Chapter 7

A Statistical Model for Biometric Verification

Ahmed Awad E. Ahmed, Issa Traoré

Information Security and Object Technology (ISOT) Laboratory,
Department of Electrical and Computer Engineering, University of
Victoria
P.O. Box 3055 STN CSC Victoria, B.C. V8W 3P6 CANADA
aahmed@ece.uvic.ca, itraore@ece.uvic.ca

Traditional biometrics technologies such as fingerprints or iris recognition systems require special hardware devices for biometrics data collection. This makes them unsuitable for online computer user monitoring, which to be effective should be non-intrusive, and carried out passively. Behavioral biometrics based on human computer interaction devices such as mouse and keyboards do not carry such limitation, and as such are good candidates for online computer user monitoring. We present in this chapter artificial intelligence based techniques that can be used to analyze and process keystroke and mouse dynamics to achieve passive user monitoring.

Contents

7.1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
7.2. Biometrics Modes and Metrics . . . . . . . . . . . . . . . . . . . . . . . . . 142
7.3. Mouse Dynamics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
7.3.1. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
7.3.2. Detection Process . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
7.3.3. Silence Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
7.4. Keystroke Dynamics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
7.4.1. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
7.4.2. Free Text Detection Using Approximation Matrix Technique . . . . 151
7.4.3. Free Text Detection based on Keyboard Layout Mapping . . . . . . 154
7.4.4. Free Text detection based on Sorted Time Mapping . . . . . . . . . 156
7.5. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159

Bibliography . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
7.1. Introduction

Biometrics can be defined as a set of distinctive, permanent and universal features recognized from human physiological or behavioral characteristics [17], [18]. As such, biometrics systems are commonly classified into two categories: physiological biometrics and behavioral biometrics. Physiological biometrics, which include finger-scan, iris-scan, retina-scan, hand-scan, and facial-scan use measurements from the human body. Behavioral biometrics such as signature or keystroke dynamics use measurements based on human actions [3], [4], [13], [14]. Due to their strong variability over time, so far, behavioral biometric systems have been less successful compared to physiological ones [3]. Despite such limitation, behavioral biometrics such as mouse dynamics and keystroke dynamics, carry the greatest promise in the particular field of online computer user monitoring [7]. For such application, passive or non-intrusive monitoring is essential. Unfortunately most biometrics systems require special hardware device for biometrics data collection, restricting their use to only networks segments where such devices are available. Behavioral biometrics such as mouse dynamics and keystroke dynamics are appropriate for such context because they only require traditional human-computer interaction devices.

In this chapter, we present some techniques for extracting and analyzing mouse and keystroke dynamics data for online computer user monitoring. While the use of mouse dynamics for online monitoring is straightforward, the use of keystroke dynamics for such purpose faces the important challenges underlying the need for free-text detection.

7.2. Biometrics Modes and Metrics

Biometric systems operate in two modes, the enrollment/identification mode. In the first mode, biometric data is acquired using a user interface or a capturing device, such as a fingerprints scanner. Raw biometric data is then processed to extract the biometric features representing the characteristics, which can be used to distinguish between different users. This conversion process produces a processed biometric identification sample, which is stored in a database for future identification/verification needs. Enrolled data should be free of noise and any other defects that can affect its comparison with other samples. In the second mode, biometric data is captured, processed and compared against the stored enrolled sample. According to the type of application, a verification or identification process will be conducted on the processed
sample as follows.

**Verification process:** conducts one-to-one matching by comparing the processed sample against the enrolled sample of the same user. For example, user authentication at login: the user declares his identity by entering his login name. He then confirms his identity by providing a password and biometric information, such as his signature, voice password, or fingerprint. To verify the identity, the system will compare the user’s biometric data against his record in the database, resulting with a match or non-match.

**Identification process:** matches the processed sample against a large number of enrolled samples by conducting a 1 to N matching to identify the user; resulting in an identified user or a non-match.

