Proposing a Biometric Verification Method for Students Attendance using Mouse Movements

MIHAILESCU Marius Iulian[1], PAU Valentin Corneliu[2]

[1] University of Titu Maiorescu, Romania,
Department of Computer Science, Faculty of Computer Science,
Calea Vacaresti nr. 187, 0400511 Bucharest, Romania, E-Mail2: mihmariusiulian@gmail.com

[2] University of Titu Maiorescu", Romania,
Departament of Computer Science, Faculty of Computer Science,
Calea Vacaresti nr. 187, 0400511 Bucharest, Romania, E-Mail1: v_pau@utm.ro

DOI: 10.6007/IJARBSS/v3-i11/368 URL: http://dx.doi.org/10.6007/IJARBSS/v3-i11/368

Abstract – The biometric authentication checks a user based on its unique characteristics – who you are. Behavioral biometrics has proven very useful in authentication a user. The mouse dynamics has a unique pattern for movements. The paper present the benefits of the user verification system based on mouse dynamics, a method very accurate and efficient enough for future usage. The idea proposed in this article represents a method which has been developed for POSDRU 61434: Modern Education and Quality for Future, a project founded by European Union.

Keyword: User Verification, Angle-Based Metrics, User Verification, Biometric.

I.INTRODUCTION

E-learning systems represent the future of learning and the systems are gaining more and more popularity every day and insecure also; therefore the security aspects in e-learning system are very sensible. The lack of the appropriate tools to properly check the users' behavior stands as one of the most important problems in learning management systems (LMS).

The main goal of this paper is to present a new e-learning model as a web application with the possibility to check the presence at any time. The multimodal biometrics included in the model are used for identification, authentication and checking the users. The model is based on two types of biometrics: behavioral biometric characteristics (keystroke and mouse movement's dynamics) and physical biometric characteristics (face recognition). Also we include another option for face recognition, such as on-line and off-line signature which can be replaced in any time in the application or it can be used in combination with the face recognition process. The model it can be used to check in real-time the presence of the students in different sensitive levels of the e-learning process.

The huge impact of the Information and Communication Technology (ICT) has influenced all the people around the planet. Biometrics are divided in two general categories: physiological (face, fingerprint, hand, iris and DNA) and behavioral (keystroke, mouse movement, signature, voice). The proposed model combines three biometric characteristics: face recognition, signature, mouse movement and keystroke. The article will focus only on mouse movements characteristic. One of the goal of this article is to present the advantages and disadvantages of this method. Keystroke and mouse movement describes an individual's behavior with a computer-based pointing device, such as a mouse or a touch-pad.

II. THE MOVEMENT MEASUREMENT AND CHARACTERIZATION OF MOUSE MOVEMENT

Two sets of data are collected, the first from an environment which is controllable, named the controllable set and the second from an online forum in the field, named the field set. The first case involves 30 users of different ages, educational backgrounds, and occupations, which participate personally in the data collection. In order for them to behave as naturally as possible, we set intentionally a normal environment, in which we record mouse movement data during their routine computing activities, like surfing the Internet, programming, word processing, playing games or online chatting. The tool we use to monitor their mouse moving activities is a logging tool RUI [16].

For the second set, the field set, with the help of JavaScript code to record the mouse movements of more than 1,000 users, passively submitted to the web server via AJAX requests. These users are anonymous but can be identified using unique login names. In any case, we cannot predict the amount of data which will be collected for a specific user, because a forum user could be logged in for a long time and have frequent mouse activities or perform just one click before they leave. However, we use this type of users to serve as base profile for both training and testing purposes.

We represent the raw movements of the mouse as tuples of timestamp and Cartesian coordinate pairs. Each tuple has the form (action-type*, t, x, y)*, where action-type equals the mouse action type (a *mouse-move* or *mouse-click*), t represents the timestamp of the mouse action, x is the x-coordinate, and y will be the y-coordinate. We collect timestamps in milliseconds.

A. Processing of Data

We preprocess in order to identify every pointand-click action, defined as a click after continuous movements of the mouse, which represent movements that have no pause (or a very small one) between each adjacent step. The *j*th mouse move record as *(*mouse-move*,* t_i, x_i, $y_{i)c,j}$, where t_i is the timestamp of the *i*th mouse movement, is denoted within the *i*th point-and-click action for a user *c*. Angle-based metrics are calculated based on the record belonging to each point-and-click action.

