[1] Ross A.J.Everittand Peter W.McOwan; “Java-Based Internet Biometric Authentication System”; *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol.25, No.9, September2003.

[2] Ahmed Awad E. Ahmed and IssaTraore, Member, IEEE; “A New Biometric Technology Basedon Mouse Dynamics”; *IEEE Transactions on dependable and secure computing*, Vol.4, No.3, July-September 2007.

[3] Mounashree H C, Nancy M R,NavyaM,NirupamaBLinganagoudar,Mangala C N; “Mouse Gesture for Authentication by Hidden Markov Model”; *International Journal of Computer, Information Technology & Bioinformatics (IJCITB)*, ISSN: 2278-7593, Volume-2, Issue-2, 2014.

[4] S. Suganya, G. Muthumari, and C. Balasubramanian; “Improving the Performance of Mouse Dynamics Based Authentication Using Machine Learning Algorithm”; *International Journal of Innovation and Scientific Research*; ISSN 2351-8014 Vol. 24 No. 1 Jun. 2016, pp. 202-209.

[5] Mr. Vishal S. Patil; Prof. P.R.Devale; “Biometric Validation by Storing different Patterns using Mouse Gesture Signatures”; *International Journal on Recent and Innovation Trends in Computing and Communication*; Volume: 3 Issue: 4; April 2015.

[6] Zach Jorgensen and Ting Yu; “On Mouse Dynamics as a Behavioral Biometric for Authentication”; *ASIACCS* ’11, March 22–24, 2011, Hong Kong, China. Pp476-482.

[7] AnandMotwani, Raina Jain, JyotiSondhi; “A Multimodal Behavioral Biometric Technique for User Identification using Mouse and Keystroke Dynamics”; *International Journal of Computer Applications* (0975 – 8887); Volume 111 – No 8, February 2015.

[8] Raina K. Jain, SanchetiShital D, PansareJyoti T, KawaleShubhangi K; “Mouse and Keystroke Based Behavioural Biometrics: Techniques, Problems and Solutions”; *International Journal of Modern Trends in Engineering and Research* ; e-ISSN No.:2349-9745, Date: 2-4 July, 2015.

[9] NazirahAbd Hamid, SuhailanSafei, SitiDhalilaMohdSatar, SuriayatiChuprat and Rabiah Ahmad; “Randomized Mouse Movement for Behavioral Biometric Identification,” *International Journal of Interactive Digital Media*, Vol. 1(2), ISSN 2289-4098, e-ISSN 2289-4101, 2013.

[10] G. Muthumari, R. Shenbagaraj, M. BlessaBinolin Pepsi; “Mouse Gesture Based Authentication Using Machine Learning Algorithm,” 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)

[11] MajaPusara, Carla E. Brodley; “User Re-Authentication via Mouse Movements”; VizSEC/DMSEC'04, October29,2004, Washington, DC, USA.Copyright 2004 ACM 1-58113-974-8/04/0010.

[12] B J Gorad, D.V Kodavade, “User Identity Verification Using Mouse Signature”; IOSR Journal of Computer Engineering (IOSR-JCE); e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 4 (Jul. - Aug. 2013), PP 33-36

[13] A. Kaklauskas, E. K. Zavadskas, M. Seniut, M. Krutinis, G. Dzemyda, S. Ivanikovas, V. Stankevič, Č. Šimkevičius, A. Jaruševičius; “Web-Based Biometric Mouse Decision Support System For User’s Emotional And Labour Productivity Analysis”; The 25th International Symposium on Automation and Robotics in Construction; June 26-29, 2008; PP-69-75

[14] Youssef Nakkabi, IssaTraoré and Ahmed AwadE.Ahmed; “Improving Mouse Dynamics Biometric Performance Using Variance Reduction via Extractors with Separate Features”; *IEEE Transactions On Systems, Man, And Cybernetics — Part A: Systems And Humans*, Vol.40, No.6, November 2010.

