User Authentication Based on Representative Users

Alon Schclar[1], Lior Rokach[2], Adi Abramson[2]

[1] School of Computer Science, Academic College of Tel-Aviv
P.O.B 8401, Tel Aviv 61083, Israel
alonschc@mta.ac.il

[2] Department of Information Systems Engineering, Ben-Gurion University of the Negev
P.O.B. 653, Beer-Sheva 84105, Israel
{adiabr, liorrk}@bgu.ac.il

Abstract

User authentication based on username and password is the most common means to enforce access control. This form of access restriction is prone to hacking since stolen usernames and passwords can be exploited to impersonate legitimate users in order to commit malicious activity. Biometric authentication incorporates additional user characteristics such as the manner by which the keyboard is used in order to identify users.

We introduce a novel approach for user authentication based on the keystroke dynamics of the password entry. A classifier is tailored to each user and the novelty lies in the manner by which the training set is constructed. Specifically, only the keystroke dynamics of a *small subset* of users we refer to as *representatives* is used along with the password entry keystroke dynamics of the examined user. The contribution of this approach is two-fold: it reduces the possibility of overfitting while allowing scalability to a high volume of users. We propose two strategies for constructing the subset for each user. The first selects the users whose keystroke profiles govern the profiles of all the users while the second strategy chooses the users whose profiles are the most similar to the profile of the user for whom the classifier is constructed.

Results are promising reaching in some cases 90% Area Under the Curve. In many cases, a higher number of representatives deteriorates the accuracy which may imply overfitting. An extensive evaluation was performed using a dataset containing over 780 users.

Keywords: User authentication, behavioural biometric, keystrokes biometric, computer security.

1. Introduction

Identity theft is a fraud in which criminals impersonate legitimate users by stealing their credentials, such as credit card details and passwords, or by exploiting a logged-on computer which was left unlocked by the user. Stolen identities may be used to perform a wide range of malicious activities such as on-line purchases that are performed under stolen identities. Such purchases incur losses of billions of dollars to the websites as well as to their insurance companies (Jain et al. 1999).

Currently, the most common way to enforce access control is by password, PIN (Personal Identification Number) or other predetermined passcode (Grabham and White, 2007; El-Saddik et al. 2007). The user is required to enter her credentials before she is allowed to perform her intended activity. This form of access control although effective to a certain extent, has many flaws which make it vulnerable to hacking (Peacock et al. 2004). In order to make a password hard to hack, it must adhere certain rules e.g. include at least eight characters, some of which capital letters and special characters (e.g.: @ , ? , !). Unfortunately, hard-to-hack passwords are also *hard-to-remember*. Consequently, many users choose passwords that relate to their private lives, e.g. digits from their social security number, pet's name, parent's or kids' names - making them easy to hack. Furthermore, many users write their passwords on a note which may be intercepted by hackers. This so called "memory obstacle" also drives most users to use the same username and password in several web sites. Thus, a hacker revealing a users' password from a non-secure website will gain access to many of the websites that the user has access to – hacking into some of which, such as the user's bank website, may incur devastating damage to the user. Due to these drawbacks, password-based user authentication methods provide only partial protection against hackers and thus they need to be complemented by additional authentication means, e.g., *physiological* and *behavioral* biometrics.

Behavioral biometrics such as keystroke dynamics can be used to identify the user either during log-in or throughout the time the user is logged-on (the latter is referred to

as *continuous verification* and is out of the scope of this paper). Authentication methods that employ this approach rely on the assumption that the keystroke dynamics of each user stay almost the same in each login attempt while uniquely characterizing each user (Monrose and Rubin, 1997). Commonly, the keystroke dynamics of the user are extracted during login and compared to a model that was constructed based on the user's keystroke dynamics and/or similar features of other users.

Physiological biometrics include fingerprints (Jain et al. 1999), iris patterns (Pierscionek et al. 2008), retina patterns (Jain et al. 1999), body heat (Jain et al. 1999), keyboard typing pressure (Hidetosshi and Kurihara, 2004), palm lines (Wu et a. 2006) and haptic measurements (El-Saddik et al. 2007), to name a few. Physical biometrics have many advantages e.g. they are harder to steal (although an imposter can still forge a fingerprint - Modi and Elliott, 2006) and cannot be lost or forgotten since users do not need to remember them or write them down as opposed to a password or a PIN. However, authentication systems that use these features require special hardware, making them more expensive and time consuming to develop than methods that rely on existing hardware devices (e.g. mouse and keyboard). Moreover, the accuracy of biometric-based systems may be affected by various factors: if a fingerprint is changed by a cut, a burn or its moisture level, the system may fail to identify that person; the retina may be influenced by health problems such as glaucoma and high blood pressure which are known to change the retina in subjects (Jain et al. 1999). Additionally, when physical biometrics, such as fingerprints, are stolen, not only can they be used to falsely incriminate the innocent but also the legitimate owner cannot change them to prevent future impersonation attempts whereas a compromised password can simply be replaced to prevent such attempts. Finally, acquisition of the biometric features may annoy the user since it requires interaction with special hardware.

Contrary to physiological biometrics, the acquisition of behavioral features is non-intrusive and transparent to the user. For example, in case keystroke dynamics are used, a *background* process is used to collect them from the user's keyboard usage. This makes the authentication process smoother and more user-friendly. Note that behavioral biometrics authentication systems need to store the biometric features in addition to the

password. Accordingly, encryption is required to protect them similarly to passwords since the biometric features of passwords entry may be exploited to narrow down an exhaustive search of passwords. We assume that such measures are taken.

In this paper, we propose a new approach to user authentication according to the keystroke dynamics of the password entry. In the proposed system every user is characterized by a biometric *profile* which is constructed in the following way: the users are required to type their password for a given number of times. Features are extracted from the keystroke dynamics of every password entry and are represented as a vector – one for each password entry. The feature vectors that are extracted from the password entries of a given user form her biometric profile. The biometric profiles of all users are stored in a profile database.

The authentication of a user is accomplished using a classifier that is tailored to each user. The novelty of the proposed approach is in the manner by which the training sets are constructed for the examined users. Specifically, a *small subset* of *representative users* is selected *for each examined user*. Various strategies may be employed for the selection process and the subset content depends on the examined user as well as the chosen strategy. The training set is composed of the biometric profiles of the representative users and the user for whom the model is built. The underlying assumption is that different levels of similarities can be found among the biometric profiles of all users. Accordingly, it may be sufficient to identify a user by distinguishing her password keystroke dynamics from the profiles of only a subset of users rather than the profiles of all the users. Ideally, when the selected subset represents the entire spectrum of users, the biometric profile of a hacker will resemble a representative profile which is different from the one of the user that he is trying to impersonate and thus will be classified as an imposter.

By using a subset of users instead of the entire set, we aim to achieve two goals: first, prevent overfitting, and second, facilitate scalability to handle a large number of users. The experimental evaluation shows that the first goal is achieved by the proposed method since in most cases choosing a higher number of representatives reduces the

authentication accuracy. According to Peacock et al. (2004), the goal has not been addressed although being a key feature of biometric authentication systems. In the proposed approach, even though only a small subset of users is used to build the authentication model, the model may still be used to authenticate the user among a high volume of users. Specifically, in our experiments the largest number of representatives that was used to construct a model was less than 7% of the total number of 783 users and in many cases the model that achieved the best results contained less than 5% of the total number of users. Note that the entire set of the users is only needed during the selection of the representatives. After the selection, the entire set is no longer needed and the construction and authentication relies only on the representatives.

We propose two strategies for choosing the representatives. Both employ clustering to the keyboard dynamic features of all the users in order to detect inter-profile similarities. Each cluster contains a subset of feature vectors that are similar to one another. Thus, each cluster represents a unique keystroke behavior that may be used to characterize a number of users in the dataset (provided the number of clusters is smaller than the number of users). Due to this similarity, a unique user profile may be used to represent each cluster instead of using all the feature vectors in the cluster. Employing this assumption, the first strategy chooses a unique user to represent each cluster. In order to do so the centroids of the clusters are calculated as well as the centroids of every user's feature vectors. The centroid of a set of vectors is defined as their mean. Given a cluster, the user that is chosen to represent it is the one whose feature vectors' centroid is the closest to cluster's centroid. Theoretically, this process may result is a user representing more than one cluster. However, when this occurs, the user is assigned to only one of the clusters where different users are chosen so that every cluster is represented by a unique user. This is required in order to obtain a subset whose diversity is as high as possible and it is accomplished by applying the Hungarian matching algorithm (Harold, 1955) which is described in details in Sec. 3.1.

