User Authentication Through Mouse Dynamics

Chao Shen, *Student Member, IEEE*, Zhongmin Cai, *Member, IEEE*, Xiaohong Guan, *Fellow, IEEE*, Youtian Du, *Member, IEEE*, and Roy A. Maxion, *Fellow, IEEE*

***Abstract*—Behavior-based user authentication with pointing devices, such as mice or touchpads, has been gaining attention. As an emerging behavioral biometric, mouse dynamics aims to address the authentication problem by verifying computer users on the basis of their mouse operating styles. This paper presents a simple and efficient user authentication approach based on a fixed mouse-operation task. For each sample of the mouse-operation task, both traditional holistic features and newly defined procedural features are extracted for accurate and fine-grained characterization of a user's unique mouse behavior. Distance-measurement and eigenspace-transformation techniques are applied to obtain feature components for efficiently representing the original mouse feature space. Then a one-class learning algorithm is employed in the distance-based feature eigenspace for the authentication task. The approach is evaluated on a dataset of 5550 mouse-operation samples from 37 subjects. Extensive experimental results are included to demonstrate the efficacy of the proposed approach, which achieves a false-acceptance rate of 8.74%, and a false-rejection rate of 7.69% with a corresponding authentication time of 11.8 seconds. Two additional experiments are provided to compare the current approach with other approaches in the literature. Our dataset is publicly available to facilitate future research.**

***Index Terms*—Biometric, mouse dynamics, authentication, eigenspace transformation, one-class learning.**

I. Introduction

THE quest for a reliable and convenient security mechanism to authenticate a computer user has existed since the inadequacy of conventional password mechanism was realized, first by the security community, and then gradually by the public [31]. As data are moved from traditional localized computing environments to the new Cloud Computing paradigm (e.g., Box.net and Dropbox), the need for better authentication has become more pressing. Recently, several large-scale password leakages exposed users to an unprecedented risk of disclosure and abuse of their information [47], [48]. These incidents seriously shook public confidence in the security of the current information infrastructure; the inadequacy of password-based authentication mechanisms is becoming a major concern for the entire information society.

Of various potential solutions to this problem, a particularly promising technique is mouse dynamics. Mouse dynamics measures and assesses a user's mouse-behavior characteristics for use as a biometric. Compared with other biometrics such as face, fingerprint and voice [20], mouse dynamics is less intrusive, and requires no specialized hardware to capture biometric information. Hence it is suitable for the current Internet environment. When a user tries to log into a computer system, mouse dynamics only requires her to provide the login name and to perform a certain sequence of mouse operations. Extracted behavioral features, based on mouse movements and clicks, are compared to a legitimate user's profile. A match authenticates the user; otherwise her access is denied. Furthermore, a user's mouse-behavior characteristics can be continually analyzed during her subsequent usage of a computer system for identity monitoring or intrusion detection. Yampolskiy *et al.* provide a review of the field [45].

Mouse dynamics has attracted more and more research interest over the last decade [2]–[4], [8], [14]–[17], [19], [21], [22], [33], [34], [39]–[41], [45], [46]. Although previous research has shown promising results, mouse dynamics is still a newly emerging technique, and has not reached an acceptable level of performance (e.g., European standard for commercial biometric technology, which requires 0.001% false-acceptance rate and 1% false-rejection rate [10]). Most existing approaches for mouse-dynamics-based user authentication result in a low authentication accuracy or an unreasonably long authentication time. Either of these may limit applicability in real-world systems, because few users are willing to use an unreliable authentication mechanism, or to wait for several minutes to log into a system. Moreover, previous studies have favored using data from real-world environments over experimentally controlled environments, but this realism may cause unintended side-effects by introducing confounding factors (e.g., effects due to different mouse devices) that may affect experimental results. Such confounds can make it difficult to attribute experimental outcomes solely to user behavior, and not to other factors along the long path of mouse behavior, from hand to computing environment [21], [41].

Manuscript received March 28, 2012; revised July 16, 2012; accepted September 06, 2012. Date of publication October 09, 2012; date of current version December 26, 2012. This work was supported in part by the NSFC (61175039, 61103240, 60921003, 60905018), in part by the National Science Fund for Distinguished Young Scholars (60825202), in part by 863 High Tech Development Plan (2007AA01Z464), in part by the Research Fund for Doctoral Program of Higher Education of China (20090201120032), and in part by Fundamental Research Funds for Central Universities (2012jdhz08). The work of R. A. Maxion was supported by the National Science Foundation under Grant CNS-0716677. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the National Science Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sviatoslav Voloshynovskiy.

C. Shen, Z. Cai, X. Guan, and Y. Du are with the MOE Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China (e-mail: cshen@sei.xjtu.edu.cn; zmcai@sei.xjtu.edn.cn; xhguan@sei.xjtu.edu.cn; ytdu@sei.xjtu.edu.cn).

R. A. Maxion is with the Dependable Systems Laboratory, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: maxion@cs.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2012.2223677

It should be also noted that most mouse-dynamics research used data from both the impostors and the legitimate user to train the classification or detection model. However, in the scenario of mouse-dynamics-based user authentication, usually only the data from the legitimate user are readily available, since the user would choose her specific sequence of mouse operations and would not share it with others. In addition, no datasets are published in previous research, which makes it difficult for third-party verification of previous work and precludes objective comparisons between different approaches.

A. Overview of Approach

Faced with the above challenges, our study aims to develop a mouse-dynamics-based user authentication approach, which can perform user authentication in a short period of time while maintaining high accuracy. By using a controlled experimental environment, we have isolated inherent behavioral characteristics as the primary factors for mouse-behavior analysis. The overview of the proposed approach is shown in Fig. 1. It consists of three major modules: (1) mouse-behavior capture, (2) feature construction, and (3) training/classification. The first module serves to create a mouse-operation task, and to capture and interpret mouse-behavior data. The second module is used to extract holistic and procedural features to characterize mouse behavior, and to map the raw features into distance-based features by using various distance metrics. The third module, in the training phase, applies kernel PCA on the distance-based feature vectors to compute the predominant feature components, and then builds the user's profile using a one-class classifier. In the classification phase, it determines the user's identity using the trained classifier in the distance-based feature eigenspace.

B. Purpose and Contributions of This Paper

This paper is a significant extension of an earlier and much shorter version [40]. The main purpose and major contributions of this paper are summarized as follows:

- We address the problem of unintended side-effects of inconsistent experimental conditions and environmental variables by restricting users' mouse operations to a tightly-controlled environment. This isolates inherent behavioral characteristics as the principal factors in mouse behavior analysis, and substantially reduces the effects of external confounding factors.
- Instead of the descriptive statistics of mouse behaviors usually adopted in existing work, we propose newly-defined procedural features, such as movement speed curves, to characterize a user's unique mouse-behavior characteristics in an accurate and fine-grained manner. These features could lead to a performance boost both in authentication accuracy and authentication time.
- We apply distance metrics and kernel PCA to obtain a distance-based eigenspace for efficiently representing the original mouse feature space. These techniques partially handle behavioral variability, and make our proposed approach stable and robust to variability in behavior data.
- We employ one-class learning methods to perform the user authentication task, so that the detection model is built solely on the data from the legitimate user. One-class methods are more suitable for mouse-dynamics-based user authentication in real-world applications.
- We present a repeatable and objective evaluation procedure to investigate the effectiveness of our proposed approach through a series of experiments. As far as we know, no earlier work made informed comparisons between different features and results, due to the lack of a standard test protocol. Here we provide comparative experiments to further examine the validity of the proposed approach.
- A public mouse-behavior dataset is established (see Section III for availability), not only for this study but also to foster future research. This dataset contains high-quality mouse-behavior data from 37 subjects. To our knowledge, this study is the first to publish a shared mouse-behavior dataset in this field.
- This study develops a mouse-dynamics-based user authentication approach that performs user authentication in a short time while maintaining high accuracy. It has several desirable properties:
  1. it is easy to comprehend and implement;
  2. it requires no specialized hardware or equipment to capture the biometric data;
  3. it requires only about 12 seconds of mouse-behavior data to provide good, steady performance.

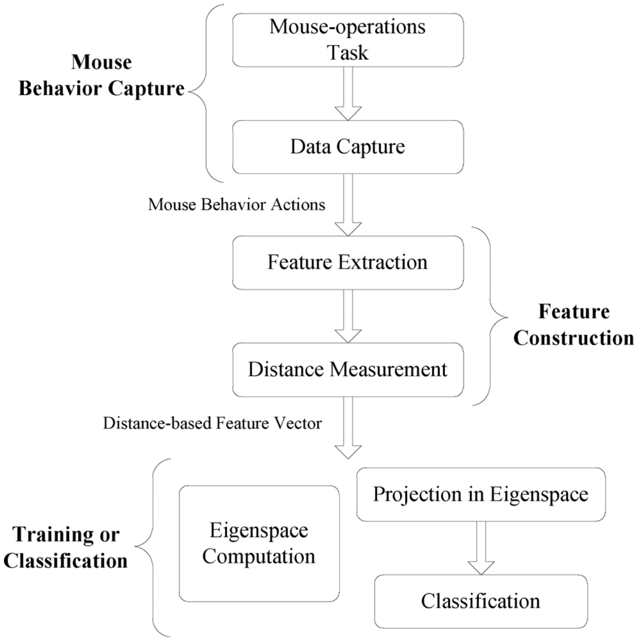


Fig. 1. Overview of approach.

