Indian Institute of Technology, Kharagpur
Department of Electronics and Electrical Communication Engineering

Biometric Authentication using Mouse Dynamics

A Report by

Soumya Sanyal

Under the supervision of:

Prof. Sudipta Mukhopadhyay

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Technology (Hons.) at
Indian Institute of Technology, Kharagpur

Certificate

This is to certify that the thesis titled **Biometric Authentication using Mouse Dynamics** submitted by **Soumya Sanyal** to the Department of Electronics and Electrical Communication Engineering in partial fulfillment for the award of the degree of **Bachelor of Technology** is a bonafide record of work carried out by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission.

Prof. S. Mukhopadhyay
Department of E & ECE
Indian Institute of Technology, Kharagpur
May 2016

Declaration

I hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources. The guidelines provided by the Institute were strictly adhered to during all times.

Soumya Sanyal
Roll No. 12EC10056
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
May, 2016

Acknowledgements

I am deeply grateful to my supervisor, **Professor Sudipta Mukhopadhyay**, for having given me the opportunity to work as part of his research group. Without his help, encouragement and patient support this thesis would never have materialized. I am also grateful to **Mr. Ankit P. Deogirikar** for sharing his expert views and previous work in this field for reference. Lastly, I thank all the members of the research group under Professor S. Mukhopadhyay for supporting me and having their faith in me.

Abstract

Mouse dynamics is one of the many techniques used for identifying individual users based on their mouse operating characteristics. Although previous work has reported some promising results, mouse dynamics is still a newly emerging technique and has not reached an acceptable level of performance. One of the major reasons is the issue of behavioral variability. The mouse data which gets collected over the running sessions of data collection are highly variable. This leads to direct classification of users based on raw data features a large issue. This study aims at addressing the problem of behavioral variability by extracting patterns from the mouse dynamics for particularly useful sessions only. This will support the user classification. Thus, the main objective is to obtain stable mouse characteristics that can be used for further classification using various one-class classification algorithms to perform the task of continuous user authentication.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Need of Biometric Authentication in PC/Laptops

Today most computer systems identify users by means of secret phrases known as passwords. However, this authentication system does nothing to protect the computer from unauthorized access once the user has started an active session. User authentication at sign on secures the workstation only against unauthorized access while the workstation is powered down. Unattended computers with an active session present a much larger security threat. In offices it is common for people to step away from their desks be it to speak to a colleague in the same office, attend a meeting or just go on a break. Users who are not tech savvy frequently leave their computers unlocked with an active session.

It is already established by [12] most attacks originate from the inside the organization that is being attacked, be it on purpose, possibly by a disgruntled employee, or by accident by a user with privileges that are higher than it is actually required for his position. This allows for three types of attacks. A user of lower clearance can gain access to a terminal with higher clearance and access files or functions of the network to which he is not supposed to have access to or a user with the same or higher clearance can conceal his identity by performing malicious actions under the guise of a coworker. Lastly a person who is not affiliated with the company in anyway can gain access to the internal network. These limitations of password based authentication lead to the introduction of authentication techniques based on biometrics. At its very core a biometric-based verification system is a pattern recognition system that acquire a persons biometric data, extracts a feature set and constructs a verification model. Said systems include the following elements: feature extraction which captures the data generated by standard input devices such as a mouse or a keyboard, feature extraction and classifier module that constructs the users signature based on his behavioral biometrics and a signature database consisting of behavioral signatures of registered users. Fig. 1.1 taken from [4] shows an example of behavioral biometric identification system architecture.

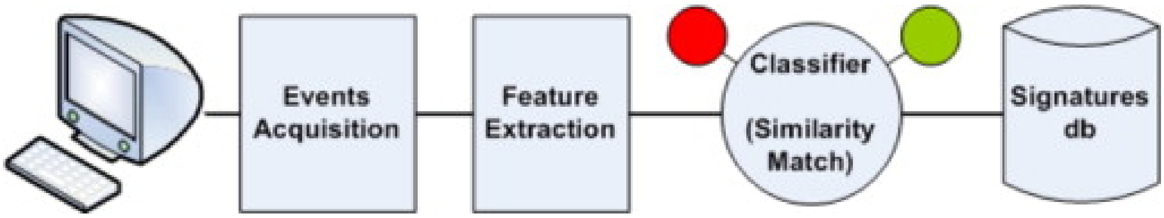

Figure 1.1: A typical framework of a behavioral biometric identification system[4]

1.2 Introduction to Biometric Authentication

Human recognition can be done by using his physiological or behavioral characteristics. It can be done by seeing his face or voice. To achieve a reliable verification the characteristics should be such that really characterizes a given person. Biometrics offer automated methods of identity verification or identification on the principle of measurable physiological or behavioral characteristics.The characteristics are measurable and unique. These characteristics are not be duplicative. Thus, biometrics play an important role in recognizing a human being.

1.2.1 Types of Biometrics

The biometrics can be broadly classified into two types:

- Physiological biometrics
- Behavioral biometrics

Physiological biometrics involve physiological characteristics of a human being used as a biometric such as voice, DNA, fingerprint, IRIS pattern or hand geometry. These biometrics are more reliable and accurate. They are not affected by any mental conditions such as stress or illness. We shall look at some of the physiological biometrics and their strengths and weakness.

1. **Iris pattern:** Iris is the annular region of the eye which controls the amount of light that enters it [3]. Iris recognition technology are used primarily in high security environments, where low error rates are essential. But the problem with Iris patterns is that its accuracy is affected by changes in lighting, its scanners are more expensive. Moreover, the recognition is difficult to perform at a distance longer than few meters.

2. **Fingerprint:** The analysis of fingerprints for matching purposes generally requires the comparison of several features of the print pattern. These patterns dont change much over the time period and differ from human to human, hence if properly recognized the biometric can be used to a great level of accuracy. The sensor is cheap and its size is also small. But still there is a chance that at the high level of security the fingerprint recognition will fail since any intruder may access a legitimate users fingerprints and can use it to login to the system.

