Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 2nd, 2008

Refinement of a Mouse Movement Biometric System

Nkem Ajufor, Antony Amalraj, Rafael Diaz, Mohammed Islam, Michael Lampe
Seidenberg School of CSIS, Pace University, White Plains, NY, 10606, USA
{rd58801p, ml81145n, na13166w, mi01252n}@pace.edu, tonyalen@gmail.com

Abstract

This study focuses on the further development and testing of an existing Mouse Movement Biometric Identification system. Building on the research and work from a previous implementation of a mouse biometric system, the study focuses on enhancing the data capture program to enrich the data with more features to identify the user, collecting more samples of data, normalization of the data into the required format and comparing the results to previous conclusions. A total of 205 data files were collected from users to further develop and test the software. All features of the Mouse Movement system are discussed as well as final recommendations for future research and improvement has been included.

1. Introduction

As computing and Internet usage continues to grow worldwide, so does the need to ensure that the users of computing systems are protected from unauthorized malicious users accessing their systems. There are a variety of threats out there and there are multiple points of entry to networks. The best form of defense is a multiple layered defense. This can be accomplished by implementing high security standards, processes, procedures and tools that together can help protect vulnerable systems.

One area lacking in the multi layered defense approach is in authentication and identification. Until recently, most systems were secured with the ubiquitous user name and password to authenticate the user and some sort of token to identify the user as they access various applications and systems. This is only one layer of defense. Once the user is authenticated, they generally are not challenged after that point and in the case of a malicious user could have free reign over the system. Security experts have pointed out that there is a necessity to provide multiple forms of authentication and identification and not rely solely on usernames and passwords. Biometrics could provide a primary as well as a secondary way to authenticate and identify users.

Biometrics is defined as and literally means "life measurement". In security applications, Biometrics refers to automated methods for identifying people based on their unique physical characteristics or behavioral traits. Biometric access control refers to any means of controlling access through human measurements which can include fingerprints, handwriting analysis, retinal scanning, hand geometry, voice printing as well as Mouse Movements.

The need for Biometrics is brought on by the need to ensure and enhance security on systems. Users need to be assured that the system that they are using is correctly and securely authenticating them for any type of application they may use. This could mean concluding a financial transaction on an online banking site or taking a test remotely from your home. Whatever the intended use may be, the main factor here is that all users are being securely authenticated and identified on the system and that people seeking to do malicious acts are kept out or caught early.

Mouse Movement Biometrics is one area of research that is ongoing in the Biometrics field. Teams from Purdue and Tufts Universities have recently done research on the possibility of re-authenticating users through their mouse movements. Data is collected into individual user profiles based on the user's mouse movements. The underlying hypothesis was that they could successfully model users' behaviors on the basis of user invoked mouse movements [2]. Normally authentication occurs in the beginning of a session. However once that session is started, there is no way to continue to find out if the user is who they say they are. One method of re-authentication is by monitoring the mouse movements of the user and comparing it to a profile [2].

This type of system could be useful in an academic on-line environment, for instance while taking a test. If we are able to identify the user based on their use of a Mouse and the Mouse Movement behavior they generate, it would be possible to verify that the proper student was taking the test and cheating by allowing someone else to take the test would be discouraged.

The study performed in this paper is similar to the research Purdue and Tufts Universities performed and focuses on enhancing the data capture program to enrich the data with more features to identify the user, collection of more samples of data for analysis, normalization of the data into the required format to provide tother groups performing testing and comparing the results to previous conclusions. A total of 205 data files were collected from users to further develop and test the software. The Mouse Movement Biometrics system is still in its infancy, therefore a lot of data was generated in order to validate our findings.

The paper is structured as follows: Section 2 describes the Mouse Movement Biometric system and the enhancements that occurred in this version of the system. Section 3 describes the experiments that were performed. Section 4 presents the results of those experiments and Section 5 presents our conclusions, discusses problems we encountered, and provides future recommendations.

2. Mouse Movements Biometric System

The Mouse Movement Biometric system contains three modules:

1. **MMSystem** - Data capture/collection using Mouse Movement System
2. **FeatureExtraction** – Generates Feature Vector data
3. **Classifier** - To identify patterns and Normalize/Classify the user's data.