The remainder of this paper is organized as follows: Section II describes related work. Section III presents a data-collection process. Section IV describes the feature-construction process. Section V discusses the classification techniques for mouse dynamics. Section VI presents the evaluation methodology. Section VII presents and analyzes experimental results. Section VIII offers a discussion and possible extensions of the current work. Finally, Section IX concludes.

II. Background and Related Work

In this section, we provide background on mouse-dynamics research, and various applications for mouse dynamics (e.g., authentication versus intrusion detection). Then we focus on applying mouse dynamics to user authentication.

A. Background of Mouse Dynamics

Mouse dynamics, a behavioral biometric for analyzing behavior data from pointing devices (e.g., mouse or touchpad), provides user authentication in an accessible and convenient manner [2]–[4], [8], [14]–[17], [19], [21], [22], [33], [34], [39]–[41], [45], [46]. Since Everitt and McOwan [14] first investigated in 2003 whether users could be distinguished by the use of a signature written by mouse, several different techniques and uses for mouse dynamics have been proposed.

Most researchers focus on the use of mouse dynamics for intrusion detection (sometimes called identity monitoring or reauthentication), which analyzes mouse-behavior characteristics throughout the course of interaction. Pusara and Brodley [33] proposed a reauthentication scheme using mouse dynamics for user verification. This study presented positive findings, but cautioned that their results were only preliminary. Gamboa and Fred [15], [16] were some of the earliest researchers to study identity monitoring based on mouse movements. Later on, Ahmed and Traore [3] proposed an approach combining keystroke dynamics with mouse dynamics for intrusion detection. Then they considered mouse dynamics as a standalone biometric for intrusion detection [2]. Recently, Zheng *et al.* [46] proposed angle-based metrics of mouse movements for reauthentication systems, and explored the effects of environmental factors (e.g., different machines).

Yet only recently have researchers come to the use of mouse dynamics for user authentication (sometimes called static authentication), which analyzes mouse-behavior characteristics at particular moments. In 2007, Gamboa *et al.* [17] extended their approaches in identity monitoring [15], [16] into web-based authentication. Later on, Kaminsky *et al.* [22] presented an authentication scheme using mouse dynamics for identifying online game players. Then, Bours and Fullu [8] proposed an authentication approach by requiring users to make use of the mouse for tracing a maze-like path.

Most recently, a full survey of the existing work in mouse dynamics pointed out that mouse-dynamics research should focus on reducing authentication time and taking the effect of environmental variables into account [21].

B. User Authentication Based on Mouse Dynamics

The primary focus of previous research has been on the use of mouse dynamics for intrusion detection or identity monitoring. It is difficult to transfer previous work directly from intrusion detection to authentication, however, because a rather long authentication period is typically required to collect sufficient mouse-behavior data to enable reasonably accurate verification. To our knowledge, few papers have targeted the use of mouse dynamics for user authentication, which will be the central concern of this paper.

Hashia *et al.* [19] and Bours *et al.* [8] presented some preliminary results on mouse dynamics for user authentication. They both asked participants to perform fixed sequences of mouse operations, and they analyzed behavioral characteristics of mouse movements to authenticate a user during the login stage. Distance-based classifiers were established to compare the verification data with the enrollment data. Hashia *et al.* collected data from 15 participants using the same computer, while Bours *et al.* collected data from 28 subjects using different computers; they achieved equal-error rates of 15% and 28% respectively.

Gamboa *et al.* [17] presented a web-based user authentication system based on mouse dynamics. The system displayed an on-screen virtual keyboard, and required users to use the mouse to enter a paired username and pin-number. The extracted feature space was reduced to a best subspace through a greedy search process. A statistical model based on the Weibull distribution was built on training data from both legitimate and impostor users. Based on data collected from 50 subjects, the researchers reported an equal-error rate of 6.2%, without explicitly reporting authentication time. The test data were also used for feature selection, which may lead to an overly optimistic estimate of authentication performance [18].

Recently, Revett *et al.* [34] proposed a user authentication system requiring users to use the mouse to operate a graphical, combination-lock-like GUI interface. A small-scale evaluation involving 6 subjects yielded an average false-acceptance rate and false-rejection rate of around 3.5% and 4% respectively, using a distance-based classifier. However, experimental details such as experimental apparatus and testing procedures were not explicitly reported.

Aksari *et al.* [4] presented an authentication framework for verifying users based on a fixed sequence of mouse movements. Features were extracted from nine movements among seven squares displayed consecutively on the screen. They built a classifier based on scaled Euclidean distance using data from both legitimate users and impostors. The researchers reported an equal-error rate of 5.9% over 10 users' data collected from the same computer, but authentication time was not reported.

It should be noted that the above two studies were performed on a small number of users—only 6 users in [34], and 10 users in [4]—which may be insufficient to evaluate definitively the performance of these approaches.

The results of the above studies have been mixed, possibly due to the realism of the experiments, possibly due to a lack of real differences among users, or possibly due to experimental errors or faulty data. A careful reading of the literature suggests that (1) most approaches have resulted in low performance, or have used a small number of users, but since these studies do not tend to be replicated, it is hard to pin the discrepancies on any one thing; (2) no research group provided a shared dataset. In our study, we control the experimental environment to increase the likelihood that our results will be free from experimental confounding factors, and we attempt to develop a simple and efficient user authentication approach based on mouse dynamics. We also make our data available publicly.

III. Mouse Data Acquisition

In this study, we collect mouse-behavior data in a controlled environment, so as to isolate behavioral characteristics as the principal factors in mouse behavior analysis. We offer here considerable detail regarding the conduct of data collection, because these particulars can best reveal potential biases and threats to experimental validity [27]. Our data set is available [1].

A. Controlled Environment

In this study, we set up a desktop computer and developed a Windows application as a uniform hardware and software platform for the collection of mouse-behavior data. The desktop was an HP workstation with a Core 2 Duo 3.0 GHz processor and 2 GB of RAM. It was equipped with a 17″ HP LCD monitor (set at 1280 × 1024 resolution) and a USB optical mouse, and ran the Windows XP operating system. Most importantly, all system parameters relating to the mouse, such as speed and sensitivity configurations, were fixed.

The Windows application, written in C#, prompted a user to conduct a mouse-operation task. During data collection, the application displayed the task in a full-screen window on the monitor, and recorded (1) the corresponding mouse operations (e.g., mouse-single-click), (2) the positions at which the operations occurred, and (3) the timestamps of the operations. The Windows-event clock was used to timestamp mouse operations [28]; it has a resolution of 15.625 milliseconds, corresponding to 64 updates per second.

When collecting data, each subject was invited to perform a mouse-operations task on the same desktop computer free of other subjects; data collection was performed one by one on the same data-collection platform. These conditions make hardware and software factors consistent throughout the process of data collection over all subjects, thus removing unintended side-effects of unrelated hardware and software factors.

B. Mouse-Operation Task Design

To reduce behavioral variations due to different mouse-operation sequences, all subjects were required to perform the same sequence of mouse operations. We designed a mouse-operation task, consisting of a fixed sequence of mouse operations, and made these operations representative of a typical and diverse combination of mouse operations. The operations were selected according to (1) two elementary operations of mouse clicks: single click and double click; and (2) two basic properties of mouse movements: movement direction and movement distance [2], [39]. As shown in Fig. 2, movement directions are numbered from 1 to 8, and each of them is selected to represent one of eight 45-degree ranges over 360 degrees. In addition, three distance intervals are considered to represent short-, middle- and long-distance mouse movements. Table I shows the directions and distances of the mouse movements used in this study. During data collection, every two adjacent movements were separated by either a single click or a double click. As a whole, the designed task consists of 16 mouse movements, 8 single clicks, and 8 double clicks.

It should be noted that our task may not be unique. However, the task was carefully chosen to induce users to perform a wide variety of mouse movements and clicks that were both typical and diverse in an individual's repertoire of daily mouse behaviors.

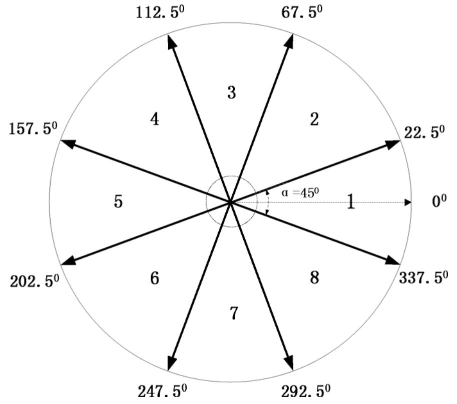


Fig. 2. Mouse movement directions: sector 1 covers all operations performed with angles between −22.5 degrees and +22.5 degrees.

TABLE I
MOUSE MOVEMENTS IN THE DESIGNED MOUSE-OPERATION TASK

Movement No.	Direction	Distance (Pixels)
1	↑	100
2	→	400
3	↙	700
4	↗	700
5	↓	400
6	↖	524
7	↘	100
8	←	100

C. Subjects

We recruited 37 subjects, many from within our lab, but some from the university at large. Our sample of subjects consisted of 30 males and 7 females. All of them were right-handed users, and had been using a mouse for a minimum of two years.

D. Data-Collection Process

All subjects were required to participate in two rounds of data collection per day, and waited at least 24 hours between collections (ensuring that some day-to-day variation existed within the data). In each round, each subject was invited, one by one, to perform the same mouse-operation task 10 times. A mouse-operation sample was obtained when a subject performed the task one time, in which she first clicked a start button on the screen, then moved the mouse to click subsequent buttons prompted by the data-collection application.