The second strategy chooses a different set of representatives. Let u be a user for whom a classifier is constructed. The users that are chosen as representatives are those whose profiles are similar to u's profile. This is achieved in the following manner: First,

the centroid $c(u)$ of u's feature vectors is calculated. Next, the cluster k whose centroid is the closest to $c(u)$ is found. Then, the centroid of the feature vectors of each user is calculated. The users that are chosen as representatives are those whose feature vectors' centroids are the closest to the centroid of the cluster k (the cluster that is associated with the user u).

Given a user, *half* of her feature vectors together with *half* of the feature vectors of each of the representatives are used to train the user's classifier. The remaining halves together with the feature vectors of the rest of the non-representatives users are used as the test set where the remaining half of the user is used to check whether the classifier identifies her and the rest simulates imposters. Unfortunately, an organization is vulnerable to attacks that come from both users that belong to the organization (internal) and users that are external to it. In this sense, we regard the representatives as internal users and use half of their feature vectors for the training set while the rest of the users simulate users that are external to the organization. Note that due to the low number of representatives, the vast majority of test users are part in the training and so this construction simulates an *open world* setting where most imposters are external to the organization.

The rest of the paper is organized as follows: in Section 2 we describe the general framework and various aspects of user authentication systems that use behavioral biometrics. In Section 3, we give a formal description of the proposed approach. Experimental settings and results are provided in Section 4. We conclude in Section 5 with a description of the various challenges and open problems that need further investigation in order to make this approach fully operational.

2. Behavioral Biometric Authentication System (BBAS)

BBASs provide a cheap and effective security approach that complements password-based authentication methods. Most BBASs use keyboard or mouse behavioral characteristics. In this paper we focus on keyboard behavioral characteristics. Combining regular password authentication with biometric authentication can provide a security suite that may be more water-proof than systems that only rely on passwords. Namely, even if

a password is stolen by a hacker, the password needs to be typed in the same manner it is typed by its rightful owner.

The evaluation of such BBASs commonly uses a predefined phrase simulating a password (e.g., Killourhy and Maxion (2009a) and Killourhy and Maxion (2009b)). This phrase is entered by the users that take part in the evaluation and the extracted features are used for the construction of the authentication model. In Section 2.1 we describe current state-of-the-art BBASs.

Keyboard-based BBAS's, or KBBAS's for short, have many advantages and receive an increasing amount of attention for the following reasons:

(a) They do not require special hardware;
(b) Their operation does not require special attention from the user as opposed to retina scan, for example, in which the user is required to place her head in a retina scanner;
(c) Their development is easier compared to other biometric authentication methods; and
(d) Keyboard ubiquity makes the data collection process cheap and accessible.

Nonetheless, a reliable and effective KBBAS needs to overcome the following obstacles: (i) Keyboard-based biometrics are yet not reliable as physical biometrics such as the iris, fingerprint etc; (ii) Keyboard behavioral characteristics may change after a period of time due to fatigue and may also be influenced by the physical status of the user and his state of mind; (iii) The keyboard that is used to characterize the user plays an important role since users may type differently on different types of keyboards (laptop, desktop, ergonomic/non-ergonomic keyboards); and (iv) The keystroke rate depends on the user posture: standing, sitting, etc.

Generally, biometric-based user authentication systems consist of the following modules:

- *Event recording module* - captures events generated by user interaction with the input devices e.g. keyboard and mouse.

- *Feature Extraction module* - extracts features from the captured events such as the time each key was pressed and organizes them in a vector.
- *Classifier* - during the construction, the classifier is trained according to the feature vectors. During identification, the constructed classifier is used to confirm the identity of the user according to feature vectors extracted from her keystrokes. A wide variety of classifiers may be chosen e.g. decision trees (Breiman et al. 1993), Artificial Neural Networks and Support Vector Machines (Vapnik, 2000) to name a few.
- *Database* - Contains the behavioral characteristics of the users together with the classifiers.

Figure 1 depicts how these components are used during enrollment of a user.

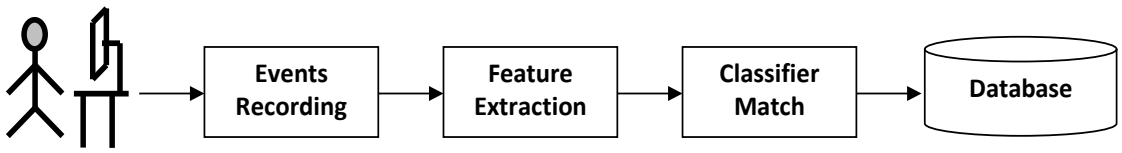

Figure 1: User enrollment in behavioral biometric user authentication systems.

2.1 Keyboard-based dynamics

Keystroke dynamics can be described by several features which are extracted from the typing rhythm of the user. These features are extracted from data which are recorded by the event recording module. Usually, each keystroke is represented by two timestamps: the moment that the key was pressed and the moment that it was released. *Dwell time* refers to a single keystroke and it is defined as the time that passed between the moment the key was pressed and the moment that it was released (Fig. 2a). Given *two* consecutive keystrokes, the following features can be defined:

- *Latency time* measures the time between the moment the first key was released and the second key was pressed (Fig. 2b).
- *Flight time* measures the time between the moment the first key was pressed and the moment the second key was pressed (Fig. 2c).

- *Up to up time* measures the time between the moment the first key was released and the moment the second key was released (Fig. 2d).

The latency time is also referred to as the *digraph latency time* and also as the interval time in some papers (Obaidat and Sadoun, 2000; Cho et al. 2000). It is not necessary to use all of these features. In fact, in this paper, as in many other papers, we only use the latency and dwell times since the other features can be derived from them.

KBBASs can be distinguished according to the training data that they use: *static* (Fixed) or *non-static* (free) text where the proposed method in this paper falls into the former category. Techniques that use static data, characterize the user keyboard behavior

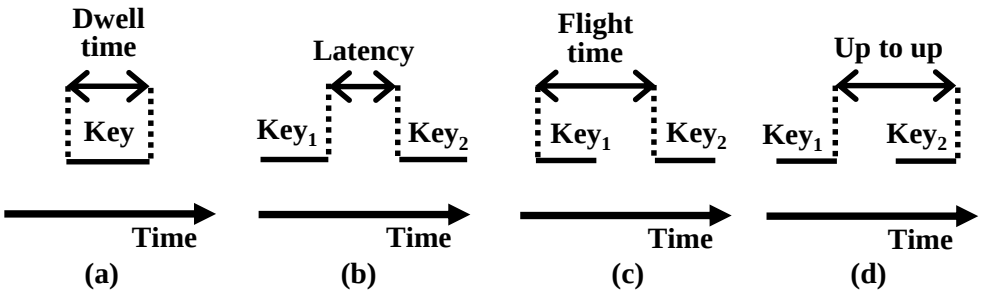

Figure 2: Keystroke features (a) Dwell time; (b) Latency; (c) Flight time; (d) Up to up.

based on features which are extracted from predetermined text that the users are required to enter. Methods that use non-static text, extract the keyboard behavior from any text content that is entered by the user where no limitations are imposed on the text and are out of the scope of this paper. Most methods extract the above features from sequences of two, three or any number of characters which are commonly known as digraphs, trigraphs and n-graphs, respectively.

In order to evaluate the performance of an authentication method the following performance metrics are used:

- *True Positive Rate (TPR)* - the ratio between the number of legitimate interactions that were correctly labeled and the total number of legitimate interactions.
- *False Acceptance Rate (FAR)* - the ratio between the number of attacks that were erroneously labeled as legitimate interactions and the total number of attacks.