B. Metrics

Three fine-grained angle-based metrics are defined in order for us to analyze the movement data of the mouse: direction, curvature angle, curvature distance. Speed and acceleration differ in the case of newly-defined metrics (unlike the conventional metrics), and can accurately define the unique behavior of a user's mouse in movement, whatever its running platform is.

- Direction. We take any two consecutive recorded points A and B, we record the traveled distance from *A* to *B*, defined as the angle between that line $\overrightarrow{AB}$ and the horizontal.
- Curvature Angle. The angle of curvature is *∠ABC,* if any three consecutive recorded points A, B, and C are considered; angle between line from A to B ($\overrightarrow{AB}$) and line from B to C ($\overrightarrow{BC}$).
- Curvature Distance. For any three recorded points A, B, and C, take into account the length of the line connecting A to C ($\overrightarrow{AC}$). The curvature distance is the length's ratio of $\overrightarrow{AC}$ to the distance from point B to the line $\overrightarrow{AC}$. Observe that this metric is the ratio of two distances. For clarification, we give the definition of two metrics of traditional mouse movement, speed and pause-and-click.
- Speed. The speed is calculated for each point-and-click action, as we divide the ratio of the total traveled distance for that action by the total time which took for the action to complete.
- Pause-and-Click. The time between the end of the movement and the click event (in other words between pointing to an object and actually clicking on it) is measured for each point-and-click action.

C. The Characterization of Mouse Movement

When we analyzed our data we discovered that it may be non-constructive to make a comparison between two users using each of them very different machines, because the data can be affected by the entire environment of the user: the OS, the size and resolution of the screen, the size of the font, the sensitivity of the mouse pointer, the brand of mouse, and even the available amount of space of the mouse pad on the desk. It's not a good idea to use metrics like speed and acceleration, to perform a comparison between the users of arbitrary platforms, because they can be influenced by differences in screen resolution and pointer sensitivity. Not only that, but what a user is reading can influence metrics like pause-and-click. Let's take the case of a Wikipedia article. A user will take longer before clicking, unlike, for example, when clicks on a "Submit" button, as the first case captivates more than the last, the outcome of which is expected.

Using angle-based metrics makes a good case for arbitrary user comparison. The relatively platform are independent given the fact that direction and angle of curvature don't depend on the screen size or any other user's environmental element. In the same manner, the distance of the curvature represents a ratio of distances on the screen, and thus being self-adjusting for the user's specific environment. We can compare across platforms a ratio to another user's.

The Unique Character of Angle-Based Metrics Across Users

We can consider the uniqueness of the distinctive feature of angle-based metrics, given that same user has resembling angle-based results on different platforms and different angle-based results appear to separate users which have similar platform.

Given that the CDF curves of different users are closely coupled in speed and pause-and click on the same environment, there is a distinct gap between the same user's two curves for different environments. Although, it can be quite difficult to uniquely differentiate people with the help of these metrics, considering that the nearest matching curves for each user, under the same environment, is the curve of the *other* one, but this makes angle-based metrics superior for user verification in what concerns speed and pause and- click, along with the above discussed platform independence. Bear in mind that we only compare the mouse dynamics difference between a pair of users, although similar observation applies to other users.

The Distance between Distributions

We can verify if the angle-based features of a user remain relatively stable across different types of mice, platforms and time by comparing the *distances* between two probability distributions with those of other controllable users.

As a definition we can consider angle-based features as continuous variables. Their range is divided into discrete intervals, named *bins*, and probability density functions (PDFs) are calculated in regards to each bin. We take two distributions: one as PDF $\{p_1, p_2, ..., p_n\}$, where p_i is the probability of falling into the *i*th bin; another as PDF $\{q_1, q_2, ..., q_n\}$. The deviation accumulated from each other over all bins represents the *distance* between the two distributions:

$$D(p,q) = \sum_i |p_i - q_i|$$

Note that distance here depends on the size of each interval, so more the interval is divided, more subtle are the differences reflected by the distance. Nevertheless, it is advised to avoid a very small bin so not to enlarge noise.

Using different mouse on a different machine at different time we have the possibility to measure the distances of a user from the others and from itself. We compute the PDF of every user over 1,000 randomly selected curvature angles from its data, and loops for 10 times. We represent the height of each bar as the average distance from the target user (called user 1) in setting A, and standard deviation over the 10 times is represented by each error bar. Data 1-A, 1-B, and 1-C are all from user 1. We list in Table 1 the details of these three settings. In addition to this, we record data 1-B two and half months later than data 1-A, and data 1-C two days later than data 1-B [13-16].