Additionally, subjects were instructed to use only the external mouse device, and they were advised that no keyboard would be needed. Subjects were told that if they needed a break or needed to stretch their hands, they were to do so after they had accomplished a full round. This was intended to prevent artificially anomalous mouse operations in the middle of a task. Subjects were admonished to focus on the task, as if they were logging into their own accounts, and to avoid distractions, such as talking with the experimenter, while the task was in progress. Any error in the operating process (e.g., single-clicking a button when requiring double-clicking it) caused the current task to be reset, requiring the subject to redo it.

[1] The mouse-behavior dataset is available from: http://nskeylab.xjtu.edu.cn/projects/mousedynamics/behavior-data-set/.

TABLE II
MOUSE DYNAMICS FEATURES

Category	Mouse Features	Definitions	Units
Holistic features	Single-click statistics	Mean and standard deviation of overall time of single click operation.	Millisecond (ms)
	Double-click statistics	Mean and standard deviation of one overall time and three interval times of double click operation.	Millisecond (ms)
	Movement offset	The distance between the practical mouse trajectory and the ideal mouse trajectory (e.g., movement in a straight line) for each movement.	Pixels
	Movement elapsed time	The time between the starting point and the ending point of a mouse movement.	Millisecond (ms)
Procedural features	Speed curve against time	Movement speed as the ratio of the traveled distance between front and back neighbor points, divided by the interval time between these two points, for each movement.	Pixels/ms
	Acceleration curve against time	Movement acceleration as the ratio of the speed value between front and back neighbor points, divided by the interval time between these two points, for each movement.	Pixels/ms^2

Subjects took between 15 days and 60 days to complete data collection. Each subject accomplished 150 error-free repetitions of the same mouse-operation task. The task took between 6.2 seconds and 21.3 seconds, with an average of 11.8 seconds over all subjects. The final dataset contained 5550 samples from 37 subjects.

IV. FEATURE CONSTRUCTION

In this section, we first extract a set of mouse-dynamics features, and then we use distance-measurement methods to obtain feature-distance vectors for reducing behavioral variability. Next, we utilize an eigenspace transformation to extract principal feature components as classifier input.

A. Feature Extraction

The data collected in Section III are sequences of mouse operations, including left-single-clicks, left-double-clicks, and mouse-movements. Mouse features were extracted from these operations, and were typically organized into a vector to represent the sequence of mouse operations in one execution of the mouse-operation task. Table II summarizes the derived features in this study. We characterized mouse behavior based on two basic types of mouse operations—mouse click and mouse movement. Each mouse operation was then analyzed individually, and translated into several mouse features. Our study divided these features into two categories:

- **Holistic features**: features that characterize the overall properties of mouse behaviors during interactions, such as single-click and double-click statistics;
- **Procedural features**: features that depict the detailed dynamic processes of mouse behaviors, such as the movement speed and acceleration curves.

Most traditional features are holistic features, which suffice to obtain a statistical description of mouse behavior, such as the mean value of click times. They are easy to compute and comprehend, but they only characterize general attributes of mouse behavior. In our study, the procedural features characterize in-depth procedural details of mouse behavior. This information more accurately reflects the efficiency, agility and motion habits of individual mouse users, and thus may lead to a performance boost for authentication. Experimental results in Section VII demonstrate the effectiveness of these newly-defined features.

B. Distance Measurement

The raw mouse features cannot be used directly by a classifier, because of high dimensionality and behavioral variability. Therefore, distance-measurement methods were applied to obtain feature-distance vectors and to mitigate the effects of these issues. In the calculation of distance measurement, we first used the Dynamic Time Warping (DTW) distance [6] to compute the distance vector of procedural features. The reasons for this choice are that (1) procedural features (e.g., movement speed curve) of two data samples are not likely to consist of the exactly same number of points, whether these samples are generated by the same or by different subjects; (2) DTW distance can be applied directly to measure the distance between the procedural features of two samples without deforming either or both of the two sequences in order to get an equal number of points. Next, we applied Manhattan distance to calculate the distance vector of holistic features. The reasons for this choice are that (1) this distance is independent between dimensions, and can preserve physical interpretation of the features since its computation is the absolute value of cumulative difference; (2) previous research in related fields (e.g., keystroke dynamics) reported that the use of Manhattan distance for statistical features could lead to a better performance [23].

1) Reference Feature Vector Generation: We established the reference feature vector for each subject from her training feature vectors. Let $\mathbf{X} = \{\mathbf{x}_i\}, i = 1, 2, \ldots, n$, be the training set of feature vectors for one subject, where $\mathbf{x}_i$ is a d-dimensional mouse feature vector extracted from the ith training sample, and n is the number of training samples. Consider how the reference feature vector is generated for each subject:

Step 1: we computed the pairwise distance vector of procedural features and holistic features between all pairs of training feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. We used DTW distance to calculate the distance vector of procedural features $\mathbf{d}_{i,j}^{P}$ for measuring the similarity between the procedural components of the two feature vectors, and we applied Manhattan distance to calculate the distance vector of holistic features $\mathbf{d}_{i,j}^{H}$.

$$\begin{aligned} \mathbf{d}_{i,j}^{P} &= \mathrm{DTW}\left(\mathbf{x}_i^{P}, \mathbf{x}_j^{P}\right), \\ \mathbf{d}_{i,j}^{H} &= \left|\mathbf{x}_i^{H} - \mathbf{x}_j^{H}\right|. \end{aligned} \tag{1}$$

where $\mathbf{x}_i^{P}$ represents the procedural components of $\mathbf{x}_i$, and $\mathbf{x}_i^{H}$ represents the holistic components.

Step 2: we concatenated the distance vectors of holistic features and procedural features together to obtain a distance vector $\mathbf{d}_{i,j}$ for the training feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ by

$$\mathbf{d}_{i,j} = \left[\mathbf{d}^P_{i,j}, \mathbf{d}^H_{i,j}\right]. \tag{2}$$

Step 3: we normalized $\mathbf{d}_{i,j}$ to get a scale-invariant feature vector:

$$\bar{\mathbf{d}}_{i,j} = \left\{ \bar{d}^l_{i,j} | \bar{d}^l_{i,j} = \frac{d^l_{i,j} - \mu_l}{\sigma_l}, l = 1 \cdots d \right\}. \tag{3}$$

where $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_d\}$ is the mean of all pairwise distance vectors from the training set, and $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \ldots, \sigma_d\}$ is the corresponding standard deviation.

Step 4: for each training feature vector, we calculated the arithmetic mean distance between this vector and the remaining training vectors, and found the reference feature vector $\mathbf{x}_{\text{ref}}$ with minimum mean distance.

$$\mathbf{x}_{\text{ref}} = \mathbf{x}_k, k = \arg\min_i \frac{1}{K-1} \sum_{j=1, j\neq i}^{n} \sqrt{\sum_{l=1}^{d} \left(\bar{d}^l_{i,j}\right)^2}. \tag{4}$$

2) Feature-Distance Vector Calculation: Given the reference feature vector for each subject, we then computed the feature-distance vector between a new mouse feature vector and the reference vector. Let $\mathbf{x}_{\text{ref}}$ be the reference feature vector for one subject; then for any new feature vector $\mathbf{x}_i$ (either from the legitimate user or an impostor), we can compute the corresponding distance vector by (1), (2) and (3).

In this paper, we used all mouse features in Table II to generate the feature-distance vector. There are 10 click-related features, 16 distance-related features, 16 time-related features, 16 speed-related features, and 16 acceleration-related features, which were taken together and then transformed to a 74-dimensional feature-distance vector that represents each mouse-operation sample.

C. Eigenspace Computation: Training and Projection

It is usually undesirable to use all components in the feature vector as input for the classifier, because much of data will not provide a significant degree of uniqueness or consistency. We therefore applied an eigenspace-transformation technique to extract the principal components as classifier input.

1) Kernel PCA Training: Kernel principal component analysis (KPCA) [37] is one approach to generalizing linear PCA to nonlinear cases using kernel methods. In this study, the purpose of KPCA is to obtain the principal components of the original feature-distance vectors. The calculation process is illustrated as follows:

For each subject, the training set represents a set of feature-distance vectors drawn from her own data. Let $\mathbf{D}_i$ $(\mathbf{D}_i \in \Re^d, i = 1, 2, \ldots, n)$ be the ith feature-distance vector in the training set, and n be the number of such vectors. We first mapped the measured vectors into the hyperdimensional feature space by the nonlinear mapping $\Phi : \mathbf{D}_i \in \Re^d \rightarrow \mathbf{Z}_i \in \Re^h$. Then we can obtain the mean $\mathbf{m}_\Phi$ and sample covariance $\sum_\Phi$ of such a training set by

$$\mathbf{m}_\Phi = \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{D}_i) \tag{5}$$

$$\Sigma_\Phi = \frac{1}{n} \sum_{i=1}^{n} (\Phi(\mathbf{D}_i) - \mathbf{m}_\Phi)(\Phi(\mathbf{D}_i) - \mathbf{m}_\Phi)^T \tag{6}$$

Here we centered the mapped point with the corresponding mean as $\bar{\Phi}(\mathbf{D}_i) = \Phi(\mathbf{D}_i) - \mathbf{m}_\Phi$. The principal components were then computed by solving the eigenvalue problem:

$$\lambda \mathbf{V} = \Sigma_\Phi \mathbf{V} = \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi}(\mathbf{D}_i)^T \mathbf{V} \bar{\Phi}(\mathbf{D}_i) \tag{7}$$

where $\lambda > \mathbf{0}$ and $\mathbf{V} \neq 0$. Then, by defining a kernel matrix

$$K_{ij} := (\bar{\Phi}(\mathbf{D}_i) \cdot \bar{\Phi}(\mathbf{D}_j)) = \mathbf{k}(\mathbf{D}_i, \mathbf{D}_j) \tag{8}$$

we computed an eigenvalue problem for the coefficients α_i, that is now solely dependent on the kernel function

$$\lambda \boldsymbol{\alpha} = K \boldsymbol{\alpha} \quad (\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T) \tag{9}$$

For details, readers can refer to B. Schölkopf *et al.* [37].