3. **Face:** The ability of distinguishing one individual from another is possessed by virtually every human. Facial metrics technology relies on the measurement of the specific facial features. The accuracy of the face recognition systems improves with time. It requires a person to sit at a proper distance from the camera. Also, the illumination factor affects the performance.

Behavioral biometrics involve the behavioral characteristics of a human being. These biometric characteristics are acquired over time by an individual, and are at least partly based on acquired behavior. Thus, it is something known to an individual and can be exploited for authentication purposes. The best known behavioral biometrics are listed below:

1. **Keyboard Dynamics:** This is a behavioral biometric which is characterized by the way a user presses a key on the keyboard or the pattern of typing, which differs from individual to individual. The authentication in keyboard can be carried out in two ways using fixed or free text [10]. For the case of fixed text, all the participants are asked to type same kind

of text data. Whereas, free text is carried out in a continuous manner where the analysis is done throughout the active login period. The biometric does not involve any special hardware for data collection. However, the only difficulty with it is that the behavior of a user doesnt remain constant throughout the active session.

2. **Mouse Dynamics:** This behavioral biometric is characterized by the way an individual moves the mouse or clicks on the screen of the desktop/laptop. Mouse actions like mouse movements, clicks, drag and drop etc. can be used as useful features. This behavioral biometric also has issue with variability of features over time. Related work [15] on mouse dynamics has suggested that it can still be used as a biometric.

1.2.2 Requirements of a Biometric

Before using any kind of human behavioral or physiological characteristics as a biometric, some properties need to be satisfied:

1. **Universality:** Every person should have the characteristics. Although,some exceptional cases can be handled like dumb, deaf or mute people.

2. **Uniqueness:** No two person should have the same biometric characteristics. For e.g. Identical twins are hard to discriminate using DNA analysis.

3. **Permanence:** The biometric should not be variant with time. Behavioral biometrics can be highly variable with time.

4. **Collect-ability:** The characteristics must be measurable quantitatively and obtaining the characteristics should be easy. For example, face recognition systems are not intrusive and obtaining of a face image is easy.

5. **Performance:** Acceptable accuracy should be achieved after identification/ verification.

6. **Circumvention:** This property indicates to how difficult it is to fool the system by fraudulent techniques. For example, a face recognition system should be able to detect actual faces and a picture of a face shown before the camera.

1.2.3 Performance Metrics for evaluating Biometric Authentication

The performance of a biometric authentication system can be measured using the following performance metrics:

1. **FAR:** False acceptance rate is the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs which are incorrectly accepted. In case of similarity scale, if the person is imposter in real, but the matching score is higher than the threshold, then he is treated as genuine.

$$FAR = \frac{\text{Total false positive}}{\text{Total false positive} + \text{Total true negative}}$$

2. **FRR:** False rejection rate is the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected. FRR might increase due to environmental conditions or incorrect use. This may lead to the frustration of the user being rejected several times.

$$FRR = \frac{\text{Total false negative}}{\text{Total false negative} + \text{Total true positive}}$$

3. **ROC curve:** Receiver operating characteristic is a plot between true positive rate and false positive rate. ROC graph is sometimes called the sensitivity vs (1 - specificity) plot as shown in Fig. 1.2. It is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting TPR(true positive rate) v/s FPR (false positive rate), at various threshold settings. TPR is also known as sensitivity and FPR is one minus the specificity or true negative rate.