2.1 Data Capture

Raw data was captured by running the MMSystem application on a pc or laptop and having the user click on a 5x5 grid of 25 buttons. The collection of data is handled by a background process and therefore hidden from the user. The MMSystem application can be configured to have a blank window, a grid of buttons or a tic-tac-toe games as its interface but the main program used for the data capturing process was the 5 x 5 grid, displayed in Figure 1.

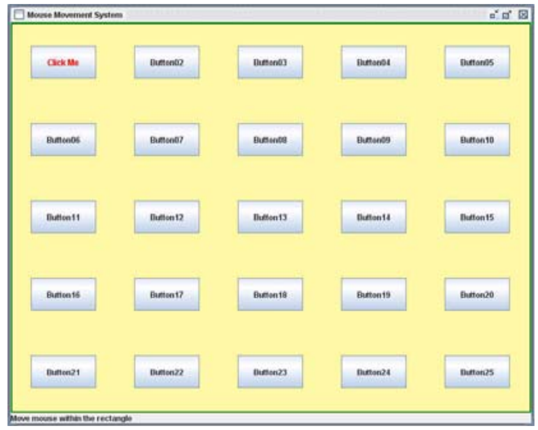

Figure 1 . Data capture screen

The MMSystem application instructs a user to click on the buttons successively until all the buttons are clicked. Currently the system is configured to prompt for the clicks in a predetermined order. Once the user completes the exercise and exits the application, the data is captured and stored in a CSV format in an excel file as a raw data format. A sample breakdown structure of some of the information captured is displayed in Table 1. The data captured, provides us with the screen size of the system used, which in this case the screen size was 1280 x1024. Other information captured is the movement of the mouse in relation to the screen as well as when the mouse was clicked (in this case – a left mouse click). Below is the screenshot of the sample data file in its raw format:

Table 1. A sample raw data file

User screen Size	X- co-ordinate/width	1280	X- co-ordinate/height	1024
timer	50			
mouseMoved	1.19251E+12	618	527	
mouseMoved	1.19251E+12	638	565	
mouseMoved	1.19251E+12	638	577	
mouseMoved	1.19251E+12	638	587	
mouseMoved	1.19251E+12	259	268	
mousePressed :left cli	1.19251E+12	47	30	
mouseReleased	1.19251E+12	47	30	
mouseMoved	1.19251E+12	50	31	
mousePressed :left cli	1.19251E+12	25	9	

The following is a brief description about the sample data file shown in Table 1:

- Raw data is collected by clicking the 5X5 - 25 click button program
- Each time the user clicks and moves the Mouse, the background program collects

and stores the raw data respectively in a .csv file

- Mouse event, whether it is a move, drag and drop or click (pressed or released) is recorded
- Time of the event in milliseconds is recorded
- X and Y co-ordinates of the Mouse pointer on the user screen is recorded
- Upon the completion of the task, a text file is generated in a CSV format

This study required the Mouse Movement system to be enhanced in order to capture more user characteristics other than just the user's name. A front-end registration process was created that captured pertinent information regarding the user. The following screenshots demonstrate the new interface.