Generally speaking, the first few eigenvectors correspond to large eigenvalues and most information in the training samples. Therefore, for the sake of providing the principal components to represent mouse behavior in a low-dimensional eigenspace, and for memory efficiency, we ignored small eigenvalues and their corresponding eigenvectors, using a threshold value T_s

$$R_k = \sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{n} \lambda_i > T_s , \tag{10}$$

where R_k is the accumulated variance of the first k largest eigenvalues with respect to all eigenvalues. In this study, T_s was chosen as 0.95 for all subjects, with a range from 0 to 1. Note that we used the same T_s for different subjects, so k may be different from one subject to another. Specifically, in our experiments, we observed that the number of principal components for different subjects varied from 12 to 20, and for an average level, 17 principal components are identified under the threshold of 0.95.

2) Kernel PCA Projection: For the selected subject, taking the k largest eigenvalues and the associated eigenvectors, the transform matrix $\mathbf{V}^k = [V_1 \; V_2 \ldots V_{\text{k}}]$ can be constructed to project an original feature-distance vector $\mathbf{D}$ into a point $\mathbf{P}$ in the k-dimensional eigenspace:

$$\mathbf{P} = \mathbf{V}^k \cdot \bar{\Phi}(\mathbf{D}) = \sum_{i=1}^{n} \alpha_i^k \mathbf{k}(\mathbf{D}_i, \mathbf{D}) \tag{11}$$

As a result, each subject's mouse behavior can be mapped into a manifold trajectory in such a parametric eigenspace. It is well-known that k is usually much smaller than the dimensionality of the original feature space. That is to say, eigenspace analysis can dramatically reduce the dimensionality of input samples. In this way, we used the extracted principal components of the feature-distance vectors as input for subsequent classifiers.

V. Classifier Implementation

This section explains the classifier that we used, and introduces two other widely-used classifiers. Each classifier analyzes mouse-behavior data, and discriminates between a legitimate user and impostors.

A. One-Class Classifier Overview

User authentication is still a challenging task from the pattern-classification perspective. It is a two-class (legitimate user versus impostors) problem. In the scenario of mouse-dynamics-based user authentication, a login user is required to provide the user name and to perform a specific mouse-operation task which would be secret, like a password. Each user would choose her own mouse-operations task, and would not share that task with others. Thus, when building a model for a legitimate user, the only behavioral samples of her specific task are her own; other users' (considered as impostors in our scenario) samples of this task are not readily available. In this scenario, therefore, an appropriate solution is to build a model based only on the legitimate user's data samples, and use that model to detect impostors. This type of problem is known as one-class classification [43] or novelty/anomaly detection [25], [26]. We thus focused our attention on this type of problem, especially because in a real-world situation we would not have impostor renditions of a legitimate user's mouse operations anyway.

B. Our Classifier—One-Class Support Vector Machine

Traditional one-class classification methods are often unsatisfying, frequently missing some true positives and producing too many false positives. In this study, we used a one-class Support Vector Machine (SVM) classifier, introduced by Scholkopf *et al.* [36], [38]. One-class SVMs have been successfully applied to a number of real-life classification problems, e.g., face authentication, signature verification and keystroke authentication [1], [23].

In our context, given l training samples $\{\mathbf{x}_i \in \Re^d\}$ belonging to one subject, $i = 1, \ldots, l$, each sample has d features (corresponding to the principal components of the feature-distance vector for that subject). The aim is to find a hyperplane that separates the data points by the largest margin. To separate the data points from the origin, one needs to solve the following dual quadratic programming problem [36], [38]:

$$\begin{aligned} \max_\beta \quad & W(\beta) = \sum_{i=1}^{l} \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & 0 \le \beta_i \le \frac{1}{vl}, i = 1, \ldots, l; \sum_{i=1}^{m} \beta_i = 1 \end{aligned} \tag{12}$$

where $\beta = \{\beta_i\}$ is the vector of l nonnegative Lagrangian multipliers to be determined, v is a parameter that controls the trade-off between maximizing the number of data points contained by the hyperplane and the distance of the hyperplane from the origin, and $k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function. We allow for nonlinear decision boundaries. Then the decision function

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} \beta_i k(\mathbf{x}_i, \mathbf{x}_j) - \rho\right) \tag{13}$$

will be positive for the examples $\mathbf{x}_i$ from the training set, where ρ is the offset of the decision function.

In essence, we viewed the user authentication problem as a one-class classification problem. In the training phase, the learning task was to build a classifier based on the legitimate subject's feature samples. In the testing phase, the test feature sample was projected into the same high-dimensional space, and the output of the decision function was recorded. We used a radial basis function (RBF) $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma > 0$ in our evaluation, after comparative studies of linear, polynomial, and sigmoid kernels based on classification accuracy. The SVM parameter v and kernel parameter γ (using LibSVM [11]) were set to 0.06 and 0.004 respectively. The decision function would generate "$+1$" if the authorized user's test set is input; otherwise it is a false rejection case. On the contrary, "-1" should be obtained if the impostors' test set is the input; otherwise a false acceptance case occurs.

C. Other Classifiers—Nearest Neighbor and Neural Network

In addition, we compared our classifier with two other widely-used classifiers, KNN and neural network [12]. For KNN, in the training phase, the nearest neighbor classifier estimated the covariance matrix of the training feature samples, and saved each feature sample. In the testing phase, the nearest neighbor classifier calculated Mahalanobis distance from the new feature sample to each of the samples in the training data. The average distance, from the new sample to the k nearest feature samples from the training data, was used as the anomaly score. After multiple tests with k ranging from 1 to 5, we obtained the best results with $k = 3$, detailed in Section VII.

For the neural network, in the training phase a network was built with p input nodes, one output node, and $\lfloor 2p/3 \rfloor$ hidden nodes. The network weights were randomly initialized between 0 and 1. The classifier was trained to produce a 1.0 on the output node for every training feature sample. We trained for 1000 epochs using a learning rate of 0.001. In the testing phase, the test sample was run through the network, and the output of the network was recorded. Denote s to be the output of the network; intuitively, if s is close to 1.0, the test sample is similar to the training samples, and with s close to 0.0, it is dissimilar.

VI. Evaluation Methodology

This section explains the evaluation methodology for mouse behavior analysis. First, we summarize the dataset collected in Section III. Next, we set up the training and testing procedure for our one-class classifiers. Then, we show how classifier performance was calculated. Finally, we introduce a statistical testing method to further analyze experimental results.

A. Dataset

As discussed in Section III, samples of mouse-behavior data were collected when subjects performed the designed mouse-operation task in a tightly-controlled environment. All 37 subjects produced a total of 5550 mouse-operation samples. We then calculated feature-distance vectors, and extracted principal components from each vector as input for the classifiers.

B. Training and Testing Procedure

Consider a scenario as mentioned in Section V-A. We started by designating one of our 37 subjects as the legitimate user, and the rest as impostors. We trained the classifier and tested its ability to recognize the legitimate user and impostors as follows:

Step 1: We trained the classifier to build a profile of the legitimate user on a randomly-selected half of the samples (75 out of 150 samples) from that user.

Step 2: We tested the ability of the classifier to recognize the legitimate user by calculating anomaly scores for the remaining samples generated by the user. We designated the scores assigned to each sample as *genuine scores*.

Step 3: We tested the ability of the classifier to recognize impostors by calculating anomaly scores for all the samples generated by the impostors. We designated the scores assigned to each sample as *impostor scores*.

This process was then repeated, designating each of the other subjects as the legitimate user in turn. In the training phase, 10-fold cross validation [24] was employed to choose parameters of the classifiers. Since we used a random sampling method to divide the data into training and testing sets, and we wanted to account for the effect of this randomness, we repeated the above procedure 50 times, each time with independently selected samples drawn from the entire dataset.

C. Calculating Classifier Performance

To convert these sets of classification scores of the legitimate user and impostors into aggregate measures of classifier performance, we computed the false-acceptance rate (FAR) and false-rejection rate (FRR), and used them to generate an ROC curve [42]. In our evaluation, for each user, the FAR is calculated as the ratio between the number of false acceptances and the number of test samples of impostors; the FRR is calculated as the ratio between the number of false rejections and the number of test samples of legitimate users. Then we computed the average FAR and FRR over all subjects.

Whether or not a mouse-operation sample generates an alarm depends on the threshold for the anomaly scores. An anomaly score over the threshold indicates an impostor, while a score under the threshold indicates a legitimate user. In many cases, to make a user authentication scheme deployable in practice, minimizing the possibility of rejecting a true user (lower FRR) is sometimes more important than lowering the probability of accepting an impostor [46]. Thus we adjusted the threshold according to the FRR for the training data. Since calculation of the FRR requires only the legitimate user's data, no impostor data was used for determining the threshold. Specifically, the threshold is set to be a variable ranging from $[-1, 1]$, and will be chosen with a relatively low FRR using 10-fold cross validation on the training data. After multiple tests, we observe that setting the threshold to a value of 0.1 yields a low FRR on average[2]. Thus, we show results with a threshold value of 0.1 throughout this study.

