Correspondence

Improving Mouse Dynamics Biometric Performance Using Variance Reduction via Extractors With Separate Features

Youssef Nakkabi, Issa Traoré, and Ahmed Awad E. Ahmed

Abstract—**The European standard for access control imposes stringent performance requirements on commercial biometric technologies that few existing recognition systems are able to meet. In this correspondence paper, we present the first mouse dynamics biometric recognition system that fulfills this standard. The proposed system achieves notable performance improvement by developing separate models for separate feature groups involved. The improvements are achieved through the use of a fuzzy classification based on the Learning Algorithm for Multivariate Data Analysis and using a score-level fusion scheme to merge corresponding biometric scores. Evaluation of the proposed framework using mouse data from 48 users achieves a false acceptance rate of 0% and a false rejection rate of 0.36%.**

Index Terms—**Biometric fusion, biometric systems, fuzzy clustering, human computer interaction, mouse dynamics, variance reduction (VR).**

I. Introduction

One of the most recent biometric technologies proposed in the literature is based on mouse dynamics analysis [1], [2], [8], [11], [20]. Mouse dynamics biometrics is analogous in many respects to keystroke dynamics in the sense that in contrast with traditional biometric technologies (e.g., iris, fingerprints, face) [6], data collection is performed using standard hardware devices (e.g., mouse, keyboard) that are readily available in any basic computing environment. As a result, mouse dynamics biometrics can be collected and processed unobtrusively and then used to passively monitor computer users with the purpose of detecting session hijacking and masquerade attacks.

Like keystroke dynamics, along with several other behavioral biometrics [4], mouse dynamics biometrics is characterized by strong variability which tends to affect its accuracy. Human behavior is characterized by uncertainties related to a number of factors (e.g., environmental, emotional, etc.). The strong variability of mouse dynamics data, for a human, can be explained by the inability of a person to reproduce the same action with a high degree of accuracy.

A variety of techniques have been proposed for modeling user behavior based on mouse dynamics ranging from basic statistics to machine learning techniques. While approaches based on basic statistics have consistently performed poorly, encouraging results have been obtained with machine learning techniques, such as decision trees and neural networks. Despite these encouraging results, none of the existing approaches have been able to meet the European standard for access control. The European standard requires, for commercial biometric systems, a false acceptance rate (FAR) of less than 0.001% and a false rejection rate (FRR) of less than 1%. This correspondence paper takes as its point of departure the feature set and data set provided in [1] and investigates, for the first time, the use of fuzzy logic for mouse dynamics biometric analysis. Fuzzy logic provides more adequate answers (than classical logic) when dealing with the uncertainty underlying human actions. A fundamental and challenging aspect of fuzzy logic is the extraction of a fuzzy membership function. We use the Learning Algorithm for Multivariate Data Analysis (LAMDA) [17], an unsupervised learning technique, for such a purpose.

Supervised learning may be used to discriminate between users in a closed setting in which mouse data can be collected for all the users. The main weakness of a supervised approach, however, is that all users' data must be collected before classification can proceed. Even so, the approach may be hindered by the nonuniform class problem as the number of classes or users increases [20]. When public access to hosts is not restricted, as is the case in many operational environments, unsupervised learning is more suitable for mouse dynamics biometric analysis. In this case, we do not need the impostor's profile *a priori* to detect him. A normal profile is built for each authorized user during enrollment and compared against a current behavior to establish whether such behavior is genuine or intrusive.

Emerging approaches to improve biometric performance consist of reducing the overall variance of the system by combining different modalities (i.e., multimodal biometrics) [12], [21] or by using several classifiers (i.e., the ensemble approach) [7], [15]. These approaches exploit the assumed independence of the combined objects for variance reduction (VR). The comparison of several VR techniques has shown that their effectiveness is related to the degree of independence of the combined objects: more independence leads to more effectiveness [19]. The following ranking based on increasing order of effectiveness was established:

1) VR via multiple synthetic samples;
2) VR via multiple classifiers;
3) VR via multiple extractors with concatenated features;
4) VR via multiple extractors with separate features;
5) VR via multiple real samples;
6) VR via multiple modalities.

Based on this ranking, VR via multiple real samples and VR via multiple extractors appear to be the most effective techniques when focusing on single biometric modality. However, VR via multiple real samples may not always be suitable for real-time analysis due to the potentially huge amount of data involved. This is a key limitation when applied to important mouse dynamics analysis, such as intrusion detection or continuous user authentication. In this context, VR via extractor seems a better alternative.

Although VR via extractors with concatenated features approach has been used extensively in the literature in other areas, it has never been applied to the particular field of mouse biometric analysis. Existing mouse dynamics biometric analysis approaches proposed in the literature are based on VR via extractors with concatenated features. Similarly, LAMDA classification has never been applied to the field of mouse dynamics biometric analysis.

Using the features defined in [1] and corresponding data, we demonstrate, as our main contribution, a new technique for mouse biometric analysis based on VR via extractors with concatenated features and LAMDA classification. This new technique addresses the performance gap observed in existing approaches. The proposed approach yields the lowest error rates currently reported in the literature.

Manuscript received December 31, 2008; revised September 1, 2009. Date of publication July 26, 2010; date of current version October 15, 2010. This work was supported in part by a Collaborative Research and Development Grant from the National Science and Engineering Research Council and in part by a Bell University Labs Grant from Bell Canada. This paper was recommended by Associate Editor C. Cao.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada (e-mail: ynakkabi@ece.uvic.ca; itraore@ece.uvic.ca; aahmed@ece.uvic.ca).

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/TSMCA.2010.2052602

The rest of the paper is structured as follows. In Section II, we summarize and discuss related work. In Section III, we give an overview of the mouse dynamics biometric features used and introduce the proposed VR approach. In Section IV, we give an overview of the LAMDA classification approach. In Section V, we discuss the mouse dynamics biometric analysis using LAMDA. In Section VI, we evaluate the proposed approach by using the initial data set presented in [1] and also by using a more recently collected extended version of the data set [16]. Finally, in Section VII, we make some concluding remarks.

II. Related Work

Research on using mouse dynamics for biometric analysis is relatively recent which explains the limited amount of published literature on this topic. Through an exploratory study on mouse dynamics analysis, Hocquet *et al.* conducted a controlled experiment involving ten users, where users were asked to click on a moving square as quickly as possible [11]. Hocquet *et al.* first extracted features such as speed, acceleration, angular velocity, curvature, and the derivative of the curvature curve. Then, by computing basic statistics such as the maximum, the minimum, the average, the standard deviation, and the difference between the maximum and the minimum, they achieved an equal error rate (EER) of 37.5%.

