On Mouse Dynamics as a Behavioral Biometric for Authentication

Zach Jorgensen and Ting Yu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
{zjorgen,tyu}@ncsu.edu

ABSTRACT

The idea of using one's behavior with a pointing device, such as a mouse or a touchpad, as a behavioral biometric for authentication purposes has gained increasing attention over the past decade. A number of interesting approaches based on the idea have emerged in the literature and promising experimental results have been reported; however, we argue that limitations in the past experimental evaluations of these approaches raise questions about their true effectiveness in a practical setting. In this paper, we review existing authentication approaches based on mouse dynamics and shed light on some important limitations regarding how the effectiveness of these approaches has been evaluated in the past. We present the results of several experiments that we conducted to illustrate our observations and suggest guidelines for evaluating future authentication approaches based on mouse dynamics. We also discuss a number of avenues for additional research that we believe are necessary to advance the state of the art in this area.

Categories and Subject Descriptors

K.6.5 [**Computing Milieux**]: Security and Protection—*authentication, unauthorized access*

General Terms

Experimentation, Security, Verification

Keywords

Mouse dynamics, behavioral biometrics, authentication, mouse movement, pointing devices, human computer interaction

1. INTRODUCTION

Mouse dynamics describe an individual's behavior with a computer-based pointing device, such as a mouse or a touchpad. Recently, mouse dynamics have been proposed as a behavioral biometric, under the premise that mouse behavior is relatively unique among different people. Mouse dynamics are not the only behavioral biometric to be proposed that is based on one's behavior with a human computer interaction (HCI) device. Keystroke dynamics, which measure an individual's unique typing rhythms, have been the subject of considerable research over the past few decades and their use for authentication has shown promising results [9]. Given that much of the interaction between humans and computers these days involves the use of a pointing device, using mouse dynamics for authentication was an obvious next step.

In the context of authentication, biometrics have several advantages over traditional authentication techniques that verify identity based on something one knows (e.g. a password) or something one has (e.g. a hardware token). In particular, biometrics cannot be forgotten, stolen, or misplaced. Additionally, HCI-based behavioral biometrics have the advantage that they are less obtrusive than other biometrics and do not require special hardware in order to capture the necessary biometric data.

While authentication with keystroke dynamics has been studied extensively over the past three decades, mouse dynamics has just recently begun to gain interest over the last decade. In general, mouse dynamics seems to hold promise as an authentication technique, with some recently proposed approaches ([2], [5]) reporting error rates better or comparable to other well-established biometrics such as voice and face recognition. However, we believe that this may be an overly optimistic assessment of the state of the art due to limitations in the experimental evaluations that have appeared in the literature. More specifically, we have observed that many of the existing approaches require an impractical amount of mouse data to be collected before an authentication decision can be made with reasonable accuracy. We have also observed that in many of the past evaluations, environmental variables that could potentially influence mouse dynamics were not properly controlled from one test subject to the next. As a consequence, it is unclear whether the results of those evaluations actually reflect detectable differences in mouse behavior among test subjects, or differences among their computing environments. To illustrate and explore these observations, we collected data from a group of volunteer subjects under tightly controlled conditions and performed several experiments on this data using implementations of two representative mouse dynamics techniques from the literature. Additionally, we performed experiments to test the feasibility of utilizing mouse dynamics for authentication in remote access scenarios, such as web-based applications. In contrast to the local access sce-

ASIACCS '11, March 22–24, 2011, Hong Kong, China.

nario, it is common for a user to remotely access the same application from different computers, which may impact authentication accuracy.

Specifically, we make the following contributions in this paper:

1. We review existing authentication approaches based on mouse dynamics and identify several pitfalls in the experimental methodology used to evaluate these techniques. For instance, we show that environmental variables (including pointing device type) were not properly controlled across test subjects in some evaluations, which can significantly impact results. To the best of our knowledge, this is the first work to call attention to these important limitations.