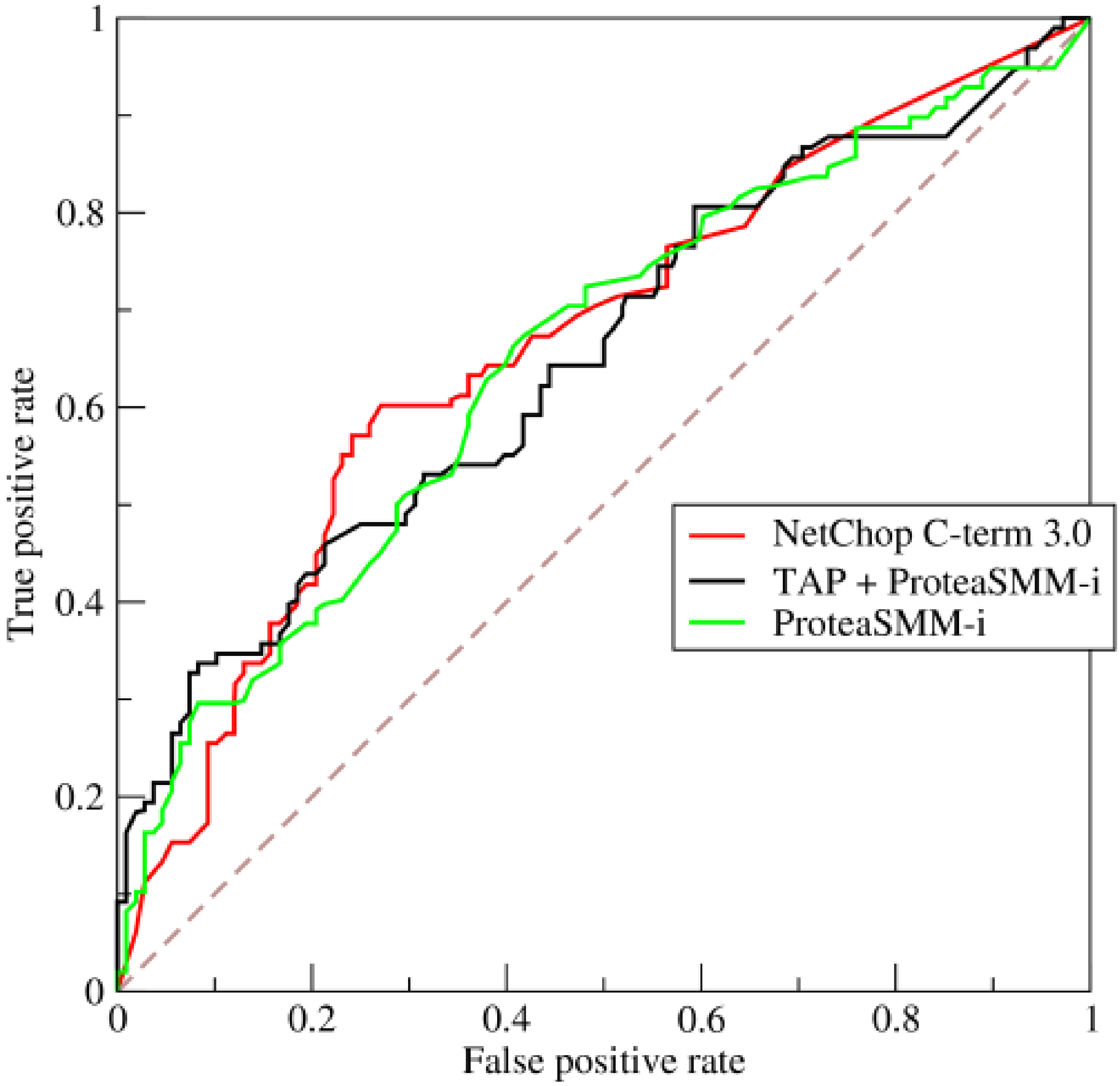

Figure 1.2: A sample ROC Curve of three predictors of peptides

Source: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

4. **EER:** Equal error rate or crossover error rate is the rate at which both accept and reject errors are equal. The value of the EER can be easily obtained from the ROC curve. The EER is a quick way to compare the accuracy of devices with different ROC curves. In general, the device with the lowest EER is most accurate.

5. **Sensitivity:** It measures the percentage of intended user, which are correctly identified.

$$\text{Sensitivity} = 1 - \text{FRR}$$

6. **Specificity:** It measures the percentage of imposter which are correctly rejected.

$$\text{Specificity} = 1 - \text{FAR}$$

   A perfect classifier would be 100% specific and 100% sensitive.

7. **Time to correct reject:** For any biometric system,it is the time after which 1^{st} accurate reject occurs. It is the time required to correctly detect an intrusion at the 1^{st} time in seconds. Ideally, the value of TCR should be zero. However, in practical situations number of samples required by the classifier for good accuracy decides the time required by the system for 1^{st} correct rejection.

1.3 Problem Definition

The objective of the thesis is to create a continuous user authentication system for PCs/laptops to prevent threat against intruder, using biometrics involving mouse dynamics. The scope of the thesis is to address the behavioral variability of users while using mouse on a PC/laptop.

1.4 Overview of Mouse Dynamics

For getting an idea about mouse dynamics we need to look into different types of mouse actions which can occur from user interaction with the PC through a mouse. The different types of possible mouse actions are listed below:

1. **Mouse Move:** Mouse move is a simple movement involving no clicks. Mouse move can be between two click events or non-click events.

2. **Drag and Drop:** It is the action which starts by a mouse button held down followed by a movement and finally the button released. Generally, it is used to move/copy a file to a particular location.

3. **Point and Click:** It is a movement of the mouse ending in a click.

4. **Silence:** This action suggests no mouse movement.

In order to capture these kind of mouse actions we would require the data collection software to capture events like:

- Mouse move
- X,Y co-ordinates of the screen
- Left button down
- Left button up
- Right button down
- Right button up

From these events mouse actions given below can be extracted:

- Left Click
- Left Double Click
- Right Click
- Drag and Drop

Once a mouse action is captured, the useful characteristics of the action can be saved and can be used later for analyzing the verification of a wide range of users.

1.5 Understanding Continuous Authentication

User authentication based on the mouse dynamics biometric can be divided mainly into two tasks of interest[14]. One task is static authentication, which checks the user only once, typically at login time. Another is continuous authentication, which checks the user continuously throughout the session. The main strength of mouse dynamics biometric technology is in its ability to constantly monitor the legitimate and illegitimate users based on their session based usage of a computer system. In this thesis, continuous authentication for mouse dynamics is addressed.

1.6 Factors affecting the Performance of Continuous Mouse Dynamics based Biometric

There are many security techniques in the market for the PC/laptops. There has been an extensive amount of work done for user authentication systems but a whole lot is remaining when it comes to mouse dynamics based PC authentication. Many security applications may require a continuous authentication where biometrics like keyboard dynamics, mouse dynamics and face recognition can complete the demand. But using mouse dynamics as a behavioral biometric has many challenges. Some of the challenges are as follows:

1. **Environmental conditions:** The environmental factors like height of the chair, distance between mouse and body, usage of new mouse can vary the mouse dynamics vastly.

2. **Physical conditions of a user:** Anger, illness or tiredness and similar moods of a user can change the mouse dynamics.

3. **Application scenario:** So far discussed papers have concentrated on only limited applications. The main reason is that the mouse dynamics are greatly dependent on the application scenario since some users are used to certain kind of applications like word file, Internet Explorer, games etc.

4. **Skill level of a user:** The skill level of a user varies with the type of application. Moreover, the user gets accustomed to certain type of operations.

5. **Noise:** The collected data can contain improper readings due to hardware or software errors.

6. **GUI/mouse setting:** Screen resolution or pointer speed can vary the dynamics by a great margin.

7. **Nature of features:** The mouse is a behavioral biometric, so the features may also be varying over time.

Chapter 2

Literature Survey

2.1 Background and Related Work

An extensive amount of work has been done in the field of biometric authentication using mouse dynamics. Here, only the ones relevant to continuous user authentication are discussed upon.

Pusara and Brodley [9] proposed a re-authentication scheme in which raw mouse data was pre-processed and grouped into data points, each corresponding to a summary of mouse events over a window of configurable size. They set up a personalized model for each user using C5.0 decision trees. Using data from 11 users, collected on their own personal computers under a free environment, an average false-acceptance rate (FAR) of 1.75% and average false-rejection rate (FRR) of 0.43% were reported. The verification time ranged from 1 minute to 15 minutes because the parameters were chosen independently for each user. This result suggests mouse dynamics may reach a practically useful level, but a sample size of 11 users is relatively small and the issue of behavioral variability is not addressed.

