A New Biometric Technology Based on Mouse Dynamics

Ahmed Awad E. Ahmed and Issa Traore, *Member*, *IEEE*

Abstract—In this paper, we introduce a new form of behavioral biometrics based on mouse dynamics,[1] which can be used in different security applications. We develop a technique that can be used to model the behavioral characteristics from the captured data using artificial neural networks. In addition, we 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. Specifically, three series of experiments are conducted. The main experiment, in which 22 participants are involved, reproduces real operating conditions in computing systems by giving participants an individual choice of operating environments and applications; 284 hours of raw mouse data are collected over 998 sessions, with an average of 45 sessions per user. The two other experiments, involving seven participants, provided a basis for studying the confounding factors arising from the main experiment by fixing the environment variables. In the main experiment, the performance results presented using receiver operating characteristic (ROC) curves and a confusion matrix yield at the crossover point (that is, the threshold set for an equal error rate) a false acceptance rate (FAR) of 2.4649 percent and a false rejection rate (FRR) of 2.4614 percent.

Index Terms—Biometrics, mouse dynamics, security monitoring, network security, human computer interaction.

1 Introduction

Recent years have seen an increasing interest in biometric systems; the underlying technology has improved, and the costs involved have been reduced considerably [10]. Biometrics is defined as the automated use of a collection of factors describing human behavioral or physiological characteristics to establish or verify a precise identity [10]. Physiological biometrics—including finger scan, iris scan, retina scan, hand scan, and facial scan—use measurements from the human body. Behavioral biometrics such as signature or keystroke dynamics use measurements based on human actions. Behavioral biometric systems have experienced less success when compared to physiological systems mainly because of their strong variability over time [2]. A common limitation of most biometric systems is the need for special hardware devices for biometric data collection. A system's scope is restricted to the networks or organizations where these devices are available. Thus, they are insufficient for securing organizations that conduct business with a large and varied population. Keystroke dynamics is an exception, since it can be implemented using regular keyboards [2], [3], [6], [12]. Keystroke verification can be done statically or dynamically. In the static approach, the system checks the user once, typically at authentication time. In the dynamic approach, the system checks the user continuously throughout the session. Since static verification occurs only once, the attacker may still take control of the session later. In contrast, dynamic verification, which is done throughout the session, prevents such an attack. Dynamic verification, however, is still subject to variations within a user's session. Most keystroke dynamics research to date has studied the use of fixed text, restricting its applicability (like all existing biometrics) to access control and eliminating the possibility of using this approach for passive user monitoring. Passive monitoring, however, is a desirable process in computer intrusion detection [1], [5], [11].

In this paper, we introduce a new biometric system based on mouse dynamics; this system addresses most of the shortcomings mentioned above. Similar to keystroke dynamics, mouse dynamics does not require a special hardware device for data collection. In contrast with existing keystroke dynamics, mouse dynamics are collected passively and verified throughout the session. Consequently, mouse dynamics biometrics may be suitable for intrusion detection, in addition to access control. In contrast with keystroke dynamics, which has been widely studied in computer security, previous work on mouse dynamics has been limited mainly to user-interface design improvement [4], [13], [17]. In our research, we target the biometric identification problem by focusing on extracting the behavioral features related to the user and using these features for computer security. Mouse dynamics biometrics involves a signature that is based on selected mouse movement characteristics, which are computed using statistical techniques such as neural networks. In order to establish our signature generation algorithm and validate our biometric system, we have conducted a main experiment involving 22 participants and two smaller experiments involving seven participants. The smaller experiments explored the effects of the confounding factors related to the large experiment. For performance,

1. Patent Pending. Priority Date: 2 May 2003, PCT Filling Date: 3 May 2004, PCT/CA2004/000669. USPTO Application No. 10/555408, 1 Nov. 2005. CIPO Application No. 2535542, 1 Nov. 2005.

• *The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada. E-mail: {aahmed, itraore}@ece.uvic.ca.*

Manuscript received 14 Sept. 2006; revised 11 Apr. 2007; accepted 22 May 2007; published online 12 June 2007.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0131-0906.
Digital Object Identifier no. 10.1109/TDSC.2007.70207.

1545-5971/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

we achieved a false acceptance rate (FAR) of 2.4649 percent and a false rejection rate (FRR) of 2.4614 percent. Although such performance does not yet meet the European standard [15] for acceptable commercial biometrics, it is better than the results achieved with some existing and well-established commercial biometrics such as voice and face recognition systems [14]. Closely related to our work are two recent papers by Hocquet et al. and Pusara and Brodley, respectively. Hocquet et al. report the results of an exploratory study, involving 10 participants, on the use of mouse dynamics for user authentication [7]. Participants in their study are asked to play a game in which they have to click as quickly as possible on a moving square during a fixed length of time. They collect the mouse coordinates and compute several features such as speeds, accelerations, angular velocities, curvature, and the derivative of the curvature curve. Next, they extract from these data the maximum, the minimum, the average, the standard deviation, and the difference between the maximum and minimum. Authentication of an individual is made by comparing such information against some threshold. The performance achieved consists of an equal error rate of 37.5 percent. In our work, we use a different experimental approach, and furthermore, we use a more powerful data analysis technique, allowing us to achieve far better performance.

Pusara and Brodley propose a user reauthentication scheme based on mouse movements [16]. They collect the raw cursor movement data and extract features such as distance, angle, and speed between data points. Using decision tree classifiers, a separate model of normal behavior is built for each application, based on a set of adjustable parameters, which are user specific. The parameters are adjusted by searching through the parameter space to lower false positives and false negatives for each user. Their work is significantly different, however, because their detection model is application dependent. In their approach, they propose to generate a different model for each user per application. Such a scheme cannot realistically be applied for passive monitoring because developing separate models for all possible applications running on a host can quickly become daunting and unmanageable. In contrast, in our main experiment, participants have total freedom to choose which applications they use and how they use them, more accurately reflecting real-life scenarios.

The rest of the paper is organized as follows: In Section 2, we summarize the experimental hypothesis and approach followed in order to evaluate the framework. In Section 3, we describe the characteristics of mouse dynamics and introduce the factors that are relevant to our study. We also describe the architecture of the detector used to collect and process mouse dynamics. In Sections 4 and 5, we describe our experiments, analyze the data collected, and discuss the results obtained. In Section 6, we make some concluding remarks and review the possible applications of mouse dynamics biometrics.

2 Experimental Hypothesis and Approach

In order for us to prove that mouse signatures can be considered as biometrics and can be used for identification purposes, we need to verify that the variations between sessions of an individual user are small compared to the variations between different users' sessions. In order to establish this hypothesis, we conducted three experiments. The first experiment studied the viability of mouse dynamics for passive monitoring and identification in computing systems. In order to reproduce real operating conditions, we gave the participants an individual choice of operating conditions and applications. Consequently, data was collected using a variety of hardware and software systems, with no restriction on the tasks accomplished by the users. The second and third experiments studied the effect of the computing environment on the results of the first experiment; in particular, the applications and operating systems involved. In both of these experiments, the operating system and application were fixed. In addition, all the users in the third experiment were required to perform exactly the same actions by using a specially designed application. In the rest of this paper, we present the mouse detector, describe the experiments, and discuss the results obtained.

3 Mouse Dynamics Detector

In this section, we describe the notion of mouse dynamics and give an overview of the architecture and components of the mouse dynamics detector.

3.1 Mouse Dynamics

Mouse dynamics can be described as the characteristics of the actions received from the mouse input device for a specific user while interacting with a specific graphical user interface. The first step in understanding the actions received from the input device is to identify the categories where those actions fall. A mouse action can be classified into one of the following categories:

- Mouse-Move (MM): general mouse movement,
- Drag-and-Drop (DD): the action starts with mouse button down, movement, and then mouse button up,
- Point-and-Click (PC): mouse movement followed by a click or a double click, and
- Silence: no movement.

An example of raw mouse data is shown in Table 1. Each record consists of the following four fields: the type of action, the traveled distance in pixels, the elapsed time in seconds, and the movement direction. Eight directions are considered: As shown in Fig. 1, the directions are numbered from 1 to 8. Each of the eight directions covers a set of mouse movements performed within a 45-degree area. For instance, direction number 1 represents all actions performed with angles between 0 degree and 45 degrees, whereas direction number 2 is responsible for all actions performed between 45 degrees and 90 degrees.

The characteristics of mouse dynamics can be described by a set of factors generated as a result of analyzing the recorded mouse actions. These factors represent the components of what we have termed a mouse dynamics signature for a specific user, which can be used in verifying the identity of the user. Different approaches can be used in each category to collect the factors characterizing it. Some examples of the type of factors collected from each analysis were the following:

- The average speed was calculated for each distance traveled.

TABLE 1
Sample Raw Data Collected

Type of action	Distance (Pixels)	Time (Seconds)	Direction
MM	26	1	6
MM	839	3	4
Silence	-	3	-
MM	2	0.75	3
MM	7	1.25	6
Silence	-	5	-
PC	238	1	6
PC	14	0.5	1
PC	177	2.25	3
MM	17	1	8
MM	282	3.25	7
MM	6	1.25	7
MM	2	0.5	3
Silence	-	2	-
PC	354	2	7
MM	193	1	1
Silence	-	3	-
DD	307	2	5
PC	353	2.25	3
DD	1	0.25	3
Silence	-	5	-
MM	10	0.5	2
Silence	-	2	-
PC	443	0.75	3
PC	308	2	7

Each row contains the characteristics of an intercepted mouse movement.

- The average speed was calculated in each of the eight movement directions.
- The average traveled distance for a specific period of time, with respect to different movement directions, was calculated.

From such data, we were able to build a usage pattern for the different directions.

The mouse signature developed in this work consists of seven measured factors illustrated in Table 2. Each factor corresponds to a vector of numbers as illustrated later in the paper. For each factor, we need to study its reproducibility and its discrimination capability.

3.2 Detector Architecture and Settings

In order to acquire and process any biometric data, a detection unit needed to be developed. To assist in the design of our detector, prior to this work, we conducted an exploratory study involving five users, none of whom were involved in the validation experiments described later.

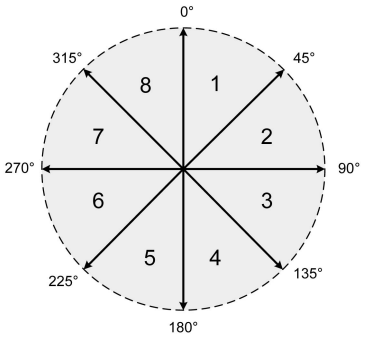


Fig. 1. Mouse movement directions. For instance, direction number 1 represents all actions performed with angles between 0 degree and 45 degrees.