2. To quantitatively examine the impact of environmental variables on existing schemes, we designed rigorous experiments to evaluate these approaches when environmental variables are tightly controlled across subjects. We also tested existing approaches under different access scenarios. In a nutshell, our results showed that (a) higher error rates are observed when variables are tightly controlled, and (b) existing approaches are unlikely to be effective under certain common remote access scenarios.

3. Based on insight gained from our experimental results we argue that, while promising, mouse dynamics based authentication is not yet suitable for practical deployment. We also discuss a number of avenues for important future work to improve the viability of this technique.

The remainder of the paper is organized as follows. In the next section, we provide some general background information on mouse dynamics authentication systems. In Section 3, we present an overview of existing techniques from the literature, followed by a discussion of the shortcomings of past experimental evaluations in Section 4. In Sections 5 and 6, we present and discuss our experiments. Finally, we offer a discussion of several important avenues for additional work in Section 7, and then conclude the paper with some final remarks in Section 8.

2. BACKGROUND

In this section we provide a general overview of mouse dynamics authentication (henceforth abbreviated MDA) systems.

2.1 Enrollment and Verification

Like all biometric authentication systems, MDA systems involve an enrollment phase and a verification phase. Before the system can be used for the first time, one is required to enroll, by providing the system with a sample of one's typical mouse behavior, which is stored for use during verification. We use the term *enrollment signature* to denote this stored behavioral data. During verification, the active user's mouse activity is captured and compared to the enrollment signature.

MDA systems can be classified according to their mode of verification. *Static* approaches collect and verify a user's mouse data at specific times (e.g. at login time), while *continuous* approaches collect and verify the user's mouse data repeatedly throughout the entire session.

2.2 Acquisition and Feature Extraction

Mouse data is captured from the user by a software program that intercepts the low-level mouse events generated by the pointing device, including raw movement, button up and button down events. Associated with each event may be several attributes, including: a timestamp, cursor coordinates and event type. Since raw mouse events are typically too low-level to be useful for behavioral analysis they are usually aggregated and converted into higher-level abstractions (e.g. point-and-clicks, drag-and-drops, etc. [2]) from which meaningful behavioral patterns can be detected. From the high-level abstractions, various statistical and kinematic features can be extracted and used for behavioral comparison.

2.3 Behavioral Comparison

Behavioral patterns are detected from a stream of mouse events at the feature level. Various algorithms have been used in the literature to detect and compare behavioral patterns. These range from simple distance metrics to complex machine learning algorithms like neural networks.

2.4 Evaluation Metrics

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

- *False Acceptance Rate (FAR)*–the probability that the system will incorrectly label the active user as the same user that produced the enrollment signature.

- *False Rejection Rate (FRR)*–the probability that the system will incorrectly label the active user as an impostor, when in fact it is not.

- *Equal Error Rate (EER)*–the error rate when the system's parameters (such as the decision threshold) are set such that the FRR and FAR are equal. The lower the EER the more accurate the system.

- *Verification time*–the time required by the system to collect sufficient behavioral data to make an authentication decision.

3. EXISTING APPROACHES

In this section we provide a brief survey of MDA approaches proposed over the past decade. In the course of our literature review, we observed some recurring problems among the existing techniques, primarily regarding how they were evaluated. In the survey, we merely note these problems as we discuss each technique and defer a more thorough discussion of the problems to Section 4.

3.1 Continuous Verification Approaches

A number of MDA approaches for continuous authentication have been proposed in the literature. In an approach by Ahmed and Traore, low-level mouse events are aggregated as higher-level actions such as point-and-clicks or drag-and-drops, characterized by action type, distance, duration and direction [1], [2]. Consecutive actions over some time frame are grouped into *sessions*, over which 39 mouse dynamics-related features are computed. Authentication involves training a neural network on mouse data from a given user, which can then be used to classify observed mouse behavior at authentication time. An authentication experiment on mouse data collected from the *personal computers* of 22 users achieved an average EER of around 2.46%. The data sessions used in the experiment were around 17 minutes in length[1], implying a verification time of at least 17 minutes. An additional experiment with 7 users, all using the same computer and mouse for data collection resulted in an FRR of 6.25% and an FAR of 1.25%; however, the data sessions in this experiment were 30 minutes in length–nearly twice the length of the sessions in the main experiment.