Ahmed and Traore [1, 2] aggregated low-level mouse events as higher-level actions such as point-and-clicks. They defined seven feature vectors. Biometric analysis was conducted by concatenating these feature vectors into a 39 dimensional global feature vector and using a neural network for model training and classification. The proposed method was assessed with 22 subjects, achieving an average equal error rate of 2.46%. The length of data session used in the experiment was around 17 minutes (though not specified explicitly in the paper). A supplementary experiment with 7 participants, each of whom was asked to provide three sessions with a period of 30 minutes for each session using the same hardware and soware application, resulted in an FRR of 6.25% and FAR of 1.25%.

Recently, Nan et al. [17] presented a user verification system based on mouse dynamics using newly defined angle-based metrics, which is able to re-authenticate a user with high accuracy. Note that in the approaches of Ahmed et al. and Nan et al., both the impostors' and the legitimate users' mouse feature samples were used for training. This is not realistic since it might be impossible to collect a large amount of data from all potential impostors in practice.

Gamboa and Fred [5, 6] presented a continuous authentication approach, in which every movement was considered as a 'stroke', to capture and extract the characteristics of mouse behavior. Each stroke was characterized by a 63-dimensional feature vector including spatial parameters such as angle and curvature, and temporal parameters such as velocity, and acceleration. This feature space was reduced to the best subset of features for each user through a greedy feature selection process. The authentication decisions were made based on the average classification outcome of a sequence of individual strokes using a statistical model. Experiments on data from 50 users collected under a free environment, found that sequences of 1 stroke, 50 strokes and 100 strokes yielded an equal error rate of 48.9% , 2% and 0.7% respectively (equivalent to verification time about 2 seconds, 50 seconds and 100 seconds respectively). However, in this approach the test data was also used for feature selection, which may lead to an overly optimistic result of the detector's performance.

Observation from the above survey is that most of the previous research uses both legitimate user's and impostors' samples to train their models, which are not practical in realistic applications.

2.2 Current Scenario

As described in [15], current methods of user identity verification based on mouse movements are not efficient enough to achieve the European Standard for Access Control Systems requirements, which recommends an FRR of less than 1% and FAR of under 0.001%. Table 2.1, taken from [17], shows the effectiveness of current mouse based user verification methods. To this end, today's behavioral biometrics systems employ both mouse based behavioral biometrics as well as ones that are keystroke based. The problem with such systems is not of a technical nature but more of a social one because for a keystroke identification system to function it has to record all of the users input. Systems that identify users through the dynamics of their keystrokes in essence fall into the category of key loggers and a user has to trust that the system will not record his passwords or private messages and relay them to a malicious third party. Because of this it is important to achieve a fully functional biometric identification system whilst solely relying on the data collected by observing the movements of the mouse. Up until recently, mouse based verification systems were implemented using neural networks which are computationally very heavy and therefore degraded the performance of the machine they were running on and consumed more resources than a background process normally should.

The method demonstrated by Zheng et al. [17] uses support vector machines (SVM) [7] that have already been successfully used in a method for recognizing handwritten digits [16], which also fall under the category of behavioral biometrics. One of the key features of it is the small number of user actions that are required in order to identify a user.

The work done by Shen et al. [14] developed a simple continuous authentication approach using pattern-growth based mining methods to extract behavioral patterns, employing a one-class learning algorithm. An FAR and FRR of 7.78% and 9.45% respectively was achieved by them.

Table 2.1: Comparison of existing user verification methods

Ref.	**FRR**	**FAR**	**Data Required**	**Settings**	**Notes**
[1]	2.45%	2.46%	2000 mouse actions	Continuous	Free mouse movements
[8]	0%	0.36%	2000 mouse actions	Continuous	Free mouse movements
[5]	2%	2%	50 mouse strokes	Static	Mouse movements from a game
[9]	1.75%	0.43%	Not specified	Continuous	Applies to certain applications
[13]	11.2%	11.2%	3600 mouse actions	Continuous	Free mouse movements
[11]	4%	3.5%	Not specified	Static	Mouse movements from a game
[4]	9.5%	17.66%	30 mouse actions	Continuous	Free mouse movements
[17]	1.3%	1.3%	20 mouse actions	Continuous	Free mouse movements
[14]	9.45%	7.78%	Not specified	Continuous	Free mouse movements

Chapter 3

Continuous User Authentication using Mouse Dynamics

3.1 Data Acquisition

Previously, for data acquisition a graphical user interface (GUI) was built which consisted of a virtual keyboard. The users were supposed to enter a text displayed on the GUI using the mouse clicks on the virtual keyboard and the corresponding features were extracted from the mouse dynamics. The logic behind using such an interface was to mimic a login system of a PC. This worked well for the case of static one time user authentication. But for continuous user authentication we need a constant monitoring of the user activity noting each mouse movements as the time passes by. For this data needs to be collected in run-time. Also, we need to do away with any GUI and build a proper mouse logger. The objective was to come up with a cross-platform application which can be used for mouse data acquisition. Initial tries included using python to develop such an application for data collection. But due to various dependency issues while testing the app on windows platform, later the development language was shifted to java. After some research the required application was built which works for mostly used platforms like Windows, Mac OS and Linux. Various java native hooks were used for capturing the system-wide data from the mouse.

The mouse data collected by the application is stored in a format as mentioned in Table 3.1.

A defined format helps in data filtering and feature extraction processes.