D. Statistical Analysis of the Results

To evaluate the performance of our approach, we developed a statistical test using the *half total error rate* (HTER) and confidence-interval (CI) evaluation [5]. The HTER test aims to statistically evaluate the performance for user authentication, which is defined by combining false-acceptance rate (FAR) and false-rejection rate (FRR):

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2} \tag{14}$$

Confidence intervals are computed around the HTER as $\mathrm{HTER} \pm \sigma \cdot Z_{\alpha/2}$, and σ and $Z_{\alpha/2}$ are computed by [5]:

$$\sigma = \sqrt{\frac{\mathrm{FAR}(1-\mathrm{FRR})}{4 \cdot \mathrm{NI}} + \frac{\mathrm{FRR}(1-\mathrm{FAR})}{4 \cdot \mathrm{NG}}} \tag{15}$$

$$Z_{\alpha/2} = \begin{cases} 1.645 & \text{for } 90\% \text{ CI} \\ 1.960 & \text{for } 95\% \text{ CI} \\ 2.576 & \text{for } 99\% \text{ CI} \end{cases} \tag{16}$$

where NG is the total number of genuine scores, and NI is the total number of impostor scores.

VII. Experimental Results and Analysis

Extensive experiments were carried out to verify the effectiveness of our approach. First, we performed the authentication task using our approach, and compared it with two widely-used classifiers. Second, we examined our primary results concerning the effect of eigenspace transformation methods on classifier performance. Third, we explored the effect of sample length on classifier performance, to investigate the trade-off between security and usability. Two additional experiments are provided to compare our method with other approaches in the literature.

A. Experiment 1: User Authentication

In this section, we conducted a user authentication experiment, and compared our classifier with two widely-used ones as mentioned in Section V-C. The data used in this experiment consisted of 5550 samples from 37 subjects. Fig. 3 and Table III show the ROC curves and average FARs and FRRs of the authentication experiment for each of three classifiers, with standard deviations in parentheses. Table III also includes the average authentication time, which is the sum of the average time needed to collect the data and the average time needed to make the authentication decision (note that since the latter of these two times is always less than 0.003 seconds in our classifiers, we ignore it in this study).

Our first observation is that the best performance has a FAR of 8.74% and a FRR of 7.96%, obtained by our approach (one-class SVM). This result is promising and competitive, and the behavioral samples are captured over a much shorter period of time

[2] Note that for different classifiers, there are different threshold intervals. For instance, the threshold interval for neural network detector is [0, 1], and for one-class SVM, it is $[-1, 1]$. For uniform presentation, we mapped all of intervals to $[-1, 1]$.

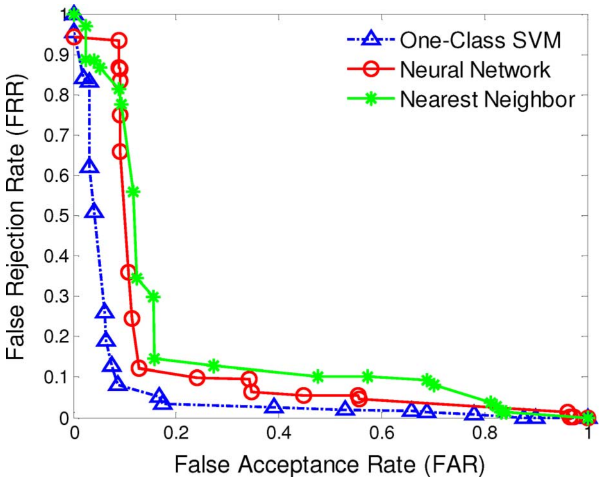


Fig. 3. ROC curves for the three different classifiers used in this study: one-class SVM, neural network, and nearest neighbor.

TABLE III
FARs AND FRRs OF USER AUTHENTICATION EXPERIMENT (WITH STANDARD DEVIATIONS IN PARENTHESES)

Classifier	FAR (%)	FRR (%)
1-class SVM	8.74 (4.26)	7.96 (4.23)
Neural Network (BP)	12.78 (5.67)	12.22 (5.12)
3-Nearest Neighbor (Standard)	15.67 (10.74)	14.53 (10.62)
Average authentication time: 11.8 seconds		

compared with previous work. It should be noted that our result does not yet meet the European standard for commercial biometric technology, which requires near-perfect accuracy of 0.001% FAR and 1% FRR [10]. But it does demonstrate that mouse dynamics could provide valuable information in user authentication tasks. Moreover, with a series of incremental improvements and investigations (e.g., outlier handling), it seems possible that mouse dynamics could be used as, at least, an auxiliary authentication technique, such as an enhancement for conventional password mechanisms.

Our second observation is that our approach has substantially better performance than all other classifiers considered in our study. This may be due to the fact that SVMs can convert the problem of classification into quadratic optimization in the case of relative insufficiency of prior knowledge, and still maintain high accuracy and stability. In addition, the standard deviations of the FAR and FRR for our approach are much smaller than those for other classifiers, indicating that our approach may be more robust to variable behavior data and different parameter selection procedures.

Our third observation is that the average authentication time in our study is 11.8 seconds, which is impressive and achieves an acceptable level of performance for a practical application. Some previous approaches may lead to low availability due to a relatively-long authentication time. However, an authentication time of 11.8 seconds in our study shows that we can perform mouse-dynamics analysis quickly enough to make it applicable to authentication for most login processes. We conjecture that the significant decrease of authentication time is due to procedural features providing more detailed and fine-grained information about mouse behavior, which could enhance performance.

TABLE IV
HTER PERFORMANCE AND CONFIDENCE INTERVAL AT DIFFERENT CONFIDENCE LEVELS

Algorithm	HTER (%)	Confidential Interval (%) around HTER for		
		90%	95%	99%
1-class SVM	8.35	±2.72	±3.24	±4.25
Neural Network (BP)	12.50	±3.21	±3.83	±5.03
3-Nearest Neighbor	15.10	±3.50	±4.17	±5.48

Finally, we conducted a statistical test, using the HTER and CI evaluation as mentioned in Section VI-D, to statistically evaluate the performance of our approach. Table IV summarizes the results of this statistical evaluation at different confidence levels. The result shows that the proposed approach provides the lowest HTER in comparison with the other two classifiers used in our study; the 95% confidence interval lies at $8.35\% \pm 3.24\%$.

B. Experiment 2: Effect of Eigenspace Transformation

This experiment examined the effect of eigenspace-transformation methods on classifier performance. The data used were the same as in Experiment 1. We applied a one-class SVM classifier in three evaluations, with the inputs respectively set to be the original feature-distance vectors (without any transformations), the projection of feature-distance vectors by PCA, and the projection of feature-distance vectors by KPCA. Fig. 4 and Table V show the ROC curves and average FARs and FRRs for each of three feature spaces, with standard deviations in parentheses.

As shown in Fig. 4 and Table V, the authentication accuracy for the feature space transformed by KPCA is the best, followed by the accuracies for feature spaces by PCA and the original one. Specifically, direct classification in the original feature space (without transformations) produces a FAR of 15.45% and FRR of 15.98%. This result is not encouraging compared to results previously reported in the literature. However, as mentioned in Experiment 1, the samples may be subject to more behavioral variability compared with previous work, because previous work analyzed mouse behaviors over a longer period of observation. Moreover, we observe that the authentication results of $\mathrm{FAR} = 10.67\%, \mathrm{FRR} = 10.12\%$ by PCA, and $\mathrm{FAR} = 8.74\%, \mathrm{FRR} = 7.96\%$ by KPCA are much better than for direct classification. This result is a demonstration of the effectiveness of the eigenspace transformation in dealing with variable behavior data. Furthermore, we find that the performance of KPCA is slightly superior to that of PCA. This may be due to the nonlinear variability (or noise) existing in mouse behaviors, and KPCA can reduce this variability (or noise) by using kernel transformations [29]. It is also of note that the standard deviations of FAR and FRR based on the feature space transformed by KPCA and PCA are smaller than those of the original feature space (without transformations), indicating that the eigenspace-transformation technique enhances the stability and robustness of our approach.

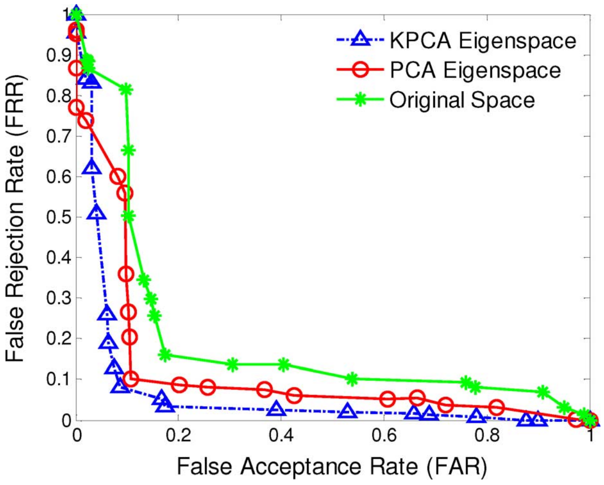


Fig. 4. ROC curves for three different feature spaces: the original feature space, the projected feature space by PCA, and the projected feature space by KPCA.