In order to evaluate the accuracy of a biometric system, the following metrics must be computed:

- **False Acceptance Rate (FAR),** the ratio between the number of occurrences of accepting a non-authorized user compared to the number of access trials.
- **False Rejection Rate (FRR),** the ratio between the number of false alarms caused by rejecting an authorized user compared to the number of access trials.
- **Failure to Enroll (FTE),** the ratio characterizing the number of times the system is not able to enroll a user’s biometric features. This failure is caused by poor quality samples during enrollment mode.
- **Failure to Capture (FTC),** the ratio characterizing the number of times the system is not able to process the captured raw biometric data and extract features from it. This occurs when the captured data does not contain sufficient information to be processed.

FAR and FRR values can vary significantly depending on the sensitivity of the biometric data comparison algorithm used in the verification/identification mode; FTE and FTC represent the sensitivity of the raw data processing module.

In order to tune the accuracy of the system to its optimum value, it is important to study the effect of each factor on the other. If the system is designed to minimize FAR to make the system more secure, FRR will increase; on the other hand, if the system is designed to decrease FRR by increasing the tolerance to input variations and noise, FAR increases. By considering the current utilization of biometrics in the market, we note that they are widely used for secure identification and verification purposes. Other security systems, such as intrusion detection systems,
Ahmed Awad E. Ahmed, Issa Traoré

do not use this technology because the application requires continuous monitoring and recording of biometric information, which can sometimes be unavailable during the detection process. Mouse dynamics biometrics, which we introduce in this paper, happens to be a good fit for such an application [1]. This biometric system does not need any training, operating totally transparent from the user, as its enrollment mode does not require the user to present any particular biometric information by conducting specific actions.

Figure 7.1 shows a generic architecture, which covers the components involved in the implementation of a behavioral biometric based detector. The flow of control and data is shown for the three possible scenarios: enrolling a user, verifying his identity, and identifying an unknown user.

7.3. Mouse Dynamics

7.3.1. Overview

In contrast with keystroke dynamics, which has been widely studied in computer security, previous works on mouse dynamics have, so far, been limited to user interface design improvement [16], [15], [12]. In particular, mouse movement analysis has been the purpose of extensive research works. Studies have been conducted to establish the applicability of Fitts’ law in predicting the duration of a movement to a target based on the size of the
target and the distance from the starting point to the target [16]. According to Fitts’ law, the mean movement time for a movement with distance $A$ to a target with width $W$ is as follows: $MT = a + b \log_2(2A/W)$ where $a$ and $b$ are empirically determined parameters [16].

Researches conducted on mouse dynamics have focused mainly on the formalization of the measured data after fixing the environment variables. In our research, we target the biometric identification problem by focusing on extracting the behavioral features related to the user and using these features in computer security. Mouse dynamics is a new behavioral biometric introduced at the Information Security and Object Technology (ISOT) research lab at the University of Victoria [1], [2]. Mouse Dynamics can be described as the characteristics of the actions received from the mouse input device for a specific user, while interacting with a specific graphical user interface. The raw data collected for each mouse movement consist of the distance, time, and angle. We broadly refer to the angle as the direction of movement. Silence periods between mouse movements also carry valuable information that can be used for biometric recognition.

The user characteristics can be described by a set of factors generated as a result of analyzing the recorded mouse actions. Those factors represent the Mouse Dynamics Signature, which can be used in verifying the identity of the user.

The architecture described in Figure 7.1. can be applied for the implementation of any detector based on this biometric. Figure 7.2. shows sample data provided to the behavior-modeling component for processing. Each point in the figure represents a movement of the mouse with a specific distance, which was completed in a specific time. By examining the output of the data collection and processing component and comparing it to its output for different sessions for the same user, one can find a pattern characterizing the data. In some cases one can even differentiate between readings for two different users.