- *False Rejection Rate (FRR)* - the ratio between the number of legitimate interactions that were misclassified as attacks and the total number of legitimate interactions.
- *Area Under the Curve (AUC)* – measures the area under the ROC curve. An ROC curve is a graphical representation of the tradeoff between the TPR (y-axis) and the FAR (x-axis) given a classifier which produces a probability for each predicted label. A high AUC is sought after since it corresponds to a better performance. In this paper the AUC is used to measure the accuracy of the proposed methods.
- *Equal Error Rate (EER)* – The rate at which both the false acceptance rate and the rejection rates are equal when plotting the FAR vs. the FRR in a similar manner to the construction of the ROC curve.

A curve that is related to the ROC curve plots the FAR vs. the FRR. This curve is useful for the evaluation of authentication systems since FAR corresponds to malicious users who are logged into the system while FRR corresponds to legitimate users being blocked from accessing the system which may antagonize the users. We aim to minimize both however, usually the FRR increases with the decrease in the FAR and thus ERR describes the point both achieve the best measure *with respect to one another*.

In the following we describe currently available keyboard-based authentication techniques which are based on static text that is used for the username and password.

Yong et al. (2005) use parallel decision trees (DTs) to authenticate users according to a fixed phrase containing 37 characters. A Monte Carlo approach is used to attain sufficient training data resulting in eight training subsets that are used to construct eight decision trees for each user. The user is authenticated if at least three DTs do so; otherwise the user is rejected. The average FRR was 9.62% and the average FAR was 0.88%.

Bleha et al. (1999) authenticate users according to two different types of passwords: names and a fixed phrase. Latencies were used to represent each password entry and the nearest neighbor and Bayesian classifiers were both used to authenticate a user. A

password entry was rejected only if both classifiers rejected the user (the indecision error i.e. when one classifier accepted the user while the other rejected her, was 1.2%). In both classifiers thresholds were used and their values were lowered if the user was rejected in the first attempt. In order to evaluate their approach 10 users were used as legitimate users and 22 as imposters. They achieved FRR of 3.1% and FAR of 0.5%.

Monrose et al. (1999) propose a user authentication scheme using a hardened password – a combination of the textual password along with its keystroke entry dynamics. The method required hackers to perform a more extensive exhaustive search to discover a password even if they got hold of the password file. The hardened password could also be used to encrypt the user's files. The method adjusts to changes in the user's keystroke dynamics by replacing older dynamics with new ones. The experimental evaluation used an eight character password, 20 users and a total of 481 logins in which the correct password was entered. However, since the evaluation was based on a password guessing procedure which is unique to their paper, we do not include its results.

Hosseinzadeh and Krishnan (2008) use an up-to-up keystroke latency (UUKL) feature and compare its performance with the key hold-down time (KD) and down-to-down keystroke latency (DDKL) features using a Gaussian mixture model (GMM)-based verification system that utilizes an adaptive and user-specific threshold based on the leave-one-out method (LOOM). Their results show that the UUKL feature significantly outperforms the KD and DDKL features. Furthermore, the inclusion of the UUKL feature achieved an equal error rate (EER) of 4.4% based on a database of 41 users.

Revett et al. (2005) describe an authentication algorithm based on rough sets. The users were asked to enter a 14 character passphrase composed of three words. They examined the digraphs of each passphrase and extracted: (a) the time between consecutives keystrokes; (b) time for entering each word in the passphrase; (c) total time to enter the passphrase; and (d) time spent to enter half of the passphrase. These attributes were discretized using an entropy/MDL algorithm and used to derive a set of authentication rules. The experimental evaluation included approximately 100 users which were split into two groups: legitimate (10 users) and imposters. Recently,

Killourhy and Maxion (2009a; 2009b) compared between 14 anomaly detection techniques for user authentication which included: one-class SVM, Fuzzy logic, Neural Networks (standard and auto-associative), Nearest neighbor using various metrics e.g. the Mahalanobis, the Euclidean and the Manhattan distances. The evaluation used data that was collected from 51 users who entered a 10 character predefined password 400 times in 8 different sessions (the dataset is available on-line - Killourhy and Maxion, 2009b). The model that achieved the best results (ERR=9.62%) used the nearest neighbor inducer with the scaled Manhattan distance.

3. The Proposed Algorithm

Let $U = \{u_i\}_{i=1}^{N}$ be the set of N users from whom keystroke dynamics are collected. Every user is required to enter a predefined password for M times. The same password is entered by all users so that only the biometric authentication capabilities of the proposed approach are evaluated. Features are extracted from each password entry and a total of $N \cdot M$ feature vectors are formed. We denote this set by $\Sigma = \{s_{i,j}\}$ where $s_{i,j} \in R^D$ is the feature vector of the j-th password entry of user i, D is the number of features that are collected and $i = 1, \ldots N;\ j = 1, \ldots, M$.

We denote by $R(u) \subset U$ the set of representative users that are chosen for the construction of user u's classifier. In order to choose $R(u)$, we first partition Σ into K clusters where K is given as a parameter to the algorithm and K-1 is the number of representatives we look for (we always include u in the set of representatives as mentioned in Section 1). We assume that two close feature vectors (according to the Euclidean distance) indicate similarity between the keyboard dynamics of their corresponding passwords entry. We denote by $C = \{c_i\}_{i=1}^{K}$ the centroids of the obtained clusters.

Two strategies for selecting $R(u)$ are proposed. The first method chooses a unique user representative from each cluster. We refer to this approach as the *cluster representative* (CR) approach. The second approach, selects the users whose biometric profiles are the most similar to that of the examined user. We refer to this approach as the inner-cluster nearest-neighbor approach (ICR).

3.1 Choosing *R(u)* as cluster representatives (CR)

We calculate the centroids of the feature vectors of each user. The representative of the *i*th cluster, denoted by r_i, is chosen as the user whose feature vectors' centriod is the closest to c_i i.e.

$$r_i = \arg\min_{u \in U} \left\| c_i - \frac{1}{M} \sum_{j=1}^{M} s_{u,j} \right\| \quad (1)$$

The set of u's representatives is given by $R(u) = \{r_i\}_{i=1}^{K}$. Choosing $R(u)$ in this manner may result in users that are selected more than once (from a number of clusters). In this case, the number of representatives that are chosen is smaller than K which in turn may damage the diversity and the accuracy of the constructed classifier. In order to remove user repetitions, we apply the Hungarian matching algorithm (Harold, 1955) to the set of users and clusters. This algorithm matches each cluster with a *unique* user.

The general matching problem takes a bipartite graph $(V_1 \cup V_2, W)$ where $N = |V_1| = |V_2|, V_1 \cap V_2 = \emptyset$ and $W = \{w(v_i, v_j)\}_{i,j=1,\ldots,N}; v_i \in V_1,\ v_j \in V_2$. The weight of an edge $w(v_i, v_j)$ is the cost for matching v_i to v_j. The Hungarian matching algorithm finds for each vertex in $v_i \in V_1$ a vertex $v_{j_i} \in V_2$ such that $\sum_{i=1}^{N} w(v_i, v_{j_i})$ is minimal. In our settings, we set V_1 to U where each vertex v_i corresponds to the user u_i. We set V_2 to be the set of clusters that are represented by their centroids. The weight of an edge between a vertex (user) $u_i \in V_1$ and a vertex (centroid) $c_k \in V_2$ is set to be the distance between c_k and the centroid of u_i's feature vectors.

A precondition of the original matching problem is that $|V_1| = |V_2|$. Since the number of clusters is substantially smaller than the number of users we add to V_2 a set of $|U| - K$ dummy vertices which we denote by $\Delta = \{\delta_p\}_{p=1,\ldots,|U|-k}$ such that $V_2 = C \cup \Delta$. We set an infinite weight to the edges between the users in V_1 and the dummy vertices i.e. $w(u_i, \delta_p) = +\infty$, p = 1,…, |*U*|−*K*. Due to Eq. 1 and the infinite weight of the edges connecting the vertices in V_1 to the dummy vertices, each user will be matched to a non-dummy vertex. We denote by u_{i_k} the user that is matched to cluster c_k. Figure 3 illustrates the construction of the bipartite graph and an example of a matching result.

Recall that half of u's feature vectors together with half of the feature vectors of the representative users form the training set of u's classifier. Accordingly, u must be included in $R(u)$. In case $R(u)$ does not include u we add it to $R(u)$ replacing the representative whose cluster centroid is the closest to the centroid of the feature vectors of u (in Section 4 we examine whether maintaining the original representative affects the authentication accuracy).