TABLE 2
Factors Involved in a Mouse Signature

Factors	Ranges	Units
Movement Speed compared to traveled Distance (MSD)	25-800	Pixels/Sec
Average Movement speed per movement Direction (MDA)	25-800	Pixels/Sec
Movement Direction Histogram (MDH)	0-100	%
Average movement speed per Types of Actions (ATA)	25-800	Pixels/Sec
Action Type Histogram (ATH)	0-100	%
Traveled Distance Histogram (TDH)	0-100	%
Movement elapsed Time Histogram (MTH)	0-100	%

Users were asked to deploy our data collection software on their machines and conduct their usual activities without any restriction. We collected an average of 5,000 records per user over a one-week period of time. The study was exploratory in that it aimed at suggesting hypotheses rather than testing hypotheses formed a priori. We studied the raw mouse data collected from the five users in order to confirm our intuitions on how this data can be processed for the purpose of our work. This study helped us in both designing the detector and identifying the proper limits used in various system components. In general, a detection unit consists of a biometrics interception device, a data processing module, and a database to store biometrics information.

The detection unit translates the biometrics information into representative data, stores and compares different results, and outputs the result of user identity verification. Fig. 2 shows the design of the mouse dynamics detector. The system consists of three units: data interception unit, behavior analysis unit, and behavior comparison unit. The data interception unit is responsible for transparently intercepting and converting all mouse movements and actions into meaningful information. The behavior analysis unit is responsible for analyzing the processed data, identifying working sessions, and modeling the data to produce the mouse signature. The behavior comparison

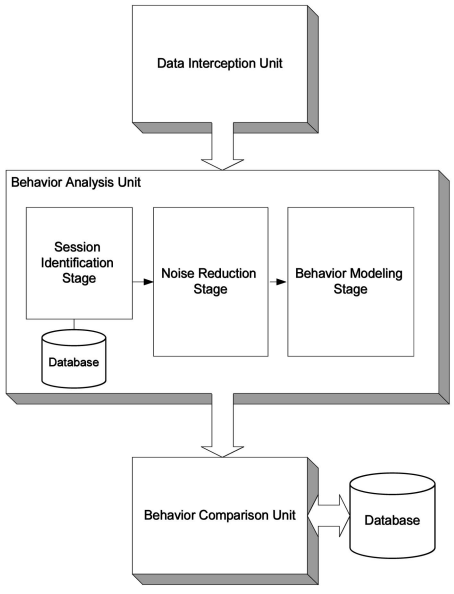


Fig. 2. Mouse dynamics detector architecture.

unit is responsible for comparing the generated signature to the reference signature of the user. This unit maintains a database of all reference signatures calculated for all known system users.

One of the parameters affecting the accuracy of this detector is the screen resolution. If the reference signature has been calculated on a specific resolution and the detection process has been done on a different resolution, this will affect the range of the data collected and will be reflected in the results. Another parameter is the operating system mouse pointer speed and acceleration settings; any changes to these settings can affect the calculated figures and, consequently, the user behavior itself. For example, if the mouse pointer speed is slow, the user will need more than one action to move the pointer along a distance. If the speed is medium, the user can move the pointer with only a single action. The mouse button configuration will also affect the detector. In order to achieve reproducible results, these factors should be fixed for each user on a specific workstation; the detector can force those settings in the operating system and verify the consistency of the settings over the detection period. Hardware characteristics such as the workstation speed and the mouse input device type and speed can also affect the interception.

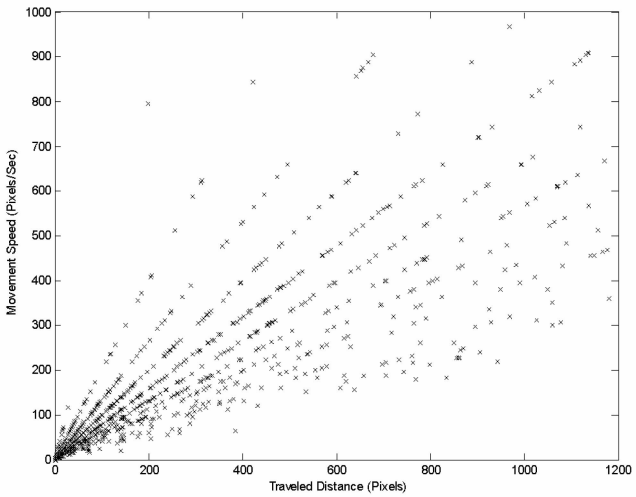


Fig. 3. Graph showing a relationship between speed and distance based on sample intercepted data. Like many behavioral biometrics, the raw mouse data shows strong variability over time. The data interception unit collects mouse actions periodically, producing the pattern of rays; in our implementation, actions are collected every quarter second. Speed is calculated by dividing distance over time, which is quantized to multiples of 0.25 of a second, producing this pattern.

3.3 Data Collection and Processing

In this section, we describe in detail the main tasks performed by the data interception unit.

3.3.1 Data Interception

The data collected by the detector is a list of actions such as an MM event, left-button-down event, left-button-up event, and so on. Such events do not provide meaningful information for analyzing the behavior. Consequently, it becomes the responsibility of the interception software to translate those events into meaningful actions. For example, a set of actions that is considered to be a good input to the behavior analysis unit could be represented by the following series of events, measured in milliseconds:

- a mouse movement from a position to another position,
- a period of silence, and
- another mouse movement ended by a click or a double click.

The interception software continuously feeds the behavior analysis unit every time mouse actions are detected on the monitored workstation. Fig. 3 illustrates the relationship between speed and distance based on the sample data intercepted. The x-axis represents the traveled distance, and the y-axis represents the movement speed. Each point on this figure represents an intercepted mouse action. For simplicity of the example, we ignore the effects of the type of action and movement direction. The screen resolution used for this session was 1,024 $\times$ 768. As the figure illustrates, traveled distances are less than or equal to 1,200 pixels, and actions with smaller distances occur more often than those with longer distances. The data interception unit monitors actions periodically; in our implementation, actions are monitored every quarter second. Speed is calculated by dividing distance over time, which is quantized to multiples of 0.25 of a second; this explains the pattern of rays (with blanks in between) in Fig. 3. In addition, observe in Fig. 3 that the movement speed increases and becomes more variable as the traveled distance increases. The challenge is how to convert this data into a model representing the user's behavior. Furthermore, we must determine how to extract factors so that the user's mouse dynamics signature can be constructed.

3.3.2 Session Identification

The session identification stage presents the data in batches to be available for processing. This module identifies the start and end of the session as follows:

- Session start is determined if an action is received for a specific user and there were no current sessions in effect for this user.
- Session end is determined if the number of actions collected in the current session reached the maximum limit or the duration of the session exceeded its maximum limit (a detailed analysis of the effect of the session length is presented in Section 5.3).

A session tag will be associated with each session; this tag contains session information such as user name, machine name, Internet Protocol (IP) address, start time/date, and end time/date.

3.3.3 Noise Reduction

The accuracy of the behavior modeling stage can be affected by the nature of the data provided. The data collected from each user can fall in different ranges because of the effect of an environment setting such as screen resolution. The collected data can also contain incorrect readings generated because of hardware or software error, such as reporting a long movement performed in a very short time, an action impossible for a human user to produce. A filtration process is needed to eliminate the effect of those factors before submitting the data to the modeling process. As an example, we consider the movement speed compared to the traveled distance graph shown in Fig. 3, which illustrates how the

session data appeared before the filtration takes place. Two filters are applied before sending the data to the behavior modeling stage. The first filter restricts the input data on the x-axis to a specific range, eliminating any data above or below that range; in our implementation, this range was determined to be 25–900 pixels. The second filter is used to eliminate any y-axis (that is, movement speed) reading that highly deviates from the mean of the y data, because such a reading can affect the modeling process. The lower limit of the first filter (25 pixels) was selected because a large number (on the average, 26.1 percent of the total number of actions in a session) of very short distances with very low elapsed times were observed for all users (involved in the exploratory study). These readings do not have any significant effect on the generated model; however, omitting them increases the speed of the modeling process. For instance, based on a sample of 12,000 records (collected in the exploratory study), by filtering these readings, we observe a decrease of 32.14 percent in the time needed to build a model. The upper limit (900 pixels) was selected as the least upper bound of the maximum distances obtained for all users. This margin was set to eliminate the effect of different users producing sessions based on different screen resolutions. The physical movement of the mouse device gets amplified to a cursor movement on the screen. The small distance moved on the mouse pad is mapped to a larger distance (measured in pixels) on the screen. This mapping is not affected by the screen size or resolution since the resulting movement distance remains in screen pixels. Hence, our adopted approach of dropping the outliers will have no effect on this mapping. An alternative approach is to normalize the data to a specific screen resolution. This, however, will affect the mapping, making it unsuitable to compare between different users' data using the same measuring units. The limit of the second filter was determined by studying the histogram of the movement speeds observed for all users. We found that speeds over 800 pixels/seconds rarely occur (on the average, 1.47 percent of the total number of actions in a session); speeds beyond this limit can be considered as noise, since it is not practical to expect such high speeds as a result of a normal human computer interaction.

3.4 Behavior Modeling

According to the characteristics of the factors, various statistical packages can be used to process the output of the noise reduction stage and generate a pattern characterizing user behavior. As indicated earlier, the mouse dynamics signature consists of the combination of seven factors organized in five categories: movement speed (MSD), movement direction (MDA, MDH), action type (ATA, ATH), traveled distance (TDH), and elapsed time (MTH). In the remainder of this section, we analyze each of these factor categories and illustrate the modeling techniques used.

3.4.1 Movement Speed

For the Movement Speed compared to Traveled Distance (denoted MSD) factor, we use neural networks to approximate the collected data to a curve that can be used to identify the user behavior. The Levenberg-Marquardt algorithm was used for training (using Matlab v7.0). In our design, a split-sample validation technique was used to evaluate different neural network configurations. The result of this test concluded that a network with a single hidden layer containing five nodes is sufficient to produce a good result. Fig. 4 provides some examples of mouse signatures based on the MSD factor. The two curves in Fig. 4a are for the same user; observe that the two curves are close to each other and that the difference between them is low. Fig. 4b illustrates two curves for two different users. The difference between these curves is high, indicating a high difference in the behaviors and a high possibility that the behaviors belong to two different users.

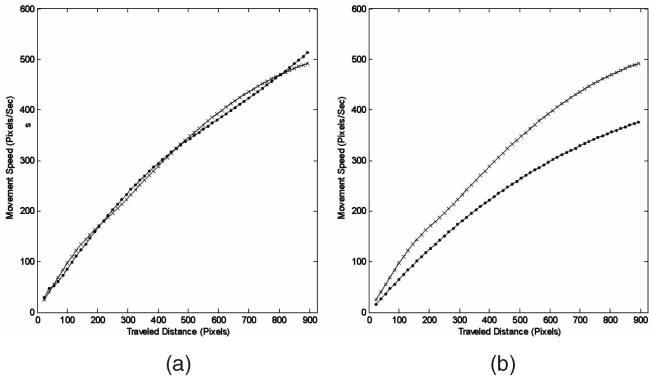


Fig. 4. MSD curves over different sessions: In (a), the sessions belong to the same user, whereas in (b), they belong to different users.