Hashia *et al.* [10] extracted from the raw mouse data a feature vector consisting of 144 parameters including the average, standard deviation, minimum and maximum of parameters, such as mouse movement speed, deviation, and angle. To enroll a user, the standard deviation of each of the features over the enrollment sample was computed and stored. Verification was performed by checking whether each of the feature values of the monitored sample was within 1.5 standard deviations from the corresponding enrollment value. The total number of matching parameters was compared with a threshold computed during enrollment to accept or reject the user. Experimental evaluation of the approach with 15 users yielded $EER = 15\%$.

The poor performances achieved in [11] and [10] underscore the need for using a more sophisticated biometric analysis technique, more powerful than the basic statistics used in these papers, such as machine learning. Pusara and Brodley [20] extracted a total of 105 features from the raw mouse data used to build the user profile. A supervised learning approach based on C5.0 decision tree classification was used to verify the identity of the users. Experimental evaluation of the approach involved 18 volunteers yielding $FAR = 3.06\%$ and $FRR = 27.5\%$. A close examination of the results revealed that most of the errors were linked to users with limited mouse events (e.g., mouse clicks). By eliminating those users and applying a smoothing filter on detection accuracy, the authors reported $FAR = 1.75\%$ and $FRR = 0.43\%$ for 11 users. Unlike in our work, Pusara and Brodley used a monolithic classifier to model their complete set of features. Most importantly, while we used an unsupervised learning scheme, Pusara and Brodley used in their study a supervised learning approach. As discussed earlier, supervised learning is not suitable for mouse dynamics biometric analysis when public access to host is allowed, and this is typically the case for online operational environments.

References reporting encouraging results with unsupervised learning include [1], [2] and [8]. Ahmed and Traoré [1], [2] defined seven biometric factors, each consisting of a vector of numbers or features. Biometric analysis was conducted by concatenating the feature vectors corresponding to the seven factors into a 39-dimensional global feature vector and using neural network for modeling and classification. In the user enrollment mode, a feedforward multilayer perceptron neural network was used to learn the user behavior based on sample mouse data. The status of the trained network was used to represent the user profile and was stored in a signature database. In the detection mode, the stored status of the trained neural network was loaded, and the monitored session data was applied to the neural network to output what is referred to by the authors as the confidence ratio (CR). The CR is a percentage number that represents the degree of likeness of the two behaviors being compared. The proposed approach was evaluated with 22 participants through free and controlled experiments, achieving $EER = 2.46\%$.

We illustrate our approach using the same biometric features and data set proposed by Ahmed and Traoré [1]. A key difference between the two approaches is that, while in [1] a monolithic classifier is used to model the entire feature space, we develop a separate classifier for each of the biometric factors. Furthermore, combining our feature modeling approach with LAMDA classification allows us to achieve better performance than using a neural network.

Gamboa and Fred [8] collected mouse movements using a memory game. They extracted a 63-dimensional feature vector involving spatial parameters such as angle and curvature, and temporal parameters such as duration, position, velocity, and acceleration. The features were represented using statistical models, in which the learning consisted of estimating the class conditional probability density from each user's data. More specifically, two different statistical models for the features were tested. The two models were a nonparametric multimodal model based on Parzen density estimation and a unimodal parametric model. The unimodal parametric model was based on the Weibull distribution and assumed statistical independence between features. The parameters of the Weibull distribution were estimated for each user over their corresponding enrollment sample using maximum likelihood. Both statistical models achieved the same performance results except that the unimodal parametric model appeared to be more scalable. The Sequential Forward Selection (SFS) algorithm was used for classification. SFS is a sequential classifier that uses a greedy strategy to select the best single feature and then adds one feature at a time to the feature vector. Each time a feature is selected, the feature vector is fed to a classifier that minimizes the equal error rate of the system. The sequential classifier will accept or reject the claimed identity when the probability distribution of the user is greater than a limit that is adjusted to operate at the crossover point, corresponding to the equal error rate. Experimental evaluation of the proposed approach, based on a population of 50 volunteers, yielded $EER = 2\%$.

Similar to our work, Gamboa and Fred assume, for pragmatic purpose, statistical independence of the features and use unsupervised learning. They use, however, a density-based learning technique which requires a large number of training samples for reliable estimation of the density functions. This may not be practical for biometric systems like mouse dynamics due to the time and effort involved in collecting training samples in operational environments. In such situations, our approach based on LAMDA classification performs better. Furthermore, while the approach proposed by Gamboa and Fred used for each user only a small subset of the features (5 out of 63 on average), a monolithic classifier was used to model the selected features which is in contrast to our proposed approach.

III. VR Approach

VR via multiple extractors techniques extract and combine multiple features from raw biometric data. These techniques rely on the premise that different features capture different or complementary capabilities that, when combined, may improve the overall accuracy of the biometric system. There are two flavors of VR via multiple extractors, namely: VR via multiple extractors with concatenated features and VR via multiple extractors with separate features. In the former case, extracted features are normalized to the same range

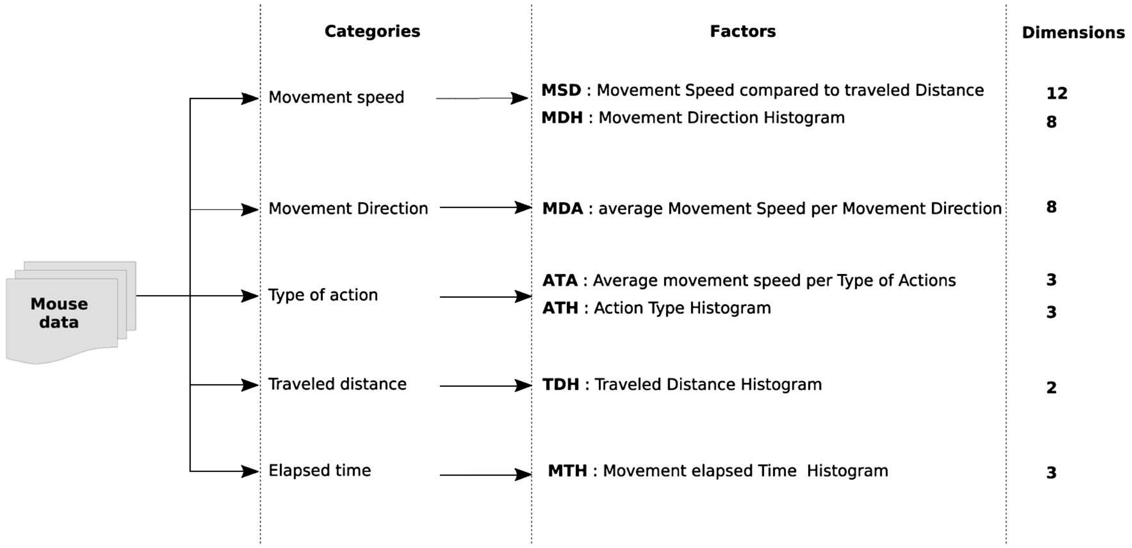

Categories
Factors
Dimensions
Mouse data
Movement speed
MSD : Movement Speed compared to traveled Distance
12
MDH : Movement Direction Histogram
8
Movement Direction
MDA : average Movement Speed per Movement Direction
8
Type of action
ATA : Average movement speed per Type of Actions
3
ATH : Action Type Histogram
3
Traveled distance
TDH : Traveled Distance Histogram
2
Elapsed time
MTH : Movement elapsed Time Histogram
3