We can conclude that the distances from user 1 to itself at different machines, using different mice, and over different times are the two smallest in the figure and they are smaller than the distances from user 1 to any other users which have the same setting. This draws the conclusion that the behavior of a user based on angle has a relatively stable inherent pattern across different settings and times, which makes it different from the behaviors of other users and capable of being distinguished. Although the distance values, between user 1-A and other users are very close, this does not mean they have similar behaviors, as here the distance represents a cumulus of deviations at different bins.

In order to achieve accurate results during measurement, we must keep in mind the following: we should configure to the same level the polling rates of mouse recorders at different platforms and we must also collect sufficient mouse events before characterizing the movement of a user's mouse, in order to create a profile of its movement. We observed that an amount of 1,000 mouse actions, usually collectable in 2 hours, constitutes enough data to elaborate a pretty accurate profile of the mouse behavior of an user.

Number of Mouse Clicks in a Real Session

If we choose during one hour 1,074 real users on an online forum for the field set, the results will be an average of 15.14 clicks per session of user, but we have to keep in mind that the value is smaller than the real number of clicks in an average user session, because of the fact that the data was gathered over a window of one hour, so users that were logged in before the beginning of the window or remained logged in after its ending would actually have more clicks than measured, which leads to the conclusion that actual average is much higher [4].
Considering that the average number of mouse clicks per session is round 15, we can conclude that a verification system that monitors mouse dynamics is able to identify a user with high accuracy in less than 15 clicks.

III.THE ARCHITECTURE OF THE SYSTEM

The four elements of the verification system we proposed are the recorder, preprocessor, classifier, and decision maker. The basic task of the recorder is clear. Meanwhile, based on the recorded raw data, the preprocessor computes the angle-based metrics, which leaves the other two components, the classifier and the decision maker, to be discussed in this section.

A. SVM Classifier

Based on the mouse movement dynamics of the users, Support Vector Machines (SVMs) are used here as classifier with the purpose to differentiate them. These types of programs were used with success to solve real-life classification problems, like handwritten digit recognition [10], object recognition [11], text classification [12], and image retrieval [9] using a simpler and thus faster scheme than neural networks.

Mainly, SVMs map the feature vectors to a high dimensional space and compute a hyper plane, separating the training vectors from different classes and maximizing this separation by making the margin as large as possible. Using a set of support vectors allows SVMs to classify data and, by outlining a hyper plane in feature space, to determine the members of the set of training inputs [3].

If we consider a binary classification problem, where l training samples $\{x_i, y_i\}$, $i = 1, . . . , l$, we observe that each sample has certain d features, written as a d-dimensional vector x_i ($x_i \in \mathbb{R}^d$), and a class label y_i with one of two values ($y_i \in \{-1, 1\}$). We can express a hyper plane in *d*-dimensional space as w・ x + *b* = 0, where w represents a constant vector in *d* dimensions, and *b* represents a scalar constant. Our purpose is to find a hyper plane which does more than separating the data points, but as well *maximizes* the separation. We named *margin* the distance between the dashed lines, and the *support vectors are* the vectors (points) that constrain the width of the margin.

We know *Quadratic Programming Problem (QP)* as a minimization problem, which has many algorithms in course of being studied. If we analyze the situation better, we see that data points cannot be entirely separated, as we thought. In order to solve that, SVMs use a “kernel trick”. The system pre-processes the data so the problem becomes of large dimensions, where, in the new space, they are linearly separable. The classifier we use is the following equation, considering a mapping z = φ(x), and a *kernel function* $K(x_a, x_b) = \varphi(x_a) \cdot \varphi(x_b)$ is:

$$f(x) = sign\left(\sum_i a_i\, y_i\big(K(x_i, x)\big) + \ b\right)$$

Gaussian Radial Basis Function (RBF) is a very known choice of kernel function: $K(x_a, x_b) = \exp(-\gamma\, ||x_a - x_b||^2)$, where $\gamma > 0$, and is a parameter named tunable. We consider RBF to be a normal first option, because is very general and easy – efficient computationally [6]. In conclusion, we can solve a classification problem using SVMs this way: (1) select a kernel function (2) set the penalty parameter *C* and, if necessary, the kernel parameters, (3) resolve the quadratic programming problem, and (4) create the discriminant function from the support vectors. The user verification problem is considered to be a two-class classification problem, so we learn to build a classifier based on the movements of the user’s mouse.

In order to built the prototype we utilized, in our proof-of-concept implementation, the open source SVM package LIBSVM 3.0 [6]. LIBSVM is an integrated tool to classify the support vectors. So we can find the best parameter *C* and *γ,* we used the default RBF kernel and the cross-validation. We classify here all impostors as +1, and normal data as -1.