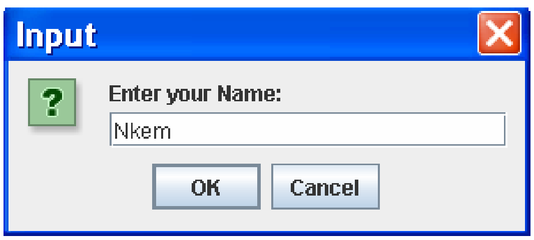

Figure 2. Name input screen

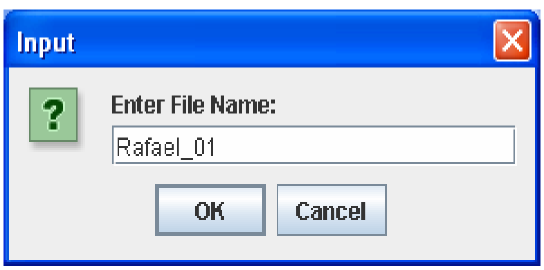

Figure 3. File Name screen

Figure 4. Gender input screen

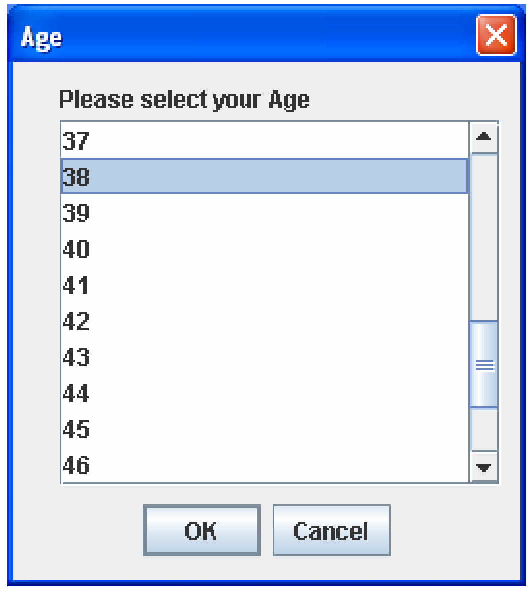

Figure 5. Age input screen

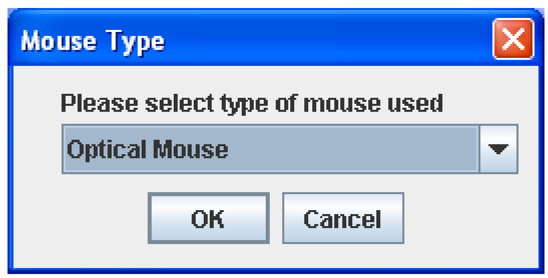

Figure 6. Mouse type

Figure 7. Hand used

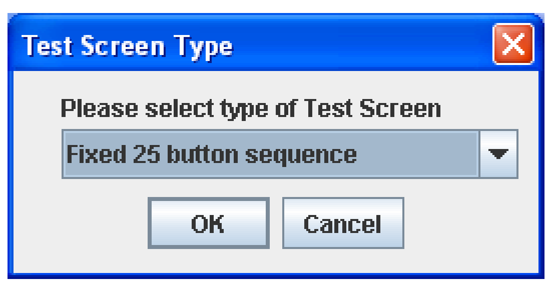

Figure 8. Screen selection

2.2 Feature Extraction

Mouse Movement is required in all graphical user interfaces such as the Windows operating system which is in use everywhere. Using Mouse Movement Biometrics the characteristics generated are unique enough to identify an individual user. With Mouse Movement, there are several features which can be used to create a pattern. These patterns can then be used to create profiles that are stored for later comparison. Currently the system collects data on two Mouse events: Mouse Movement and left Mouse click.

Feature Vector Data Extraction involves feeding the raw data that was collected in the Data Capture program and applying some calculations to extract feature characteristics that help to identify user behavior. These measurements help us in creating a feature vector, which in turn refers to a user profile or a user signature. The following section describes features definitions and how they are computed.

2.2.1 Mouse Movement System Features

We begin by using the Data Capture module (MMSystem) to capture raw data from users. After capturing the raw data, it is processed creating Mouse curves and Mouse clicks. Each Mouse curve and click is associated with the following set of features: Size of the curve, length of the curve, speed of the curve acceleration of the curve, duration of the click and curvature of the curve.

2.2.1.1 Size of a curve:

Size of a curve is defined, as the total number of continuous points that constitutes the curve.

Size of the curve n = $\sum_{i=4}^{n} (p_i)$

(1)

P = a Mouse data point. The curve that is less than four points is ignored. [10]

2.2.1.2 Length of a Mouse curve:

Length of the Mouse curve is defined as the sum of the distance between all adjacent curve co-ordinates. A Mouse curve c with n points has a length of [10]:

Length(c) =

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

(2)

2.2.1.3 Total time of the Mouse curve:

Total time of the Mouse curve is defined as the total time taken to complete the Mouse curve. A Mouse curve c with n point has a total time of:

Total Time (c) = $\sum_{i=2}^{n} (t_i - t_{i-1})$

(3)

Where t is the time stamp on the Mouse point [10].