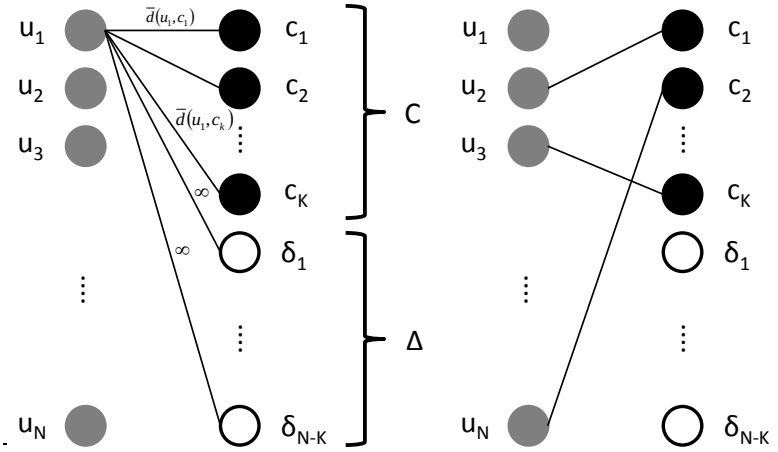

Figure 3: Graph construction for the Hungarian matching algorithm. Left: Bipartite graph construction for the Hungarian matching algorithm in case a user represents more than one cluster. Right: Matching result example - $u_{i_1} = u_2, u_{i_2} = u_N, u_{i_K} = u_3$.

3.2 The inner-cluster nearest-neighbor approach for choosing $R(u)$ (ICR)

In contrast to the cluster representative approach, in this approach $R(u)$ contains u and the users whose biometric profiles are the most similar to u's biometric profile. First, we find the cluster $k(u)$ whose centroid $c_{k(u)}$ is the closest to the centroid of u's feature vectors i.e.

$$k(u) = \arg\min_{i=1,\dots,K} \left\| c_i - \frac{1}{M}\sum_{j=1}^{M} s_{u,j} \right\|$$

Next, we find the $K-1$ users whose feature vectors' centroids are the closest to $c_{k(u)}$. Thus, $R(u)$ consists of u and the *K-1* users $v_1, \dots, v_{K-1}$ who achieve the lowest values of $\left\| c_{k(u)} - \frac{1}{M}\sum_{j=1}^{M} s_{v_i,j} \right\|$ where $i = 1, \dots, K-1$.

3.3 Random selection of representatives

A third method constructs $R(u)$ from *u* and *K-1* randomly selected users which differ from *u*. We refer to this method as the *random* representative selection method and it is used for comparison with the CR and ICR methods.

4. Experimental results

One of the major obstacles in the development of user authentication methods is obtaining data for performance evaluation. To date, no benchmark dataset consisting of hundreds of users is available. One of the known benchmark datasets which is available on-line was constructed by Killourhy and Maxion (2009a; 2009b). This dataset consists of 51 users who entered the phrase `'.tie5Roanl'` 400 times in 8 different sessions. However, this dataset does not meet our requirements since we needed a larger number of users to choose representatives from. Furthermore, we put emphasis on a small number of samples for each user to reduce the data collection burden on the users. Thus, we constructed a dataset, whose size is an order of a magnitude larger than most datasets that are used by current state-of-the-art methods.

The dataset was constructed in the following manner: *817* users were asked to enter the phrase `'password'` *10* times. The same keyboard was used for all password entries and the sampling resolution was in milliseconds. A feature vector describing the keystroke dynamics of each passphrase entry was constructed. The vector consisted of two parts: (a) the dwell times of the characters; and (b) the latencies between each pair of consecutive characters; the 'enter' key that was pressed at the end was also considered part of the phrase. The structure of the feature vector is illustrated in Fig. 4. Phrases whose entry included corrections (up to 3) were also included in the datatset (provided the final phrase was correct). We allowed corrections since they are quite common in password entry, however, we limited the number of corrections to 3 in order to avoid the contamination of the dataset with entries that were probably the result of either lack of concentration or interest on behalf of certain users. For example, if a user typed 'passq', pressed the backspace to erase the 'q' and then typed 'word' – this keystroke sequence was included in the dataset.

In order to construct the feature vectors, all the keys that were pressed by all the users were found including characters that were a result of a typo. The first part of the feature vector included an entry for the dwell time of each of the found characters. If a character was not pressed in a given phrase entry, its corresponding dwell time entry was set to zero. Next, all pairs of consecutive keys that were pressed were found – again, including typos. The second part of the feature vector included an entry for the latency of each pair of this kind. If a pair did not occur in the phrase entry, its corresponding latency was set to zero. If a character or a consecutive pair of characters occurred more than once in a given phrase entry, their corresponding dwell and latency in the feature vector were set to the average of their occurrence. Initially, the length of each feature vector was 169 (dwell times and latencies) indicating various typos throughout the dataset. However, if 'password' was entered without typos, only 9 dwell entries and 8 latencies were non zero. Consequently, we had to *clean* the dataset from abnormal entries. Specifically, we excluded users and their features vectors if one of their feature vectors included: (a) features that were non-zero in very few vectors; and (b) entry time longer than *5* seconds. The filtered dataset contained *783* users – each having *10* feature vectors where each vector was composed of at most *23* non-zero features (when there were no corrections, the size of the feature vector was 23 where 6 features contained zero).

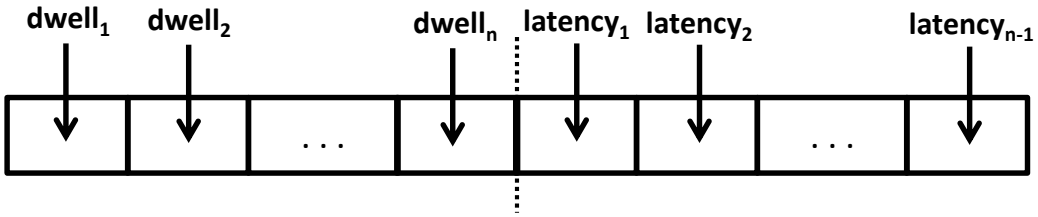

Figure 4: Structure of a feature vector

All experiments were performed using the WEKA (Frank et al. 2005) software package. The feature vectors of every user were divided into two *disjoint* sets of 5 feature vectors each. The methods were evaluated for 10 users that were randomly selected. The training set of each evaluated user was composed of 5 of her feature vectors and 5 feature vectors of each of her representatives. The test set was composed of: (a) the 5 remaining feature vectors of the user; (b) the remaining 5 feature vectors of each of the representatives; and (c) the 10 feature vectors of each of the non-representative users.

Three inducers were used where the emphasis was put on their simplicity: Naïve Bayes, nearest neighbor (WEKA's IBK with $K = 1$) and the AdaBoost ensemble using the C4.5 decision tree (J48 in WEKA) as the ensemble core inducer. We used the *k*-means algorithm to cluster the data. Various numbers of representatives were tested. For each randomly chosen user u, an inducer I and a number of clusters K - a classifier was constructed based on the training vectors of the users in $R(u)$.

The accuracy of the proposed methods was evaluated according to the area under the curve (AUC) criterion. In the following we evaluate various aspects of the proposed approach.

Experiment 1: Influence of the number of clusters/representatives on the accuracy

The underlying assumption of the proposed algorithms is that the keystroke dynamics of all the users can be characterized by the profiles of a small number of users (representatives) due to similarities in the keystroke behavior of the users. In order to examine this assumption, authentication models based on the CR, ICR and Random methods for representative selection were constructed using various numbers of clusters for each of the 10 tested users. The AdaBoost-C4.5 inducer was used for all models. Figure 5 shows that the number of clusters has statistically significant influence on the accuracy of the random representative selection method ($F(22, 720) = 5.3588$, $p < 1\%$). However, increasing the number of clusters mostly deteriorates the accuracy which may be accounted to either overfitting or local AUC maxima. It can be seen that for the CR method the same level of accuracy is obtained for 15 and 50 representatives indicating that increasing the number of representatives is not needed to obtain a certain level of accuracy. The same phenomenon can be observed in the ICR method where the best performance is obtained for 20 representatives.