Twelve points $\{(x_i, y_i)|0 \leq i \leq 11\}$ over the MSD curve, obtained through periodic sampling, are used to represent this factor in the signature. Let $y = f(x)$ denote the function characterizing the curve, and let x_{min} and x_{max} denote the lower and upper bounds of the x parameters, respectively. The points (x_i, y_i) are determined as follows:

$$(x_0 = x_{\min}; y_0 = f(x_0)) \quad \text{and for } 1 \leq i \leq 11$$
$$\left(x_i = \frac{(x_{\max} - x_{\min})}{11} + x_{i-1}; y_i = f(x_i)\right).$$

3.4.2 Direction of Movement

The analysis of the impact of the direction of movement resulted in the definition of two different factors. The first factor involves the calculation of the average speed in each of the eight directions of movement. This factor is referred to as the Average Movement speed per movement Direction (denoted MDA). The second factor involves calculating the percentage of actions occurring in each of the eight movement directions. This factor is referred to as the Movement Direction Histogram (denoted MDH). Fig. 5a shows the distribution of the average movement speed against the direction of movement for two different users, User 1 and User 2. Observe that movements in directions 2, 3, 6, and 7 are performed with a higher speed than movements in directions 1, 4, 5, and 8. Fig. 5b shows the histograms of the performed actions in each direction for two different users, User 1 and User 2. Observe that some directions involve more actions than others.

Furthermore, there is usually a direction that involves more actions than any of the other directions. User 1 performs more actions in the third direction, whereas User 2's actions dominate more in the fourth direction. MDA and MDH factors are each represented by eight

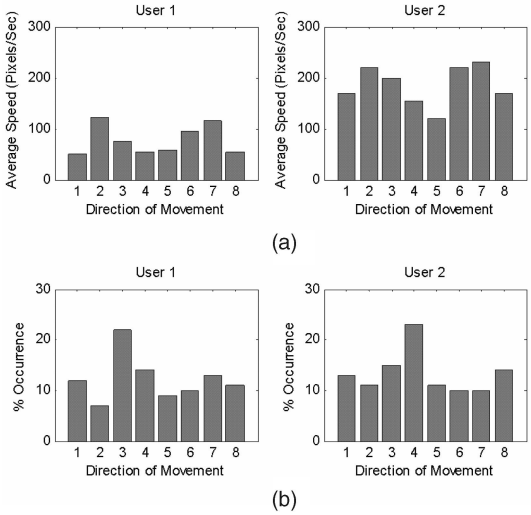


Fig. 5. (a) Average speed for different movement directions for two different users. Notice that movements in directions 2, 3, 6, and 7 are performed with higher speed than movements in directions 1, 4, 5 and 8. (b) Histogram of the directions of movement for two different users. Notice that some directions involve more actions than others. Each of the histograms was created using 2,000 actions (of all types) collected over one user session.

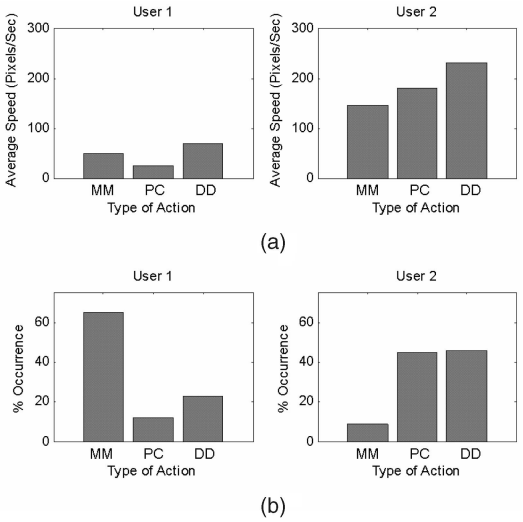


Fig. 6. (a) Average speed for different types of actions for two different users. (b) Histogram of the types of actions for two different users. Each of the histograms was created using 2,000 actions collected over one user session.

numbers (corresponding to the directions of movement) to be added to the user's signature.

3.4.3 Type of Action

We conduct this analysis based on the fact that the type of action the user is performing affects his or her behavior.

The following three types of actions are considered: PC, DD, and MM. The type of action is studied with respect to the movement speed (denoted ATA) and the distribution of the performed actions over the three types of actions (denoted ATH). Fig. 6a shows the relation between the movement speed and the type of performed action for the three recognized types of actions. Two pieces of information can be extracted from the bar chart: the range of each entry and the ratios between the entries. It is possible to depend on this factor for identification if the ratios between the entries are constant. For instance, the speed of movement for User 1 in Fig. 6a is at its lowest level for the PC type of action compared to other types of actions. Fig. 6b shows the histogram of the types of actions for two different users. Behavioral differences can be easily detected for the two users, and ratios between entries can easily be identified. In the figure, we notice that User 1 performs more MMs than any other type of action, whereas User 2 performs fewer MMs compared to other types of actions such as PC, and DD, which are approximately the same. ATA and ATH factors are each represented by three numbers (for the three types of actions) to be added to the user's signature.

3.4.4 Traveled Distance

The histogram of the traveled distance (denoted TDH) illustrates the distribution of the number of actions performed by the user within different distance ranges.

Usually, the number of actions performed with short distances is higher than those performed with long distances. This distribution, however, differs from one user to another. Fig. 7 shows a comparison between two users. User 2 depends more on short distances in performing actions; 64 percent of his or her actions are within the 0–100 pixel range. User 1 performs only 33 percent of his or her actions in the same range. As the probability of occurrence of large distances is usually low (below 15 percent), it is possible to depend only on the first two bars of the graph to represent this factor in the signature.

3.4.5 Movement Elapsed Time

The elapsed time is the time spent by the user to perform an action; it depends on the traveled distance and the type of the performed action. The movement elapsed time histogram (denoted MTH) illustrates the distribution of the number of actions performed by the user within different time ranges during a user's session.

Fig. 8 shows MTH diagrams for two users. Each histogram shows the distribution for actions, performed

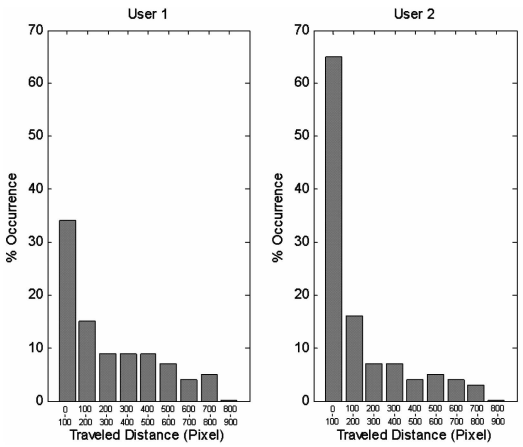


Fig. 7. Traveled distance histograms for two different users. Each of the histograms was created using 2,000 actions (of all types) collected over one user session. Each bar represents a range of distances. Notice that the number of actions performed with short distances is higher than those performed with long distances. Only the first two bars of the graph may represent this factor.

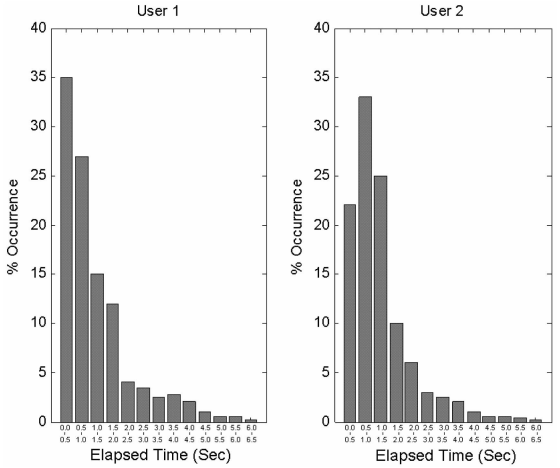


Fig. 8. Elapsed time histograms for two different users. Each of the histograms was created using 2,000 actions (of all types) collected over one user session. Each bar represents a specific time range. The first three bars of the graph are used to represent this factor.

within or before 6.5 seconds, during one user's session that included 2,000 actions of all types. The first two bars of the graph provide significant behavioral information. See the following examples:

- For User 1, the first bar in the graph (0–0.5 seconds) represents around 35 percent of the total number of actions, whereas for User 2, this represents only 22 percent of his or her actions.
- For User 1, the mode of the distribution is the first bar of the graph (0–0.5 seconds), whereas for User 2, it is the second bar (0.5–1.0 seconds).

For more accuracy, the first three bars of the graph are used to represent this factor in the user signature.

3.5 Behavior Comparison

The behavior comparison approach adopted in this work consists of using neural networks to detect differences between behaviors. The neural network automatically upweights the most reliable discriminating factors and improves performance by eliminating the remaining factors. Specifically, a neural network is trained for each user during the enrollment process, and a different combination of training data is prepared for each user, whereas the design of the networks is the same. The status of the trained network is stored in the signatures database. In the detection mode, the behavior detection unit loads the legitimate user's stored neural network status. The saved status is then applied to the network, and the monitored behavior resulting from session analysis is applied to the neural network. We refer to the output of the network as the confidence ratio (CR), a percentage representing the degree of similarity of the two behaviors. In order to use this number in the detection process, a threshold limit is set to determine if the obtained CR is sufficient to ensure the identity of the user.

The neural network used in the detection process is a feed-forward multilayer perceptron network consisting of three layers. The input layer consists of 39 nodes, the total number of inputs representing the factors involved in the mouse dynamics signature. Table 3 shows the inputs to the network consisting of a set of numbers describing the signature. The hidden and output layers consist, respectively, of 40 nodes and 1 node. The expected output range is from 0 to 100. Inputs and outputs are normalized so that they fall in the [-1, 1] range. The output of the network can be defined as

TABLE 3
Inputs for Behavior Comparison Neural Network

Factors	MSD	MDA	MDH	ATA	ATH	TDH	MTH
Neural Network Inputs	12	8	8	3	3	2	3

The network takes a total of 39 inputs (12 points over the curve for MSD, eight directions for MDA and MDH, three types of actions for ATA and ATH, and the first two and three histogram bar readings for TDH and MTH, respectively). The output of the network corresponds to the CR expressed as a percentage.

$$CR = \left(\left(\sum_{j=1}^{N+1} w_{2j} \cdot \frac{1}{1 + e^{\left(\sum_{i=1}^{N} w_{ij} s_i \right) - b_{1j}}} \right) - b_{21} \right) \times 100,$$

where S_i represent the inputs to the network, w_{kl} and b_{kl} represent, respectively, the weight and bias of node l of layer k ($k = 1$ for the hidden layer, and $k = 2$ for the output layer), and N is the number of nodes in the input layer ($N = 39$). The back propagation algorithm is used to train the network. To expedite the learning process, we use the Levenberg-Marquardt algorithm for training (using Matlab v7.0); on a dual Xeon 2.8-GHz server with 2 Gbytes of RAM, the training of the detection network takes on the average 0.39 seconds. The training data is designed as follows:

- Positive training consists of data collected from five sessions for the user trained for an output of 100, meaning a 100 percent confidence in identity.
- Negative training consists of data collected from other users based on five sessions per user with an output of 0, meaning a 0 percent confidence in identity.