[15] Govardhini.S, Kowshika.A; “An Effective User Recognition Using Mouse Gesticulation”; *International Journal of Advanced Research inComputer Science and Software Engineering*; Volume 3, Issue 11, November 2013; ISSN: 2277 128X; pp 456 – 460.

[16] Haitao Tang and Wang Mantao; “User Identity Authentication Based on the Combination of Mouse and Keyboard Behavior”; *International Journal of Security and Its Applications*; Vol. 10, No. 6 (2016) pp.29-36.

[17] Chethan D C, Mr.Sundaresh M P; “Design and Development of a Biometric System Using Mouse Gesture Dynamics,” *International Journal of Advanced Research in Computer Engineering & Technology* (IJARCET), 2014.

[18] Cheng-Jung Tsai, Ting-Yi Chang, Yu-Ju Yang, Meng-Sung Wu, Yu-Chiang Li; “An Approach For User Authentication On Non-Keyboard Devices Using Mouse Click Characteristics and Statistical-Based Classification”; *International Journal of InnovativeComputing, Information and Control*, ISSN 1349-4198, Volume 8, Number 11, November 2012 pp. 7875-7886.

[19] Saurabh Singh, Dr. K.V.Arya; “Mouse Interaction based Authentication System by Classifying the Distance Travelled by the Mouse”; *International Journal of Computer Applications* (0975 – 8887), pp-45-48, Volume 17– No.1, March 2011.

[20] ShivaniHashia, “Authentication by Mouse Movements”, San Jose State University.

[21] Dr. Jien (Morris) Chang, Mr. Kuan-Hsing Ho and Mr. Chi-Chen Fang, Capturing Cognitive Fingerprints from Keystroke Dynamics for Active User Authentication, all of the Department of Electrical and Computer Engineering, Iowa State University of Science and Technology.

[22] Mounashree H C, Nancy M R, Navya M, NirupamaBLinganagoudar, Mangala C N; "Mouse Gesture for Authentication by Hidden Markov Model"; International Journal of Computer, Information Technology & Bioinformatics (IJCITB),ISSN: 2278-7593, Volume-2, Issue-2, 2014.

[23] Nan Zheng, Aaron Paloski, and Haining Wang; An Efficient User Verification System via Mouse Movements; Department of Computer Science, The College of William and Mary Williamsburg, VA 23185, USA;

[24] Joseph S. Valacich, Jeffrey L. Jenkins, Jay F. Nunamaker, Jr, Salim Hariri, John Howie; Identifying Insider Threats through Monitoring Mouse Movements in Concealed Information Tests; University of Arizona.

[25] G. Muthumari, R. Shenbagaraj, M. BlessaBinolin Pepsi; "Authentication of User Based On Mouse-Behavior Data Using Classification"; International Journal of Innovative Research in Science, Engineering and Technology; Volume 3, Special Issue 3, March 2014

[26] Chethan D C, Mr.Sundaresh M P; Design and Development of a Biometric System Using Mouse Gesture Dynamics; International Journal of Advanced Research in Computer Engineering & Technology (IJARCET).

[27] Mounashree H C, Nancy M R,NavyaM,NirupamaBLinganagoudar,Mangala C N; MouseGesture for Authentication by Hidden Markov Model; International Journal of Computer, Information Technology & Bioinformatics (IJCITB), ISSN: 2278-7593, Volume-2, Issue-2.

[28] Chee-Hyung Yoon, Daniel Donghyun Kim; User Authentication Based On Behavioral Mouse Dynamics Biometrics; Department of Computer Science, Stanford, Stanford, CA 94305.

[29] Cheng-Jung Tsai, Ting-Yi Chang, Yu-Ju Yang, Meng-Sung Wu, Yu-Chiang Li; An Approach For User Authentication On Non-Keyboard Devices Using Mouse Click Characteristics and Statistical-Based Classification; International Journal of InnovativeComputing, Information and Control, ISSN 1349-4198, Volume 8, Number 11, November 2012 pp. 7875-7886.

[30] Adam Weiss, Anil Ramapanicker, Pranav Shah, Shinese Noble and Larry Immohr; Mouse Movements Biometric Identification: A Feasibility Study; Faculty Research Day, CSIS, Pace University, May 4th, 2007.