2.2.1.4 Mouse speed over a pre-defined action

Mouse Movement is obtained when the user moves the Mouse from point A to point B without pressing any Mouse buttons. Movement ends when there is no input from the Mouse for a pre-defined period of time. Drag and Drop is defined when the user moves Mouse from point A to point B while pressing the left Mouse button. The data capture module has the facility to collect the data at different time intervals. We compute the Mouse speed from the distance traveled and the time it takes to reach the next point. The Mouse speed may not be uniform through out the movement of the mouse. At the start of movement it may be at a low speed and at middle of movement it accelerates to a faster speed. With these characteristics each user creates a unique profile on his/her Mouse Movements technique [10].

Average speed of the curve: Average speed of the curve is defined as the average speed taken to complete the curve. The speed between two points is computed as the distance traveled over time. A Mouse curve c with n points has an average speed [10]:

Average speed (c) =

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

(4)

2.2.1.5 Angle of Mouse Movement

Another interesting feature to consider is the angle of Movement along with the speed of the Movement. Depending upon the Movement of the Mouse in a particular direction, the speed of the Mouse can vary [10].

2.2.1.6 Acceleration

Acceleration is the time rate of change of the velocity with respect to magnitude or direction, which is the derivative of velocity with respect to time [11]. Readings from the Mouse Movement literature [1] [2] indicates that as we move from one button to the next there will be an acceleration and deceleration of the Mouse. The system takes acceleration as the time and divides it by velocity [10].

2.2.1.7 Mouse click duration

Mouse click is obtained when the user presses the left Mouse button and releases it. The duration of a click is the time difference between the Mouse press and Mouse release events in a click. Due to the effective motor skills of a person, there can be significant difference on click duration for different individuals.

Once each Mouse click and curve measurements are computed, a user Mouse profile is created using the mean, average and standard deviation of the all the individual features. This computation results in a feature vector. The following measurements are included in a feature vector [10]:

a) The average and standard deviation of these Mouse click durations
b) The average and standard deviation of these transition times
c) The average and standard deviation of the curvature measurements
d) The average and standard deviation of the transition velocities
e) The average and standard deviation of the transition accelerations

2.3 Classification

The Classifier application checks the validity of the features extracted and classifies the patterns involved in the Mouse Movement characteristics of a user. The Classifier applies the K-Nearest Neighbor (KNN) method to identify unknown Mouse profiles from a set of known user profiles. The nearest neighbor algorithm is a simple classification algorithm. The test data set is classified according to the classification of nearest neighbor from a database of known classification, i.e. a training set. In the general version of this algorithm namely k-nearest neighbor, it outputs k nearest samples from the training set. In the current implementation we used Euclidean distance to find the nearest neighbor. The feature extraction module provides an n-dimensional feature vector of a user. This n-dimensional feature vector is used to compute the distance of an unknown entity and a set of known entities. In N dimensions, the Euclidean distance d between two points, p and q is:

$$\textit{Distance } d = \sqrt{\sum_{i=2}^{n}(p_i - q_i)^2} \quad (5)$$

Where pi (or qi) is the coordinate of p (or q) in dimension i.

The classifier program takes the feature vector of a user as input. The feature vector is then normalized to create a normalized feature vector using the following formula. The normalized feature x' of a feature x is:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (6)$$

Min and max are the minimum and maximum of the measurement over all samples from all subjects. This provides measurement values in the range 0-1 to give each measurement a roughly equal weight.

The classifier program can classify the normalized features using two different methods. In the identification method, the unknown test is classified against a set of known user profiles. In the Leave One Out method, a cross validation is done for all the files in the training set. In the Leave One Out method, one file is compared against rest of the files and the process is repeated for all files [10].

3. Experiments

A total of 205 sets of data were collected to perform further testing of the MMSystem as well as provide to backend data groups who performed their own analysis of the data. Each user was requested to use the data capture training program to collect 5 sets of data a week. The training program comprised of a 25 click button screen arranged in a 5X5 grid. The buttons were enabled for clicking in a particular order. All users used the same order format for clicking on the buttons. The order allowed us to obtain Mouse curves of different lengths; some curves are long and some are short.

The data obtained from the data capture training program (MMSystem) is the input to the feature extractor module. The feature extractor module parses the raw data into Mouse curves and Mouse clicks. The number of data points in each curve is different depending upon the length of the curve. Each raw data file provides many curve and click details for each user and each user can have many files.