Table 3.1: Mouse Data Logging Format

Notation	**Meaning**
MC, n, t:	*Mouse Clicked, Click count, Relative time*
MP, n, t:	*Mouse Pressed, Button ID, Relative time*
MR, n, t:	*Mouse Released, Button ID, Relative time*
MM, x, y, t:	*Mouse Moved, x-coordinate, y-coordinate, Relative time*
MD, x, y, t:	*Mouse Dragged, x-coordinate, y-coordinate, Relative time*
MWM, x, y, w, a, s, t:	*Mouse Wheel Moved, x-coordinate, y-coordinate, Wheel rotation sense, Amount of scrolling, Scroll type, Relative time*

Various other parameters are also collected for each run of the application. Factors like the logging time, user's IP address, OS and architecture used by the user, the version of the OS and the screen resolution are logged. These features can later be used to check for feature variations based on the time when data is collected, the operating system used etc. A sample log is shown in Table 3.2.

Table 3.2: User Statistics: sample collection

LOGGING TIME:	20151128_003812
CLIENT IP:	10.109.11.55
USERNAME:	soumyamac
OS:	Mac OS X
ARCHITECTURE:	x86_64
VERSION:	10.11.1
RESOLUTION:	1280.0 800.0

3.2 Feature Selection

3.2.1 Feature Extraction:

The mined frequent-behavior patterns cannot be used directly by a detector or classifier. Instead, dynamic characteristics are extracted from these patterns. Some characteristics that can be extracted are:

- **Click Time:** It is the time required for the user to click a button. It is the time gap between Left Button Down and Left Button Up Event (or for the Right button case).

- **Pause Time:** It is the amount of time spent pausing between pointing to an object and actually clicking on it.

- **Horizontal Velocity:** Horizontal Velocity is change in X coordinate value for the given change in time.

  $$HorizontalVelocity = \frac{\partial x}{\partial t}$$

- **Vertical Velocity:** Vertical Velocity is change in Y coordinate value for the given change in time.

  $$VerticalVelocity = \frac{\partial y}{\partial t}$$

  Where x and y defines the coordinates of a point and t defines the time of that point recorded.

- **Straightness:** The straightness feature characterizes the nature of movement of the mouse move. It was seen in the previous chapter that the feature was not discriminating.

From the logged data, the characteristics mentioned above are calculated. For calculation of these values a session is maintained. The concept of a session is explained below. Now, a user may have many log files corresponding to the number of times he used the application to log the mouse data. The characteristics are calculated for all such log files for a user. Now two features are defined corresponding to each characteristic calculated. One is the mean of all the values and the other is standard deviation. Thus, for each characteristic calculated we get two features, the mean and the standard deviation. The features found from the characteristics are mentioned in the table below:

Table 3.3: Feature Table

Characteristic	**Mean**	**Std. Deviation**
Click Time	Feature 1	Feature 2
Horizontal Velocity	Feature 3	Feature 4
Vertical Velocity	Feature 5	Feature 6
Pause Time	Feature 7	Feature 8
Straightness	Feature 9	Feature 10

3.2.2 Feature Plots

For analysis of the viability of the features their probability density functions are plotted using a gaussian fit for the interpolation. The plots are plotted such that the red plot is for the legitimate user's data and the blue plot is for the imposter user's data. The plots for individual features are shown from Fig. 3.1 to Fig. 3.10 :

