STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LIX**, Special Issue 1, 2014
10th Joint Conference on Mathematics and Computer Science, Cluj-Napoca, May 21-25, 2014

IDENTITY INFORMATION REVEALED FROM MOBILE TOUCH GESTURES

MARGIT ANTAL, LÁSZLÓ ZSOLT SZABÓ, AND ZSOLT BOKOR

Abstract. Due to the powerful sensors incorporated, the new generations of smartphones have become capable of many sophisticated biometrics. Touchscreen based biometrics is a new type of biometrics showing great potential. In this paper we review the studies already conducted in this direction, then present our study aimed to find the best method for touch data based identification. We collected a large touch dataset from 71 users using 8 different mobile devices: tablets and phones. Touch data were divided in strokes and several classification schemes like k-NN, Random Forests and SVM were investigated on this dataset. Measurements show that several strokes are required for accurate user identification. Besides different classification results, statistical analysis of the collected data is presented.

1. Introduction

Biometrics technology is now increasingly adopted in a wide range of security systems. These systems identify users by their measurable human characteristics like fingerprint, hand geometry, voice, signature, keystroke dynamics, gait and s.o. Biometrics systems are divided into identification and verification systems. An identification system determines a person's identity by performing one-to-many comparison against a biometric dataset. Identification systems are widely used in criminal or frequent customers tracking. A verification or authentication system performs one-to-one comparison and matches the claimed identity against the current biometrics sample. Verification systems are used for access control for doors, cars, computers, mobile devices, ATMs and s.o. The development of pattern recognition significantly influenced biometrics research. Several types of biometrics have been intensively studied like fingerprint [13], face [15] , voice [11], palmprint [12], keystroke dynamics [2] [10] [14], gait [9].

Received by the editors: May 1, 2014.
2010 *Mathematics Subject Classification.* 68T10.
1998 *CR Categories and Descriptors.* H5.2 [**Information Interfaces and Presentation**]: Subtopic – *User Inter-faces Input devices and strategies, evaluation*.
Key words and phrases. Security, Biometrics, Identification.

Most desktop computers and mobile devices offer only entry-point based authentication schemes which usually require providing a username/password combination. However, after the authentication step has been carried out, the device could be used by other unauthorized users. If software were able to continuously track the identity of the user, this abnormal usage pattern would be impossible. Continuous keystroke dynamics (typing rhythm) [3] is a proposal to solve such usage anomalies. Unfortunately typing rhythm based continuous authentication schemes are not suitable for smartphones, hence mobile devices, compared to desktop computers, are more vulnerable concerning their security. Touch-based biometrics is proposed instead of keystroke dynamics in a few recent studies. Damopoulos et. al. [5] suggested a touchlogger software to detect the owner's usage patterns and to block unauthorized access to the device. The paper presents classification measurements on a non-public dataset containing data from 18 users. [1] and [6] results show that adding biometric touch information to screen unlock patterns can enhance the security of the method. Only one study [7] presents results about the amount and type of touch data necessary for high accuracy authentication.

The main objective of our research is to study whether vertical and horizontal scrolling touch data (strokes) can be used for user identification. A second objective is to determine the amount of touch scrolling data necessary for accurate identification.