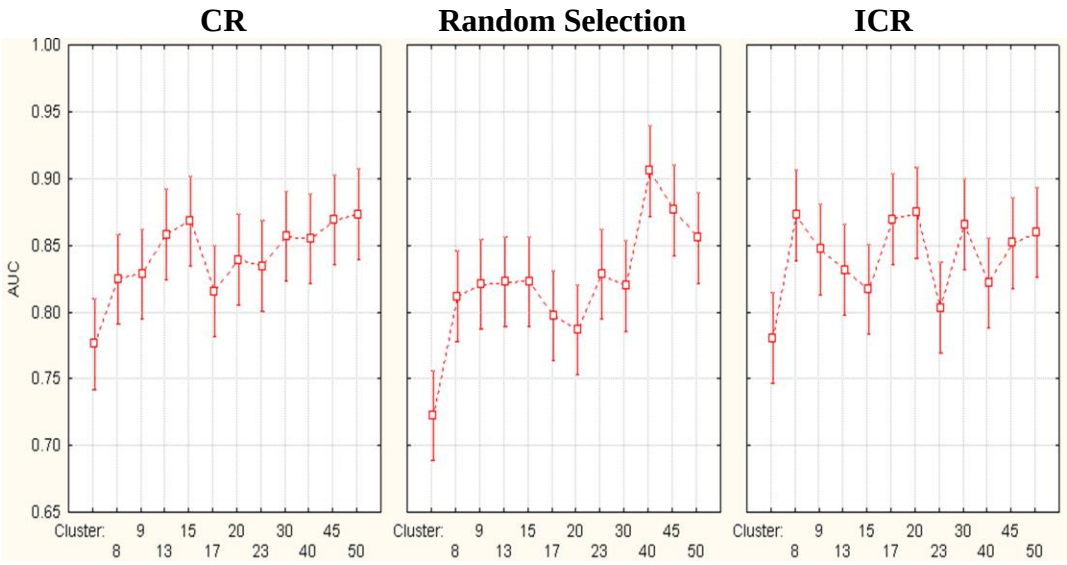

Figure 5: Impact of number of clusters on the cluster representative selection method when AdaBoost-C4.5 inducer is used. Current effect: F(22,720)=5.3588, p<1%. The vertical bars denote 0.95 confidence intervals. *Left*: The CR method. *Middle*: Random selection. *Right*: The ICR method.

Experiment 2: Influence of the number of clusters/representatives on the inducer

We also examined the impact that the number of clusters has on the inducers that were used. The CR representative selection method was used to construct the authentication models based on the nearest neighbor (IB1), the Naïve Bayes and the AdaBoost-C4.5 inducers. It can be seen in Fig. 6 that: (a) the nearest neighbor classifier (IBK) exhibits statistically significant inferiority to the other inducers; (b) the number of clusters has higher influence (manifested as bigger fluctuations) on the AdaBoost inducer than the other inducers; and (c) the AdaBoost accuracy increases with the number of clusters (substantial increase in the AUC was achieved when the number of clusters grew from 5 to 8 and from 23 to 40). Figure 7 provides a closer look on the results obtained by the AdaBoost-C4.5 inducer. The effect that the number of clusters has on the authentication accuracy of the CR method for one of the test users is shown in Fig. 8. The EER values are marked with circles and it can be seen that the best results are obtained for 50 clusters. The EER values for all the test users are summarized in Table 1. It can be seen that for 6 out of the 10 users the accuracy reduces when the number of representatives is increased indicating that choosing a higher number of representatives may result in

overfitting. The variation in EER values may be attributed to inconsistencies in the profiles of the chosen users.

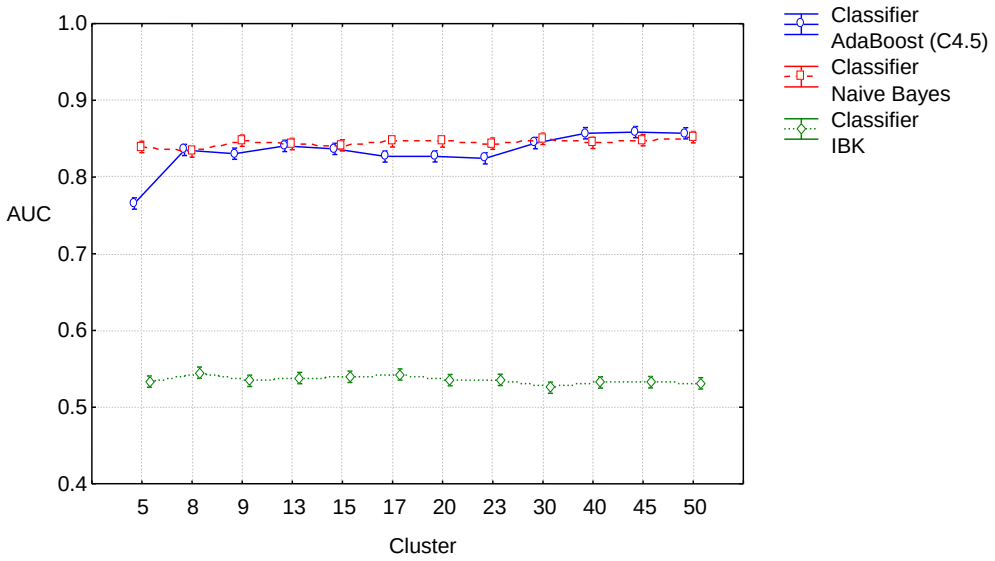

Figure 6: Impact of the number of clusters on the inducer type when the CR method is used. F(22, 3240) = 15.236, p = 0.

Experiment 3: Exclusion of the examined user's cluster representative

As described in Section 3.1, when constructing $R(u)$ using the CR approach, u's cluster representative $r_{k(u)}$ is replaced by u. We examined whether adding u to $R(u)$ without removing $r_{k(u)}$ affects the accuracy of the classifier. When $r_{k(u)}$ is not removed, the size of $R(u)$ is $K+1$. Figure 9 shows that removing $r_{k(u)}$ improves the results without statistical significance.

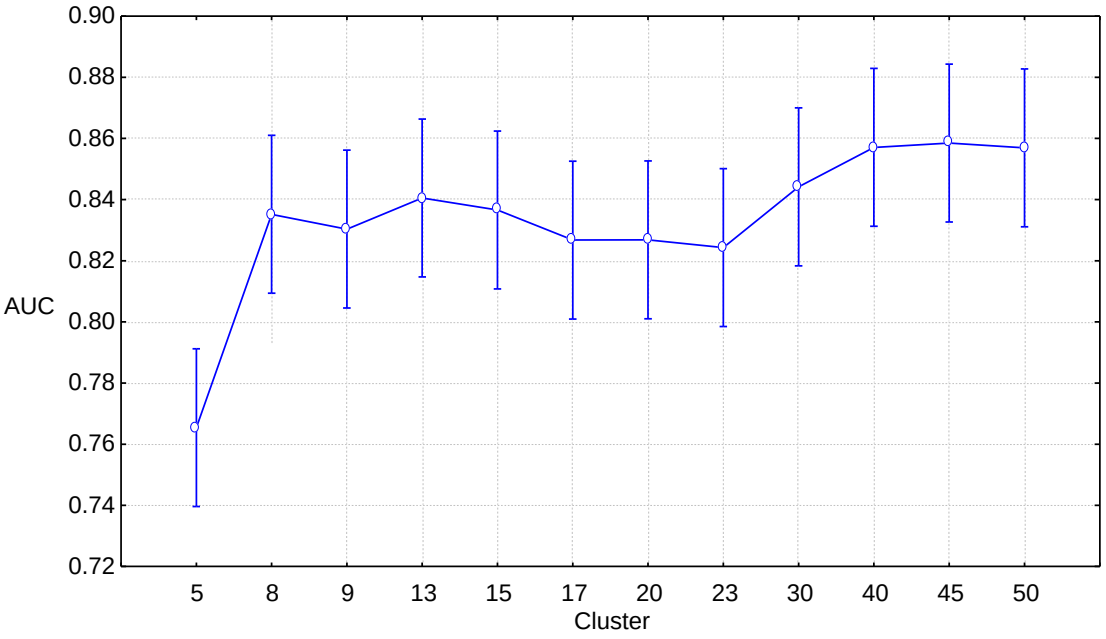

Figure 7: Effect of the number of clusters on the CR authentication accuracy when the AdaBoost-C4.5 inducer is used. F(11, 1296)=3.5304, p=0.00007. The vertical bars denote 0.95 confidence intervals.