Fig. 1. Mouse dynamics factors introduced in [1] and [2] and their dimensions. The factors are organized in five categories. Each factor consists of a vector of numbers corresponding to a specific feature group. All the factors are modeled as histograms or averages, except the MSD factor, which is represented as a curve computed using a neural network. Action types include MM, DD, PC, and silence.

and then concatenated and fed to a common classifier. In the latter case, each feature is processed through its own separate classifier, and then corresponding scores are merged using an appropriate decision fusion technique. We present in this section the feature set used in this correspondence paper and the proposed VR approach.

A. Mouse Dynamics Biometric Features

Mouse dynamics correspond to the actions generated by the mouse input device for a specific user while interacting with a graphical user interface. The actions can be grouped into four categories, namely: mouse-move (MM), drag-and-drop (DD), point-and-click (PC), and silence mode. From these four categories, the following raw data can be extracted, namely: the time elapsed, the distance traveled, and the direction or angle of each movement. Such information can be used to construct a mouse dynamics signature for a specific user. Different factors characterizing each category can be used including, for instance, the following:

- average speed for each distance;
- average speed for each movement direction;
- average traveled distance for a specified period of time, with respect to each movement direction.

Ahmed and Traoré [1] use a set of seven biometric factors to establish the user mouse signature and present a study for their reproducibility and discrimination capacity. These factors, grouped into five categories as shown in Fig. 1, are represented either by histograms or averages; however, one of the factors is approximated by a curve constructed using a neural network. Each factor provides a number of features contributing to a 39-dimensional global feature space. Although any kind of feature set could be used with our approach, we will focus this correspondence paper on the aforementioned features.

B. Proposed Approach

The analysis approach proposed in [1] falls in the category of VR via extractors with concatenated features. The feature vectors corresponding to the different biometric factors are concatenated into a 39-dimensional feature vector and used as the input to a neural network. The neural network in turn gives as output a percentage number known as the CR. The CR assesses the degree of similarity or resemblance between two behaviors. In this approach, the neural network gives the same importance to all the features in the decision-making process, as illustrated in Fig. 2(a). From a practical point of view, when dealing with human behavior, some of the features may carry more weight than others for each user.

In our approach, as illustrated in Fig. 2(b), we construct a separate model for each of the seven factors through an associated classifier and merge the resulting scores through a suitable biometric fusion scheme. As indicated earlier, we use fuzzy logic to model the features. The idea behind using fuzzy logic is to build a fuzzy model that takes into account human uncertainty. For each feature group, the elementary model will give as output a value between 0 and 1. This value reflects the adequacy of the features compared to the user reference model. In fuzzy logic, all these values are represented by a fuzzy membership's function. The extraction of such a function remains an important task in fuzzy logic. Generally, a knowledge expert or a fuzzy clustering analysis allows this construction. In our case, a knowledge expert is not available; so we propose to use LAMDA classification for this purpose. In the next section, we give an overview of the LAMDA algorithm.

IV. LAMDA Classification

LAMDA is a conceptual clustering and classification methodology that computes the degree of adequation of an object to a class with all the partial or marginal information available [17]. The difference between this algorithm and the classical clustering and classification approaches is that LAMDA models the total indistinguishability or homogeneity inside the context or universe from which the information is extracted. This is done by means of a special class, the so called *non-informative class* (NIC), which accepts all objects under the same status. Therefore, the adequation degree of the objects of NIC acts as a minimum threshold to assign an element to a significant class. Hence, the minimum threshold is not fixed arbitrarily but is automatically determined by the proper context. Algorithm 1 summarizes the main steps of LAMDA clustering. Given a set of objects $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$, where an object is represented by an m-dimensional vector $\vec{x}_k = \{x_1^k, x_2^k, \ldots, x_m^k\}$, the algorithm starts by creating an initial class consisting of one of the objects selected at random. For

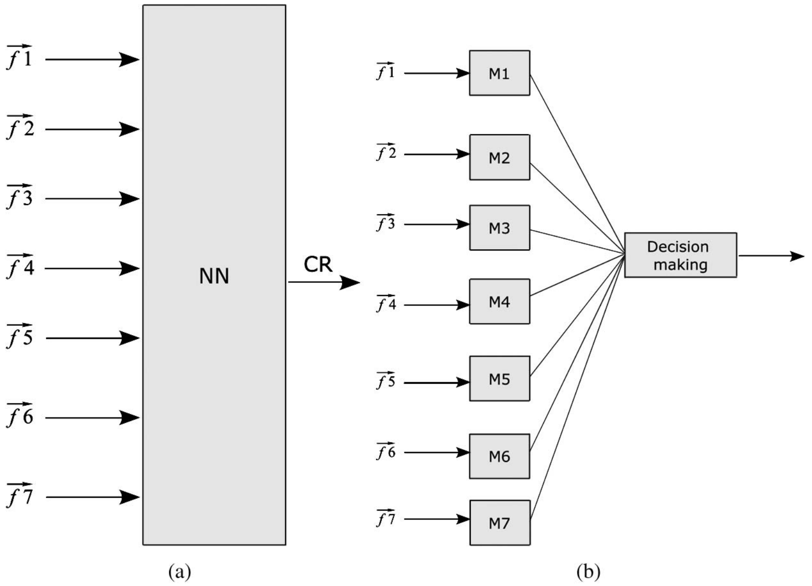


Fig. 2. Analysis model. (a) Approach used in [1]; all the seven factors, each represented by a separate feature vector $\vec{f}_{\text{i}}$, are submitted to the same analysis model consisting of a neural network (NN) classifier providing as output the CR. (b) Our proposed approach; a separate model M_{i} is constructed using LAMDA for each of the seven factors; the outcomes of the models are combined using a suitable biometric fusion scheme.