TABLE V
FARs and FARs for Three Different Feature Spaces (With Standard Deviations in Parentheses)

Classifier	FAR (%)	FRR (%)
1-class SVM (+ KPCA)	8.74 (4.26)	7.96 (4.23)
1-class SVM (+ PCA)	10.67 (6.67)	10.12 (7.45)
1-class SVM	15.45 (11.46)	15.98 (10.53)

C. Experiment 3: Effect of Sample Length

This experiment explored the effect of sample length on classifier performance, to investigate the trade-off between security (authentication accuracy) and usability (authentication time). In this study, the sample length corresponds to the number of mouse operations needed to form one data sample. Each original sample consists of 32 mouse operations. To explore the effect of sample length on the performance of our approach, we derived new datasets with different sample lengths s_n by applying bootstrap sampling techniques [13] to the original dataset, to make derived datasets containing the same numbers of samples as the original dataset. The new data samples were generated in the form of multiple consecutive mouse samples from the original dataset. In this way, we considered classifier performance as a function of the sample length using all bootstrap samples derived from the original dataset. We conducted the authentication experiment again (using one-class SVM) on six derived datasets, with $s_n = 32, 80, 160, 320, 480$ and 800 operations.

Table VI shows the FARs and FRRs at varying sample lengths, using a one-class SVM classifier. The table also includes the authentication time in seconds. The FAR and FRR obtained using a sample length of 32 mouse operations are 8.74% and 7.96% respectively, with an authentication time of 11.8 seconds. As the number of operations increases, the FAR and FRR drop to 6.97% and 6.68% for the a data sample comprised of 80 mouse operations, corresponding to an authentication time of 29.88 seconds. Therefore, we may conclude that classifier performance almost certainly gets better as the sample length increases. Note that 60 seconds may be an upper bound for authentication time, but the corresponding FAR of 4.69% and FRR of 4.46% are still not low enough to meet the needs of the European Standard for commercial biometric technology [10]. We find that after observing 800 mouse operations, our approach can obtain a FAR of 0.87% and a FRR of 0.69%, which is very close to the European standard, but with a corresponding authentication time of about 10 minutes. This long authentication time may limit applicability in real systems. Thus, a trade-off must be made between security and user acceptability, and more investigations and improvements should be performed to secure a place for mouse dynamics in more pragmatic settings.

TABLE VI
FARs and FRRs of Different Sample Lengths

Sample length (Number of mouse operations)	FAR (%)	FRR (%)	Authentication time (seconds)
32	8.74	7.96	11.80
80	6.97	6.68	29.88
160	4.69	4.46	59.49
320	3.33	2.12	118.14
480	1.67	1.27	295.14
800	0.87	0.69	588.62

D. Comparison

User authentication through mouse dynamics has attracted growing interest in the research community. However, there is no shared dataset or baseline algorithm for measuring and determining what factors affect performance. The unavailability of an accredited common dataset (such as the FERET database in face recognition [32]) and standard evaluation methodology has been a limitation in the development of mouse dynamics. Most researchers trained their models on different feature sets and datasets, but none of them made informed comparisons among different mouse feature sets and different results. Thus two additional experiments are offered here to compare our approach with those in the literature.

1) Comparison 1: Comparison With Traditional Features: As stated above, we constructed the feature space based on mouse clicks and mouse movements, consisting of holistic features and procedural features. To further examine the effectiveness of the features constructed in this study, we provide a comparative experiment. We chose the features used by Gamboa *et al.* [17], Aksari and Artuner [4], Hashia *et al.* [19], Bours and Fullu [8], and Ahmed and Traore [2], because they were among the most frequently cited, and they represented a relatively diverse set of mouse-dynamics features. We then used a one-class SVM classifier to conduct the authentication experiment again on our same dataset with both the feature set defined in our study, and the feature sets used in other studies. Hence, the authentication accuracies of different feature sets can be compared.

Fig. 5 and Table VII show the ROC curves and average FARs and FRRs for each of six feature sets, with standard deviations in parentheses. We can see that the average error rates for the feature set from our approach are much lower than those of the feature sets from the literature. We conjecture that this may be due to the procedural features providing fine-grained information about mouse behavior, but they may also be due, in part, to: (1) partial adoption of features defined in previous approaches

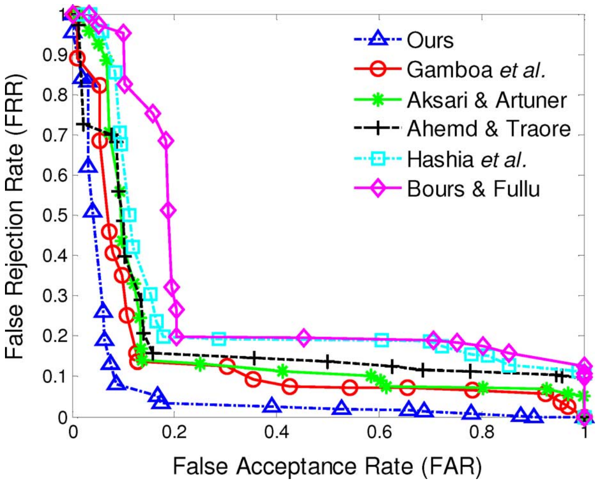


Fig. 5. ROC curves for six different feature sets: the feature set in our study, and the features sets in other studies.

TABLE VII
RESULTS OF COMPARISON WITH SOME TRADITIONAL FEATURES (WITH STANDARD DEVIATIONS IN PARENTHESES)

Classifier: one-class SVM		
Feature sets source	**FAR (%)**	**FRR (%)**
Our work	8.74 (4.26)	7.96 (4.23)
Gamboa *et al.* [17]	12.87 (9.46)	13.63 (9.73)
Aksari and Artuner [4]	13.76 (11.24)	13.85 (10.35)
Ahmed and Traore [2][3]	15.89 (7.63)	15.67 (6.92)
Hashia *et al.* [19]	17.85 (12.35)	19.86 (13.56)
Bours and Fullu [8]	20.32 (15.68)	19.65 (13.48)

[3] Note that this approach [2] is initially applied to intrusion detection, and we extracted parts of features closely related to mouse operations in our dataset. The reason for this decision is that we want to examine whether the features employed in intrusion detection can be used in user authentication.

because of different data-collection environments; (2) using different types of thresholds on the anomaly scores; (3) using less enrollment data than was used in previous experiments. The improved performance based on using our features also indicates that our features may allow more accurate and detailed characterization of a user's unique mouse behavior than was possible with previously used features. Another thing to note from Table VII is that the standard deviations of error rates for features in our study are smaller than those for traditional features, suggesting that our features might be more stable and robust to variability in behavior data.

One may also wonder how much of the authentication accuracy of our approach is due to the use of procedural features or holistic features. We tested our method using procedural features and holistic features separately, and the set of procedural features was the choice that proved to perform better. Specifically, we observe that the authentication accuracy of $\mathrm{FAR} = 10.32\%$, $\mathrm{FRR} = 9.78\%$ by using the set of procedural features is much better than for the set of holistic features, which have a FAR of 19.58% and a FRR of 17.96%. In combination with the result when using all features, it appears that procedural features may be more stable and discriminative than holistic features, which suggests that the procedural features contribute more to the authentication accuracy.

The results here only provide preliminary comparative results and should not be used to conclude that a certain set of mouse features is always better than others. Each feature set has its own unique advantages and disadvantages under different conditions and applications, so further evaluations and comparisons on more realistic and challenging datasets are needed.

2) Comparison 2: Comparison With Previous Work: Most previous approaches have either resulted in poor performance (in terms of authentication accuracy or time), or have used data of limited size. In this section, we show a qualitative comparison of our experimental results and settings against results of previous work (listed in Table VIII).

Revett *et al.* [34] and Aksari and Artuner [4] considered mouse dynamics as a standalone biometric, and obtained an authentication accuracy of ERR around 4% and 5.9% respectively, with a relatively-short authentication time or small number of mouse operations. But their results were based on a small pool of users (6 users in [34] and 10 users in [4]), which may be insufficient to obtain a good, steady result. Our study relies on an improved user authentication methodology and far more users, leading us to achieve a good and robust authentication performance. Ahmed and Traore [2] achieved a high authentication accuracy, but as we mentioned before, it might be difficult to use such a method for user authentication since the authentication time or the number of mouse operations needed to verify a user's identity is too high to be practical for real systems. Additionally, Hashia *et al.* [19] and Bours and Fulla [8] could perform user authentication in a relatively-short time, but they reported unacceptably high error rates (EER of 15% in [19], and EER of 26.8% in [8]).

In our approach we can make an authentication decision with a reasonably short authentication time while maintaining high accuracy. We employ a one-class classifier, which is more appropriate for mouse-dynamics-based user authentication. As mentioned in Experiment 3, we can make an authentication decision in less than 60 seconds, with corresponding error rates are FAR of 4.49% and FRR of 4.46%. Although this result could be improved, we believe that, at our current performance level, mouse dynamics suffice to be a practical auxiliary authentication mechanism.

In summary, Comparison 1 shows that our proposed features outperform some traditional features used in previous studies, and may be more stable and robust to variable behavior data. Comparison 2 indicates that our approach is competitive with existing approaches in authentication time while maintaining high accuracy. More detailed statistical studies on larger and more realistic datasets are desirable for further evaluations.

VIII. DISCUSSION AND EXTENSION FOR FUTURE WORK

Based on the findings from this study, we take away some messages, each of which may suggest a trajectory for future work. Additionally, our work highlights the need for shared data and resources.