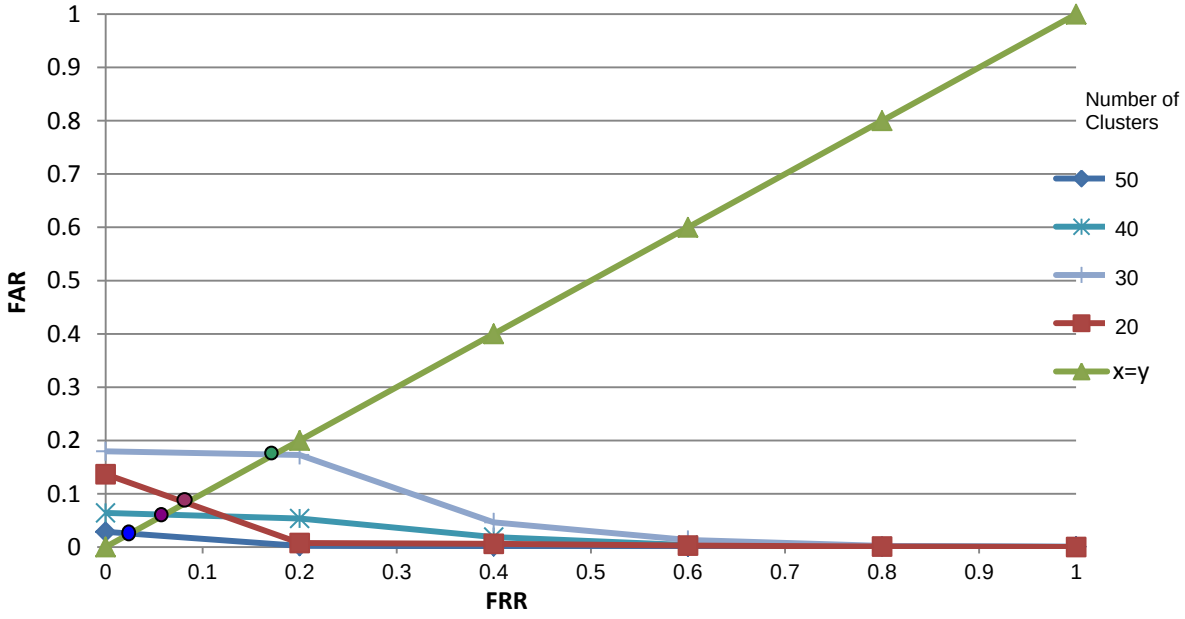

Figure 8: Effect of the number of clusters on the CR authentication accuracy for one of the test users. The EER values are circled.

Table 1: The EER values that were obtained by the CR method for the test users.

Users	Number of Clusters			
	20	**30**	**40**	**50**
U1	0.211	0.378	0.658	0.203
U2	0.083	0.179	0.063	0.025
U3	0.255	0.224	0.844	0.205
U4	0.121	***0.014***	0.0088	0.0089
U5	0.385	0.310	***0.226***	0.319
U6	0.021	0.0022	0.010	0.005
U7	0.179	0.197	***0.145***	0.322
U8	0.591	***0.047***	0.107	0.167
U9	***0.045***	0.151	0.158	0.164
U10	***0.027***	0.125	0.037	0.076

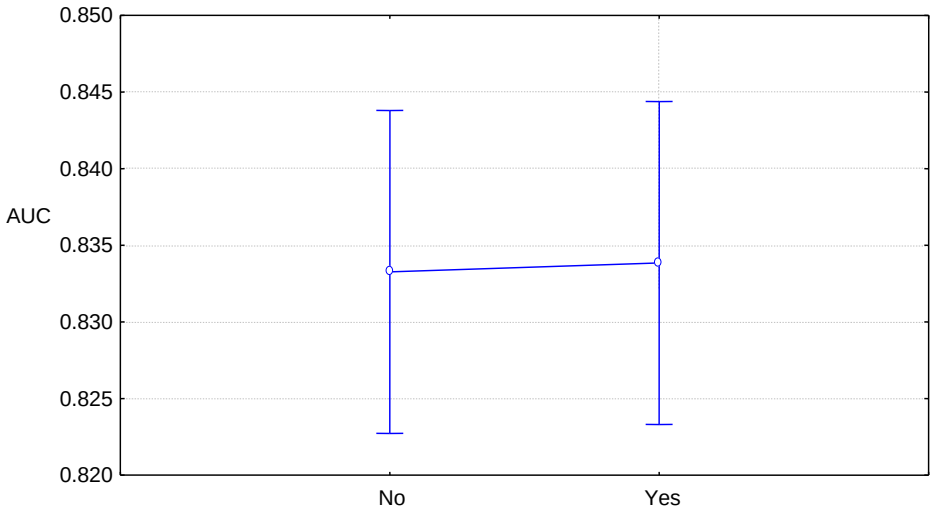

Figure 9: Influence of removing *u*'s cluster representative. F(1, 1296)=0.00587, p=0.93895. The vertical bars denote 0.95 confidence intervals.

Experiment 4: Influence of the human factor on the accuracy

The different manners by which users interact with the keyboard are utilized by BBASs to confirm their identity. On that note, we examined whether the authentication accuracy is influenced by the human factor i.e. by the typing manner of the test users. Figure 10 illustrates the authentication accuracy in term of the AUC criterion for the 10 examined users. It can be seen that the human factor is statistically significant. Namely, the

proposed method produces very accurate results for some users (such as in the case of the username *u4* which obtained AUC of 95%) while producing less accurate results for others e.g. the AUC of user *u1* is only 67%. This may be accounted for the different consistency levels in the typing manner of the users.

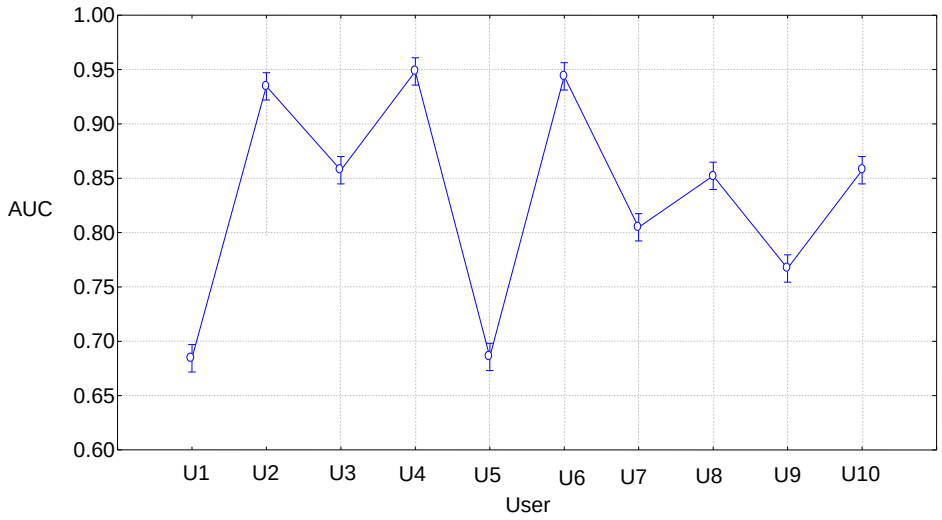

Figure 10: Influence of the human factor on the authentication accuracy. F(9, 720)=233.33, p=0. The vertical bars denote 0.95 confidence intervals.

Experiment 5: Accuracy comparison between internal and external attacks

Commonly, a distinction is made between users that belong to the organization (internal) and users that are external to the organization. Recall that (Section 1) the cluster representatives are considered as the internal users while the remaining users constitute the external ones. In this experiment we evaluated the accuracy of the proposed approach when applied to *only* external users. We compared the results with those obtained for only the internal users. In order to do so, two test sets were used. The first consisted of only the test feature vectors of the representative users while the second was composed of only the feature vectors of the non-representative users. The classifiers were constructed using the CR method and the AdaBoost-C4.5 inducer and the results are shown in Fig. 11. It can be seen that the authentication is *only* slightly less accurate when only applied to external users where the differences are statistically insignificant. This shows that the proposed method is equally effective for both external and internal attacks.