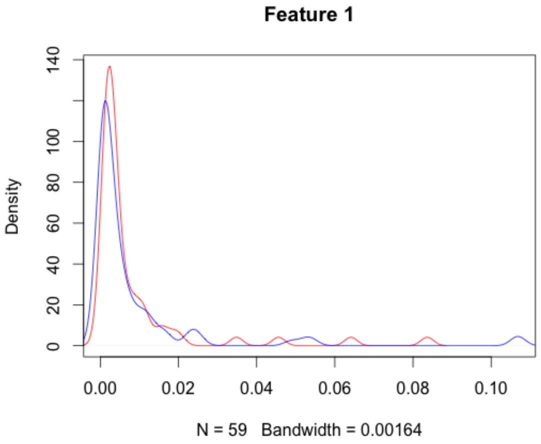

Figure 3.1: PDF of Feature 1

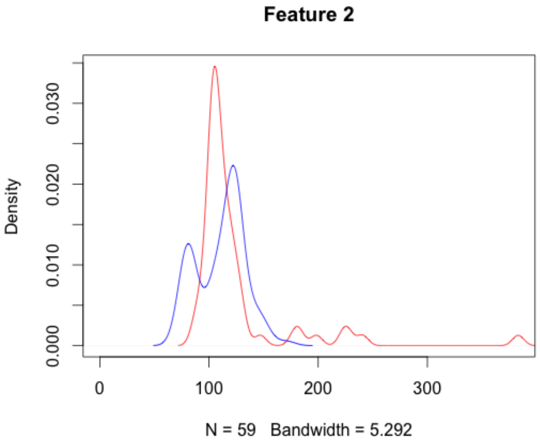

Figure 3.2: PDF of Feature 2

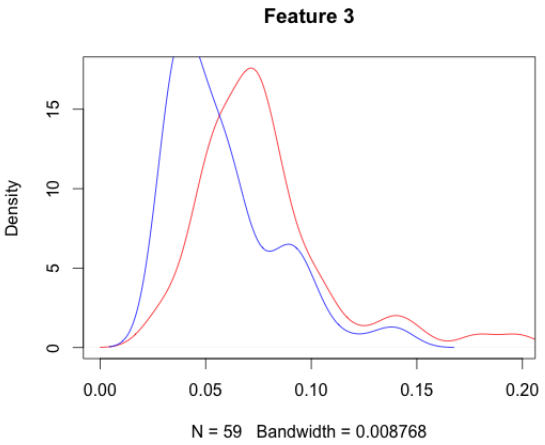

Figure 3.3: PDF of Feature 3

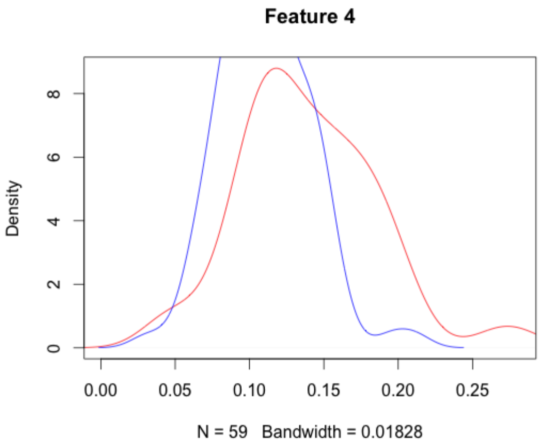

Figure 3.4: PDF of Feature 4

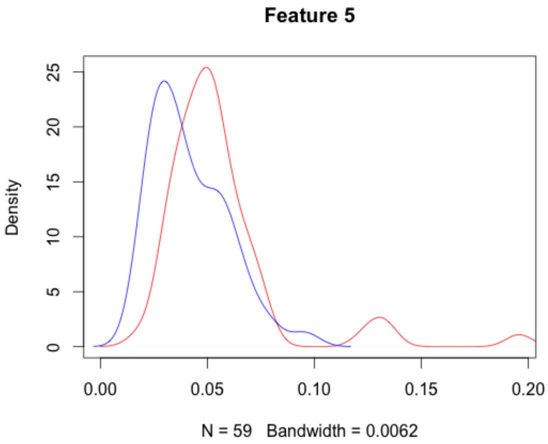

Figure 3.5: PDF of Feature 5

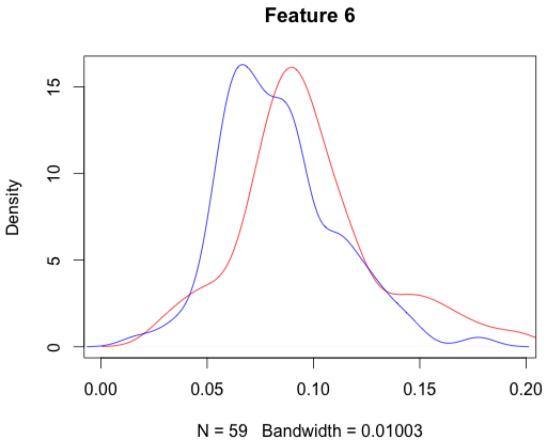

Figure 3.6: PDF of Feature 6

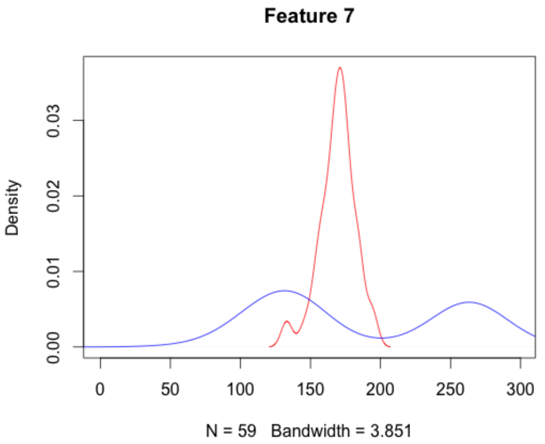

Figure 3.7: PDF of Feature 7

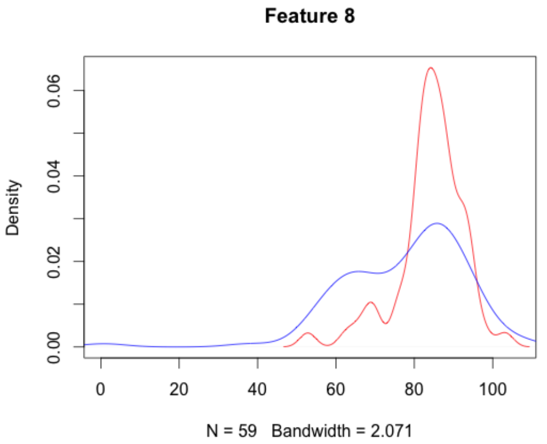

Figure 3.8: PDF of Feature 8

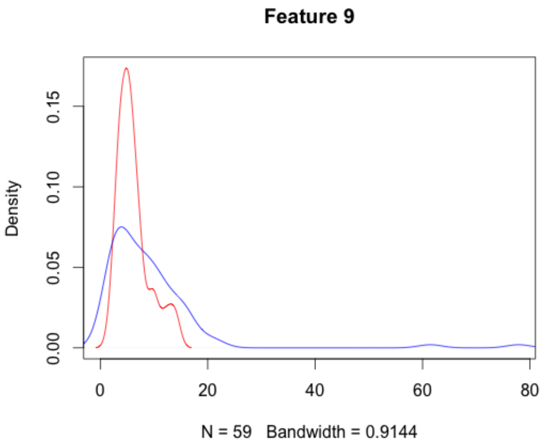

Figure 3.9: PDF of Feature 9

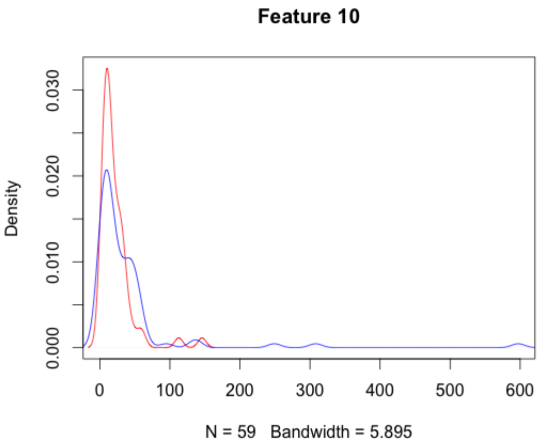

Figure 3.10: PDF of Feature 10

3.3 Concept of User Session

A unique concept of session is introduced here in this work. Usually, for continuous user authentication, when data collection is done using some logger, there are times when there is some lag in the data collection. This lag can be for reasons like the user is out for some personal timeout, he is watching a video or listening to some music etc. So, if a user's log file has two records after a specified *session break time*, then it is considered a new session of data logging. Also, with regards to the concept of session, another term is defined as the *minimum actions per session*. This parameter sees that every session has some minimum number of logged records. If a session does not have these many number of records then that session is marked invalid and is not considered for feature extraction process. This is done because only after a certain length of valid mouse movements can we go about extracting features from the session. Also, a term called *minimum response time* is defined to be equal to the minimum time difference between two consecutive records. If the time difference exceeds this limit then it is presumed that the continuity of the mouse movement is broken and a new mouse stroke is started. With these added constraints, the features are extracted.