Pusara and Brodley presented an approach in which raw mouse data is preprocessed and grouped into data points, each corresponding to a summary of mouse events over a window of configurable size [10]. Features such as frequency, angle, distance and speed are extracted for each window. Parameters for the technique were chosen on a per-user basis using a validation set and the C5.0 decision tree algorithm [11] was used for classification. Using data from 11 users, collected on their *own personal computers*, the technique achieved an average FAR of 0.43% and an average FRR of 1.75% in an authentication experiment. Since parameters were chosen independently for each user, detection time varied by user, ranging from one minute to 14.5 minutes depending on the user.

Gamboa and Fred introduced a continuous authentication approach that aggregates raw mouse events into *strokes*– the movement events occurring between two clicks [4], [5]. Each stroke is characterized by 63 spatial, temporal and statistical features, though this feature space is reduced to the best subset of features, for each user through a greedy feature selection process. According to the paper, feature selection was performed on the same test data used to evaluate the approach; we remark that using test data for feature selection is known to bias the training model toward the test data, resulting in an overly optimistic estimate of a classifier's performance [7]. A statistical model employing the Weibull distribution as the parametric model was used for classification and authentication decisions were based on the average classification outcome of a sequence of individual strokes. The detection time was therefore dependent on the sequence length used. Experiments on data from 50 users, trying various sequence lengths found that sequences of 50 strokes (equivalent to about 50 seconds of mouse data) yielded an EER of 2%. The paper did not explicitly state whether subjects used different computers during the experiment.

Schulz presented a continuous authentication approach in which the mouse event stream is segmented into curves, characterized by length, curvature and inflection-based features [14]. An enrollee's behavioral signature is formed by computing histograms from the curve characteristics of multiple curves. Verification is accomplished by computing the Euclidean distance between a user's enrollment signature and the mouse activity observed during verification. An evaluation of the system on mouse data from 72 anonymous users running mouse recording software on their *personal machines*, yielded an average EER of 24.3%, over groups of 60 curves and an EER of 11.2% with 3600 curves per group.

3.2 Static Verification Approaches

Several static MDA approaches have also been proposed in the literature. Hashia et al. present an approach in which enrollment involves moving the mouse pointer between pairs of dots shown sequentially on the screen [8]. Features computed from the user's movement between each pair of dots comprise the enrollment signature. Authenticating involves the same series of dot-to-dot movements, which are compared against the enrollment signature. Experiments involving 15 students yielded an equal error rate of 15%. All test subjects used the same computer and mouse.

Gamboa et al. extended their continuous verification approach into a static verification scheme for web-based applications [6]. In the proposed scheme, an authenticating user enters a username and pin number via an on-screen virtual keyboard embedded in the login web page, using only the mouse. Mouse movements are recorded through javascript embedded in the web page and are sent to an authentication server, which grants access based on the entered credentials as well as the corresponding mouse movements. The underlying mouse dynamics techniques are the same as in their continuous authentication approach. Testing the system with 50 subjects yielded an equal error rate of 6.2% for 15-digit pin numbers. As with their continuous approach, the test data was used for feature selection, which results in an overly optimistic estimate of the classifier's accuracy. The paper also overlooked the scenario in which a user enrolls using one computer and later attempts to authenticate using a different computer; such a scenario is common in the context of web-based applications.

Bours and Fullu proposed a static approach in which the authenticating user utilizes the mouse to trace a winding maze-like path while mouse movements are recorded and used to compute velocity vectors for each segment of the path [3]. Edit distance is used to compare the verification data to the enrollment data. Experiments with 28 subjects yielded an EER of around 27%. Test subjects ran the experiments using their *own computers*.