A. Success Factors of Our Approach

The presented approach achieved a short authentication time and relatively-high accuracy for mouse-dynamics-based user

TABLE VIII
COMPARISON WITH PREVIOUS WORK

Source Study	Results			Data Collection			Training data source	Application
	FAR (%)	FRR (%)	Authentication time	Hardware (Computer)	Users	Environment		
Our work	8.74	7.69	11.8 seconds	Same	37	Controlled	Owner	User authentication
	4.69	4.46	59.49 seconds					
	3.33	2.12	118.14 seconds					
Hashia *et al.* [19]	15	15	20 seconds	Same	15	Controlled	Owner	User authentication
Gamboa *et al.* [17]	6.2	6.2	Not reported[4]	N/A	50	Uncontrolled	Owner and Impostor	User authentication
Bours and Fulla [8]	26.8	26.8	Not reported[4]	Different	28	Uncontrolled	Owner and Impostor	User authentication
Revett *et al.* [34]	3.5	4	39.7 seconds	N/A	6	Uncontrolled	Owner and Impostor	User authentication
Aksari and Artuner [4]	5.9	5.9	Not reported[4]	Same	10	Controlled	Owner and Impostor	User authentication
Ahmed and Traore [2]	2.46	2.46	1033 seconds[5]	Different	22	Uncontrolled	Owner and Impostor	Intrusion detection

[4] Authentication time was not explicitly reported in [4], [8], [17]; instead, they required the user to accomplish a number of mouse operations for each authentication (15 clicks and 15 movements for [17]; 10 clicks and 9 movements for [4]; 18 short movements without pauses for [8]).

[5] Authentication time was not explicitly stated in [2]; however, it can be assumed by data-collection progress. For example, it is stated in [2] that an average of 12 hours 55 minutes of data were captured from each subject, representing an average of 45 sessions. We therefore assume that average session length is 12.55 × 60/45 = 17.22 minutes = 1033 seconds.

authentication. However, it is quite hard to point out one or two things that may have made our results better than those of previous work, because (1) past work favored realism over experimental control, (2) evaluation methodologies were inconsistent among previous work, and (3) there have been no public datasets on which to perform comparative evaluations. Experimental control, however, is likely to be responsible for much of our success. Most previous work does not reveal any particulars in controlling experiments, while our work is tightly controlled. We made every effort to control experimental confounding factors to prevent them from having unintended influence on the subject's recorded mouse behavior. For example, the same desktop computer was used for data collection for all subjects, and all system parameters relating to the mouse were fixed. In addition, every subject was provided with the same instructions. These settings suggest strongly that the differences in subjects were due to individually detectable mouse-behavior differences among subjects, and not to environmental variables or experimental conditions. We strongly advocate the control of potential confounding factors in future experiments. The reason is that controlled experiments are necessary to reveal causal connections among experimental factors and classifier performance, while realistic but uncontrolled experiments may introduce confounding factors that could influence experimental outcomes, which would make it hard to tell whether the results of those evaluations actually reflect detectable differences in mouse behavior among test subjects, or differences among computing environments.

We had more subjects (37), more repetitions of the operation task (150), and more comprehensive mouse operations (2 types of mouse clicks, 8 movement directions, and 3 movement distance ranges) than most studies did. Larger subject pools, however, sometimes make things harder; when there are more subjects there is a higher possibility that two subjects will have similar mouse behaviors, resulting in more classification errors.

We proposed the use of procedural features, such as the movement speed curve and acceleration curve, to provide more fine-grained information about mouse behavior than some traditional features. This may allow one to accurately describe a user's unique mouse behavior, thus leading to a performance improvement for mouse-dynamics-based user authentication.

We adopted methods for distance measurement and eigenspace transformation for obtaining principal feature components to efficiently represent the original mouse feature space. These methods not only overcome within-class variability of mouse behavior, but also preserve between-class differences of mouse behavior. The improved authentication accuracies demonstrate the efficacy of these methods.

Finally, we used a one-class learning algorithm to perform the authentication task, which is more appropriate for mouse-dynamics-based user authentication in real applications.

In general, until there is a comparative study that stabilizes these factors, it will be hard to be definitive about the precise elements that made this work successful.

B. *Opportunities for Improvement*

While previous studies showed promising results in mouse dynamics, none of them have been able to meet the requirement of the European standard for commercial biometric technology. In this work, we determined that mouse dynamics may achieve a pragmatically useful level of accuracy, but with an impractically long authentication time. This long authentication time may limit the applicability of the technique in real systems. Therefore, how can current knowledge be turned toward improving our results? Four factors are readily apparent.

First, our operating environment was relatively impoverished; only mouse clicks and point-and-click movements are considered in this study for the sake of simplicity and efficiency. By enriching the environment to include all kinds of mouse operations (e.g., drag-and-drop), more information will be available as input for a classifier.

Second, to deal with variable behavior data, one method that seems to be more effective is to clean the raw mouse data of extraneous noise. With higher quality data, it might be possible to make authentication decisions over small quantities of data, in turn demanding less user involvement for authentication.

Third, we did not accommodate the idiosyncrasies of user mistakes. During the data-collection process, if any errors were detected, the subjects were prompted to redo the designed mouse-operation task. It is likely that these operation errors will also confer some uniqueness or consistency to the user, so this information may be used to advantage in identifying a user.

Finally, we believe that a trade-off must be made between security and user acceptability for mouse-dynamics-based user authentication. A straightforward way to handle this trade-off is to keep the number of mouse operations high when the user enrolls for the first time in the system, and to reduce this number during the testing stage. Under this perspective, an alternatively appealing research direction is the incorporation of recently introduced tactics from "anytime-algorithms" theory [9], [44], with the aim of avoiding collecting data beyond the number necessary for the system to reach an accurate decision.

C. Shared Data and Methodology

We have observed that in most previous evaluations, the datasets and evaluation methodologies are different from one study to another. Specifically, most approaches in the literature (1) trained on different feature sets; (2) had different sizes of training data; (3) used different evaluation procedures; (4) adopted different types of threshold on anomaly scores; and (5) tested their approaches on different datasets. These factors would make it difficult to compare results with each other. Thus there is a critical need for the creation of a widely recognized and public dataset and standard evaluation methodologies.

To our knowledge, this study is the first to establish a publicly available dataset of mouse dynamics. Not only would such a dataset and the repeatable method allow for the comparison of existing and future approaches, it would significantly reduce the overhead for new researchers in this field.

D. Limitations

This study has shown promising authentication performance using mouse dynamics in a controlled experimental environment, but in practice we are aware that such a controlled approach may be affected by intrinsic behavioral variability, in contrast with other physiological biometric characteristics, such as face or fingerprint patterns [20]. Behavioral variability occurs between two immediately consecutive samplings, even if the subject providing the samples strives to maintain a uniform regime of mouse operations. Real-world variability often comes from (1) hardware-level factors (e.g., mouse device type, computer type); (2) software-level factors (e.g., operating system, screen resolution, mouse speed and sensitivity configuration, mouse-event sampling rate, perceptual delays caused by high CPU load); (3) environmental factors (e.g., distance between monitor and body, height of the chair, positions of the mouse pad); and (4) psychological and physiological state of the subject (e.g., the subject may be fatigued, distracted or distressed). Thus it is reasonable to wonder whether these factors would change mouse behavioral characteristics if the state of these factors is different at enrolment time than at authentication time, or even would have some impact on a classifier's performance. Ideally, we would test all potential confounding factors, to see whether they actually do confound the experimental results. However, to do so would require an exponential amount of data (in the number of factors). Further, how these factors would affect mouse-dynamics-based user authentication is still an open question and will be investigated in our future work.

IX. Conclusion

Mouse dynamics is a newly emerging behavioral biometric, which offers a capability for identifying computer users on the basis of extracting and analyzing mouse click and movement features when users are interacting with a graphical user interface. Many prior studies have demonstrated that mouse dynamics has a rich potential as a biometric for user authentication.

In this study, we highlighted the challenges faced by mouse-dynamics-based user authentication, and we developed a simple and efficient approach that can perform the user authentication task in a short time while maintaining high accuracy. Holistic features and procedural features are extracted from the fixed mouse-operation task to accurately characterize a user's unique behavior data. Then distance-based feature construction and parametric eigenspace transformation are applied to obtain the predominant feature components for efficiently representing the original mouse feature space. Finally, a one-class classification technique is used for performing the user authentication task. This approach is evaluated on a newly-established dataset collected in a controlled experimental environment, consisting of 5550 data samples from 37 subjects. Extensive experiments have demonstrated the validity of the proposed approach, with a false-acceptance rate of 8.74%, a false-rejection rate of 7.69%, and an authentication time of 11.8 seconds. These results suggest that mouse dynamics can provide a significant enhancement for traditional authentication systems. Although not yet meeting the European standard for commercial biometric technology, this work has been shown encouraging progress on mouse-dynamics-based user authentication.

Acknowledgment

The authors are grateful to K. S. Killourhy and S.-H. Lau for their insightful comments and helpful advice, and to J. Wang for running the experiments. Thanks also to several anonymous reviewers for their helpful comments. Z. Cai is the corresponding author of this paper.

References

[1] S. Abe, *Support Vector Machines for Pattern Classification*. New York: Springer, 2005.

[2] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Trans. Depend. Secure Comput.*, vol. 4, no. 3, pp. 165–179, Jul./Sep. 2007.

[3] A. A. E. Ahmed and I. Traore, "Anomaly intrusion detection based on biometrics," in *Proc. IEEE Information Assurance Workshop*, West Point, NY, 2005, pp. 452–453.

[4] Y. Aksari and H. Artuner, "Active authentication by mouse movements," in *Proc. 24th Int. Symp. Computer and Information Science*, Guzelyurt, 2009, pp. 571–574.

[5] S. Bengio and J. Mariethoz, "A statistical significance test for person authentication," in *Proc. Speaker and Language Recognition Workshop*, Toledo, Spain, 2004, pp. 237–244.