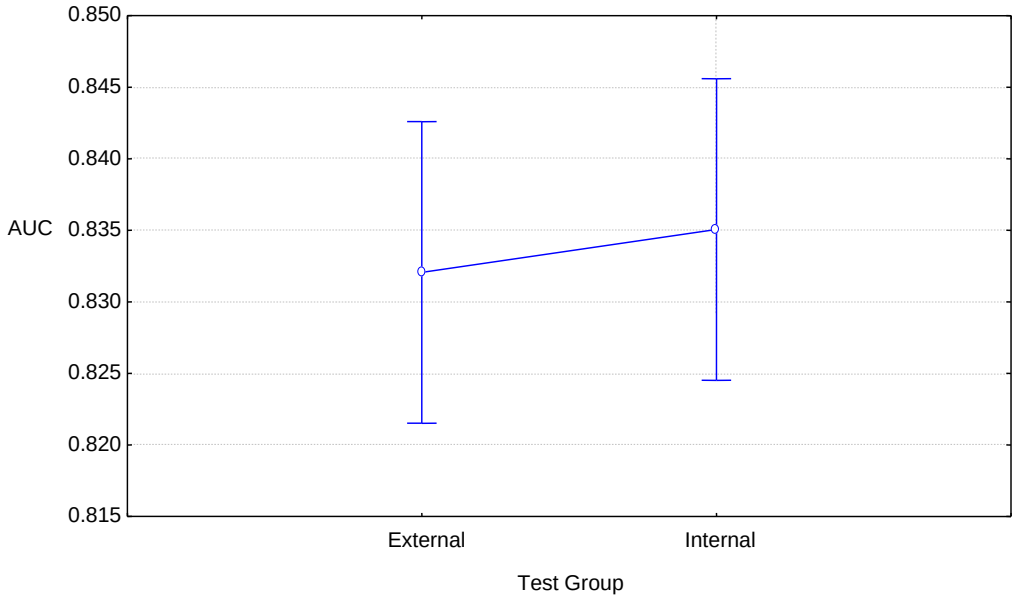

Figure 11: Authentication of external and internal users when the CR method and the AdaBoost-C4.5 inducer are used. Current effect: F(1, 1296)=0.15638, p=0.69258. The vertical bars denote 0.95 confidence intervals.

Experiment 6: Comparison between the representative selection methods

Figure 12 compares the overall performance of the different representative selection methods when the AdaBoost-C4.5 inducer is used. Both the CR and ICR methods produce results that are significantly better than randomly selection of the representatives where CR is slightly better than ICR. In Fig. 13 the results for one of the test users is displayed when 30 representatives are selected. In this case, the CR method is better than both the ICR and the Random methods.

Table 2 summarizes the average AUC results for (a) the inducers; (b) the representative selection methods; (c) the number of clusters; and (d) the test groups. The results indicate that the CR method is better than the other two methods when using the Naïve Bayes and AdaBoost-C4.5 inducers. Furthermore, in 9 out of the 18 cases, the best accuracy is not achieved by the highest number of representatives which may be attributed to overfitting due to a high number of representatives.

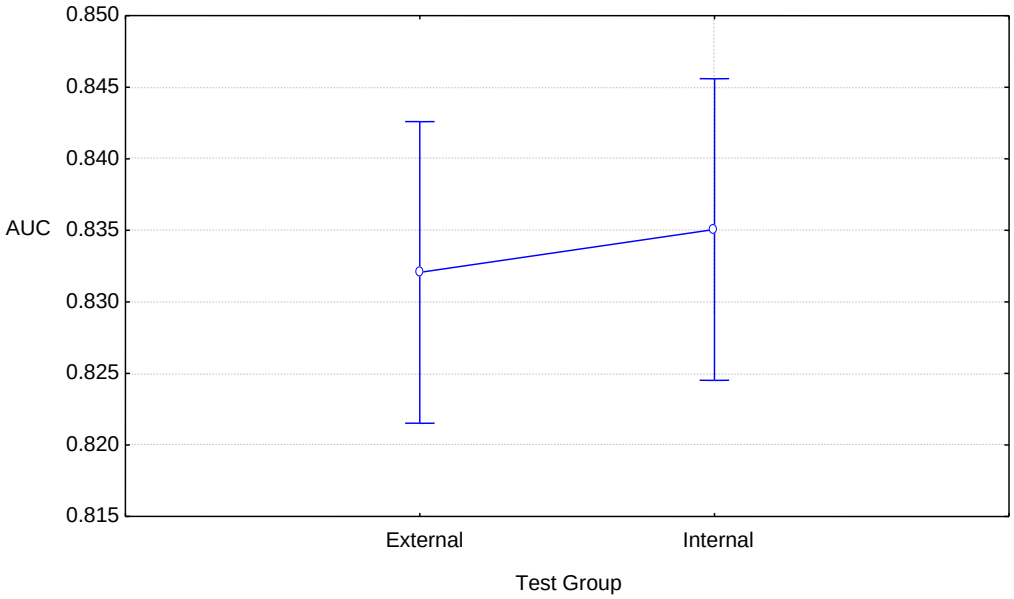

Figure 12: Overall results for the selection methods Cluster representatives (CR), Inner-Cluster representatives (ICR) and Random (R). F(2, 720) = 8.7312, p = 0,00018. The vertical bars denote 0.95 confidence intervals.

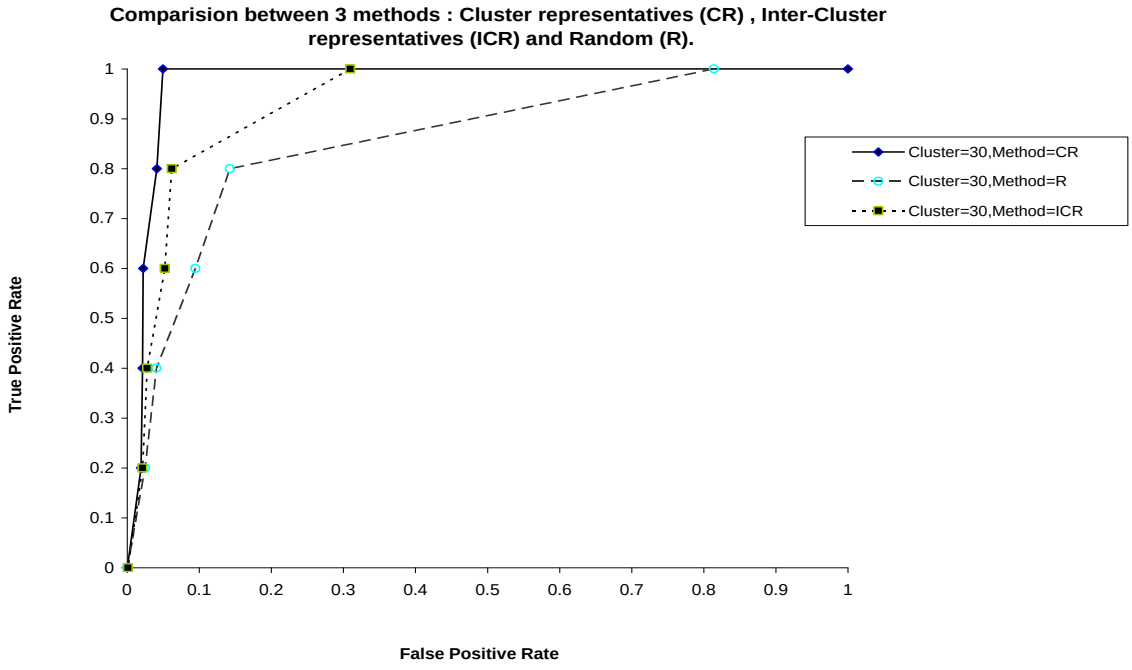

Figure 13: TPR and FPR results achieved by the selection methods for one of the tested users cluster representatives (CR), Inner-Cluster representatives (ICR) and Random (R). The CR method is superior to both the ICR and Random methods.