In real life, the sessions used for training are the first sessions of the user as they are used for enrollment; the (other) users whose sessions are used for negative training are selected from the list of known (legitimate) users. There is no restriction on their number; however, the higher the number of users, the better the training. For the purpose of the experiments described later in this paper, we selected the users and the training sessions randomly. To illustrate the detection process, Table 4 shows sample training data for five different users. The sample data consists of five sessions per user; for each session, all of the seven factors representing the calculated signature are shown. The output shown (column P) was set to train the network for the first user (that is, User 1) as the legitimate user. During the training phase, the output of the network will be set to 100 for User 1's sessions and to 0 for the other four users' sessions. Column CR shows the CRs calculated for all of the 25 sessions after the network has been trained. As shown in the table, the CR values are very high for User 1 and very low for the other users, indicating that the trained network is able to differentiate between the five users.

TABLE 4
Training Data for Five Different Users

	MDA								MDH								TDH		MTH		
User 1	245.4	306.6	307.7	139.3	182.0	261.2	267.0	248.4	9.865	11.99	14.79	15.69	12.33	13.45	11.77	9.977	30.38	17.93	12.55	9.865	36.54
	184.8	284.9	264.0	120.7	146.9	282.1	326.6	261.9	10.90	11.76	15.82	17.43	10.90	12.40	10.16	10.48	34.11	17.32	12.51	9.411	40
	266.4	302.2	252.7	170.1	199.3	276.4	261.1	196.8	10.38	10.06	16.31	15.89	9.639	12.28	12.5	12.81	27.75	20.02	13.24	7.944	37.5
	179.5	268.1	256	168.4	190.9	285.9	271.9	233.9	8.437	11.74	14.70	18.35	10.03	10.83	10.94	14.82	31.24	16.87	12.08	7.411	37.4
	252.6	301.5	272.9	175.5	165.1	297.7	268.2	247.0	7.330	13.56	19.03	13.34	9.299	13.56	12.58	11.16	25.82	16.30	12.80	8.424	35.55
User 2	197.7	295.6	295.4	217.4	206.3	279.3	275.8	260.9	7.781	9.570	17.71	15.02	6.887	9.749	19.05	14.13	22.45	18.33	12.34	5.724	42.57
	211.7	227.5	200.1	230.9	250.1	207.9	229.0	293.8	12.1	12.53	17.37	11.66	8.729	12.53	13.31	11.66	31.28	17.891	12.79	11.06	41.4
	153.1	168.8	212.4	185.6	169.7	177.2	243.6	228.3	11.35	13.21	18.04	14.03	7.943	10.24	14.69	10.39	42.09	17.07	10.31	18.63	42.24
	242.0	249.6	302.7	229.1	256.3	264.1	384.4	270.4	8.514	12.40	21.10	9.782	7.880	10.32	20.56	9.329	26.63	14.94	12.95	9.329	44.56
	172.8	238.5	239.6	176	225.5	233.2	232.5	211.9	11.02	12.22	18.19	10.38	7.536	14.43	16.17	9.926	28.86	18.29	11.58	6.801	39.89
User 3	165.4	169.1	170.7	161.5	138.0	175.8	199.6	157.1	7.782	15.47	20.99	8.235	9.230	13.93	16.65	7.601	34.38	19.91	13.66	5.701	31.04
	163.2	200.5	184.7	159.5	153.8	190.2	211.0	188.8	13.60	11.35	12.32	11.74	15.36	13.30	9.686	12.52	30.62	20.93	13.99	8.023	32.29
	147.3	191.2	193.3	156.7	150.6	216.4	213	166.4	12.22	12.13	15.56	10.88	13.27	11.17	12.22	12.41	28.17	19.96	14.04	6.208	28.65
	145.1	148.9	120.1	137.8	137.9	160.5	152.3	165.9	9.110	12.95	18.77	10.75	10.42	10.86	16.79	10.20	42.7	15.91	12.29	6.805	31.50
	162.3	207.6	202.4	195.0	152.7	193.7	242.3	203.2	10.49	14.75	14.66	10.49	11.31	11.85	15.29	11.04	31.49	19.00	11.49	8.144	31.76
User 4	231.0	258.7	203.9	151.0	139.8	316.2	253.7	177.3	12.13	16.06	15.72	14.56	10.63	11.67	10.86	8.208	30.75	18.72	10.75	4.971	31.21
	244.2	272.4	250.4	212.1	251.2	285.3	254.7	226.4	14.68	17.29	12.91	11.97	10.62	14.06	8.958	9.375	25.31	19.89	11.45	8.437	38.02
	249.8	319.4	304.6	220.2	184.9	367.9	303.4	199.0	15.19	17.23	9.977	12.47	12.81	14.62	8.730	8.843	26.64	15.07	12.47	8.616	36.16
	266.5	245.8	241.6	243.9	229.6	332.0	319.1	243.9	11.35	17.39	17.60	11.04	9.687	12.70	12.70	7.395	24.89	18.22	12.81	7.5	39.16
	276.6	334.6	266.6	187.2	187.8	293.8	348.8	230.6	12	14.97	14.17	12	11.42	12	14.74	8.571	23.65	15.77	11.77	5.828	37.14
User 5	238.0	325.6	256	176.2	230.1	232.9	193.5	184.7	9.153	11.61	14.61	12.70	9.016	11.06	18.85	12.84	30.32	19.67	12.56	7.513	31.42
	215.8	295.7	232.5	172.6	211.1	245.4	182.3	215.3	10.43	10.43	13.46	13.91	11.44	11.22	14.36	14.59	32.54	17.05	17.05	10.88	36.7
	238.1	231.7	267.7	230.4	201.6	222.7	204.4	182.6	10.41	10.19	14.25	12.5	10.30	13.70	15.46	13.04	31.68	18.53	11.18	8.114	37.17
	139.3	170.9	210.2	169.4	188.3	194.5	236.0	159.2	9.154	14.96	20.33	10.29	6.866	11.62	19.19	7.482	46.21	16.37	7.922	12.06	43.48
	210.9	264.0	224.4	224.0	196.5	191.3	183.0	177.5	8.117	10	14.82	14.11	9.529	10.70	16.70	15.88	32.35	20.58	12.11	6.470	38.11

	MSD												ATA			ATH			P	CR
User 1	45.12	108.2	170.4	230.7	288.0	341.7	391.0	435.8	476.0	511.5	542.6	569.6	232.6	226.3	278.4	73.31	4.484	22.08	100	99.69
	216.3	226.7	228.2	228.4	228.4	228.4	228.4	228.4	228.4	228.4	228.4	228.4	198.8	230.7	337.4	74.75	4.919	20.21	100	99.92
	46.22	109.3	157.8	214.4	275.2	334.7	388.0	432.1	466.3	491.3	509.0	521.1	222.7	210.4	276.3	65.36	5.508	29.02	100	99.59
	38.33	100.5	160.3	216.7	268.8	316.1	358.2	395.2	446.6	474.1	497.5	517.1	209.4	171.7	273.0	58.95	5.359	35.57	100	99.88
	31.20	106.4	179.3	244.9	300.5	345.0	379.2	404.6	422.9	436.1	445.9	455.4	238.2	306.1	302.5	78.55	4.376	16.95	100	99.81
User 2	38	260.5	260.5	260.5	260.5	260.5	260.5	260.5	260.5	260.5	260.5	260.5	198.5	125.2	298.0	29.07	5.366	65.47	0	0.084
	46.61	110.3	171.9	230.1	283.8	332.4	375.5	413.1	445.4	472.9	496.0	515.2	164.3	115.4	261.5	26.53	4.753	68.62	0	0.103
	195.4	195.4	195.4	195.4	195.4	195.4	195.4	195.4	195.4	195.4	195.4	195.4	147.9	128.0	213.1	21.01	4.751	74.16	0	0.23
	41.41	107.7	182.2	252.9	311.4	356.5	394.2	432.8	479.4	533.7	585.8	625.4	209.9	138.0	334.5	28.08	4.891	66.93	0	0.24
	41.48	102.6	159.4	211.7	259.5	302.8	341.8	376.8	407.9	435.5	460	481.4	165.5	122.8	262.6	28.95	10.66	60.29	0	0.12
User 3	36.61	91.21	141.7	186.6	224.8	256.3	283.4	347.3	362.5	374.1	382.8	388.4	125.8	134.6	210.6	40.54	7.149	52.21	0	0.12
	40.17	102.9	155.7	198.6	232.4	258.5	278.2	337.6	348.4	356.4	362.2	366.5	147.1	130.2	221.8	47.45	7.240	45.20	0	0.12
	41.70	94.62	143.9	188.8	228.9	263.9	294.2	319.9	341.4	359.3	374.1	286.0	137.1	146.6	228.5	46.60	7.831	45.46	0	0.043
	34.22	86.08	132.8	174.4	211.3	243.7	271.9	296.4	317.5	335.7	351.2	364.4	107.4	143.6	183.4	48.84	4.720	46.32	0	0.024
	44.63	97.11	154.1	209.2	256.7	293.7	320.4	338.6	396.3	403.9	408.7	411.6	179.4	150.4	223.2	43.62	8.687	47.60	0	0.007
User 4	47.83	92.87	137.4	181.5	225.2	268.3	310.8	352.8	394.2	435.0	475.1	515.0	175.7	165.6	279.3	54.68	4.161	41.04	0	0.009
	251.8	251.8	251.8	251.8	251.8	251.8	251.8	251.8	251.8	251.8	251.8	251.8	232.6	131.3	280.1	49.37	3.229	47.29	0	0.16
	272.7	272.7	272.7	272.7	272.7	272.7	272.7	272.7	272.7	272.7	272.7	272.7	240.0	205.0	316.1	51.92	3.401	44.55	0	0.308
	46.21	105.0	164.6	223.6	281.0	335.6	386.7	433.5	475.8	513.4	546.3	575.0	254.1	168	283.8	47.08	3.437	49.37	0	0.048
	36.57	103.9	166.9	225.5	279.8	329.6	375.3	416.8	454.5	488.5	519.2	546.7	244.8	196.1	310.3	51.08	4.457	44.34	0	0.002
User 5	226.4	226.4	226.4	226.4	226.4	226.4	226.4	226.4	226.4	226.4	226.4	226.4	173.0	386.2	280.7	53.14	2.868	43.85	0	0.206
	33.33	106.5	172.6	231.7	284.0	329.9	369.7	404.1	433.7	458.9	480.4	498.6	185.3	294.0	249.2	50.16	2.581	47.13	0	0.031
	40.63	103.5	162.4	216.9	267.0	312.6	353.8	390.7	423.7	452.9	478.7	501.4	188.0	149.1	262.8	47.69	4.166	48.02	0	0.003
	37.56	105.0	167.6	224.9	276.9	323.6	365.1	401.7	433.9	461.9	486.2	507.4	196.7	88	228.4	28.34	19.89	51.67	0	0.044
	42.63	92.87	149.5	207.7	261.6	307.3	343.0	369.3	387.7	555.6	563.8	568.7	174.9	198.8	251.2	53.76	6.705	39.41	0	0.243

The sample data consists of five sessions per user, each represented rowwise. The signature calculated for each session is shown. Each of the seven factors is represented by a set of columns corresponding to the numbers involved. As an example, cell [2, 4] in the first table shows that the average speed for actions performed in direction number 4 (representing the fourth number in the MDA factor) for the second session for User 1 is 120.7 pixels/seconds. The output (P) shown was set to train the network for the first user (that is, User 1). The CRs for all the included sessions after the network has been trained for the first user are shown as well. Notice that the trained network is able to differentiate between the five users.