[6] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. Advance in Knowledge Discovery in Database: Papers From the 1994 AAAI Workshop*, Jul. 1994, pp. 359–37.

[7] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Computing Surveys*, vol. 44, no. 4, 2012, to be published.

[8] P. Bours and C. J. Fullu, "A login system using mouse dynamics," in *Proc. 5th Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, 2009, pp. 1072–1077.

[9] R. Brooks, T. Arbel, and D. Precup, "Anytime similarity measures for faster alignment," *J. Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 378–389, Jun. 2008.

[10] *European Standard EN 50133-1: Alarm Systems. Access Control Systems for Use in Security Applications*, Part 1: System requirements, Standard Number EN 50133-1:1996/A1:2002, Technical Body CLC/TC 79, 2002, CENELEC, European Committee for Electrotechnical Standardization (CENELEC).

[11] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, Apr. 2011.

[12] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ: Wiley, 2001.

[13] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.

[14] R. Everitt and P. W. McOwan, "Java-based internet biometric authentication system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1166–1172, Sep. 2003.

[15] H. Gamboa and A. Fred, "A behavioral biometric system based on human computer interaction," in *Proc. SPIE*, Orlando, FL, 2004, pp. 381–392.

[16] H. Gamboa and A. Fred, "An identity authentication system based on human computer interaction behaviour," in *Proc. 3rd Int. Pattern Recognition on Information Systems Workshop*, Angers, France, 2003, pp. 46–55.

[17] H. Gamboa, A. L. N. Fred, and A. K. Jain, "Web biometrics: User verification via web interaction," in *Proc. Biometrics Symp.*, Baltimore, MD, 2007, pp. 1–6.

[18] I. Guyon and A. Elisseeff, "An introduction to variables and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[19] S. Hashia, C. Pollett, and M. Stamp, "On using mouse movements as a biometric," in *Proc. Int. Conf. Computer Science and Its Applications*, Singapore, 2005, pp. 143–147.

[20] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 125–143, Jun. 2006.

[21] Z. Jorgensen and T. Yu, "On mouse dynamics as a behavioral biometric for authentication," in *Proc. 6th ACM Symp. Information, Computer and Communication Security*, Hong Kong, 2011, pp. 476–482.

[22] R. Kaminsky, M. Enevand, and E. Andersen, Identifying Game Players With Mouse Biometrics University of Washington, Tech. rep., Aug. 2008.

[23] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proc. Int. Conf. Dependable Systems & Networks*, Estoril, Portugal, 2009, pp. 125–134.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. Artificial Intelligence*, Montreal, Canada, 1995, pp. 1137–1143.

[25] M. Markou and S. Singh, "Novelty detection: A review, Part I: Statistical approaches," *Signal Process.*, vol. 83, pp. 2481–2497, 2003.

[26] M. Markou and S. Singh, "Novelty detection: A review, Part II: Neural network based approaches," *Signal Process.*, vol. 83, pp. 2499–2521, Dec. 2003.

[27] R. A. Maxion, "Making experiments dependable," *Dependable and Historic Computing*, ser. Lecture Notes in Computer Science, vol. 6875, pp. 344–357, 2011.

[28] Microsoft developer network, EVENTMSG Structure, 2011 [Online]. Available: http://msdn2.microsoft.com/en-us/library/ms644966(VS.85).aspx

[29] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, G. Rätsch, M. S. Kearns, S. A. Solla, and D. A. Cohn, "Kernel PCA and de-noising in feature spaces," in *Proc. Advances in Neural Information Processing Systems*, Denver, CO, 1999, pp. 536–542.

[30] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Gen. Comput. Syst.*, vol. 16, no. 4, pp. 351–359, Feb. 2000.

[31] *National Strategy for Trusted Identities in Cyberspace: Why We Need It*, Nat. Inst. Standards and Technol., 2011 [Online]. Available: www.nist.gov/nstic/NSTIC-Why-We-Need-It.pdf

[32] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithm," *Image Vis. Comput.*, vol. 16, no. 10, pp. 295–306, Apr. 1998.

[33] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proc. 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, Washington, DC, 2004, pp. 1–8.

[34] K. Revett, H. Jahankhani, S. T. de Magalhes, and H. M. D. Santos, "A survey of user authentication based on mouse dynamics," in *Proc. 4th Int. Conf. Global E-Security*, London, 2008, pp. 210–219.

[35] K. N. Ross, "Sample design for educational survey research," *Eval. Educ. Int. Prog.*, vol. 2, pp. 105–195, 1978.

[36] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computat.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[37] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigen-value problem," *Neural Computat.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[38] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. Advances in Neural Information Processing Systems*, Denver, CO, 1999, pp. 526–532.

[39] D. Schulz, "Mouse curve biometrics," in *Proc. Biometrics Symp.: Special Session on Research*, Baltimore, MD, 2006, pp. 1–6.

[40] C. Shen, Z. M. Cai, and X. H. Guan, "Can it be more practical? Improving mouse dynamics biometric performance," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Chicago, IL, 2011, pp. 853–856.

[41] C. Shen, Z. M. Cai, X. H. Guan, H. L. Sha, and J. Z. Du, "Feature analysis of mouse dynamics in identity authentication and monitoring," in *Proc. IEEE Int. Conf. Communication (ICC)*, Dresden, Germany, 2009, pp. 1–5.

[42] J. A. Swets and R. M. Pickett, *Evaluation of Diagnostic Systems: Methods From Signal Detection Theory*. New York: Academic, 1982.

[43] D. M. J. Tax, "One-Class Classification: Concept-Learning in the Absence of Counter-Examples," Ph.D. Dissertation, Delft Univ. Technology, Delft, The Netherlands, Jun. 2001.

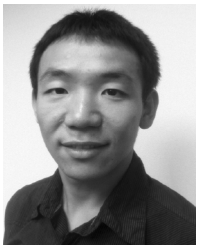
[44] K. Ueno, X. Xi, E. Keogh, and D.-J. Lee, "Anytime classification using the nearest neighbor algorithm with applications to stream mining," in *Proc. IEEE 6th Int. Conf. Data Mining (ICDM)*, Hong Kong, 2006, pp. 623–632.

[45] R. V. Yampolskiy and V. Govindaraju, "Behavioral biometric: A survey and classification," *Int. J. Biometrics*, vol. 1, no. 1, pp. 81–113, Jun. 2008.

[46] N. Zheng, A. Paloski, and H. M. Wang, "An efficient user verification system via mouse movements," in *Proc. ACM Conf. Computer and Communications Security*, Chicago, IL, 2011, pp. 139–150.

[47] [Online]. Available: http://dazzlepod.com/csdn/

[48] [Online]. Available: http://www.guardian.co.uk/world/2011/oct/27/sweden-hacking-twitter-hijack



Chao Shen (S'09) received the B.S. and M.S. degrees in automatic control from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree with the Systems Engineering Institute and SKLMS Laboratory, Xi'an Jiaotong University.

He is also a research scholar in the Computer Science Department at Carnegie Mellon University. His research interests include insider/intrusion detection, behavioral biometric, and measurement and experimental methodology.

Zhongmin Cai (M'09) received the B.S. degree in automatic control from Xi'an Jiaotong University, Xi'an, China, in 1998, and the Ph.D. degree in system engineering from Xi'an Jiaotong University, Xi'an, China, in 2004.

He is currently an Associate Professor in the School of Electronic and Information Engineering, Xi'an Jiaotong University of China. His research interests include Internet security and machine learning.

Xiaohong Guan (S'89–M'93–SM'94–F'07) received the B.S. and M.S. degrees in automatic control from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 1993.

From 1985 to 1988, he was with the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, China. From 1993 to 1995, he was a senior consulting engineer at PG&E. From January 1999 to February 2000, he was with the Division of Engineering and Applied Science, Harvard University, Cambridge, MA. Since 1995, he has been with the Systems Engineering Institute, Xi'an Jiaotong University, Xi'an, China, where he is also currently a Cheung Kong Professor of Systems Engineering and the Dean of School of Electronic and Information Engineering. He is also with the Department of Automation, Tsinghua National Laboratory for Information Science and Technology and the Center for Intelligent and Networked Systems, TNLIST, Tsinghua University, Beijing, China. His research interests include allocation and scheduling of complex networked resources, network security, and sensor networks.

Youtian Du (M'09) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2002, and the Ph.D. degree in automatic control from Tsinghua University, Beijing, China, in 2008.

He is currently an Assistant Professor in the School of Electronic and Information Engineering, Xi'an Jiaotong University of China. His research interests include web multimedia understanding, social network, and machine learning.

Roy A. Maxion (F'08) is a Research Professor in the Computer Science and Machine Learning Departments at Carnegie Mellon University. He is also director of the CMU Dependable Systems Laboratory where the range of activities includes computer security, behavioral biometrics, insider/masquerader detection, usability and keystroke forensics in addition to general issues of hardware/software system reliability and information assurance. A primary interest/concern is the correctness and completeness of experimental methodologies and measurement techniques.

Dr. Maxion has been program chair of the International Conference on Dependable Systems and Networks, member of the executive board of the IEEE Technical Committee on Fault Tolerance, the United States Defense Science Board, the European Commission AMBER advisory board, and other professional organizations. He has consulted for the U.S. Department of State as well as for numerous industry and government bodies. He has been on the editorial boards of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and the *International Journal of Security and Networks*. He is presently on the editorial boards of *IEEE Security & Privacy*, and the *International Journal of Biometrics*. Dr. Maxion is an elected member of the IFIP 10.4 working group on dependable computing and fault tolerance.