Revett et al. presented Mouse-Lock, in which a user authenticates by manipulating a graphical, combination lock-like GUI interface featuring icons, arranged on a circular dial [12]. The entered combination, along with some timing based features, are used to make an authentication decision. A small-scale evaluation involving six subjects yielded an average FAR and FRR of around 3.5% and 4% respectively.

4. LIMITATIONS OF EXISTING WORK

In the previous section, we described a number of existing MDA systems that appear to achieve reasonably low error rates. However, we have identified several recurring problems with respect to how these techniques have been experimentally evaluated. We believe that these issues have resulted in an overly optimistic view of the effectiveness of the current techniques. In this section, we discuss these issues and some of their implications; in Section 5 we present an overview of our study.

4.1 Impractical Verification Time

The first limitation that we observed is that many of the existing approaches, particularly those providing continuous verification, require a significant amount of mouse data to be captured before a reasonably accurate authentication decision can be made. This is clearly not practical for an

[1] The session length was not explicitly stated in [2]; however, it is stated that an average of 12 hours 55 minutes of data was captured from each subject, representing an average of 45 sessions. We therefore assume that average session length is 12:55/45=17.22 minutes.

online system, as even a few minutes is more than enough time for an adversary to compromise a system. We have also observed that when these techniques are referenced in other papers, their error rates are often cited without mention of the corresponding verification time. We note that the approach of [5] may at first appear to be an exception, as this technique was reported to achieve an EER of 2% with less than a minute of verification data; however, it is likely that the invalid feature selection step used in that approach resulted in an overly optimistic estimate of its accuracy.

4.2 Uncontrolled Environmental Variables

Another limitation that we observed is that environmental variables, which could potentially influence mouse dynamics, are not properly controlled from one test subject to the next. Instead, the trend has been to collect mouse data from test subjects by installing recording software on each subject's personal computer (indeed this appears to have been the case [2], [10], [14]). The problem with using a different machine for each subject is that each test machine could differ with respect to any number of software-related variables (e.g. screen resolution, pointer speed and acceleration settings, and mouse polling rate, etc.) or with respect to hardware-related variables, particularly the type of pointing device used. As a consequence, it is unclear whether the results of these prior evaluations actually reflect detectable differences in mouse behavior among test subjects, or differences in the configurations among their machines.

4.3 Remote Access Scenario

Aside from providing host-based authentication for local access, MDA techniques could also potentially be employed in remote access scenarios, such as web-based applications. To properly evaluate their effectiveness in this scenario, we must consider the fact that, in practice, a given user may access a single web-based application from multiple computers. This could lead to the situation in which the enrollment data for a user is collected under a different computing environment than the data used for verification. Thus, to be effective in this scenario, the authentication system must be capable of recognizing a user's behavior across different computing environments. To the best of our knowledge, the effectiveness of mouse dynamics based authentication has not been evaluated under this scenario in the existing literature.

5. OVERVIEW OF OUR STUDY

We chose to focus this study on two primary questions formulated from the limitations pointed out in the previous section.

Q1: In the absence of any effects caused by environmental variables, is pointing device behavior alone sufficient to verify user identity?

Q2: Is it possible to verify user's identity when enrollment and verification data are collected under different computing environments?

Although we do not directly address the problem of verification time (Section 4.1), the verification time used in all of our experiments is arguably closer to what is likely to be acceptable in practical applications than that used in previous experiments. The next subsection gives an overview of the experiments that we performed in an attempt to answer the above questions.

5.1 Overview of Experiments and Methods

Our experiments were performed using implementations of the continuous verification approaches of Ahmed and Traore [2] and Gamboa and Fred [5]. We chose these two approaches because they were among the most frequently cited and represented a relatively diverse set of mouse dynamics features.