### 7.3.2. Detection Process

In order to automate the detection process, however, it is important to formalize the data in such a way that it can be used for comparison. Various statistical analysis packages can be used to achieve this goal, according to the characteristic of each factor [6], [17]. We choose to use Neural Networks to approximate the collected data to a curve that can be used to identify the user behavior. Function approximation is considered one of the most common uses of neural networks. It was shown by Hecht-Nielsen that for any continuous mapping of $f$ with $n$ inputs and $m$ outputs, there must exist a three layer neural network with an input layer of $n$ nodes, a hidden layer
Ahmed Awad E. Ahmed, Issa Traoré

Fig. 7.2. Graph showing a relationship between speed and distance based on sample intercepted data.

with \(2n+1\) nodes, and an output layer with \(m\) nodes that implements \(f\) exactly. According to those results, it is expected that neural networks can approximate any function in the real world. Hecht-Nielsen established that back propagation neural network is able to implement any function to any desired degree of accuracy. For our neural network, we use a feed-forward multi-layer perceptrons (MLP) network. MLP is one of the most popular network architectures; it is widely used in various applications. Our network involves a number of nodes organized in a layered feed-forward topology consisting of an input layer, an output layer and one hidden layer. All connections between nodes are feeding forward from inputs toward outputs. The MLP network uses a linear Post Synaptic Potential (PSP) function; the PSP function used is the weighted sum function. The transfer function used in this network is the log-sigmoid function. A linear transfer function is used for the input and output layers to allow the expected input and output range. For faster training, the network is initialized with the weights and biases of a similar network trained for a straight line. The output of our
neural network can be described by the following equation:

$$y = \sum_{j=1}^{N} \left( \frac{w_{2j}}{1 + e^{\sum_{i=1}^{N} w_{1i} x_i - b_{1j}}} \right) - b_{21},$$  \hspace{1cm} (7.1)$$

where $w_{ij}$ and $b_{ij}$ represent the weights and biases of the hidden and output layers respectively, $x$ is the input to the network, and $N$ represents the number of nodes in the hidden layer (which is set to $N=5$ in our design). We use the back propagation algorithm to train the network. The error criterion of the network can be defined as follows:

$$E = \frac{1}{2} \sum_{i=1}^{p} (t_i - y_i(x_i, w))^2,$$  \hspace{1cm} (7.2)$$

where $w$ represents the network weights matrix and $p$ is the number of input/output training pairs set $(x_i, y_i)$. During the behavior modeling stage, the neural network is trained with processed raw data. Input vectors and their corresponding target vectors are used. The back propagation-training algorithm is used to train a network until it can approximate a function describing the collected data. Curve over fitting is one of the common problems related to this approach; this problem occurs when the resulted curve has high curvatures fitting the data points. The main cause of this problem is when the network is over trained with a high amount of data; such a problem increases the noise effect and reduces the generalization ability of the network.

In order to avoid the over fitting problem, first the right complexity of the network should be selected. In our design, we tested different network configurations and concluded that a network with a single hidden layer containing five perceptrons produces the best results. Second, the training of the network must be validated against an independent training set. At the beginning of the training, the training error and the validation error will decrease until we reach a point where the validation error will start to increase. We call this point the stop point (corresponds to point $A$ in Figure 7.3.), where the training should stop to obtain the desired generalization. After the network-training curve reaches the stop point, the network will be fed with a test stream presenting the spectrum of the input data; the result is a curve approximation of the training data. This curve is considered as a biometric factor in the Mouse Dynamic Signature.

Figure 7.4. shows the mouse dynamics signature for three different users, the figure shows the relation between the traveled distance and the elapsed time for any performed actions. Figure 7.5. shows the same curves computed over five different sessions for the same user. Each of those curves was produced as a result of training the neural network with session
raw data similar to what is shown in Figure 7.2. The neural network approximates the collected data to a curve, which can be used to check the user identity by comparing it to a reference signature. Deviations between the curves can be used to detect sessions belonging to different users, or to recognize sessions belonging to the same user.

### 7.3.3. Silence Analysis

As mentioned above, silence periods can also be considered as valid and useful mouse biometrics data. Analysis of the silence periods can lead to the detection of a number of distinctive characteristics for each user. In this analysis we only consider the short silence periods (less than 20 sec), which happens between movements. Longer silence periods may occur as a response to a particular action like reading a document, and usually contain noise.

Figure 7.6. illustrates one of the signature factors that can be derived from silence analysis. This factor is based on the histogram of the short silence periods. Each bar in the figure represents the number of silence periods reported in a user session where the silence period is within a 2 sec interval covering a spectrum of 20 sec. The figure shows the histogram
A Statistical Model for Biometric Verification

Fig. 7.4. Mouse signature for three different users.