3.1 Generating feature vector data

The Feature Extractor module then computed the features for each data file to calculate the average and standard deviation of the curve speed, curve time and curve click duration. A measurement file was created for each of the raw data files. A Screen shot of a measurement file is shown in Table 2 as well as a screen shot of a Feature vector data file is shown in Table 3.

Table 2. A measurement file showing individual curve and click features

Type	Points	Length(pixels)	Total time(msec)	Average velocity(pixels/ms
mouse curve	7	953.4675	375	2.555318
mouse curve	12	2270.89	719	3.292494
mouse curve	9	1215.579	656	2.427516
mouse curve	10	1472.517	641	2.59591
mouse curve	10	1789.098	578	3.174683
mouse curve	10	3041.22	594	5.381281
mouse curve	10	2494.866	688	4.41675

Table 3. Screen Shot of feature vector file – Shows average and standard deviations of features.

Mouse movement biometric sample test data created by team1 on November 12th 2007										
25					Average	SD Curve	Average	SD	Average	SD
User Name	Hand Click	Type of Mouse	Program Used	# of	Curve	Speed	Curve	Curve	Click	Click
				Case	Speed		Time	Time	Duration	Duration
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.383774	1.063485	636.7143	386.5254	95.55172	48.13543
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.354256	1.083917	553.5385	243.3718	89	47.46095
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.019463	1.257007	540.8462	150.4206	88.34615	39.16892
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.811001	1.227672	561.8077	154.6371	75.07692	33.76664
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.013997	1.032733	565.4231	184.5059	95.46	25.97839
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.538493	1.304207	571.1538	319.1655	104.92	14.99578
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.520052	1.468109	518.2692	215.0745	103.2	15.87209
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.471137	0.768032	596.2308	208.8493	93.72	17.59422
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.157805	1.451746	499.3077	222.8064	99.68	22.26073
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.758012	1.506676	683.2308	500.2619	97.72	15.60363
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.044503	1.499596	584.1154	340.7263	114.8	22.99949
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.140682	1.371961	546.8462	321.3871	106.92	18.04596
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.011922	1.516506	558.8846	231.2278	102	23.90307
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.131787	1.655704	543.9615	338.8498	91.36	13.52565
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.986001	1.49176	609.1923	274.8909	91.2	12.98422
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.862105	1.737211	874.3333	1048.127	101.6923	12.77913
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.614694	1.351798	586.1481	203.4187	98.73077	35.34315
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.715189	1.32342	618.4231	226.5235	94.44	54.66647
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.863235	1.470593	585.3077	179.828	97.56	14.3326
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.0219	1.185219	584.6154	328.9136	100.64	13.6927
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.600922	1.050919	566.6296	240.4045	86.5	12.74779
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	2.762265	0.991783	540.8077	202.409	92.36	23.72116
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.304412	1.34983	490.7407	209.306	96.61538	19.36452
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.068122	1.177271	598.7308	271.2777	93.16	17.67786
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	3.230971	1.239768	537.5185	346.7905	88.26923	17.29353

3.2 Classification and Normalization of data

The feature vector file is taken as an input for the classification and Normalization process. This normalized file was also provided to backend teams for further analysis and experiments. A screen shot of a normalized feature vector file is shown in Table 4.