Table 5 shows a sample of five other sessions for User 1, and the CRs computed using his or her trained neural network. As shown by the table, the resulting CRs are (relatively) close to 100 percent, indicating the ability to recognize this user.

3.6 Evaluation Metrics and Procedures

In order to evaluate the accuracy of mouse dynamics biometrics, we compute the following:

- False Acceptance Rate (FAR), which measures the likelihood that an imposter may be erroneously accepted by the system, and
- False rejection Rate (FRR), which measures the likelihood that a genuine user may be rejected by the system.

We compute the performance metrics through a one-hold-out cross-validation test. Considering a user base of size n, the test is repeated n times by considering each time, one of the n users, say, User $j(1 \leq j \leq n)$, as an outsider. In the following, $\{CR\}_{ij}^{p}$ denotes the set of CR computed by comparing all the sessions of User j against the reference signature of User i in test round p. In each round j $(1 \leq j \leq n)$, we conduct two sets of comparisons. First, we compare all the sessions of the outsider User j against each of the reference signatures of the $n-1$ others, who for the iteration are considered as insiders. The training of the detection neural network is based on sample sessions selected randomly for each of the $n-1$ insiders. In each round, the sessions of the user considered as the outsider are excluded from the training. By comparing the sessions of

TABLE 5
Sample User Signatures

	MDA								MDH								TDH		MTH		
Session 1	263.9	326.4	318.9	117.9	221	341.1	309.4	271.2	9.955	10.17	14.87	19.79	9.39	12.08	14.09	9.507	29.30	17.56	11.07	9.172	38.47
Session 2	234.0	259.7	248.1	169.4	187.0	256.5	305.4	290.4	11.66	10.69	13.60	16.63	11.77	11.66	12.31	11.55	29.15	19.87	12.31	10.36	38.87
Session 3	263.8	308.3	235.3	132	133.5	322.6	272.3	184.9	10.94	11.35	12.29	26.21	8.108	12.29	7.162	11.48	38.51	15.94	9.729	9.324	36.21
Session 4	216.1	276	253.5	146.7	176.5	240.7	287.1	237.1	9.477	11.42	17.86	15.18	10.69	14.21	9.356	11.66	31.34	16.16	14.21	6.925	37.54
Session 5	186.1	340.3	289.7	167.0	164.3	336.4	329.7	224.6	9.718	12.05	14.48	17.29	11.27	10.30	12.73	12.05	29.83	19.04	12.92	9.523	42.85

	MSD												ATA			ATH			CR
Session 1	35.09	104.6	167.9	225.3	315.7	362.2	403.8	440.9	473.8	503.0	528.8	551.6	258.9	194.3	288.3	75.72	4.362	19.79	100
Session 2	75.78	94.87	137.4	260.1	315.6	345.1	356.9	478.2	479.6	480.0	480.2	480.2	215.1	325.4	293.7	70.62	7.019	22.24	83.46
Session 3	138.3	139.1	139.5	139.7	139.9	139.9	140.0	140.0	140.0	140.0	140.0	140.0	205.2	226.9	277.7	78.37	3.648	17.83	95.73
Session 4	35.15	97.12	159.0	219.0	275.7	327.7	374.3	415.2	450.3	480.1	504.9	525.4	221.8	193.1	253.5	72.53	4.981	22.35	100
Session 5	31.66	110.2	181.4	245.3	302.1	352.2	396.1	434.3	467.4	495.9	520.4	541.4	222.6	191.4	328.5	62.48	7.191	30.22	98.33

Sample signatures for five sessions for User 1, and the CRs calculated using a neural network trained using the data in Table 4. Notice that the CRs are (relatively) close to 100 percent, which indicates the ability to recognize this user.

the outsider User j against the signature of an insider User $i(1 \leq i \leq n, i \neq j)$, we compute the corresponding set of CRs $\{CR\}_{ij}^{j}$. Second, we compare the reference signature of each of the $n-1$ insiders against all of his remaining sessions (not involved in the computation of the signature), as well as all the remaining sessions of the $n-2$ other insiders. Considering an insider User $i(1 \leq i \leq n, i \neq j)$, we derive the set $\{CR\}_{ii}^{j}$ containing the CRs computed by comparing all of his or her remaining sessions against his or her reference signature. By comparing all the remaining sessions of insider User $k\,(1 \leq k \leq n, k \neq i, k \neq j)$ against the reference signature of User i, we compute the set $\{CR\}_{ik}^{j}$. Given User i and User $j(1 \leq i, j \leq n)$, we derive the global set of CRs over all n test rounds as $\{CR\}_{ij} = \bigcup_{p=1, p\neq i}^{n} \{CR\}_{ij}^{p}$. A misclassification is established if the CR is below the threshold limit when $(i = j)$ or above or equal to the threshold limit when $(i \neq j)$. We conduct the testing by varying the threshold limit for CR from 5 percent to 95 percent and derive for each set $\{CR\}_{ij}$ the number M_{ij} of corresponding misclassifications. For each threshold value, the overall FAR and FRR is computed based on all of the misclassifications observed over all the n testing rounds. This gives

$$FAR = \frac{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} M_{ij}}{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} count(\{CR\}_{ij})} \times 100$$

and

$$FRR = \frac{\sum_{i=1}^{n} M_{ii}}{\sum_{i=1}^{n} count(\{CR\}_{ii})} \times 100.$$

4 Main Experiment

In this section, we describe and discuss the procedure and results for our main experiment.

4.1 Method

Twenty-two participants, 16 males and 6 females, with varying computer skills and ages ranging from 13 to 48 years, were involved in this experiment. Each user was asked to install the client software on his workstation and to use the workstation for his daily use throughout the duration of the experiment. Although the users provided informed consent to participate in the experiment, data was collected transparently during the routine use of their own computers. The data collected was sent directly to a central server located in our lab. The tasks performed by the users varied from Web browsing to word processing and video game playing.

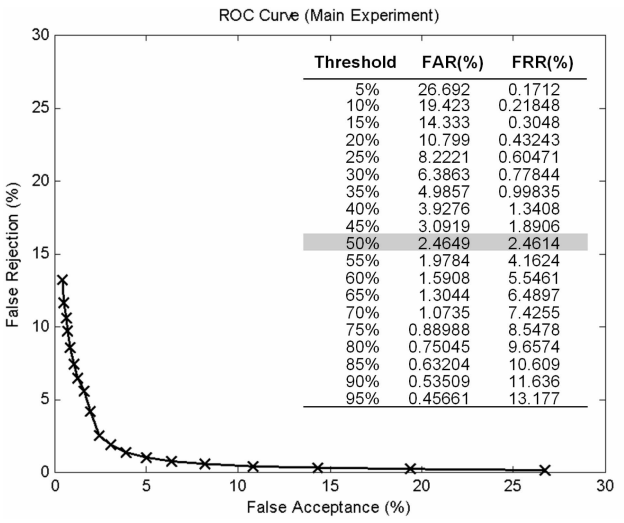
4.2 Apparatus

The experiment configuration involved the client software deployed on remote workstations connected to a central server via the Internet. The client software (responsible for monitoring mouse actions) fed a detection server (software) with the monitored data. The client software, which runs as a background job, starts monitoring user actions when the user login occurs and stops running when the user logout occurs; the software is totally transparent and does not affect any other applications.

The detection server was installed on a local area network and accepted connections from local workstations and from outside the network over the Internet to allow remote users to participate in the experiment. A large number of participants were connecting remotely to the network from their home computers; however, several of the users also connected from national or international locations during the experiment. The server software stored the collected data in an internal database, along with other information, including the user ID. The hardware configurations for the participating computers varied from a Pentium 2 266-MHz processor to a Pentium 4 1.5-GHz processor. The server configuration was a Pentium 3 450-MHz processor with 256 Mbytes of RAM, running the Windows 2000 operating system. The client workstations ran different versions of the Microsoft Windows operating system (Windows 98SE, Windows ME, Windows 2000, and Windows XP).

4.3 Data Collected

Data was collected over a total of 998 sessions with an average of 45 sessions per user. The entire experiment lasted nine weeks. Overall, 284 hours of raw mouse data was collected, with an average input of 12 hours and 55 minutes per user.



Threshold	FAR(%)	FRR(%)
5%	26.692	0.1712
10%	19.423	0.21848
15%	14.333	0.3048
20%	10.799	0.43243
25%	8.2221	0.60471
30%	6.3863	0.77844
35%	4.9857	0.99835
40%	3.9276	1.3408
45%	3.0919	1.8906
50%	2.4649	2.4614
55%	1.9784	4.1624
60%	1.5908	5.5461
65%	1.3044	6.4897
70%	1.0735	7.4255
75%	0.88988	8.5478
80%	0.75045	9.6574
85%	0.63204	10.609
90%	0.53509	11.636
95%	0.45661	13.177

Fig. 9. ROC curve showing how the FRR and FAR vary when different thresholds are used for the CR and a table showing FAR and FRR values as the threshold limit varies from 5 percent to 95 percent.

4.4 Evaluation

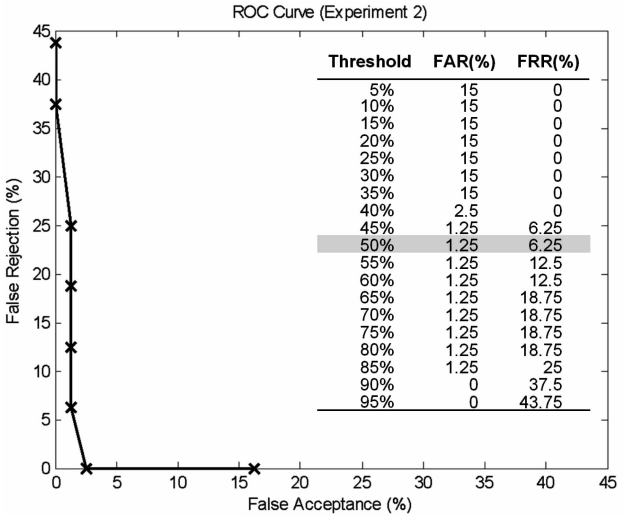
Using the evaluation technique outlined previously in Sections 3.5 and 3.6, we computed the FAR and FRR for each of the $\mathrm{n} = 22$ users separately through a one-hold-out cross-validation test. Fig. 9 shows the receiver operating characteristic (ROC) curve, illustrating the relation between the (global) FRR and FAR by varying the threshold limit, as well as selected points of the curve. The optimal operating point depends on the relative cost of a false acceptance versus a false rejection. In many cases, a false rejection is much less costly than a false acceptance; a false rejection is a source of frustration, whereas a false acceptance represents a security breach.