The design of our first experiment was geared toward answering Q1. Therefore, it was necessary to examine the effectiveness of some of the existing MDA approaches in a setting where the computing environment was tightly controlled across all test subjects. We collected mouse data from a group of users on the same computer under the same conditions while performing the same task, and used this data in an authentication experiment. Our hypothesis was that the error rates achieved in this setting would be higher than those previously reported in the literature, since in this case the authentication system would not be able to use hardware or software differences to distinguish among users.

Our second experiment was aimed at answering Q2; that is, to evaluate existing techniques in a remote access scenario, in which it is common for a single user to access the system from different computing environments at different times (e.g. accessing a web-based application such as a shopping or banking web site). Although there are many variables that could potentially differ from one machine to another, we speculated that pointing device type (e.g. mouse, touchpad, etc.) would have the greatest effect on behavior; therefore, we tested this scenario by using data collected from one pointing device for enrollment and data from the other pointing device for verification. All other variables were held constant for the two sets of data collected for a given user (and across users). Our hypothesis was that the pointing device would have a significant impact on the user's perceived behavior, which would cause an increase in false rejections.

To provide further insight regarding Q2, we performed a third experiment to determine if the mouse dynamics features defined by the two techniques could be used to identify which of the two pointing devices generated a given session of mouse data. Our reasoning behind this experiment was that if these techniques were capable of discriminating data sessions according to the pointing device that generated them, this would indicate that pointing device hardware itself exhibits a strong influence on mouse dynamics.

5.2 Data Collection

We collected data from 17 volunteer subjects using two different types of pointing devices, while performing a common web browsing task. The subjects, eight males and nine females, were all computer science students from our department. With one exception, all subjects were right handed. We set up two identical computers in our lab and equipped one with a USB optical mouse and the other with a USB touchpad. The subjects were given a specific web browsing task designed to last 30 minutes and were asked to perform that task once for each of the two pointing devices.

5.2.1 *Apparatus*

We used two identical computers for data collection so that we could collect data from multiple subjects at once; we made every effort to control software and hardware factors other than the pointing device itself from having any unintended influence on the subject's recorded mouse behavior. Both computers were Dell Dimension XPSs with Pentium 4, 3.20 GHz processors and 1GB of RAM; both were equipped with identical 21" Dell LCD monitors (set at 1280x1024 resolution). We equipped one of the computers with a USB Logitech optical mouse and the other with a USB Adesso Browser Cat 2 Button Touchpad. Both computers were loaded with a copy of Windows Vista from the same image file. The Google Chrome web browser and our custom mouse recording software were installed on both computers. The default Windows Vista drivers were used for the optical mouse and the GlidePoint 3.3 driver was used for the touchpad. Both pointing devices used a polling rate of 125hz (8ms) and the pointing speed setting in the operating system was left at the default value. Our custom mouse event logging software, implemented in C#, ran as a background process and used a Windows mouse hook to intercept all mouse events, which were written to a file.

5.2.2 *Procedure and Experimental Task*

Subjects were quickly briefed regarding the purpose of the experiment and given the instructions for completing the experimental task. Upon starting the experiment, the Chrome web browser was automatically opened to display the Amazon.com website and the mouse recording software was initiated. After 30 minutes, the subject was asked to move to the other computer, which was equipped with a different pointing device (but otherwise identical), and repeat the experimental task. The experimental task was essentially a scavenger hunt on the popular shopping web site, Amazon.com. Subjects were provided with a long list of items that can be purchased on Amazon and were instructed to browse for and locate the items using only the pointing device. The list included things such as "find 4 DVD movies, each under $10" and "find 3 books each by a different author", etc. We created two different but comparable lists of items for this experiment and each list was associated with one of the two pointing devices for the entire experiment; all subjects used the same two lists of items.

6. EXPERIMENTS AND RESULTS

Recall that the specific MDA approaches used in our experiments were those of Ahmed and Traore [2] and Gamboa and Fred [5]. For brevity, we will refer to the two approaches as *Ahmed* and *Gamboa* respectively in this section. In the following subsections, we first describe how the data was prepared for use by the two approaches, followed by the details of each of the three experiments.