3.4 Classification

User authentication is still a challenging task from the pattern classification viewpoint. It is a two class (legitimate user vs. imposter) problem, but only the patterns from the legitimate user are available in advance. Most previous research used both the legitimate user's and the imposter user's samples to train their models. Yet this is not practical in realistic applications since there may be thousands of potential imposter data samples at the risk of fatal intrusion. Therefore, a better solution is to build a model only based on the legitimate user's data samples, and then use it to detect imposter users who are using some sort of similarity measures. This type of problem is known as one-class classification or anomaly detection. So, here in this study I have used both the classification techniques to compare the results and draw a conclusion about the accuracy of the two models.

3.4.1 SVM Classifier

We 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 [29], object recognition [22], text classification [14], and image retrieval [28]. 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 hyperplane, 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 [30].

For a binary classification problem, given l training samples $x_i, y_i, i = 1, ..., l$, each sample has d features, expressed as a d-dimensional vector $x_i (x_i \in \mathbb{R}^d)$, and a class label y_i with one of the two values ($y_i \in \{-1, 1\}$). A hyperplane in d-dimensional space can be expressed as $\mathbf{wx} + b = 0$, where w is a constant vector in d dimensions, and b is a scalar constant. We aim to find a hyperplane that not only separates the data points but also maximizes the separation.

A popular choice of kernel function is the Gaussian Radial Basis Function (RBF). In practice, RBF is a reasonable first choice among other kernels, due to its generality and computational efficiency. Thus, the procedure to resolve a classification problem using SVMs is: (1) choosing a kernel function, (2) setting the penalty parameter C and kernel parameters as well, if any, (3) constructing the discriminant function from the support vectors. In particular, we view the user verification problem as a two-class classification problem, and the learning task is to build a classifier based on the user mouse movements.

3.4.2 Anomaly Detection

For proper anomaly detection we need to use only the legitimate user's log data to train our classifier. The data of the imposter user is not used for any training purpose. This is the true scenario as we will not have the fraudulent data while we use it for real life scenarios. Thus, this approach is more realistic. For anomaly detection, a standard naive approach is to fit gaussian curves for each of the features and then estimate the probability for a new test case. If the probability is less than some pre-calculated threshold $\in$, then the test sample is marked as an anomaly (imposter). So basic model is,

$$P(x) = \prod_{i=1}^{N} \mathcal{N}(x, \mu, \sigma)$$

$$Output = \begin{cases} \textit{imposter} & \textit{if P(x)} \leq \in \\ \textit{legitimate} & \textit{otherwise} \end{cases}$$

where,

$p(x)$ is the probability of the sample x belonging to the legitimate class,

N is the number of features used,

$\mathcal{N}$ is the normal distribution value at x for mean μ and variance σ^2

For this model, the values of $\mu's$ and $\sigma's$ for individual feature are calculated by considering the values of the features found from the feature extraction and taking average and standard deviation respectively. That is,

$$\mu = \frac{(\sum_{i=1}^{M} samples)}{M}$$

where M is the number of samples we have extracted for a particular feature.

3.5 Results and Discussions

The whole model is trained and tested by the data collected from 7 volunteers over their normal usage of their own laptops. Thus, varied amount of data could be collected from individual user based on the enthusiasm with which the particular user used the data logger. So, for analysis purposes, the data of one of the most volunteering person was chosen for specific analysis. On an average every 60,000 records logged amounted to 1Mb of data. The user chosen for training had 40Mb of data logged. Though this is still a very small data for behavioral analysis but this was the best data available. On this data both the models were trained. For the first model (SVMs), all the available data was used. For this model, an average accuracy of 95.06% was achieved for 1000 iterations of the model. Average FRR was 4.97% and average FAR was 4.80%. The area under the ROC curve was 0.95. This is a respectable model which can be compared equivalently to the current models published. For the second model (Gaussian fit for anomaly detection), only the data for the legitimate user is considered for the calculation of the mean and the sigma values for the individual features. Also, a threshold value is set for this model to work based on some manual experimentation. For this model, an average accuracy of 84.59% was achieved for 1000 iterations of the model. Average FRR was 23.77% and average FAR was 7.03%. The area under the ROC curve was 0.85. We observe a drop in the accuracy and the FRR values of this system. The main reason for drop is the lack of quality data which rightly estimates the user. Also, an inherent problem with this feature is that if the number of features are reduced then the threshold value is also changed accordingly and more inaccuracy is built into the system. The final values are enumerated in the following table:

Also, the ROC plot and the Sensitivity-Specificity plot of the two models are shown in Fig. 3.11 to Fig. 3.14:

Table 3.4: Results Table

Model	**Accuracy %**	**FRR %**	**FAR %**
SMVs	95.06	4.97	4.80
Gaussian Anomaly	84.59	23.77	7.03