each remaining object $\vec{x}_k$ and for each existing class C_j, LAMDA computes for every descriptor the so-called *marginal adequacy degree* $MAD_{ij}(x_i^k)$ between the values that the ith descriptor takes over $\vec{x}_k$ and the class C_j. Thus, a vector $MAD_j(\vec{x}_k)$ can be associated with object $\vec{x}_k$ for each class C_j. $MAD_j(\vec{x}_k)$ is a membership function derived from a fuzzy generalization of a binomial probability law, as expressed in the algorithm. In the expression, $\nu(x_i^k, c_{ij})$ is a distance function between the descriptor x_i^k and the attribute c_{ij} of the center of the class C_j; ρ_{ij} is the possibility of the descriptor x_i^k to belong to class C_j.

```
Algorithm 1: Data clustering with LAMDA
Input: A set of data objects $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$, where $\vec{x}_k = \{x_1^k, x_2^k, \ldots, x_m^k\}$
Output: $\Gamma = \{C_j\}$ set of classes where each class $C_j$ is represented
 by the parameters $c_{ij}$ and $\rho_{ij}$
Initialization: $\nu(x_i^k, c_{ij}) = 1 - \|x_i^k - c_{ij}\|^2$ and $\rho_{init} = 0.5, \alpha \in ]0, 1[, T_{norm}, T_{conorm}$ and $C = 1$
begin
    for $k \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $C$ do
            for $i \leftarrow 1$ to $m$ do
                $MAD_{ij}(x_i^k) = \rho_{ij}^{\nu(x_i^k, c_{ij})}.(1 - \rho_{ij})^{1-\nu(x_i^k, c_{ij})}$
            end
            $MAD_j(\vec{x}_k) = \{MAD_{ij}(x_i^k) \mid 1 \leq i \leq m\}$
            $GAD_j(\vec{x}_k) = L_\alpha(MAD_j(\vec{x}_k)) = \alpha \times$
            $T_{norm}(MAD_j(\vec{x}_k)) + (1 - \alpha) \times T_{conorm}(MAD_j(\vec{x}_k))$
        end
        $j \leftarrow \arg\max_{1 \leq l \leq C}(GAD_l(\vec{x}_k))$
        if $(GAD_j(\vec{x}_k) > 0.5)$ then
            //1 − Affect object $\vec{x}_k$ to class $j$
            $\vec{x}_k \mapsto C_j$
            //2 − Update parameters $\rho$ and $c$ for class
              //$C_j$
            for $i \leftarrow 1$ to $m$ do
                $\sum_{i=0}^{m}(\delta/\delta c_{ij})\nu(x_i^k, \hat{c}_{ij}) = 0$
                $\hat{\rho}_{ij} = 1/n \sum_{i=0}^{n} \nu(x_i^k, \hat{c}_{ij})$
            end
        else
            //Create a new class
            $\Gamma \leftarrow \Gamma \cup \{C_j\}$
            $C \leftarrow C + 1$
            //Initialize the new class parameters
            $\rho_{ij} = \rho_{init}$
            $c_{ij} = x_i^k$
        end
    end
    return $\Gamma$
end
```

Notice that if $\rho_{ij} = 0.5$, for every pair of values (x_i^k, c_{ij}), and for every distance function ν, then $MAD_{ij}(x_i^k)$ is equal to 0.5. Thus, NIC is characterized by MAD vector with $\rho = 0.5$ for every j. In the next step, the *global adequacy degree* $GAD_j(\vec{x}_k)$ for each class C_j is obtained by summarizing all the $MAD_j(\vec{x})$ through an aggregation operator $\boldsymbol{L}$. In fuzzy logic, several operators can be used to aggregate different membership functions (into another membership function). These operators, known as T_{norm} and T_{conorm} correspond to the intersection and union, respectively. Table I gives some examples of operators used in fuzzy logic frameworks. As shown in the algorithm, $GAD_j(\vec{x}_k)$ is computed as a linear combination of T_{norm} and T_{conorm}. In the given expression, α is a real value between 0 and 1.

When computing $GAD_j(\vec{x}_k)$, if the value is smaller or equal to 0.5, the object is considered as part of NIC and automatically assigned as the first element of the new class as a result. Otherwise, after computing the GAD values corresponding to all the classes, the object will be assigned to the class with the greatest GAD value. Using sample data, the ρ_{ij} and c_{ij} values for each class are estimated by minimizing a maximum likelihood criterion as expressed in the algorithm.

The worst case complexity of Algorithm 1 is $n^2 m$, where n and m are the number of objects and their dimensionality, respectively.

V. Biometric Analysis Using LAMDA

In this section, we describe in detail the main aspects of mouse biometric analysis consisting of behavior extraction during enrollment

TABLE I
T-NORM AND T-CONORM OPERATORS. THESE OPERATORS ARE USED TO AGGREGATE DIFFERENT FUZZY MEMBERSHIP FUNCTIONS INTO ANOTHER MEMBERSHIP FUNCTION

Name	Minimum	Product	Lukasiewicz	Yager	Completely reinforced
Operator	$min(x)$	$-\ln(x)$	$1-x$	$(1-x)^{\lambda}, \lambda > 0$	$\frac{1}{1+\prod_{i=1}^{n}\left[\frac{1-x_i^{\alpha}}{x_i^{\alpha}}\right]^{1/n\alpha}}$
T_{norm}	$\min_i\{x_i\}$	$\prod_{i=1}^{n} x_i$	$\max\left\{1-n+\sum_{i=1}^{n} x_i, 0\right\}$	$1-\min\left\{\left(\sum_{i=1}^{n}(1-x_i)^{1/\lambda}\right)^{\lambda}, 1\right\}$	
T_{conorm}	$\max_i\{x_i\}$	$1-\prod_{i=1}^{n}(1-x_i)$	$\min\left\{\sum_{i=1}^{n} x_i, 1\right\}$	$\min\left\{\left(\sum_{i=1}^{n}(x_i)^{1/\lambda}\right)^{\lambda}, 1\right\}$	

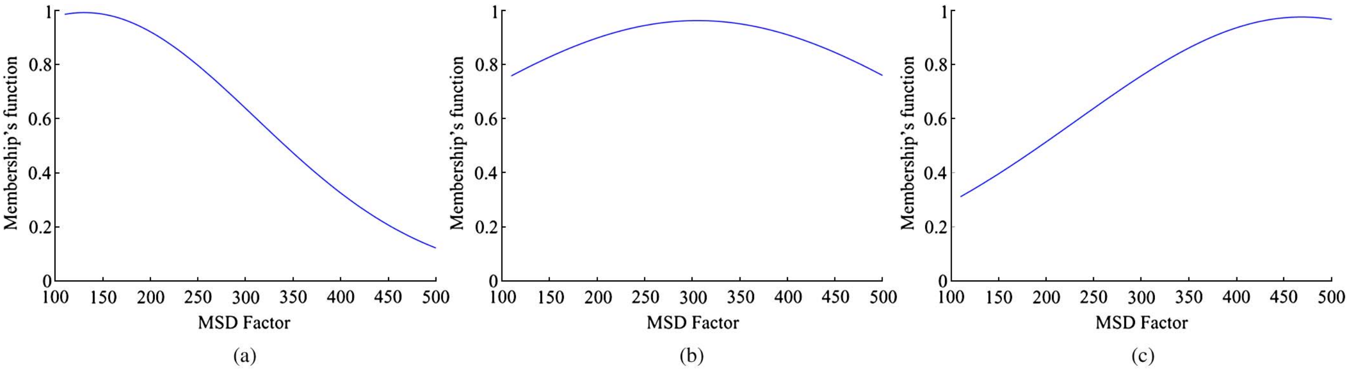


Fig. 3. Membership's functions for the MSD factor for $User1$. Classification was done using parameters $\alpha = 0.9$, the operator *completely reinforced*. (a) Class c_{11}. (b) Class c_{12}. (c) Class c_{13}.

and behavior comparison during the monitoring (or system usage) phase.