Fig. 7.5. Mouse signature computed over five different sessions for the same user.

for two different users, we can easily notice the difference in the behavior. From the first bar ($0 < t < 2$) we can notice that for the first user 45% of his silence periods are in this category, while for the second user only 8% of his silence periods last less than 2 sec.
Fig. 7.6. Silence Analysis: histogram of the silence time for two different users.

Fig. 7.7. Silence Analysis: histogram of the silence time based on three different sessions for the same user.

Figure 7.7. illustrate silence histogram computed over three different sessions for the same user. We can notice the similarity of the behavior across these different sessions.

Many other factors can be computed from the raw mouse data, and used for biometrics recognition. We refer the reader to [1], [2] for more about these parameters.
7.4. Keystroke Dynamics

7.4.1. Overview

Keystroke dynamics recognition systems measure the dwell time and flight time for keyboard actions, and use such data to construct a set of digraphs, tri-graphs or \( n \)-graphs producing a pattern identifying the user. User authentication is the most suitable application for such technology.

Since the 1980s’, many researches have been done in this area. Most of the researches focus on using this technology for user authentication or access control [3], [4], [5], [8], [10], [11]. The various works reported in the keystroke literature involves using a wide range of statistical methods to analyze keystroke dynamics. For instance, Brown and Rogers used neural networks to solve the problem of identifying specific users through the typing characteristics exhibited when typing their own names [5]. In [11], Monrose and Rubin developed a technique to harden passwords based on keystrokes dynamics. More recently, Bergadano and colleagues presented a new technique based on calculating the degree of disorder of an array to quantify the similarity of two different samples [3]. Most of these works focus on fixed text detection. Because of the time limitation of the identification process, the user is asked to type a predefined word or set of words in order to get reasonable amount of data for the identification. During the enrollment process, the user is also required to enter the same fixed text. For passive monitoring, we need to be able to detect the user without requiring him to enter a predefined message or text. So free text detection is essential for our purpose. However, free text detection presents huge challenges, which explain the limited number of related work published in the literature. So far, one of the most significant works in this area was authored recently by Guneti and Picardi who adapt for free text detection the technique based on the degree of disorder of an array introduced in [3] for fixed text detection. Still the importance of this issue warrants investigating alternative techniques. In the rest of this section, we discuss the different factors underlying this issue and propose three different techniques, which can be used to tackle them.

7.4.2. Free Text Detection Using Approximation Matrix Technique

The first approach we propose is based on digraph analysis. The approach utilizes a neural network to simulate the user behavior based on the detected digraphs.

The neural network used for this approach depicted by Figure 7.8.
Ahmed Awad E. Ahmed, Issa Traoré

Fig. 7.8. Neural network model used for free text detection based on Approximation Matrix.

A feedforward multilayer perceptrons network. The training algorithm is back propagation. The network consists of four layers: an input layer, two hidden layers, and a single node output layer. The input layer consists of \( N \) number of nodes where \( N = 2 \times n \), with \( n \) corresponding to the Number of Monitored Keyboard keys. Input to the nodes is binary 0 or 1, as each node in the input layer represents a key. The first \( n \) nodes represent the key where the action is started at, and the second \( n \) nodes represent the key where the action ends. Each batch of nodes should have only one input set to 1 while the other inputs are set to 0; the node set to 1 represents the selected key.

During the enrollment phase, a batch of \( M \) actions will be collected and fed to the behavior modeling neural network as training data. A simulation will run after the neural network has been trained with this batch, this simulation will consist of the set of non-redundant actions collected from the enrollment data. The result of this simulation will be stored for each user as well as the training data, which will be used also in the verification stage. During the verification mode a small batch of actions will be used in this stage to verify the user identity. This batch will be added to the training batch of the user’s neural network, resulting a network with different weights. The effect of the small batch on the network...
weights represents a deviation from the enrollment network. In order to measure this deviation, another simulation will run on this network with the same batch prepared for the enrollment process for the specific user. By comparing the result of this simulation to the enrollment stage result, the deviation can be specified. An approach that can be used here is to calculate the sum of the absolute difference of the two results, if this deviation is low then the collected sample is for the same user, if not then this sample is for another user.