When selecting a threshold setting, a trade-off must be made between security and user acceptability. In a typical ROC curve, there is a crossover point at which false acceptance and false rejection errors are equally likely [15]. The lower the crossover point, the more accurate the biometric system. The crossover point in the above ROC curve corresponds to a threshold limit of 50 percent. At this threshold, the overall FAR and FRR of the test for the 22 users are 2.4649 percent and 2.4614 percent, respectively. Table 6 shows the user misclassification matrix based on a threshold limit of 50 percent. Each cell $[u_{\mathrm{i}}, u_{\mathrm{j}}](1 \leq i, j \leq 22)$ in the matrix represents a test done between the reference signature of user u_i and all the sessions of user u_j (except those involved in the computation of the signature where $i = j$). Test results are presented in percentage format. The numbers in the cells show the percentage of the number of user u_j's sessions classified as user u_i's. Note that an ideal confusion matrix would have 100 percent down the diagonal, indicating perfect recognition of the user's own behavior, and 0 percent everywhere else, indicating perfect recognition of other users as impostors.

TABLE 6
User Misclassification Matrix

Classified as: i \ True Identity: j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	99.967	0	0.3523	0.189	0	0.0086	0.1289	0	0.0173	0.842	0	0	0.146	0.025	0.0257	0.33	0	0	0	0	0.008	0
2	0.008	97.19	0.318	0.4725	0.0172	0.086	0.2234	0.7647	0.5241	0.61	0	0	0.008	0	0	0.257	0.062	0	0.068	0	0	0
3	0.017	0.1633	95.463	0.043	0	0.1804	0.1804	0.6702	0.593	0.541	0	0.0172	0.713	0	0.0343	0.885	0.051	0	0	0	0	0.0085
4	0	0.275	0.5671	97.663	0	0.206	0.1203	0.8077	0.3437	1.203	0.0086	0.0086	0.258	0.017	0	0.257	0.025	0	0.034	0.004	0.025	0
5	0	0	0.0516	0	99.871	0	0	0	0.0087	0.472	0	0	0.043	0	0	0.094	0	0	0	0	0	0
6	0	0	0	0.0172	0	98.359	0.0687	0	0	0.73	0	0	0	0	0.026	0.017	0.008	0	0	0	0	0
7	0	0.1718	0.2148	0.318	0	0.0429	96.649	0.0257	0.129	1.2115	0	0.0086	0.257	0.008	0.042	0.008	0.008	0	0	0	0	0
8	0	1.2632	0.275	0.1375	0.0172	0.1633	0.1211	95.429	0.5327	0.885	0.0172	0.0258	0.077	0.085	0	0.24	0	0	0	0	0.094	0
9	0	0.335	0.9194	0.1804	0	0.1718	0.4124	0.6186	97.156	1.2115	0	0.0172	0.41	0.068	0.086	0.953	0	0	0.06	0.004	0.05	0.0085
10	0	0	0.2664	0.0172	0	0.1976	0	0.1976	0	81.586	0.0172	0	0	0	0	0.08	0	0	0	0	0	0
11	0	0.3437	0.1117	0.1633	0	0.0344	0	0.3352	0.0086	1.1428	99.905	0	0.146	0	0	0.034	0	0	0	0	0.18	0
12	0	0.0515	0.0515	0	0	0	0.06	0	0.0515	1.9161	0	99.897	0.146	0	0	0.171	0	0	0	0	0.008	0
13	0	0	0.4296	0	0	0.0945	0.2148	0	0	1.2201	0	0.0086	96.97	0.008	0	0.18	0	0	0.008	0	0.008	0
14	0	0.0172	0.1804	0	0.0344	0.0515	0.576	0	0.008	0.7391	0	0	0.069	99.73	0.008	0.06	0	0	0	0	0	0
15	0	0	0.0172	0.0086	0	0.2492	0.1375	0	0.1031	0.27	0	0	0.026	0	99.762	0.051	0	0	0	0	0	0
16	0	0.0773	0.1031	0.0086	0.0516	0	0.1203	0	0.1374	0.472	0	0	0	0	0	94.123	0	0	0	0	0	0
17	0	0.0172	0.0687	0	0	0	0.0172	0.0859	0.1117	1.177	0.0086	0	0.57	0	0.008	0.05	99.77	0	0	0	0.017	0
18	0	0	0.0172	0	0	0	0	0.0086	0	0.4554	0	0	0.008	0	0	0.008	0	100	0	0	0	0
19	0	0.0258	0	0	0	0	0	0	0.026	0.601	0	0	0.017	0	0	0.017	0	0	99.83	0	0	0
20	0	0	0	0.0687	0	0.0258	0	0	0	0.429	0.0086	0	0.033	0	0	0.025	0.008	0	0	99.988	0	0
21	0.008	0.043	0.198	0.326	0.0086	0.1032	0.06	0.8334	0.0515	1.607	0.0258	0.017	0.017	0.034	0	1.71	0.017	0	0	0	99.61	0
22	0	0.026	0.395	0.387	0	0.0258	0.91	0.2234	0.1977	0.6785	0.009	0	0.086	0.025	0.008	0.45	0.051	0	0	0.004	0	99.983

Matrix computed based on a threshold limit of 50 percent. Each cell represents a test done between the reference signature of the row user and all the sessions of the column user. The numbers in the cells are in percentage format. For instance, cell (3, 7) indicates that 0.1804 percent of User 7 sessions were misclassified as belonging to User 3. Cell (18, 18) shows that all of User 18 sessions (that is, 100 percent) are properly classified as his or hers.



Threshold	FAR(%)	FRR(%)
5%	15	0
10%	15	0
15%	15	0
20%	15	0
25%	15	0
30%	15	0
35%	15	0
40%	2.5	0
45%	1.25	6.25
50%	1.25	6.25
55%	1.25	12.5
60%	1.25	12.5
65%	1.25	18.75
70%	1.25	18.75
75%	1.25	18.75
80%	1.25	18.75
85%	1.25	25
90%	0	37.5
95%	0	43.75

Fig. 10. ROC curve obtained by fixing the operating conditions and a table showing FAR and FRR values as the threshold limit varies from 5 percent to 95 percent.

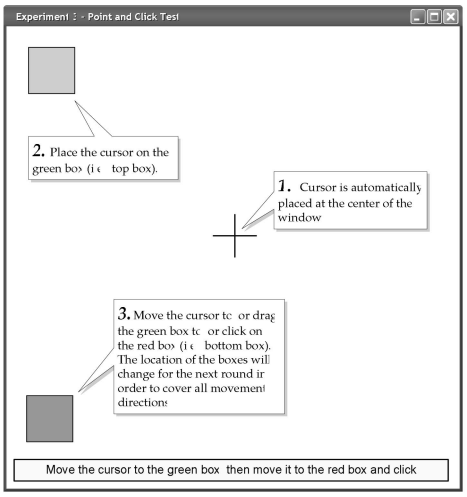


Fig. 11. User interface used for Experiment 3. The interface directs the user to repeat the steps described 100 times per session for each of the three types of actions. The current snapshot is for a PC action (as expressed by the instructions in the textbox) in direction 4.

5 Effect of Environment Variables

In the experiment described previously, we gave the participants an individual choice of operating conditions and applications. As a consequence, data was collected using a variety of hardware and software systems. This was necessary in order to reproduce the characteristics of real-life computing environments, which represent the target of the proposed biometric system. Questions remain about the exact impact of this diversity on the results obtained. For instance, what if the perceived difference between the signatures of two different users was simply because they were using different software applications? As an initial attempt to explore these questions, we conducted two small experiments in which we fixed the operating environment and asked seven different users to use a specific application or perform specific actions. The (same) seven users, selected randomly from the 22 previous participants, were involved in both small experiments. In this section, we describe and analyze the results of these experiments labeled Experiments 2 and 3. We also analyze, at the end of this section, the impact of session length on the performance of our biometrics system.

5.1 Experiment 2

In this section, we describe the approach used for Experiment 2 and discuss the obtained results.

5.1.1 Method

This experiment was conducted locally in our laboratory and involved seven participants. The participants were asked to provide three sessions with a period of 30 minutes for each session using the same hardware and software application. The users were asked to limit their use to browsing the Web using Internet Explorer (Version 6.0), without any further restrictions.

5.1.2 Apparatus

The experiment was conducted using a client workstation (Pentium 4 3.0-GHz processor, 512 Mbytes of RAM, 19-inch display, and an external mouse) running Windows XP and connected over a local area network. The client software described earlier was deployed on the workstation and used for data collection, as well as Internet Explorer (Version 6.0). The client communicated with a server located on the same network. The hardware configuration of the server was a Dual Xeon 2.8-GHz processor with 2 Gbytes of RAM, running Windows Server 2003.

5.1.3 Results

The number of sessions collected in this experiment was 49 (3 to 10 sessions per user); some users provided more than the three sessions requested. The collected data was analyzed using the same detection technique implemented in the main experiment. Only one session, selected randomly, was used as the reference signature for each user. Fig. 10 shows the ROC curve obtained by varying the threshold for the CR, as well as selected points over the curve. When setting the threshold limit to 50 percent as in the main experiment, the performance results obtained are $\mathrm{FRR} = 6.25$ percent and $\mathrm{FAR} = 1.25$ percent. These results suggest that at least on a small population size, tighter control of the environmental variables does not lessen the promise of mouse dynamics to become a new behavioral biometric.

5.2 Experiment 3

We present in the following the method and apparatus for Experiment 3 and discuss the corresponding results.

5.2.1 Method

In this experiment, the (same seven) users were asked to perform the same set of actions using the same machine; all the users were assigned the same tasks in the same order. Specifically, we developed a fixed user interface as illustrated in Fig. 11. Each user was asked to perform a specific action between two rectangles. In each round, the program forces the user to perform the action in a specific direction by changing the position of both rectangles; the distances between the rectangles are equal. The software records the time the user spends to perform the action. Each user was required to perform at least 100 actions in order to

TABLE 7
Drag-Drop Sessions for Seven Different Users

Users' Sessions \ Movement Direction	1	2	3	4	5	6	7	8
User 1	113.83	94.95	74.984	116.59	104.23	89.684	105.14	85.919
User 2	106.81	137.58	77.09	128.62	110.87	121.69	146.6	74.48
User 3	95.76	89.28	65.15	103	97.23	82.14	122.52	73.74
User 4	187.7	142.32	137.76	212.5	196.87	148.92	208.87	153.75
User 5	91.31	138.87	90.71	135	81.28	85.61	84.46	67.14
User 6	122	95.44	83.66	117.62	120.06	88.74	145.06	115.4
User 7	100.73	84.76	63.84	107.44	112.83	88.17	108.88	73.8

The number in each cell represents the mean elapsed time (over a session, in multiples of 10 ms) needed to perform a DD action in one of the eight directions of movement. For instance, cell (1, 1) shows that User 1 takes on the average $(113.83 \times 10 =)1,138.3$ *ms to perform DD in direction 1.*

complete a session. The test was repeated three times for the MM, PC, and DD categories of actions. All environment variables were fixed for the experiment.

5.2.2 Apparatus

The hardware configuration of the machine used consisted of the following: a Pentium 2 450-MHz processor, 128 Mbytes of RAM, a regular two button external mouse, and the Windows 2000 operating system. We installed our data collection software, as well as an interactive software client used for operating the experiment by guiding the user through the required set of actions (see Fig. 11). The results were recorded locally.