A. Behavior Extraction

Mouse dynamics biometric samples are organized into sessions of specific length, involving seven factors, each represented by a separate feature vector as mentioned earlier. Using our proposed biometric analysis approach, each of the factors is processed separately through an associated classifier. Although different classifiers are developed for different factors, the same classification technique (i.e., LAMDA) is used for all of them. For each legal user, the classifiers are derived during the enrollment phase. For each factor, two models are derived, each represented by its center and a separate fuzzy membership function. The first model, which matches genuine or "self" behavior, is derived through positive training using exclusive sample data from the user. The second model, which characterizes imposter or "nonself" behavior, is derived through negative training using sample data from other legal users. The constructed pair of models is stored and used as a reference profile for the user.

To derive the self model, we use four (real) sessions from the user; to obtain the nonself model, we use four (real) sessions from four different users (one session per user). We use LAMDA classification and follow the same process in both cases. Given the initial sessions, we expand the training data set by generating synthetically new samples, each obtained by computing the arithmetic average of each pair of real samples. Therefore, given the four samples used to derive self or nonself models, the training data set will be expanded to 13 sessions overall. The expanded data set is then processed using LAMDA. LAMDA is an unsupervised classification technique, so we do not need to specify the number of classes at the beginning. The algorithm determines a number of classes according to the information hidden in the data.

All the obtained classes will be grouped in one state called a reference state. To build a reference state for a user, we start by allocating to this state only the class with the largest number of data objects, and we start a new classification with the data remaining in the other classes. We repeat this process until all the data are allocated to a reference state. This methodology allows the construction of a user model in the form of a fuzzy decision tree (FDT). The obtained FDT is a rooted tree T involving L levels $(L \geq 1)$, each consisting of exactly two nodes, except the root level. Let n_i^j denote the jth node from level i in T, such that $0 \leq i \leq L-1$; $j = 1$ if $i = 0$, and $j \in \{1, 2\}$ otherwise. Each node will be associated with both a set of data and the set of classes extracted from such data using LAMDA. Let $data(n_i^j)$ and $\Gamma_{n_i^j}$ denote the set of data and the set of classes associated with node n_i^j, respectively. We will also occasionally use the notation $data(C_k)$ to denote the set of data involved in class C_k. Algorithm 2 summarizes the different steps leading to the construction of the FDT; the worst case complexity of the algorithm is n^3m, where n and m are the number of objects and their dimensionality, respectively. Fig. 3 illustrates the membership's functions for the MSD factor extracted from user's sessions; let us refer to him as $User1$. The initial classification realized on MSD for this user gives three classes. Each membership's function is described by a specific curve that characterizes how the data are dispersed in the class. Fig. 4 shows the self-decision tree derived for the user for the MSD factor using Algorithm 2.

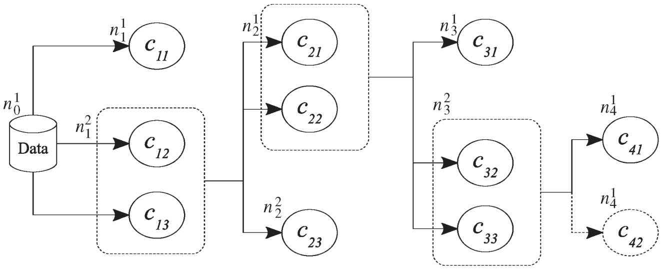


Fig. 4. Self FDT model for the MSD factor for $User1$. The FDT is represented as a rooted tree involving $L = 4$ levels, each consisting of two nodes, except the root level.

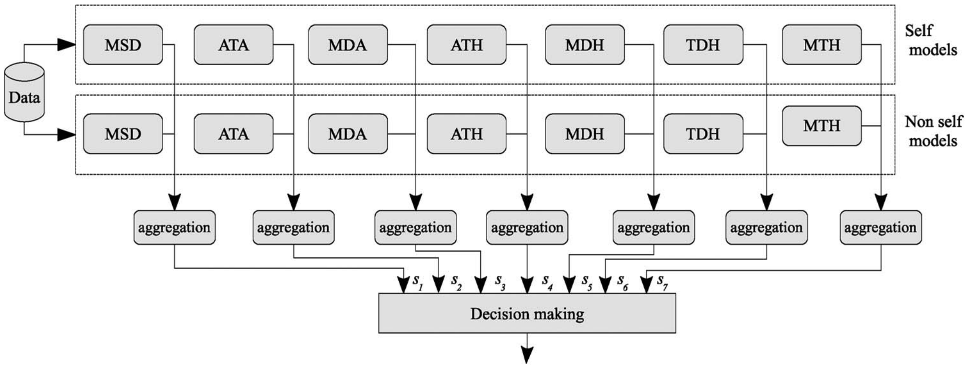


Fig. 5. Decision Fusion Architecture. The membership values for *self* and *nonself* are aggregated using T-norm and T-conorm operators. The outcome is compared to a threshold yielding a bit vector $[s_1, \ldots, s_7]$. Final decision is made by comparing the sum of the elements of the bit vector to a predefined threshold.

```
Algorithm 2: Construction of the fuzzy decision tree model using LAMDA
Input: A set of objects $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$
Output: A rooted-tree $T$
Initialization:
Let $X'$ denote a set of data objects: $X' \leftarrow X$
Let $T$ be a rooted tree with a single node $\{n_0^1\}$: $data(n_0^1) \leftarrow X$
$i \leftarrow 0, j \leftarrow 1$
begin
    while $(X' \neq \emptyset)$ do
        $\Gamma_{n_i^j} \leftarrow LAMDA(X')$
        $C_k \leftarrow \arg\max_{C \in \Gamma_{n_i^j}} |C|$
        Construct new node $n_{i+1}^{j+1}$ and edge $n_i^j n_{i+1}^{j+1}$ in $T$ such that
        $data(n_{i+1}^{j+1}) = data(C_k)$
        //Remove the data associated with $C_k$ from X'
        $X' \leftarrow X' \backslash data(C_k)$
        Construct new node $n_{i+1}^j$ and edge $n_i^j n_{i+1}^j$ in $T$ such that
        $data(n_{i+1}^j) = data(X')$
        $i \leftarrow i + 1$
    end
    return $T$
end
```