6.1 Data Preparation

The data collection procedure produced two sets of raw data: one from the optical mouse and the other from the touchpad. Two versions of these datasets were created by preprocessing the data according to the requirements of the two MDA approaches used in this study. The *Ahmed* approach specifies that the raw event stream is to be segmented into actions and then groups of consecutive actions are aggregated into sessions, over which various features are computed. By contrast, in the *Gamboa* approach the raw event stream is segmented into strokes and features are computed directly over each stroke; classification is then done at the stroke level, and the classification of a sequence of consecutive strokes is determined according to the average classification of the constituent strokes. Thus, in a sense, both approaches can be viewed as making a single authentication decision based on a sequence of individual actions. For brevity, we will subsequently use the term *session* to denote either a group of consecutive strokes or a group of actions.

Since different users produced different amounts of data, we trimmed each user's data, to 325 actions (225 strokes for *Gamboa*) for each pointing device, which was the minimum action count across all users. After the trimming process, each user's data was divided into five equal-length sessions of 65 actions (25 strokes) each.

We opted not to perform the per-user feature selection step for *Gamboa* as specified in [5]. In that paper, the feature selection step was performed using each user's test set, which is known to bias the classifier to the test data and produce unrealistic results. We found that using the training set for feature selection gave comparable or worse results than doing no feature selection at all and that using part of the data as a validation set was not feasible due to the limited amount of data available to us. Therefore, we omitted feature selection and used all features for every user.

6.2 Experiment 1: controlled environment

Our first experiment investigated the effectiveness of the two MDA approaches when all environmental variables were constant across users. The procedure was the same for both of the approaches and was performed as follows: let n denote the number of users. The five sessions belonging to each user were divided such that the *first two* sessions were designated for training and the *last three* for testing. For each user $u(1 \leq u \leq n)$, two classification tasks were performed. First, a classifier was trained with u's training sessions as positive examples and the training sessions of the other $n-1$ users as negative training examples. The classifier was then tested on u's three test sessions. Second, for each of the other $n-1$ users, $v(1 \leq v \leq n; v \neq u)$, in turn, a classifier is trained using u's training sessions and the training sessions of the other $n-2$ users (that is, excluding user v's training sessions), and then tested on v's and u's test sessions.

A given test session is considered misclassified for user u if the classifier outputs a score below some threshold t, for positive test sessions, or above the threshold for negative test sessions. The FAR is then calculated as FA/TN, where TN is the number of test sessions belonging to the $n-1$ other users and FA is the number of those test sessions for which the classification score was above the threshold t. The FRR is calculated as FR/TP where TP is the number of test sessions belonging to u and FR is the number of those test sessions for which the classification score was below t. The threshold t is set independently for each user such that the FAR and FRR are equal (or as close as possible). The resulting error rates for all n users are averaged to get the FAR and FRR for the entire experiment. The experimental procedure was performed separately on the mouse and touchpad datasets and the results are given in Table 1.

6.3 Experiment 2: remote access scenario

The purpose of Experiment 2 was to investigate whether data collected from a given user on one of the pointing devices could be used to verify that user's identity based on data collected from the same user on the other pointing device. The procedure was as follows. For each user $u(1 \leq u \leq n)$, two classification tasks were performed. First, the classi-

Table 1: Results of Experiment 1

	Gamboa				*Ahmed*			
	Mouse		Touchpad		Mouse		Touchpad	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
Avg.	21%	21.5%	20%	21.5%	30.3%	37.1%	29.7%	34.3%
Std.	14.3%	13.4%	13.3%	13.4%	9.8%	17.7%	17.4%	24.2%