Figure 7.9. shows the detector architecture and the flow of data in enrollment and detection modes. In enrollment mode extracted monographs and digraphs are encoded with a mapping algorithm. This process is needed in order to convert key codes into another representation, which is relevant and meaningful as an input to the neural network.

Since this detector is based on free input text it is very important to be able to evaluate if the collected data is enough during enrollment mode. The aim of this research is to develop a technique to help in minimizing the amount of data needed for the enrollment process, by extracting the needed information from the information detected so far.

In order to approximate unavailable digraphs, we use a matrix-based approximation techniques. Specifically we use a pair of matrix named coverage matrix and approximation matrix. Coverage matrix is a two
Ahmed Awad E. Ahmed, Issa Traoré

dimensional matrix, which is used to store the number of occurrences of the observed graphs in the enrollment mode. Keeping track of such information helps in different areas such as in evaluating the overall coverage of the enrollment process and the development of a customized enrollment scenario, which can be used in case of low coverage. Approximation matrix, which is a two dimensional matrix represents the relations between the keys and how close or far they are from each other; the matrix will be initialized with numbers representing the relative distances between the keys on the keyboard.

Figure 7.10. illustrates how the approximation process is performed. Let's assume that an approximation for the $EB$ digraph is needed. We can detect that directly from its value corresponding to 0 in the coverage matrix, depicted by Figure 7.10.b. The approximation matrix, depicted by Figure 7.10.a will be used to locate alternative entries (for each key), which have the lowest distance in the matrix; in this case these correspond to $(D,F)$ and $(G,H)$ respectively. From this step we can enumerate the tentative approximations, which correspond in this case to $DG$, $DH$, $FG$, and $FH$. In the next step the distance of each combination will be calculated from the approximation matrix (underlined numbers in figure 7.10.a), where they will be sorted according to their closeness to the original distance of the approximated digraph ($AppMatrix(EB) = 3$). The sorted result is $(FH, DG, DH, FG)$. The Coverage matrix will be used to make the final decision out of the sorted result. The matrix in Figure 7.10.b shows only the weights of the tentative combinations. Notice that digraph FH has a coverage of 30, which means that it is a good candidate (the best fit in this case, since it is also the closest fit in the approximation matrix). The second alternative $DG$ also has good coverage, while $DH$ has a relatively low coverage.

7.4.3. **Free Text Detection based on Keyboard Layout Mapping**

One of the important factors to be considered in the enrollment phase is the amount of data needed to enroll the user and create a signature modeling his behavior. The aim is to minimize the enrollment time as much as possible without affecting the accuracy of the system in detection mode. Since the key codes do not reflect the relation between the keys like their absolute or relative positions, the pre processing stage (Figure 7.9.) should include a mapping mechanism in order to convert those sets of keystrokes into numbers, which are suitable to train the network with. Such numbers should reflect a specific characteristic for each key and its relation to other keys. The mapping method used in the previous technique is a binary mapping since each key is represented by its own network input, and the
A Statistical Model for Biometric Verification

Fig. 7.10. Example of how to approximate unavailable digraphs.

effect of other keys was removed by restricting the input to only one high input at a time. This technique is considered to involve high computation power requirement, since the number of nodes in the neural network is large.

The second approach we propose is based on a keyboard layout mapping technique. The function of this technique is to replace each key code by its location on a previously identified keyboard layout. In our implementation we use the QWERTY keyboard layout depicted by Figure 7.11.
Fig. 7.11. QWERTY Keyboard layout can be used to encode keys as input to the neural network.

Each key will be presented by a pair of numbers representing its $x$ and $y$ location. Neural networks can still be used to model the relation between the keys and the time used to move between them.

Figures 7.12.a and 7.12.b show the neural networks used for the digraph and monograph analysis respectively. The networks in this case are lighter than the one used in the previous technique. The numbers of input nodes are 2 and 4 for each network respectively as only 2 network inputs are needed to represent a key. The hidden layers consist of 5 nodes for the monograph network and 12 nodes for the digraph network.