B. Behavior Comparison

In the monitoring phase, behavior comparison takes place by matching received (or monitored) session data against the models corresponding to the different factors, as shown in Fig. 5. Algorithm 3 illustrates the steps for computing the membership value for each of the factors using corresponding FDTs. The membership value captures the similarity of the monitored session to the reference models. The computation yields a $K \times 2$ matrix $V = [V_{ij}]_{\substack{1 \leq i \leq K \\ 1 \leq j \leq 2}}$, where the rows represent the K factors $(K = 7)$ and the columns represent the values of the membership's functions for self $(j = 1)$ and nonself $(j = 2)$ behaviors, respectively. The worst case complexity of Algorithm 3 is $n^2 m$.

```
Algorithm 3: Membership value calculation based on FDT model for a given factor
Input: A data object $\vec{x}_k = \{x_1^k, x_2^k, \ldots, x_m^k\}$ ; A Fuzzy Decision Tree T with depth $L (L \geq 1)$
Output: $V$ the Membership value of $\vec{x}_k$
Initialization: $V \leftarrow 1, i \leftarrow 0$
begin
    while $(i < L - 1)$ do
        $C_l \leftarrow \arg\max_{C \in \Gamma_{n_i^j}} Membership(\vec{x}_k, C)$
    end
    $V_l \leftarrow Membership(\vec{x}_k, C_l)$
    $V \leftarrow V \times V_l$
    Let $r$ be such that $r \in \{1, 2\}$ and $data(C) \subset data(n_{i+1}^r)$
    if $n_{i+1}^r$ $is\ a\ terminal\ node$ then
        return $V$
    else
        $i \leftarrow i + 1$
    end
    return$V$
end
```

We aggregate the membership values by computing a decision score, which we refer to as the CR. CR is a vector $CR = [CR_i]_{1 \leq i \leq K}$ obtained through the linear combination of the columns of matrix V via an aggregation operator $L : CR_i = \alpha T_{norm}(V_{i1}, V_{i2}) + (1 - \alpha) T_{conorm}(V_{i1}, V_{i2})$, where α is a real value between 0 and 1.

At this stage, we conduct initial decision making by comparing the CR_i against a predefined threshold CR_{th}. This will yield a bit vector $S = [s_i]_{1 \leq i \leq K}$, such that

$$s_i = \begin{cases} 1 & \text{if } CR_i \geq CR_{th} \\ 0 & \text{otherwise.} \end{cases}$$

Final decision is made via a score-level fusion scheme. Various score-level fusion schemes have been proposed in the literature, including the *sum*, *product*, *minimum*, *maximum*, and *median* rules. According to Kittler *et al.*, "under the most restrictive assumptions, the sum rule outperforms other classifier combinations" [15]. The sum rule has become the most commonly used score-level fusion scheme due to its high performance and simplicity. We use the sum rule to combine the different scores s_i obtained from the individual classifiers. To make the final decision, we compute the sum of the elements of the bit vector S and compare the obtained value with n which is here an integer threshold between 1 and K, as follows:

$$\begin{cases} \text{accept} & \text{if } n \leq \sum_{i=1}^{i=K} s_i \\ \text{reject} & \text{otherwise.} \end{cases}$$

Note that this is equivalent to an *n-out-of-K* fusion scheme, where an overall matching decision requires matching decisions from n factors out of K. Our framework, therefore, relies on at least two different thresholds that can be adjusted to determine suitable operating conditions.

The sum rule assumes statistical independence of the K scores. Such an assumption is fulfilled only partially in our approach. Although the majority of the features are linked, the scores s_i are

TABLE II
MSD FEATURES VALUES AND CORRESPONDING SCORES EXTRACTED FROM SESSIONS FOR THREE USERS

	MSD Features												Membership's value		CR	s
													Self	Nonself		
user 1	44.92	102.42	161.17	219.23	274.78	326.28	372.67	413.36	448.25	477.58	501.82	521.60	1.00	0.00	0.6	1
	41.81	112.65	169.59	221.60	273.14	323.67	371.01	413.10	448.72	477.61	500.25	517.50	1.00	0.00	0.6	1
user 2	50.13	96.91	146.24	196.35	245.37	291.56	333.60	370.67	402.43	429.07	450.79	468.36	0.29	0.71	0.54	0
	51.72	104.39	156.21	204.97	248.98	287.22	319.37	345.66	366.67	462.84	475.60	485.35	0.32	0.68	0.53	0
user 3	35.81	119.94	196.32	263.42	320.69	368.38	407.27	438.47	463.15	482.46	497.46	509.02	0.49	0.51	0.50	0
	44.83	115.94	192.79	267.14	331.73	382.80	420.25	446.24	463.58	474.85	482.05	545.01	0.10	0.9	0.58	0

The sample data consists of 2 sessions per user, each represented row-wise. The MSD feature vector consists of 12 numbers. All the membership values are computed with respect to the self and nonself (FDT) models of $User1$. CR values are computed based on $\alpha = 0.4$ and the *minimum* operator. The fusion result s is based on threshold $CR_{th} = 0.6$.

produced by separate classifiers. Like with many biometric systems, statistical independence may not fully be achieved. Such an assumption, however, represents a pragmatic approximation of the reality which yields acceptable results.

Table II presents sample values for the MSD factor and corresponding scores for three different users: $User1$, $User2$, and $User3$. Membership values are computed for each of the three users with respect to the reference model of $User1$ (see Figs. 3 and 4). As we notice, for $User1$, the self values are high, while the nonself ones are low; the opposite situation applies for $User2$ and $User3$. The CR column is the result of the aggregation of the membership's values via the *minimum* operator and a value of $alpha = 0.4$; the s values are computed using threshold $CR_{\mathrm{th}} = 0.6$.

VI. EXPERIMENTAL EVALUATION

In this section, we describe the data sets used to evaluate our approach and present corresponding results.

A. Data sets

We use the data sets developed in [1], [16] to evaluate our framework. A free experiment involving 22 users was organized in [1] to evaluate the proposed mouse dynamics analysis framework. Users were invited to use freely any kind of applications in various computing environments because this reflects operating conditions in real computing environments. The collected data consisted of 284 hours of raw mouse data over 998 sessions, with an average of 45 sessions per participant. The collected data set was expanded later in [16] by repeating the same free experiment with 26 additional users; 2184 sessions, with an average of 84 sessions per user, were collected.