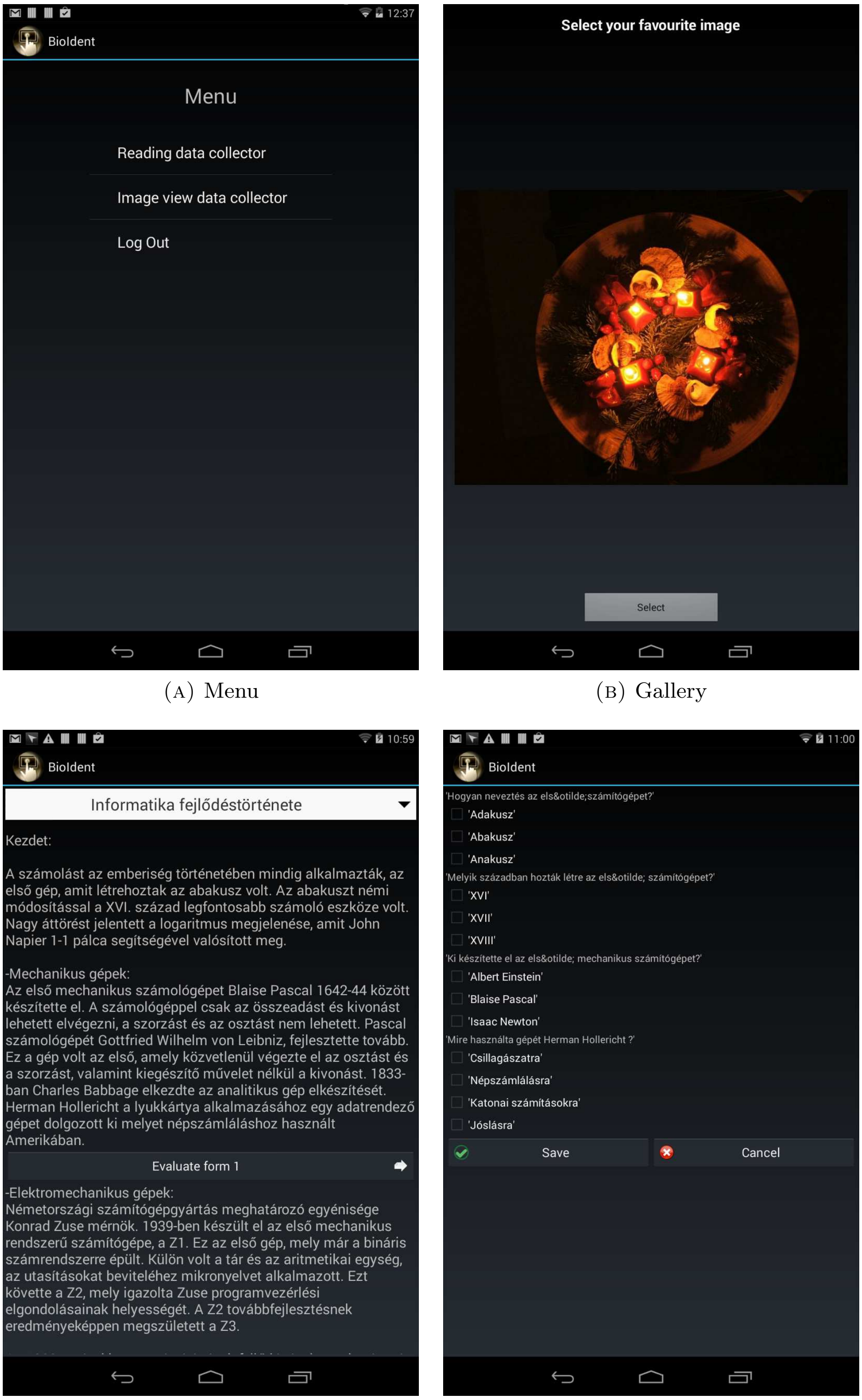
2. Methods

In order to study the suitability of vertical and horizontal strokes for user identification we designed a data collection application. Data were collected in four sessions from several Android phones and sent to a data server. Afterwards data analysis and feature extraction was performed, followed by single and multiple stroke classification experiments.

2.1. **Data acquisition.** For data acquisition a client-server application was developed. The Android client presents to users two types of tasks which require scrolling interactions. One type of task is a reading task, and the other type is an image gallery. While the reading task requires vertical scrolling, the image gallery task requires horizontal scrolling. In the first task users have to read a text and answer to some questions regarding text comprehension. The questions are inserted after each paragraph as shown in figure 1.c. Questions are shown to the user by pressing the numbered buttons inserted into the text. Figure 1.d shows an example of quiz. The texts are uploaded by the administrator to the server application and after the login operation Android clients always synchronize with the server and download the new texts. For the second task the images were uploaded to the server application side and were grouped into albums. Android clients synchronize with the server and download only the newest album. In this case the task was to select the favorite picture. Data were sent to the server continuously during the two tasks. Data collection was performed during 4 weeks, for each week a new text comprehension task and a new album were added. 71 users, 56 male and 15 female participated in this study. The average age of the participants was 29.8 with a standard deviation of 9.08. The youngest was 19 and the oldest one was 47 years old. 8 different Android devices were used having resolutions from 320x480 to 1080x1205 pixels. Each participant provided the data in multiple sessions, sometimes from multiple devices. In the registration phase several extra information were required from the participants such as the gender, the birth date and the user experience level using touchscreen. Device information such as Android version, resolution were also stored. The user experience level was quantized into 4 classes: unexperienced, moderate experienced, experienced and very experienced stated by the user.