Table 2: Results of Experiment 2

	Train on Mouse				Train on Touchpad			
	Gamboa		*Ahmed*		*Gamboa*		*Ahmed*	
	FAR	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**	**FAR**	**FRR**
Avg.	19.6%	56.9%	40.9%	46.7%	26.7%	56.9%	37.5%	56.9%
Std.	18.1%	38.7%	31.7%	35%	23.2%	30.7%	29.2%	31.8%

fier was trained with user u's two training sessions from the *mouse data set* as positive examples and the two mouse training sessions of each of the other $n-1$ users as negative training examples; the classifier was then tested on user u's three test sessions from the *touchpad data set.* Next, for each of the other users, $v(1 \leq v \leq n; v \neq u)$, user v's training examples are excluded from the training set and the classifier is retrained. The retrained classifier is then tested both on user v's and user u's own test sessions from the touchpad dataset. This whole procedure is repeated a second time using the *touchpad data set* for training and the *mouse data set* for testing. The thresholds for determining the error rates in this experiment were set to be the same as those determined for the same user in Experiment 1. The results of this experiment are shown in Table 2.

6.4 Experiment 3: detecting device type

In the final experiment, we sought to determine whether the two approaches could be used to identify the pointing device that generated a given session. We first divided the 17 users into two groups with eight users for training and nine for testing, such that no user was represented in both the training and testing set. We then labeled all of the sessions in the training and test groups according to pointing device. A classifier was then trained on the labeled training sessions and subsequently tested on the test sessions. For *Ahmed*, we used the same neural network approach for classification as in the earlier experiments. For *Gamboa*, however, we used a logistic regression classifier, due to time constraints and the inflexibility of our implementation. Using the same session length as in the previous experiments, and using a decision threshold of 0.5, *Gamboa* was able to correctly identify the generating pointing device for 96.7% of the test instances, while *Ahmed* achieved a success rate of 97.8%.

6.5 Analysis of Results

In Table 1, we see that the average error rates for both techniques are significantly higher than those originally reported in the literature. We suspect that these results are due, in part, to the tight control of environmental variables in our experiment, but they are also likely due, in part, to: (1) omitting the feature selection step for *Gamboa*, and to (2) using less enrollment data than was used in the previous experiments reported in the literature; however, we believe that the amount of data used in our experiments represents a practical enrollment time. The thing to note from these results is that although the error rates obtained under the conditions of this experiment were not especially good, they were well below 0.5 for both techniques, which suggests that there are indeed detectable behavioral differences among users, even when environmental variables are tightly controlled.

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

6.6 Limitations of our Experiments

In practice it is reasonable to assume that, over time, more training data could be collected and incorporated into the system than was available in our experiments; this would likely improve error rates. We did not collect enough data to examine the impact of enrollment time on the accuracy of the techniques.

In hindsight, Experiment 1 could have been improved by additionally collecting mouse data from each test subject as they performed the 30 minute task on their personal computers. This would have provided a better baseline for the first experiment, allowing us to make stronger claims about the extent of the impact of environmental variables on the two techniques we tested.

7. AVENUES FOR ADDITIONAL WORK

We believe that there are still significant issues to be resolved before MDA systems are truly ready to be deployed in practice. In this section we discuss important directions for new research in this area.

7.1 Creation of Common Evaluation Data

There is critical need for the creation of a common, publicly available data set for use by researchers in this area. Not only would such a data set allow for the comparison of existing and future approaches, it would significantly reduce the overhead for new researchers in this field.

7.2 Reduction of Verification Time

The practicality of MDA systems depends not only on low error rates, but also depends critically on achieving a reasonably short verification time. However, existing approaches tend to require a considerable amount of data to be collected before a reasonably accurate authentication decision can be made, which for many systems is more than enough time for an attacker to achieve her goal. One approach to decrease verification time might be to develop more effective ways to clean the raw mouse data of extraneous noise; with higher quality data, it might be possible to make authentication decisions over smaller quantities of data.

7.3 Strategies for Minimizing False Rejection-s

Strategies for minimizing or handling false rejections in a graceful manner might also increase the practicality of M-DA systems. One possibility might be to combine mouse dynamics with other types of behavioral biometrics, such as keystroke dynamics ([9]) to improve authentication accuracy. Multimodal biometric systems have been proposed in the past to address low authentication accuracy caused by noisy data or intra-class variations [13].