Note that, in [1], a controlled experiment involving seven users was also conducted to assess the impact of fixing the operating conditions for all the users on a mouse biometric recognition system. All the users were asked to use a customized application running on the same PC and perform specific actions between two rectangles displayed on the screen, such as dragging one rectangle to another or simply clicking on one of the rectangles. This yielded a $FAR = 2.245\%$ and $FRR = 0.898\%$, suggesting that even by fixing the actions and operating environments, mouse dynamics can be used to discriminate effectively between different users. Based on this finding, we focus our evaluation on the data sets collected from the free experiments which actually reflects real operating conditions under which mouse dynamics biometrics might be collected and analyzed for computer user recognition.

B. Results

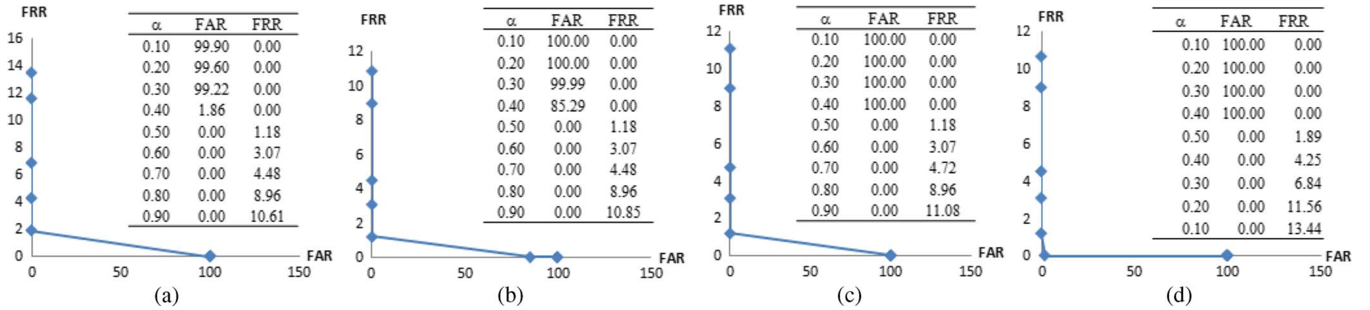
To validate our approach, we conducted one-hold cross validation tests, as described in [1], using the aforementioned data sets. For each data set of size N, the test was repeated N times, by considering in each round one of the users as an impostor to be tested against the remaining users considered as legal users. In each case, the construction of the reference models for the legal users was based solely on sample sessions from legal users; sessions from the impostor were excluded from this process. We computed FRR and FAR by varying the control parameters described earlier, namely α, the aggregation operator L, and the thresholds CR_{th} and n. The evaluation was conducted by developing and running a C program on an Intel(R) Core(TM) Duo with 2.00 GHz for each processor and 3 GB of RAM. For each user, the generation of the reference models took between 0.4 and 0.5 of a second. For each session, we needed 7 s to compute FRR and 7.8 s for FAR. These times are relatively short and present an advantage in the monitoring step.

First, we conducted our evaluation by considering just the first data set with 22 users to study the impact of the control parameters on the final result. We varied the different parameters one-by-one and studied their impact on the performance. To illustrate this process, we will focus the rest of this correspondence paper on the impact of the aggregation operators by using $n = 5$ and $CR_{\mathrm{th}} = 0.6$. For each operator, we varied α from 0.1 to 0.9. Fig. 6 shows the receiver operating characteristic (ROC) curves for four operators (minimum, product, Lukasiewicz, and Yager), illustrating the relation between FRR and FAR. The optimal operating point depends on the relative cost of a false acceptance versus a false rejection; a tradeoff must be made between security and user acceptability. The EER, which corresponds to the crossover point at which false acceptance and false rejection errors are equally likely, was obtained for a value of α ranging between 0.4 and 0.5, for each of the operators considered. Outside this range, all the operators exhibited high sensitivity, especially for FAR values which were characterized by huge variations. We noticed, by analyzing the results, that the operator *minimum* is the most promising. This operator provided a value between 0% and 1.18% for FRR and between 1.86% and 0% for FAR which prefigured that a good compromise could be found between FAR and FRR.

In the next step of our evaluation, we combined all the data sets obtained from the free experiments comprising the initial data set of 22 users and its extension of 26 users. The combined data set of 48 users was tested (through one-hold cross validation) using the *minimum* operator and by varying α in the range [0.40, 0.50]. Table III shows the results obtained. It is noticeable that $\alpha = 0.40$ and $\alpha = 0.41$ yield FRR and FAR values that meet the European standard for access control.

C. Discussions

Table IV compare our results with existing error rates published in the mouse literature. Approaches based on basic statistics perform poorly, while the ones based on machine learning achieve relatively



(a)

α	FAR	FRR
0.10	99.90	0.00
0.20	99.60	0.00
0.30	99.22	0.00
0.40	1.86	0.00
0.50	0.00	1.18
0.60	0.00	3.07
0.70	0.00	4.48
0.80	0.00	8.96
0.90	0.00	10.61

(b)

α	FAR	FRR
0.10	100.00	0.00
0.20	100.00	0.00
0.30	99.99	0.00
0.40	85.29	0.00
0.50	0.00	1.18
0.60	0.00	3.07
0.70	0.00	4.48
0.80	0.00	8.96
0.90	0.00	10.85

(c)

α	FAR	FRR
0.10	100.00	0.00
0.20	100.00	0.00
0.30	100.00	0.00
0.40	100.00	0.00
0.50	0.00	1.18
0.60	0.00	3.07
0.70	0.00	4.72
0.80	0.00	8.96
0.90	0.00	11.08

(d)

α	FAR	FRR
0.10	100.00	0.00
0.20	100.00	0.00
0.30	100.00	0.00
0.40	100.00	0.00
0.50	0.00	1.89
0.40	0.00	4.25
0.30	0.00	6.84
0.20	0.00	11.56
0.10	0.00	13.44

Fig. 6. ROC curves and sample FRR and FAR values, for the initial data set (22 users), obtained by varying α from 0.1 to 0.9 for different operators. By yielding a value between 0% and 1.18% for FRR and between 1.86% and 0% for FAR, the operator *minimum* appears to be the most promising. (a) Minimum. (b) Product. (c) Lukasiewicz. (d) Yager.