2.2. **Data cleansing.** Data acquisition was followed by data cleansing. In this process sequences of pixels were divided into strokes, a procedure which provoked several problems to solve. First of all, several users did not rise their finger between consecutive strokes. This resulted in compound strokes which had to be divided into individual strokes using pressure and time information. Secondly, one device used in data acquisition turned out to be inappropriate due to its incapacity to return the accurate pressure value. This device always returned the value 1, consequently all the strokes collected through this device were excluded from the dataset. Finally, short strokes were detected, containing no more than three pixels. These strokes were also excluded from the final dataset. After cleaning the data remained 14316 strokes (11584 horizontal and 2732 vertical), which means that we had on average 200 strokes/user.

2.3. **Feature extraction.** The collected raw data were divided into strokes. Each stroke consists of a sequence of touchscreen points between a starting and stopping point. The k.th stroke is defined as $v^k = (x_i^k, y_i^k, vx_i^k, vy_i^k, t_i^k, p_i^k, A_i^k, o_i^k)$, $i \in 1, 2, \ldots, N^k$, where x_i^k, y_i^k are the coordinates of the touching position, vx_i^k, vy_i^k are the horizontal and vertical velocities, t_i^k is the timestamp, p_i^k the pressure, A_i^k the area covered by the finger, o_i^k the orientation of the device (horizontal or vertical) and N^k denotes the number of points belonging to the stroke. From each stroke one feature vector was extracted. We used a similar terminology as in [7], but used fewer features. The elements of our feature vector are the following: (1)*stroke duration*: the time needed for a stroke expressed in milliseconds; (2)*start x*: the x coordinate of the stroke starting point; (3)*start y*: the y coordinate of the stroke starting point; (4)*stop x*: the x coordinate of the stroke ending point; (5)*stop y*: the y coordinate of the stroke ending point; (6)$direct\ end-to-end\ distance$: the length of the segment



(A) Menu (B) Gallery

(C) Reading (D) Questions

FIGURE 1. The Android client

defined by the two endpoints; (7)*mean resultant length*: a feature characterizing the straightness of the stroke [7]; (8)$up\ down\ left\ right\ flag$: orientation of the stroke; (9)$direction\ of\ end-to-end\ line$: the slope of the segment defined by the two endpoints; (10)$largest\ deviation\ from\ end{-}to{-}end\ line$: the maximum of the distances between points belonging to the stroke and the segment defined by the two endpoints; (11)*average direction*: the average slope of the segments belonging to the stroke trajectory; (12)$length\ of\ trajectory$: the length of the stroke; (13)*average velocity*: the average velocity of the stroke; (14)$mid-stroke\ pressure$: the pressure at the midpoint of the stroke; (15)$mid\ -\ stroke\ area\ covered$: the area covered by finger at the midpoint of the stroke; Additional information were available for each stroke such as information regarding the device, the task in which the stroke was produced and information regarding the user who produced the stroke. Some of these additional information were used only as attributes for labeling the class in different classifications.

2.4. **Classification.** For classification we used k-NN, SVM (SMO algorithm) and Random forests algorithms. k nearest neighbors (k-NN) is an instance based classification algorithm where a new instance label is decided by the k closest neighbors. We tested the algorithm for several odd values for parameter k and report the best accuracy. We always mention the value of parameter k which produce the best accuracy.

Support vector machines (SVM) build a linear discriminant function that separates the instances of classes. If no linear separation is possible, a kernel is applied which maps the instances into a high-dimensional feature space. In this paper we experimented with the following kernels offered by the Weka Machine learning toolkit: polynomial kernel, normalized polynomial kernel, the Puk kernel (Pearson VII function based universal kernel, [16]), and RBF kernel. Moreover, we optimized the parameters of these kernels. PUK kernel produced the best results for our data, so we report accuracies obtained using this type of kernel.