B. Decision Making

We use two mechanisms, threshold and majority vote, to improve verification accuracy.

Threshold

Threshold determines the interpretation of the SVMs’ output: a value over the threshold shows an impostor, and one under the threshold represents the certitude of a true user. So, in the process of recognition, it is sometimes more important to minimize the probability of rejecting a true user than to decrease the probability of accepting an impostor. A *decision value* is outputted for each testing sample, in a binary classification problem with labels in {+1, −1}, LIBSVM. We classify the sample as +1 if the decision value is bigger than 0, and −1 if the value is smaller than 0.

Majority Votes

Randomly we can pick *m/*2 samples of the user in order to build the profile for an authorized user, we label them as negative (non-impostor), and another random *m/*2 from the field set, which we label as positive (impostor). We further employ a simple *majority vote* decision making scheme, which has the purpose of improving and stabilizing the accuracy of the verification. More specific, we train the user's profile 2*n*+1 times before verifying if a sample belongs to the target user. Keeping in mind that the training samples are randomly selected, they are different every time, so there will be 2*n* + 1 votes about the predicted label for each testing sample. The final predicted label will be the one voted by the majority, i.e., with greater than *n* votes. The direct positive consequence is that the decision maker can significantly reduce the randomness of the results and improve accuracy of the verification with majority votes.

IV. CONCLUSIONS

As we have seen, we focus on fine-grained angle based metrics, which is characterized on two advantages over previously studied metrics. *First*, the angle-based metrics have the possibility to make the difference between users accurately with very few mouse clicks. *Second*, the angle based metrics are quite independent by the operating environment on which the user uses, in this making them suitable for online re-authentication.

REFRENCES

[1] A. A. E. Ahmed and I. Traore. Anomaly intrusion detection based on biometrics. In *Proceedings of the 2005 IEEE Workshop on Information Assurance*, 2005.

[2] J. L. Alba Castro, E. Gonzalez Agulla, E. Argones Rua,and L. Anido Rifon. " Realistic measurement of student attendance in LMS using biometrics". In To appear on the Proceedings of the International Symposium on Engineering Education and
Educational Technologies: EEET 2009, 2009.

[3] L. Hong, A. K. Jain, and S. Pankanti, "Can Multibiometrics Improve Performance?", Proc. AutoID '99, pp.59-64, October 2005.

[4] I.Sogukpinar, L. Yalçin, "User identification via keystroke dynamics",Ist. Üniv. Journal of Electrical and Electronic Engineering, vol. 4, no. 1,2004, pp. 995-1005,2007.

[5] D. Gonzalez-Jimenez and J. Alba-Castro. "Shape-Driven Gabor Jets for Face Description and Authentication". IEEE Transactions on Information Forensics and Security,2(4):769–780, 2007.

[6] S. Sanderson and J. Erbetta, "Authentication for secure environments based on iris scanning technology", in IEEE Colloquium on Visual Biometrics, vol. 8, pp.1-7, 2005.

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

[8] Chinese Academy of Sciences – Institute of Automation. Database of 756 Grayscale Face Images. Available: http://www.sinobiometrics.com, Version 1.0,2003.

[8] S. Chiasson, P. C. V. Oorschot, and R. Biddle. Graphical password authentication using cued click-points. In *12th European Symposium On Research In Computer Security (ESORICS), 2007*. Springer-Verlag, 2007.
[9] S. Chiasson, P. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security Symposium*, 2006.
[10] DTREG. SVM - Support Vector Machines. http://www.dtreg.com/svm.htm, Feb 2011.
[11] D. Florencio and C. Herley. A large scale study of web password habits. In *Proceedings of WWW 2007*, 2007.
[12] H. Gamboa and A. Fred. A behavioral biometric system based on human-computer interaction. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5404, pages 381–392, Aug. 2004.
[13] P. Gupta, S. Ravi, A. Raghunathan, and N. K. Jha. Efficient fingerprint-based user authentication for embedded systems. In *Proceedings of the 42nd annual Design Automation Conference (DAC)*, pages 244–247, 2005.
[14] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of European Conference on Machine Learning*, pages 137–142, 1998.
[15] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS ’11, pages 476–482, 2011.
[16] K. Killourhy and R. Maxion. Why did my detector do that?!: predicting keystroke-dynamics error rates. In *Proceedings of RAID’10*, pages 256–276, 2010.