### 7.4.4. Free Text detection based on Sorted Time Mapping

Another mapping technique, which can be used to prepare the data for the neural network is to sort the key codes based on associated average time, and accordingly to map them with corresponding order. For the monographs set, the average time is the dwell time. So, in this case the key codes will be sorted according to the average dwell time. Key codes will be mapped to their order in the sorted list before being fed to the neural network. In this case the mono network will have only one input node. For the digraph set the sorting order is different. Digraphs will be sorted two times: the first time they will be sorted according to the average of the times of all digraphs with the same code in the from-key; the second time they will be sorted according to the average of the to-key. A key code will be mapped to its sort order according to whether it is to or from. The input layer for the neural network in this case consists of two nodes.

The keyboard layout mapping technique doesn’t require a full coverage of all possible set of keys as it is based on a very definite reference scheme, which is the keyboard layout. So missing a number of keys during enrollment phase does not prevent this technique from working. However, for the sorting technique it is mandatory that the user provide
A Statistical Model for Biometric Verification

input at enrollment phase covering the whole set of monitored keys. This criterion makes the approximation matrix technique and the keyboard layout technique more suitable if the system is designed to hide the enrollment process from the user. The sorting technique will be very effective if a controlled enrollment procedure is used.

Figure 7.13. shows monographs signatures for two different users. The figure shows the relation between the key code, and the dwell time involved. The sequence of the keys represented by those codes is different for each user as they were sorted for each user individually. The Y-axis represents the output of the neural network after it has been trained with the sorted key order. Taking a look again at Figure 7.9., the output of the behavior-modeling component in this case is equivalent to the curve shown in Figure
7.13. for monograph analysis and to a set of curves or a 3-D matrix for digraph analysis.

Fig. 7.13. Comparing signatures (Monograph Analysis): User 2’s session compared to user 1’s reference signature.

Fig. 7.14. Comparing signatures (Monograph Analysis): User 1’s session compared to user 1’s reference signature.
Figure 7.14. shows Monograph signature calculated from one of user 1’s sessions compared to his reference signature. Figure 7.13. shows the signature calculated for one of user 2’s sessions during detection mode as a result of supplying his session data to User 1’s neural network. We can note from the figure that the difference between this curve and user 1’s reference signature is high compared to the difference between user 1’s curves.

7.5. Conclusion

Biometric systems are widely deployed to ensure security in a large spectrum of today’s industries. Some implementations rely solely on biometrics; others utilize it to increase their level of security. The choice to implement a specific biometric technology in an IT environment is ruled by a number of factors. Examples of those factors are its accuracy, cost, user acceptance, and stability on the long term.

In this chapter we present methods based on statistical and artificial intelligence techniques targeting the use of behavioral biometrics for online computer user monitoring. The techniques presented in this chapter work on a large set of features from a newly introduced biometric (mouse dynamics) and the keystroke dynamics biometric without putting any restriction on the user at any stage of operation making it suitable for online identity verification and detection applications. As those biometrics gain high acceptability they also gain a relatively high accuracy. In experiments conducted in our lab, we achieve an overall accuracy of $FAR = 0.00651$, and $FRR = 0.01312$. This is based on a number of experiments conducted locally and remotely through our lab involving a set of 22 participants.

Acknowledgments

This work was partly supported by the Natural Science and Engineering Research Council of Canada (NSERC) through an Idea-to-Innovation grant. We would like to thank all the participants in the experimental evaluation of our framework, as well the Innovation Development Corporation of the University of Victoria for their support. We would like also to thank Dr Svetlana Yanushkevich for her review and valuable comments on this work.

Bibliography

Ahmed Awad E. Ahmed, Issa Traoré

Security and Trust, October, 2005 St. Andrews, New Brunswick, Canada, pp. 91-98.


13. Parker, J. R. Composite Systems for Handwritten Signature Recognition, this issue.

14. Popel, D. V. Signature analysis, verification and synthesis in pervasive environments, this issue.