[31] NkemAjufor, Antony Amalraj, Rafael Diaz, Mohammed Islam, Michael Lampe; Refinement of a Mouse Movement Biometric System; Ivan G Seidenberg School of CSIS, Pace University, 1 Martine Ave, White Plains, NY, 10606, USA.

[32] MirkoStanić ; Continuous User Verification Using Mouse Dynamics; Agency for Science and Higher EducationZagreb, Croatia.

[33] Saurabh Singh, Dr. K.V.Arya; Mouse Interaction based Authentication System by Classifying the Distance Travelled by the Mouse; International Journal of Computer Applications (0975 – 8887), pp-45-48, Volume 17– No.1, March 2011.

[34] Zach Jorgensen and Ting Yu; On Mouse Dynamics as a Behavioral Biometric for Authentication; ASIACCS ’11, March 22–24, 2011, Hong Kong, China. Pp476-482.

[35] MajaPusara, Carla E. Brodley; User Re-Authentication via Mouse Movements; VizSEC/DMSEC'04, October29,2004, Washington, DC, USA.Copyright 2004 ACM 1-58113-974-8/04/0010.

[36] Govardhini.S, Kowshika.A; An Effective User Recognition Using Mouse Gesticulation; International Journal of Advanced Research inComputer Science and Software Engineering; Volume 3, Issue 11, November 2013; ISSN: 2277 128X; pp 456 – 460.

[37] HediehZandikarimi, Frank Lin, Celia Carlos, Justin Correa, Phil Dressner, and Vinnie Monaco; Design of a Mouse Movement Biometric System to Verify the Identity of Students Taking Multiple-Choice Online Tests; Seidenberg School of CSIS, Pace University, White Plains, New York

[38] Mr. Vishal S. Patil, Prof. P.R.Devale; “Biometric Validation by Storing different Patterns using Mouse Gesture Signatures”; International Journal on Recent and Innovation Trends in Computing and Communication; Volume: 3 Issue: 4, April 2015

[39] Ahmed AwadE.Ahmed and IssaTraore, Member, IEEE; A New Biometric Technology Basedon Mouse Dynamics; IEEE Transactions on dependable and secure computing, Vol.4, No.3, July-September 2007.

[40] Youssef Nakkabi, IssaTraoré and Ahmed AwadE.Ahmed; Improving Mouse Dynamics Biometric Performance Using Variance Reduction via Extractors with Separate Features; IEEE Transactions On Systems, Man, And Cybernetics — Part A: Systems And Humans, Vol.40, No.6, November 2010

[41] SoumikMondal, Patrick Bours; ”A study on Continuous Authentication using a combination of Keystroke and Mouse Biometrics”; ELSEVIE journal, Article in Neurocomputing, November 2016; DOI: 10.1016/j.neucom.2016.11.031

[42] Chinmayee.KS, Vanishree C; “Continuous Authentication for Mouse Gesture Recognition using Hidden Markov Model”; InternatIonal Journal of electronIcs&communIcatIon technology; Vol- 8, Issue 4 OCT - DEC 2017.

[43] I. Traore. Information security and object technology (ISOT) research lab, http://www.uvic.ca/engineering/ece/isot/datasets/#section0-3[09 December 2015].

[44] Chao Shen, “Performance Evaluation of Anomaly-Detection Algorithms for Mouse Dynamics” (Computers&Security-2014).http://nskeylab.xjtu.edu.cn/people/cshen/data-sets/behavior-data-set. [15 December 2015].

[45] Mouse Input Notifications;

http://msdn.microsoft.com/en-us/library/windows/desktop/ff468877(v=vs.85).aspx. [20 December 2016].

[46] Weka 3: Data Mining Software;https://www.cs.waikato.ac.nz/ml/weka/. [10 May 2015].

[47] JitBit Macro Recoder; https://www.jitbit.com/macro-recorder/ [02 September 2015].

[48] RUI-Recording User Input; http://acs.ist.psu.edu/projects/RUI/ [24 January 2016].