We conclude this section with Table 3 comparing the CR method (using AdaBoost) to currently available state-of-the-art methods. These results were evaluated for the inspected users.

Table 2: Area under the curve (AUC) averages. The best results are shaded.
CR - Cluster representatives (Section 3.1)**,**
ICR - inner-cluster nearest-neighbor representatives (Section 3.2)**,**
R - Random representatives.

Test Group								
Internal				External				
Method			Cluster	Method			Cluster	Classifiers
R	ICR	CR		R	ICR	CR		
78.65%	**87.44%**	83.93%	20	78.40%	83.90%	83.79%	20	**AdaBoost (C4.5)**
81.94%	86.58%	85.67%	30	81.84%	**85.05%**	85.44%	30	
90.57%	82.18%	85.49%	40	**90.40%**	80.33%	85.30%	40	
85.55%	85.95%	**87.33%**	50	85.37%	82.84%	**87.13%**	50	
53.53%	51.77%	**54.72%**	20	53.54%	52.79%	**54.73%**	20	**IBK**
53.42%	51.84%	51.86%	30	53.34%	52.84%	51.87%	30	
55.43%	51.87%	51.89%	40	55.37%	**52.86%**	51.91%	40	
55.46%	**51.93%**	51.90%	50	**55.47%**	51.92%	51.91%	50	
84.97%	83.15%	85.44%	20	84.68%	84.42%	85.16%	20	**Naïve Bayes**
84.60%	**85.19%**	85.69%	30	84.23%	84.77%	85.42%	30	
83.80%	84.08%	85.77%	40	83.38%	84.19%	85.47%	40	
85.32%	84.52%	**85.92%**	50	**84.90%**	**84.91**%	**85.61%**	50	

Table 3: Comparison of the CR method to other methods.

Method	FRR	FAR	Accuracy	EER
Bleha et al. (1999)	3.1%	0.5%	-	-
Revett et al. (2005)	-	-	97%	-
Cho et al. (2000)	1%	0%	-	-
Killourhy and Maxion, (2009b)	-	-	-	9.32%
The CR method – AdaBoost – 40 clusters	4.3%	0.4%	97.2%	14.9%

5. Conclusions and future work

We presented a novel approach for authentication of users at login according to the biometric characteristics of the password entry. A classifier is tailored to each user where the novelty lays in the way the training sets are constructed for each tested user. Namely,

only the feature vectors of a small subset of the users constitutes the training set of each user.

Choosing a small training set reduces the possibility of overfitting while allowing scalability to a large volume of users. We introduced the CR and ICR strategies for selecting the representatives. The CR strategy chooses the users whose biometric profiles govern the biometric profiles of all the users while the ICR strategy chooses the users whose biometric profiles are the most similar to the biometric profile of the examined user. Both methods employ clustering to the session data in order to find inter-user profile similarities. Both methods are superior to a simple random selection of representatives.

The results obtained in this paper show that constructing the training set using only a small set of representative users is promising. Other selection methods should be sought after and other inducers should be examined in order to further improve the results. Where possible, rigorous justification should be provided in order to theoretically corroborate the proposed methods. Additionally, choosing the number of representatives that produces the best results is still an open problem which is currently being investigated by the authors.

A common problem in user authentication is the acquisition of data for the evaluation of the proposed methods. Since the proposed approach selects representative users, a dataset with a large enough number of users was required. Unfortunately, no benchmark dataset that met our requirements was available and we had to construct our own dataset containing over 800 users. The absence of benchmark datasets makes it difficult to compare between methods since each method may have different requirements.

Furthermore, many authentication systems, e.g. in commercial websites, handle a large number (10^3-10^6) of users (Peacock et al. 2004). In the approach proposed in this paper, only a small subset of users in used to build the authentication model while the model may be used to authenticate a user among a much higher number of users. In our experiments, the largest number of representatives that was used to construct a model was less than 7% of the total number of users and, in many cases, the model that achieved the

best results contained less than 5% of the total number of users. These results indicate the ability to scale to a high volume of users. In order to corroborate this, the proposed method should be evaluated using a dataset containing 10^3-10^6 users. However, collecting keystroke dynamics from such a large number of users is a difficult task (Peacock et al. 2004). In should be noted that the proposed approach was tested using a single computer. Another challenge is to adapt it to handle multiple keyboards and remote connections where delays due to the remote connections are one of the main issues that need to be addressed.

Finally, the authors are currently investigating ways to update the classification models given new collected data - an approach which was employed by Monrose et al. (1999).

References

[1] Bleha, S., Slivinsky, C., & Hussein, B., (1999). Computer-access security systems using keystroke dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12(12), pp.1217–1222.

[2] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1993). Classification and Regression Trees. Chapman & Hall, Inc., New York.

[3] El-Saddik, A., Orozco, M., Asfaw, Y., Shirmohammadi, S. , & Adler, A. (2007). A novel biometric system for identification and verification of haptic users. *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, pp. 895–906.

[4] Frank, E., Hall, M. A., Holmes, G., Kirkby, R., Pfahringer, B., & Witten, I. H., (2005). Weka: A machine learning workbench for data mining. In O. Maimon and L. Rokach (Eds), Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, pp. 1305–1314, Springer.

[5] Grabham, N. J., & White, N. M., (2007). Validation of keypad user identity using a novel biometric technique. *Journal of Physics*: Conference Series, Vol. 76(1) 012023.

[6] Harold, W. K. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, Vol. 2, pp. 83–97.

[7] Hidetosshi, N. & Kurihara, M. (2004). Sensing pressure for authentication system. International Conference on Computational Intelligence, (ICCI 2004), December 17–19, Istanbul, Turkey.

[8] Hosseinzadeh, D., & Krishnan, S. (2008). Gaussian Mixture Modeling of Keystroke Patterns for Biometric Applications, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 38(6), pp. 816–826.

[9] Jain, A., Bolle, R., & Panakanti, S. (Eds.), (1999). Biometrics: Personal Identification in Networked Society, Kluwer Academic.

[10] Killourhy, K. S. & Maxion, R. A. (2009a). Comparing Anomaly Detectors for Keystroke Dynamics, In *Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN-2009)*, pp. 125–134, Estoril, Lisbon, Portugal, June 29-July 2.

[11] Killourhy, K. S. & Maxion, R. A. (2009b). http://www.cs.cmu.edu/~keystroke/#ref1

[12] Modi, S., & Elliott, S. J. (2006). Keystroke dynamics verification using a spontaneously generated password. *Proceedings 40th Annual 2006 International Carnahan Conference on Security Technology*, pp. 116–121, Lexington, Kentucky.

[13] Monrose, F., Reiter, M. K., & Wetzel, S. (1999). Password Hardening Based on Keystroke Dynamics. *International Journal of Information Security*, pp. 73–82.

[14] Monrose, F., & Rubin, A. (1997). Authentication via keystroke dynamics. In: *Proceedings of the Fourth ACM Conference on Computer and Communications Security*, pp. 48–56, Zurich, Switzerland, 2–4 April.

[15] Obaidat, M. S., & Sadoun, B. (2000). Keystroke Dynamics Based Authentication In *Biometrics. Personal Identification in Networked Society* (Jain, A., Bolle, R., & Pankanti, S., Eds.), pp. 213–229, Kluwer Academic Publishers.

[16] Peacock, A., Ke, X., & Wilkerson, M. (2004). Typing Patterns: A key to User Identification, *IEEE Security and Privacy*, Vol. 2(5), pp. 40–47.

[17] Pierscionek, B., Crawford, S., & Scotney, B. (2008). Iris recognition and ocular biometrics - the salient features. *International Machine Vision and Image Processing Conference IMVIP'08*, IEEE Computer Society, pp. 170-175, 3-5 September.

[18] Revett, K., Magalhães, P. S., & Santos, H. M. D. (2005). Developing a keystroke dynamics based agent using rough sets. *In 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, University of Technology of Compiegne.

[19] Vapnik, V. N. (2000). The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA.

[20] Wu, X., Zhang, D. and Wang, K. (2006). Palm line extraction and matching for personal authentication, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 36(5), pp. 978–987.

[21] Yong, S., Phoha, V. V., and Rovnyak, S.M. (2005). A parallel decision tree-based method for user authentication based on keystroke patterns, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 35(4), pp. 826–833.