5.2.3 Results

The data collected in this experiment consisted of nine sessions for each user. The total number of sessions collected in this experiment was 63.

Each session recorded how the elapsed time varies in the eight recognized directions for each of the three types of action. Samples of the data collected are presented in Tables 7 and 8. Table 7 shows seven DD sessions for seven different users. Each cell in the table records the mean elapsed time (in multiples of 10 ms) for a user to perform a DD action in a specific direction.

Table 8 shows another seven DD sessions belonging to the same user (User 1). In this experiment, we performed user identity verification by applying the same detection technique used for the main experiment. Since the nature of this experiment forces all users to perform the same set of actions in all of their sessions, some of the user signature factors are constant. In particular, all the histogram-based factors (MDH, ATH, TDH, and MTH) are constant since the number of actions performed is fixed for all movement directions and for all action types. Also, the impact of the MSD factor is negligible since the traveled distances of all performed actions are always the same. MDA and ATA are the only detectable factors in this experiment. The detection process is based on the same neural network design used in the two previous experiments. Inputs representing the nondetectable factors remain unaltered although they are constant and will have no effect on the network training. The network will be trained mainly with MDA and ATA factors.

In this experiment, for enrollment, we selected randomly one session out of the nine sessions provided by each user, whereas the remaining eight sessions were used for testing. Similar to the main experiment, a neural network was trained for each of the users; this network was used in testing mode to confirm the user's identity. Tables 9 and 10 illustrate the detection processes. Table 9 shows the mouse dynamics signature calculated for the sessions listed in Table 7; a neural network was trained for User 1 using MDA and ATA factors as inputs and the value P as output.

TABLE 8
Sample Data Collected from Seven Different Drag-Drop Sessions Performed by User 1

Users' sessions \ Movement Direction	1	2	3	4	5	6	7	8
Session 1	115.79	98.33	79.01	116.41	96.52	84.41	103.56	86.62
Session 2	105.35	95.71	65.92	101.8	101.63	74.12	94.66	80.87
Session 3	100.93	88.92	72.50	111.5	101	83.92	93.2	79.14
Session 4	126.04	104.28	76.68	125.11	113.35	119.64	111.93	92.41
Session 5	119	99.44	72.97	123.33	104.58	95.80	98.70	95.89
Session 6	107.87	84.01	75.63	116.62	104	80.89	105.43	82.67
Session 7	121.8	93.96	82.18	121.33	108.52	89.01	128.47	83.83

The number in each cell represents the mean elapsed time (over a session, in multiples of 10 ms) to perform a DD action in one of the eight directions of movement by the user.

TABLE 9
Mouse Signature Factors Extracted from Each of the Seven Sessions Listed in Table 7

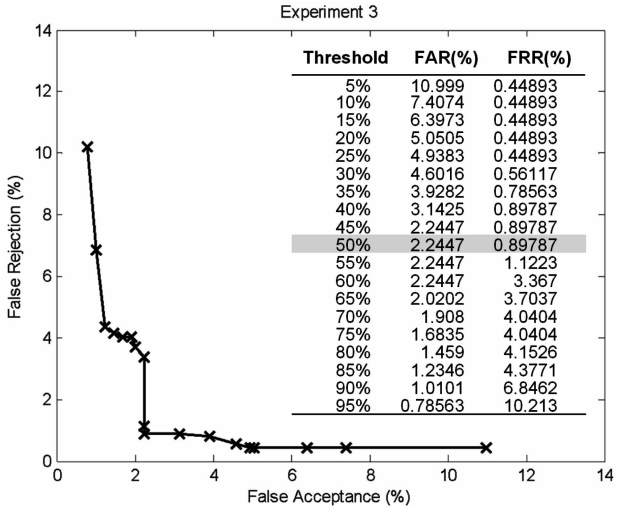
	MDA								ATA			P	CR
User 1	85.723	70.86	52.901	93.91	72.977	70.741	76.147	63.383	48.26	73.565	98.166	100	99.76
User 2	81.253	87.523	57.18	96.538	86.387	79.787	95.377	63.49	57.695	72.163	112.97	0	0.07
User 3	72.323	64.653	56.61	74.14	71.323	61.253	76.147	51.68	49.086	57.86	91.103	0	0.22
User 4	100.21	78.09	72.023	114	103.43	80.29	105.71	83.253	42.281	60.511	173.59	0	0.01
User 5	66.286	81.938	54.839	93.24	67.717	63.857	66.205	51.48	46.97	60.819	96.797	0	0.11
User 6	124.69	101.33	79.007	122.61	115.59	109.34	122.97	100.07	108.68	108.68	111	0	0.04
User 7	103.42	97.297	66.627	103.65	103.39	83.94	111.93	75.963	96.174	91.103	92.556	0	0.03

Each row shows the MDA and ATA factors representing the signature calculated for its corresponding session in Table 7. Column P shows the output values used to train User 1's network. The CR column shows the output obtained from the trained User 1's network for each particular session. Notice that the trained network was able to recognize User 1's session and reject sessions belonging to other users.

TABLE 10
Mouse Signatures Calculated for the Seven Sessions Listed in Table 8

	MDA								ATA			CR
User 1	87.227	73.297	55.357	93.85	70.407	68.983	77.753	66.19	50.025	74.793	97.581	99.40
User 2	82.923	74.16	50.07	91.48	82.06	62.797	74.303	64.643	54.721	73.685	90.007	61.14
User 3	82.273	70.16	53.187	94.7	76.75	67.923	74.36	61.497	48.961	77.469	91.389	100
User 4	91.733	77.993	53.467	96.75	76.017	80.463	80.543	68.12	52.163	73.565	108.68	86.87
User 5	85.533	71.38	53.297	98.643	77.943	72.147	74	66.707	48.26	75.395	101.21	100
User 6	84.587	68.523	54.42	98.907	81.883	66.913	78.437	65.247	52.485	77.469	94.64	97.94
User 7	88.38	70.53	55.3	95.49	74.407	70.517	83.923	62.687	48.26	73.565	103.64	86.12

Each row shows the MDA and ATA factors and the CR obtained using User 1's neural network, which was trained using the data in Table 9. The outputs obtained are all above the 50 percent threshold, which indicates that the network was able to correctly classify all the sessions as belonging to User 1.



Threshold	FAR(%)	FRR(%)
5%	10.999	0.44893
10%	7.4074	0.44893
15%	6.3973	0.44893
20%	5.0505	0.44893
25%	4.9383	0.44893
30%	4.6016	0.56117
35%	3.9282	0.78563
40%	3.1425	0.89787
45%	2.2447	0.89787
50%	2.2447	0.89787
55%	2.2447	1.1223
60%	2.2447	3.367
65%	2.0202	3.7037
70%	1.908	4.0404
75%	1.6835	4.0404
80%	1.459	4.1526
85%	1.2346	4.3771
90%	1.0101	6.8462
95%	0.78563	10.213

Fig. 12. ROC curve for Experiment 3 involving the data collected from the three tests (MM, PC, and DD), showing how the FRR and FAR vary when different thresholds are used, and a table showing sample points from the curve.

The table shows the CRs obtained for those sessions after the network had been trained. The results obtained indicate a successful training, as the network was able to establish a distinguishing pattern for User 1. Table 10 shows the signatures calculated from User 1's sessions listed in Table 8 and the CRs obtained when those sessions were tested against User 1's reference signature. The computed values for CRs indicate a successful identification of User 1's sessions, which were not included in the enrollment.

Fig. 12 shows the ROC curve for the experiment showing the relation between false negative and false positive as the threshold limit varies. Setting the threshold limit to 50 percent yields a performance of $\mathrm{FAR} = 2.245$ percent and $\mathrm{FRR} = 0.898$ percent. This suggests that even by fixing the actions performed by users, mouse dynamics can still be used to discriminate effectively among a small group of users.

5.3 Effect of Session Length

The session limit corresponds to the number of actions required to complete a session. As the number of actions in a session increases, the possibility of detecting more biometric characteristics increases. In order to analyze the effect of the session limit on the accuracy of the detection algorithm, we randomly selected seven users from the 22 participants of the main experiment (independent from the users involved in the previous small experiments). The data collected from each user was divided into a number of sessions with n actions, where n is the session limit defined for the test. We conducted five tests on the same data with $n = 1{,}000$, 1,500, 2,000, 2,500, and 3,000 actions.

The detection technique used in those tests was the same as the technique used in the main experiment. Fig. 13 shows the ROC curves obtained for the five tests. Observe that the best FRR obtained for the first two tests $(n \leq 1,500)$ are 24 percent and 19 percent, respectively. As we increase the number of actions in the sessions, this number decreases to become less than 3.29 percent for the last three tests $(n \geq 2,000)$. On the other hand, we can observe

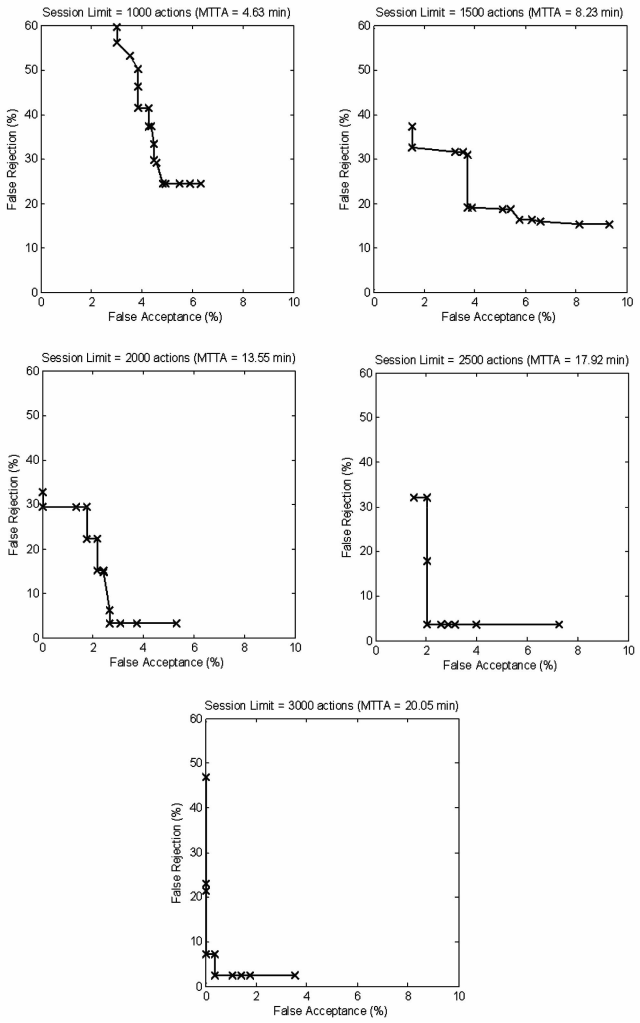


Fig. 13. ROC curves for the five tests, where n varies from 1,000 to 3,000 actions, showing how the FRR and FAR vary when different thresholds are used for the CR, as well as the MTTA expressed in minutes.

that the best FAR obtained for the best FRR range mentioned above decreases from 4.6 when $n = 1,000$ to 0.4 percent as n reaches 3,000. From these readings, we can conclude that selecting the session limit $n = 2,000$ produces relatively accurate results ($\mathrm{FAR} = 2.6489$ percent and $\mathrm{FRR} = 3.29$ percent in this case), and the accuracy of the system increases as the number of actions included in the sessions increases. The figure also shows the Mean Time To Alarm (MTTA) calculated for each of the tests. The MTTA is the mean time needed to detect an identity mismatch. This corresponds to the sum of the time needed for the user to generate session data, the time needed to generate the signature for this session, and the time needed to verify the signature. The value of MTTA is mainly dominated by the amount of time needed to collect the actions. For the $n = 2,000$ test, we obtained $\mathrm{MTTA} = 13.55$ minutes consisting of 13.543 minutes average session duration and only 0.44 seconds as the time needed to process the data and make the decision. It is important to note that MTTA decreases as n decreases; thus, a trade-off should be made between detection performance and time.