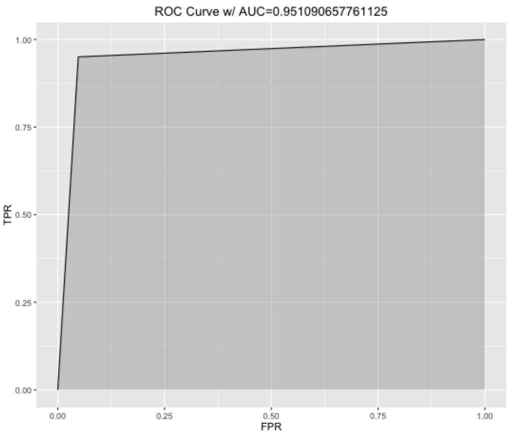

Figure 3.11: ROC Curve of Model 1

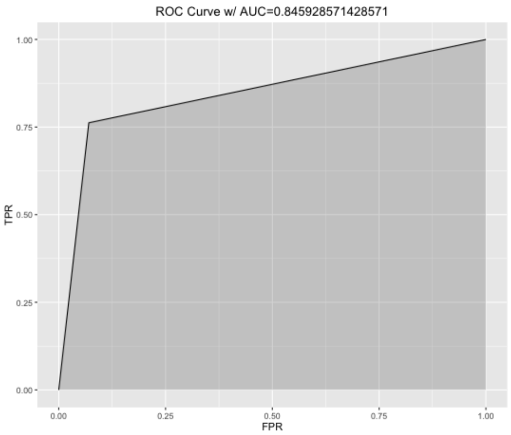

Figure 3.12: ROC Curve of Model 2

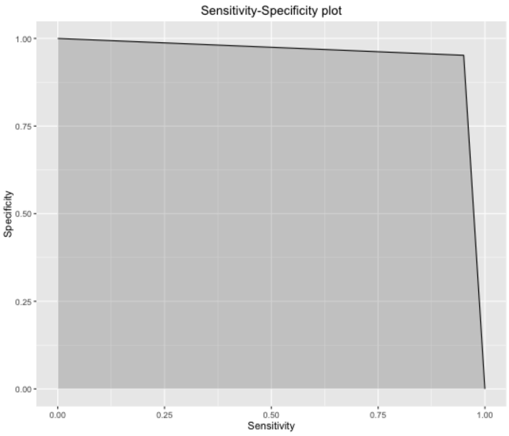

Figure 3.13: Sensitivity-Specificity plot of Model 1

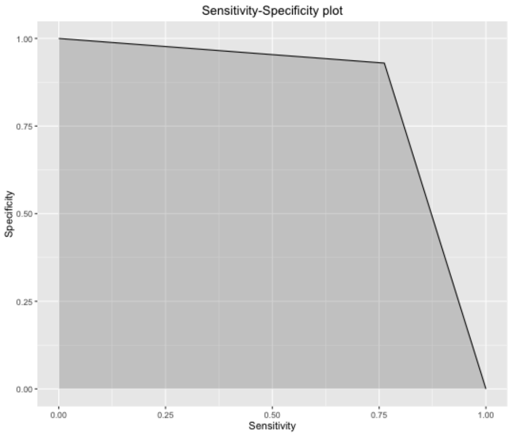

Figure 3.14: Sensitivity-Specificity plot of Model 2

3.6 Conclusion and Further Insights

Thus, from the above work we conclude that with proper feature engineering and quality data, a model can be built which can be used to uniquely identify the legitimate user from the fraud user based on their mouse dynamics using the concept of user session. Thus, this leads to an understanding that this domain can be further explored in future years to build a more robust system for session based fraud detection. There is a huge scope of improvement in the current project. The major challenge is to collect quality data for analysis and this remained the major bottleneck in the project. A more user friendly and non-intrusive way needs to be devised for data collection. Also, more experimental features can be tried and improvisation along those lines can lead to some further improvement in the model.

Whole bibliography

[1] Ahmed Awad E Ahmed and Issa Traore. “A new biometric technology based on mouse dynamics”. In: *Dependable and Secure Computing, IEEE Transactions on* 4.3 (2007), pp. 165–179.

[2] Ahmed Awad E Ahmed and Issa Traore. “Anomaly intrusion detection based on biometrics”. In: *Information Assurance Workshop, 2005. IAW’05. Proceedings from the Sixth Annual IEEE SMC*. IEEE. 2005, pp. 452–453.

[3] Rajat Kumar Das, Sudipta Mukhopadhyay, and Puranjoy Bhattacharya. “Continuous multimodal biometric authentication for PC and handheld devices”. In: *IETE Journal of Education* 52.2 (2011), pp. 59–69.

[4] Clint Feher et al. “User identity verification via mouse dynamics”. In: *Information Sciences* 201 (2012), pp. 19–36.

[5] Hugo Gamboa and Ana Fred. “A behavioral biometric system based on human-computer interaction”. In: *Defense and Security*. International Society for Optics and Photonics. 2004, pp. 381–392.

[6] Hugo Gamboa and Ana LN Fred. “An Identity Authentication System Based On Human Computer Interaction Behaviour.” In: *PRIS*. 2003, pp. 46–55.

[7] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[8] Youssef Nakkabi, Issa Traoré, and Ahmed Awad E Ahmed. “Improving mouse dynamics biometric performance using variance reduction via extractors with separate features”.

In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 40.6 (2010), pp. 1345–1353.

[9] Maja Pusara and Carla E Brodley. “User re-authentication via mouse movements”. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security.* ACM. 2004, pp. 1–8.

[10] Kenneth Revett. *Behavioral biometrics: a remote access approach.* John Wiley & Sons, 2008.

[11] Kenneth Revett et al. “A survey of user authentication based on mouse dynamics”. In: *Global E-Security.* Springer, 2008, pp. 210–219.

[12] Bruce Schneier. *Secrets and lies: digital security in a networked world.* John Wiley & Sons, 2011.

[13] Douglas Schulz et al. “Mouse curve biometrics”. In: *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the.* IEEE. 2006, pp. 1–6.

[14] Chao Shen, Zhongmin Cai, and Xiaohong Guan. “Continuous authentication for mouse dynamics: A pattern-growth approach”. In: *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on.* IEEE. 2012, pp. 1–12.

[15] Mirko Stanic. “Continuous user verification based on behavioral biometrics using mouse dynamics”. In: *Information Technology Interfaces (ITI), Proceedings of the ITI 2013 35th International Conference on.* IEEE. 2013, pp. 251–256.

[16] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory.* Vol. 1. Wiley New York, 1998.

[17] Nan Zheng, Aaron Paloski, and Haining Wang. “An efficient user verification system via mouse movements”. In: *Proceedings of the 18th ACM conference on Computer and communications security.* ACM. 2011, pp. 139–150.