Random forests [4] classifier builds up a number of decision trees following specific rules for tree growing, tree combination, self-testing and post-processing. Among the benefits of Random forests classifier we mention the following: it can be used for both regression and classification, there is no need for prior feature selection and the training phase is fast.

3. Results

3.1. **Classification results.** For classification we used a Java program based on Weka API [8]. All measurements were made using 3-fold cross-validation. For multiple stroke classification the evaluation phase consisted of computing the prediction distribution for each stroke belonging to the stroke sequence to

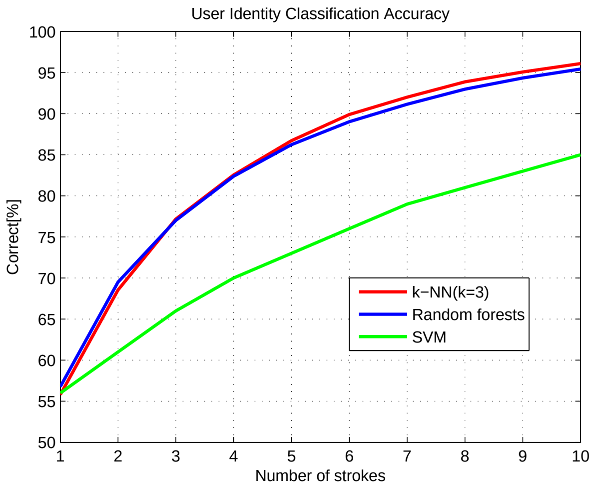


FIGURE 2. Classification of multiple strokes

be classified. The prediction distribution is an array of probabilities having N elements, where N is the number of classes.

Let us denote X the sequence of strokes to be classified:

$$X = \{x_1, x_2, \dots, x_T\}, \qquad x_i \in R^D,$$

where T is the number of strokes and D is the number of features. We obtained for each stroke the prediction distribution

$$P_i = \{p_i^1, p_i^2, \dots, p_i^N\}, p_i^k \in [0,1], k = 1 \dots N, i = 1 \dots T.$$

We computed the average probability for each class and chose the maximum one. Consequently, a sequence of strokes was classified belonging to the k^{th} class if the average probability for this class was the maximum one.

Figure 2 shows that for our 71 users dataset the k-NN and the Random forests algorithms provide almost the same accuracy starting from 56% for one stroke and ending in more than 95% for 10 strokes. This means that accurate identification cannot be made using only one stroke, but requires several strokes. The number of mistakes made by Random forests algorithm are illustrated in figure 3 with the confusion matrices obtained for one, three, five and ten strokes.

3.2. **The best features for user identification.** Feature selection is the process of selecting a subset of relevant features in order to reduce the time to build a classifier and to improve the classification rate.