TABLE 11
Biometric Recognition Error Rates Collected from the Literature

Biometric	FRR	FAR	Reference
Face	16%	16%	[14]
Voice	7%	7%	[9,14]
Keystroke	4%	0.01%	[2]
Hand	3%	0.3%	[14]
Fingerprint	2%	0.02%	[14,8]
Iris	0.25%	0.0001%	[14]
Mouse	2.4614%	2.4649%	Current

6 Concluding Remarks

We summarize in the following our results and discuss future directions of research.

6.1 Summaries and Discussion

Mouse dynamics is a behavioral biometric that could be used in different security applications. The purpose of this biometric is to model the behavior of the user and find the distinguishing factors that establish the user's identity. We introduced a software implementation for the detector and conducted experiments to establish its accuracy, achieving a FAR of 2.4649 percent and a FRR of 2.4614 percent. Such performance does not yet meet the European standard for access control, which requires a commercial biometric system to achieve a FAR of less than 0.001 percent and a FRR of less than 1 percent [15].

As Table 11 shows, however, the obtained results are better than those of some well-established biometrics such as voice and face recognition systems. On the other hand, better results are achieved with traditional biometrics such as iris and fingerprint. It is also worth noting that among all existing biometrics only iris meets the European standard. The results reported by Pusara and Brodley support the claim that mouse dynamics can be used to discriminate between different users [16]. Their experimental evaluation involved 18 users, each using Internet Explorer for two hours. The initial evaluation gave for all 18 users, on the average, a $\mathrm{FRR} = 27.5$ percent and a $\mathrm{FAR} = 3.06$ percent. After applying a smoothing filter on detection accuracy, they obtained for 11 (out of 18) users, on the average, a $\mathrm{FRR} = 0.43$ percent and a $\mathrm{FAR} = 1.75$ percent, which represent a significant improvement. As indicated earlier, our work differs because it explores multiple sets of conditions, for instance, on imposing greater control on environmental variables and also on imposing less control on environmental variables.

6.2 Future Work

Besides the seven mouse factors considered in this paper, future work will investigate additional factors and also extend the size of the user population involved in the evaluation process. The experimental approach adopted in this work can be improved through a more effective handling of confounds. For instance, the differences in the number of participants or in the amount of testing and training data used in the different experiments are themselves confounding factors. We plan in future work to address this issue by strictly minimizing the differences between the different experiments. Future work will also investigate several practical issues regarding the use of mouse dynamics. For instance, the dynamics may change as the user becomes more proficient, performs a new set of repetitive tasks, or places an object such as a beverage on the mouse pad or when the relationship between the user's chair and monitor changes. It is important to study the impact of these factors on the proposed biometrics system. Mouse biometrics cannot be used as a replacement for passwords to statically authenticate users at the beginning of a session because the data capturing process takes some time. This may be possible, however, by developing a graphical user interface to force the user to do specific actions before granting access, such as following a specific path or performing movements from and to specific areas. Different sets of actions can be requested for every login trial, such actions should be intelligently selected for each user according to the most unique and reproducible factors in his or her signature. Mouse biometrics may also be used for dynamic authentication or identity confirmation in cases where the actions of an active user raise some suspicions. Intrusion detection is the most suitable of the possible applications of this biometric. The passive and continuous user data acquisition and monitoring capabilities of mouse dynamics biometrics make it appropriate for the efficient detection of masqueraders [1]. Future work will also investigate how mouse biometrics can be integrated efficiently with existing intrusion detection techniques. Another important area for possible application of mouse dynamics biometrics is in forensics systems. By recording the mouse dynamics features of perpetrators, it may be possible to build profiles that can be used as evidence in court. Profiles may also be used to track and identify perpetrators across various sites.

Acknowledgments

The authors wish to thank Associate Editor Dr. R. Maxion for giving them the opportunity to publish this work and for his advice on improving the quality of the paper. They would like to thank and acknowledge the tremendous work done by the anonymous reviewers for their valuable comments and thorough review. They would also like to thank all the volunteers who have participated in the experimental evaluation of this work for their valuable time. The authors wish to acknowledge the relentless efforts of Mr. A. Hoole of the University of Victoria and Mrs. L. Hoole for helping improve the quality of this manuscript. The authors wish to thank Dr. J. Weber-Jahnke and Dr. K. Wu of the University of Victoria, Dr. B. Yu of the British Columbia Institute of Technology (BCIT), and Dr. I. Simplot-Ryl of the University of Lille for their comments on the initial version of this manuscript. Last but not least, the authors wish to acknowledge the support and advice provided by Dr. M. Lesperance of the University of Victoria Statistical Consulting Center on conducting the statistical analysis involved in this work. This work was supported in part by an Idea-to-Innovation (I2I) grant from the National Science and Engineering Research Council (NSERC).

References

[1] J.P. Anderson, "Computer Security Threat Monitoring and Surveillance," technical report, J.P. Anderson, Fort Washington, Pa., Apr. 1980.

[2] F. Bergadano, D. Guneti, and C. Picardi, "User Authentication through Keystroke Dynamics," *ACM Trans. Information and System Security,* vol. 5, no. 4, pp. 367-397, Nov. 2002.
[3] M. Brown and S.J. Rogers, "User Identification via Keystroke Characteristics of Typed Names Using Neural Networks," *Int'l J. Man-Machine Studies,* vol. 39, pp. 999-1014, 1993.
[4] A. Chan, R.W.H. Lau, and A. Si, "A Motion Prediction Method for Mouse-Based Navigation," *Proc. IEEE Computer Graphics Int'l Conf. (CGI '01),* pp. 139-146,
[5] D. Denning, "An Intrusion Detection Model," *IEEE Trans. Software Eng.,* vol. 13, no. 2, pp. 222-232, Feb. 1987.
[6] R. Joyce and G. Gupta, "Identity Authentication Based on Keystroke Latencies," *Comm. ACM,* vol. 33, no. 2, pp. 168-176, Feb. 1990.
[7] S. Hocquet, J.Y. Ramel, and H. Cardot, "Users Authentication by a Study of Human Computer Interactions," *Proc. Eighth Ann. (Doctoral) Meeting on Health, Science and Technology,* http://www.univ-tours.fr/ed/edsst/comm2004/hocquet.pdf, 2004.
[8] D. Maio, D. Maltoni, R. Capelli, J.L. Wayman, and A.K. Jain, "FVC2000: Fingerprint Verification Competition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 402-412, Mar. 2002.
[9] A. Martin and M. Przybocki, "The NIST 1999 Speaker Recognition Evaluation—An Overview," *Digital Signal Processing,* vol. 10, no. 1-3, pp. 1-18, 2000.
[10] V. Matyas Jr. and Z. Riha, "Toward Reliable User Authentication through Biometrics," *IEEE Security and Privacy Magazine,* vol. 1, no. 3, pp. 45-49, May/June 2003.
[11] J. McHugh, "Intrusion and Intrusion Detection," *Int'l J. Information Security,* vol. 1, pp. 14-35, 2001.
[12] F. Monrose and A. Rubin, "Authentication via Keystroke Dynamics," *Proc. Fourth ACM Conf. Computer and Comm. Security (CCS '97),* pp. 48-56, Apr. 1997.
[13] P. Oel, P. Schmidt, and A. Shmitt, "Time Prediction of Mouse-Based Cursor Movements," *Proc. Joint AFIHM-BCS Conf. Human-Computer Interaction (IHM-HCI '01),* vol. 2, pp. 37-40, Sept. 2001.
[14] L. O'Gorman, "Comparing Passwords, Tokens, and Biometrics for User Authentication," *Proc. IEEE,* vol. 91, no. 12, pp. 2021-2040, Dec. 2003.
[15] D. Polemi, "Biometric Techniques: Rev. and Evaluation of Biometric Techniques for Identification and Authentication, Including an Appraisal of the Areas where They Are Most Applicable," ftp://ftp.cordis.lu/pub/infosec/docs/biomet.doc, 2006.
[16] M. Pusara and C.E. Brodley, "User Re-Authentication via Mouse Movements," *Proc. ACM Workshop Visualization and Data Mining for Computer Security (VizSec/DMSEC '04),* Oct. 2004.
[17] T.G. Whisenand and H. Emurian, "Effects of Angle of Approach on Cursor Movement with a Mouse: Consideration of Fitts' Law," *Computer in Human Behavior,* vol. 12, no. 3, pp. 481-495, 1996.

Ahmed Awad E. Ahmed received the BSc and MSc degrees from the Department of Electrical and Computer Engineering, Ain Shams University, Cairo, in 1992 and 1997, respectively. He is a research associate and a PhD candidate in the Department of Electrical and Computer Engineering, University of Victoria. His PhD thesis presents a number of new trends in security monitoring through human-computer interaction devices. He is a member of the Security and Object Technology (ISOT) Research Laboratory at the University of Victoria and the principal investigator of Biotracker (http://www.isot.ece.uvic.ca/projects/biotracker), an intrusion detection system based on biometrics. He worked as a software design engineer, project manager, and quality assurance consultant in a number of leading software firms.

Issa Traore received the aircraft engineer degree from the Ecole de l'Air in Salon de Provence, France, in 1990, the master's degree in aeronautics and space techniques and the master's degree in automatics and computer engineering from the Ecole Nationale Superieure de l'Aeronautique et de l'Espace (ENSAE), Toulouse, France, in 1994 and 1995, respectively, and the PhD degree in software engineering from the Institute Nationale Polytechnique (INPT)-Laboratoire d'Analyse et d'Architectures des Systèmes/Centre National de la Recherche Scientifique (LAAS/CNRS), Toulouse, France, in 1998. He held a postdoctoral position at LAAS/CNRS, Toulouse, France, from June to October 1998 and was a research associate from November 1998 to May 1999 and a senior lecturer from June to October 1999 at the University of Oslo. Since November 1999, he has been with the faculty of the Department of Electrical and Computer Engineering, University of Victoria, Canada, where he is currently an associate professor and holds, since October 2003, the position of computer engineering programme director. His research interests include behavioral biometrics systems, intrusion detection systems, software security metrics, and software quality engineering. He is the founder and coordinator of the Information Security and Object Technology (ISOT) Research Laboratory (http://www.isot.ece.uvic.ca). He is a member of the IEEE.