7.4 Offline Analysis Techniques

Existing techniques could potentially be used in offline forensics analysis to provide useful insights; for example, determining whether multiple distinct individuals have been using a single account or whether a single user has been using more than one account. To our knowledge, this has not yet been explored.

7.5 Effects of Environmental Variables

We have shown in this paper that certain environmental variables (e.g. pointing device type) may significantly impact mouse behavior if the state of these variables is different at enrollment time than at verification time. Other variables that could be explored include the following: software-level variables, such as screen resolution, mouse speed and acceleration settings in the OS, mouse polling rate, perceptual delays caused by high CPU load, and etc; properties of the surface on which the mouse is being used; and the psychological state of the user (e.g the user may be fatigued, distracted or distressed). The effects of such variables on one's mouse behavior are generally unexplored.

8. CONCLUSION

In this paper, we pointed out that past evaluations of M-DA techniques did not carefully control environmental variables across test subjects. We conducted an experiment on mouse data collected from 17 volunteer subjects to try to answer the question of whether the low error rates reported in the literature were due to strongly detectable behavioral differences among test subjects, or instead due to differences in their computing environments. The results of that experiment confirmed that there are detectable behavioral differences among individuals but that the loosely controlled environmental variables in past evaluations likely contributed to the low error rates.

We also pointed out that existing continuous approaches generally require a significant amount of mouse data to be collected before an authentication decision can be reached. Consequently, we question whether mouse dynamics is really practical for continuous online authentication.

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

9. ACKNOWLEDGMENTS

We wish to thank the students who volunteered their time to participate in our data collection. We would also like to thank the anonymous peer reviewers for their helpful comments.

10. REFERENCES

[1] A. A. E. Ahmed and I. Traore. Anomaly intrusion detection based on biometrics. In *IEEE Workshop on Information Assurance and Security*, pages 452–453, 2005.

[2] A. A. E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *IEEE Transactions on Dependable and Secure Computing*, 4(3):165–179, 2007.

[3] P. Bours and C. J. Fullu. A login system using mouse dynamics. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1072–1077, 2009.

[4] H. Gamboa and A. Fred. An identity authentication system based on human computer interaction behaviour. In *3rd International Workshop on Pattern Recognition on Information Systems*, pages 46–55, 2003.

[5] H. Gamboa and A. Fred. A behavioural biometric system based on human computer interaction. In *SPIE 5404 - Biometric Technology for Human Identification*, pages 381–392, Orlando, FL USA, 2004.

[6] H. Gamboa, A. L. N. Fred, and A. K. Jain. Webbiometrics: User verification via web interaction. In *Proceedings of Biometrics Symposium*, pages 1–6, 2007.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] S. Hashia, C. Pollett, and M. Stamp. On using mouse movements as a biometric. In *Proceeding in the International Conference on Computer Science and its Applications*, volume 1, 2005.

[9] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems*, 1:351–359, 2000.

[10] M. Pusara and C. E. Brodley. User re-authentication via mouse movements. In C. E. Brodley, P. Chan, R. Lippman, and W. Yurcik, editors, *Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC 2004), 29 October 2004, Washington DC, USA*, pages 1–8. ACM, 2004.

[11] R. J. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[12] K. Revett, H. Jahankhani, S. T. de Magalhes, and H. M. D. Santos. A survey of user authentication based on mouse dynamics. In *Proceedings of 4th International Conference on Global E-Security*, volume 12 of *Communications in Computer and Information Science*, pages 210–219, London, UK, June 2008. Springer Berlin Heidelberg.

[13] A. Ross and A. Jain. Multimodal biometrics: An overview. In *12th European Signal Processing Conference*, pages 1221–1224, Vienna, Austria, 2004.

[14] D. A. Schulz. Mouse curve biometrics. In *Biometric Consortium Conference, 2006 Biometrics Symposium*, pages 1–6, 2006.