Table 4. A screen shot of a Normalized feature vector file

Mouse movement biometric normalized feature vector data created by team1 on November 21st 2007										
25					Average	SD Curve	Average	SD	Average	SD
User Name	Hand Click	Type of Mouse	Program Used	# of	Curve	Speed	Curve	Curve	Click	Click
				Case	Speed		Time	Time	Duration	Duration
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.132349	0.182314	0.198946	0.009363	0	0.819722
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.090337	0.056143	0.525508	1	0.168187	0.435383
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0	0.260347	0.306433	0.31441	0.2186	0.926259
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.138874	0	0.17509	0	0.042605	0.164547
Antony/M/40	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.12502	0.281028	0.157038	0.002829	0.228953	0.341622
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.714492	0.77591	0.048966	0.022423	0.262261	0.253992
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.78003	0.598747	0.028837	0.018954	0.198795	0.156376
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.717859	0.491743	0.062317	0.017308	0.188893	0.299866
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.6387[illegible]	0.579944	0.00881	0.01833	0.206897	0.060713
Michael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.705946	0.556787	0.030193	0.040664	0.202396	0.153292
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.637989	0.613685	0.091256	0.038331	0.358586	0.016432
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.71474	0.704186	0.151031	0.100524	0.381542	0.383257
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.848886	0.606134	0.031446	0.04952	0.339231	0.136185
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.836028	0.650839	0.058894	0.022293	0.289268	0.241923
Mohammed/M/37	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.75227	0.553744	0.052683	0.07618	0.223759	0.428209
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.232128	0.09699	0.737162	0.469208	0.564739	0.079599
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.349477	0.376055	0.724488	0.382852	0.599848	0.287394
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.296422	0.352295	1	0.733153	0.698111	0.248933
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.406102	0.663414	0.769998	0.488306	0.593996	0.208545
Nkem/F/30	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.279226	0.357677	0.560343	0.467115	0.430154	0.08865
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.69435	0.534907	0.101891	0.028634	0.119021	0
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.796765	0.838307	0.053988	0.003553	0.148451	0.25847
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	1	1	0	0.054013	0.09932	0.31199
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.908174	0.988051	0.014128	0.309971	0.181857	0.205798
Rafael/M/38	right-handed	Optical Mouse	fixed 25 button sequence/used right hand	6	0.985456	0.74826	0.045577	0.056648	0.119575	0.070933

In order to test the data two classification experiments were carried out:

a) Identifying an unknown test case from a user against the training set (K Nearest Neighbor)

b) To do a cross validation using the Leave Out One method

In the Leave Out One method, a training file is validated against the other entire training set of data files. The k-nearest neighbor algorithm is configured to input first 10 nearest neighbors. A sample classification process result for one case is shown in Table 5. This process is repeated for all other cases.

Table 5. Classification process results using Leave Out One method

Testing:Antony			
matching with the closest user:Antony placed at 1			
Antony	0.473274		
matching with the closest user:Antony placed at 2			
Antony	0.481265		
matching with the closest user:Antony placed at 3			
Antony	0.543466		
Mohammed	0.772214		
Mohammed	0.871845		
Nkem	0.905039		
Nkem	0.910634		
Nkem	0.991478		
Michael	0.99497		
Mohammed	1.004956		
Mohammed	1.035502		
Nkem	1.082849		
Michael	1.098134		
Michael	1.12669		
matching with the closest user:Antony placed at 15			

4. Results

The classifier method results from Leave Out One method was analyzed further for success rate. The screen shot of a result file is shown in Table 6. In this particular example, 35 data files were collected. 5 files each from 7 users (team members and 3rd party members). In this particular experiment we obtained a success rate of 80% for the first choice of the nearest neighbor. Matching the second choice was 77%.

Table 6. Screen shot of results based on Leave Out One method

CaseDescription	totalCases	matching	not matching	percentage
Matching first choice	35	28	7	80
Matching second choice	35	27	8	77.14286
Matching first and second choice together	35	20	15	57.14286
Matching third choice	35	14	21	40

A data set containing 205 records with 30 records from each of five subjects, 15 records for one subject, and 10 records for each of four subjects was provided to a backend data mining team for further analysis. This data was collected in the fall of 2007 and was tested in identification experiments.

Authentication tests were run on the Mouse Movement dichotomy data using the K Nearest Neighbor algorithm with k=1. The Table 7 shows the results of the Mouse Movement authentication experiments on the dichotomy data that contained 100 record sets with 5 subjects each.

Table 7. Results of Authentication Experiments on the new Mouse Movement data (collected Fall 2007).

Train	**Test**	**Accuracy**
First 5 Subjects 1000 records	**Last 5 Subjects 1000 records**	56.5%
Last 5 Subjects 1000 records	**First 5 Subjects 1000 records**	56.5%

In addition to the authentication tests on the dichotomy data, an identification test was also run on the new Mouse Movement normalized data set of 205 records.

Table 8 describes the identification experiment, which was run using K Nearest Neighbor with k=1 and the Leave-One-Out procedure. The test used a full data set as training and a full data set as test with the Leave-One-Out procedure. The table below shows that 93% accuracy was achieved on the full data set [5].