TABLE III
FAR AND FRR OBTAINED FOR COMBINED DATASET (48 USERS), WITH THE OPERATOR MINIMUM AND BY VARYING α BETWEEN 0.40 AND 0.49. IT CAN BE NOTED THAT THE PERFORMANCE RESULTS AT $\alpha = 0.40$ AND $\alpha = 0.41$ MEET THE EUROPEAN STANDARD FOR ACCESS CONTROL

α	0.40	0.41	0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49	0.45
FAR (%)	0.36	0.09	1.26	1.44	2.07	2.47	2.55	2.88	3.06	3.42	3.51
FRR (%)	0	0	0	0	0	0	0	0	0	0	0

TABLE IV
PERFORMANCE RESULTS FOR EXISTING MOUSE DYNAMICS BIOMETRIC ANALYSIS APPROACHES. APPROACHES BASED ON BASIC STATISTICS YIELD POOR RESULTS, WHILE MOST OF THE ONES USING SOME FORM OF LEARNING TECHNIQUE PERFORM REASONABLY WELL. OUR APPROACH, HOWEVER, IS THE ONLY ONE THAT MEETS THE EUROPEAN STANDARD FOR ACCESS CONTROL

References	**[1], [16]**	**[8]**	**[20]**	**[11]**	**[10]**	**Current**
Classifier	**Neural Network**	**Sequential Forward Selection**	**Decision Tree (C5.0)**	**Basic Statistics**	**Basic Statistics**	**Fuzzy Clustering**
Number of users	48	50	18	10	15	48
FAR (%)	2.46	2	1.75	37.5	15	0
FRR (%)	2.46	2	0.43	37.5	15	0.36

TABLE V
PERFORMANCE RESULTS FOR POPULAR BIOMETRICS TECHNOLOGIES. AS IT CAN BE NOTED, IRIS IS THE ONLY BIOMETRIC TECHNOLOGY THAT MEETS THE EUROPEAN STANDARD FOR ACCESS CONTROL

Biometrics	Fingerprint	Face	Voice	Iris	Keystroke
References	[14]	[14]	[14]	[18]	[9]
FAR (%)	0.1	0.1	$2-5$	0.0001	0.00489
FRR (%)	0.4	$1-2.5$	$5-10$	0.25	4.8333

low error rates. Overall, our approach based on LAMDA classification outperforms all the existing approaches.

It is also important to mention that we took the study presented in [1] as a point of departure for this correspondence paper, because to our knowledge, it is the most comprehensive study published on mouse dynamics biometric analysis. The approach proposed by the authors covers a more realistic view of the use of mouse dynamics biometrics through a series of controlled and free experiments that reflect the heterogeneity inherent in real computing environments. For instance, unlike in [1], the data collected by Pusara and Brodley was analyzed using user-specific parameters and an application-specific template; this does not reflect real operating conditions in a computing environment.

Using the data provided in [1] and the same evaluation approach, our analysis approach decreases the FAR from 2.46% to 0% and the FRR from 2.46% to 0.36%. Although in absolute, this outcome might seem insufficient, it is significant in the context of biometrics, because it shows that by using a suitable classification technique, mouse dynamics biometrics can meet the European standard for access control. As shown in Table V, to our knowledge, Iris is the only traditional biometrics that meets the European standard.

VII. CONCLUSION

A. Summary

We have presented in this paper a new approach for mouse dynamics biometric analysis based on VR via extractors with separate features and the LAMDA classification technique. The proposed approach allows us to meet the performance constraints set by the European standard for commercial biometric technology. This is a significant achievement, since most existing biometric technologies do not meet these requirements, and previous works on mouse dynamics have shown error rates well beyond the required ranges. In our future work, we will investigate whether the use of alternative classification techniques along with the generic approach used here can also impact positively mouse biometric performance.

B. Limitations and Future Work

The main limitation of our work is the fact that our performance results are based on a fixed session length inherited from the evaluation data set. The session length plays an important role, in particular, in continuous authentication applications, since it represents the window of opportunity for an attacker; the shorter the session length the better. Alternatively, short session length means less data, which could mean reduced performance. We intend to investigate the impact of session length on our system performance in future research.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief and the Associate Editor for giving them the opportunity to publish this correspondence paper and for their advice on improving its quality. They would like to thank and acknowledge the tremendous work done by the anonymous reviewers for their valuable comments and thorough review. The authors would like to thank the relentless efforts of A. Hoole of the University of Victoria for copyediting and helping improve the quality of this paper.

REFERENCES

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

[2] A. A. E. Ahmed and I. Traoré, "System and method for motion-based input device computer user profiling," Patent International Filing No. PCT/CA2004/000669, May 3, 2004.

[3] T. J. Alexander, "Biometrics on smart cards: An approach to keyboard behavioral signature," *Future Gener. Comput. Syst.*, vol. 13, no. 1, pp. 19–26, Jul. 1997.

[4] B. Guo and M. S. Nixon, "Gait feature subset selection by mutual information," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 36–46, Jan. 2009.

[5] M. Brown and S. J. Rogers, "User identification via keystroke characteristics of typed names using neural network," *Int. J. Mach. Stud.*, vol. 39, no. 6, pp. 999–1014, Dec. 1993.

[6] M. De Marsico, M. Nappi, and D. Riccio, "FARO: Face recognition against occlusions and expression variations," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 121–132, Jan. 2010.

[7] T. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Germany: Springer-Verlag, 2000, pp. 1–15.

[8] H. Gamboa and A. Fred, "An identity authentication system based on human computer interaction behaviour," *Proc. SPIE*, vol. 5404, pp. 381–392, 2004.

[9] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 3, pp. 312–347, Aug. 2005.

[10] S. Hashia, C. Pollett, and M. Stamp, "On using mouse movement as a biometric," in *Proc. 3rd ICCSA*, 2005, pp. 143–147.

[11] S. Hocquet, J. Y. Ramel, and H. Cardot, "Users authentication by a study of human computer interactions," in *Proc. 8th Annu. (Doctoral) Meeting Health, Sci. Technol.*, 2004.

[12] A. Humm, J. Hennebert, and R. Ingold, "Combined handwriting and speech modalities for user authentication," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 25–35, Jan. 2009.

[13] R. W. Ives, Y. Du, D. M. Etter, and T. B. Welch, "A multidisciplinary approach to biometrics," *IEEE Trans. Educ.*, vol. 48, no. 3, pp. 462–471, Aug. 2005.

[14] A. K Jain and S. Pankanti, "Beyond fingerprinting," Scientific American, pp. 78–81, Sep. 2008.

[15] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–238, Mar. 1998.

[16] A. Nazar, I. Traoré, and A. A. E. Ahmed, "Inverse biometrics for mouse dynamics," *Int. J. Artif. Intell. Pattern Recognit.*, vol. 22, no. 3, pp. 461–495, May 2008.

[17] N. Piera and J. Aguilar, "Controlling selectivity in nonstandard pattern recognition algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 71–82, Jan./Feb. 1991.

[18] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proc. IEEE*, vol. 91, no. 12, pp. 2021–2040, Dec. 2003.

[19] N. Poh and S. Bengio, "Non-linear variance reduction techniques in biometric authentication," in *Proc. Workshop MMUA*, 2003, pp. 123–130.

[20] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proc. ACM Workshop Vis. Data Mining Comput. Secur.*, May 2004, pp. 1–8.

[21] A. Ross, A. Jain, and J.-Z. Qian, "Information fusion in biometrics," in *Proc. 3rd Int. Conf. AVBPA*, 2001, pp. 354–359.