Table 8. Results of Identification Experiment on the new Mouse Movement data (collected Fall 2007).

Train	**Test**	**Accuracy**
Full (205 samples from 10 subjects)	**Full (205 samples from 10 subjects)**	93%

5. Conclusions and recommendations

In this particular study, much more data was generated compared with previous iterations of Mouse Movement Biometrics system research. Previous studies achieved a higher success rate of 92% for the first nearest neighbor using the leave-one-out method as opposed to 80% for the current study. 93% was achieved in another test performed by another team using the full set of 205 samples. While 100% accuracy is not probable it seems more experiments need to be performed to see if there is a more consistent accuracy rate over time and from more generated data.

This study also began the process of enhancing the Data Capture system by capturing more user characteristics other than just the user's name. A front-end registration process was created that captured pertinent information regarding the user. For this iteration of the study, only the enhancements to the data Capture module were completed. Work was begun on adding the new data fields to the Feature Extraction and Classifier modules but will need to be continued by succeeding teams.

It is recommended that succeeding teams focus on developing the Data Capture GUI to randomize the buttons to provide more varied data, add more user characteristics to classify the user, add more characteristics of the mouse such as right click or track wheel use and making the overall system more streamlined and user friendly Finally, it would be optimal if the system were to be developed to be used online with a database backend. This would allow for more data to be generated from a larger pool of users for further analysis and research .

6. References

[1] Ahmed Awad E. Ahmed, and Issa Traore, Detecting Computer Intrusions Using Behavioral Biometrics, Department of Electrical and Computer Engineering, University of Victoria, P.O. Box 3055 STN CSC Victoria, B.C. V8W 3P6 CANADA

[2] Carla E. Brodley and Maja Pusara, User Re-Authentication via Mouse Movements, Department of Computer Science, Tufts University, Medford, MA 02155 and School of Electrical and Computer Engineering Purdue University, West Lafayette, IN 47906-2035

[3] M. Villani, M. Curtin, Ngo J. Simone, H. St. Fort, C. Tappert. S. –H Cha. Keystroke Biometric Recognition Studies on Long Text Input over the Internet, CSIS Pace University, Pleasantville, New York 10570

[4] S. Bharati, R. Haseem, R. Khan, M. Ritzmann, and A. Wong, "Biometric Authentication System using the Dichotomy Model," Proc. CSIS Research Day, Pace Univ., May 2008.

[5] Clara Eusebi, Cosmin Gilga, Deepa John, Andre Maisonave, "A Data Mining Study of Mouse Movement, Stylometry, and Keystroke Biometric Data," Proc. CSIS Research Day, Pace Univ., May 2008.

[6] Sandeep Kumar Terence Sim Rajkumar Janakiraman Sheng Zhang, Using *Continuous* Biometric Verification to Protect Interactive Login Sessions, School of Computing, National University of Singapore 3 Science Drive 2, Singapore 117543

[7] Douglas A. Schulz, MOUSE CURVE BIOMETRICS, Pacific Northwest National Laboratory, U.S. Department of Energy

[8] Symantec Internet Security Threat Report Trends for January 06–June 06

[9] M. Calayonnides. Computer Forensics and Privacy – Chapter 14 Practical Encryption: Protecting Proprietary and Confidential Information Against Hostile Penetration, Artech House, Norwood, MA 02062

[10] A.Weiss, A. Ramapanikar, P.Shah, S. Nobel and L. Immohr, "Mouse Movements Biometric Identification: A Feasibility Study", Proc. Student/Faculty Research Day CSIS, Pace University, White Plains, NY, May 2007.

[11] Daniele Gunetti and Claudia Picardi, Keystroke Analysis of Free Text, University of Torino

[12] K. Calix, M. Connors, D. Levy, H. Manzar, G. McCabe, and S. Westcott, "Stylometry for E-mail Author Identification and Authentication," Proc. CSIS Research Day, Pace Univ., May 2008.

[13] Tarjani Buch, Andreea Cotoranu, Eric Jeskey, Florin Tihon, Mary Villani, "An Enhanced Keystroke Biometric System and Associated Studies," Proc. CSIS Research Day, Pace Univ., May 2008.