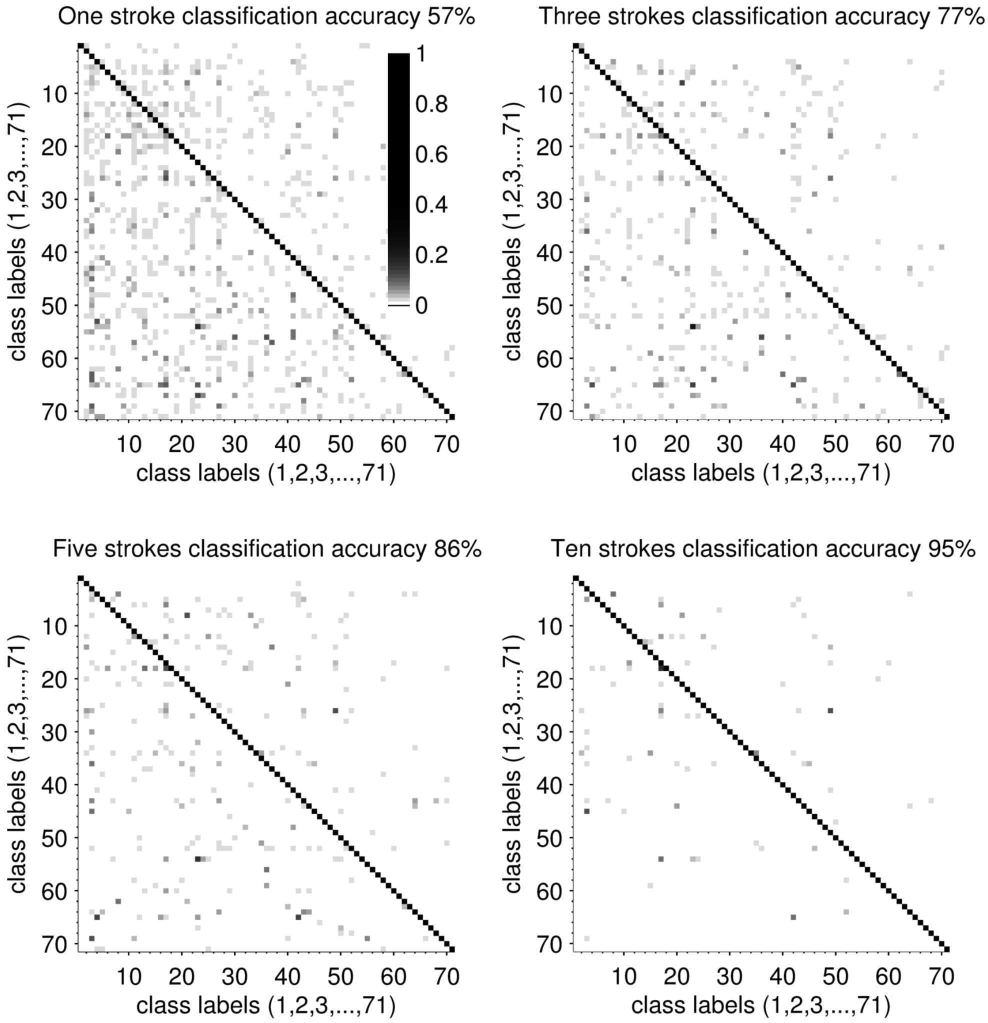


FIGURE 3. Classification of multiple strokes

Not every classifier benefits from feature selection, e.g. the decision tree based classifiers C4.5 and Random forests. These classifiers are capable of selecting and using the most relevant features.

Weka is also capable of performing feature selection (Select attributes). If we combine an attribute evaluator with a Ranker search method, we obtain the ranking of the features, i.e. feature relevance with respect to the class. In order to make our result comparable to Frank et. al. [7] result, we used GainRatioAttributeEval evaluator. This specific evaluator computes a value

Table 1. Feature ranking

Gain ratio	Feature
0.307	mid-stroke-area-covered
0.229	mid-stroke-pressure
0.186	direction-of-end-to-end-line
0.175	direct-end-to-end-distance
0.169	length-of-trajectory
0.158	average-velocity
0.157	start-y
0.143	stop-y
0.135	average-direction
0.133	stroke-duration
0.132	largest-deviation-from-end-to-end
0.132	mean-resultant-length
0.129	stop-x
0.124	start-x
0.116	up-down-left-right

between 0 and 1 for each feature (gain ratio with respect to class [17]). The larger this value is, the more it determines the user.

Using the GainRatioAttributeEval evaluator resulted in the ranking shown in Table 1. We should note that our results resemble those of Frank et. al., who found our best 3 features (mid-stroke-are-covered, mid-stroke-pressure and direction-of-end-to-end-line) to be among their best 4 features in this order.

Figure 4 shows the histograms for the best 4 features for user identity classification.

4. Conclusions

We designed an experiment for collecting touch data on Android devices in multiple sessions. Touch data were divided into strokes, then user classification measurements were performed using single and multiple strokes. Measurements show that several strokes are necessary for an accurate user identification. We obtained over 95% accuracy for 10 strokes in the case of k-NN and Random forests algorithm. Besides the classification experiments we used feature ranking in order to find the best features for user identification. We can conclude that for our dataset the most discriminative features are finger area and pressure, both measured at the middle of the stroke, direction of the line between the two endpoints of the stroke, and the length of the line between the two endpoints of the stroke. This is largely identical

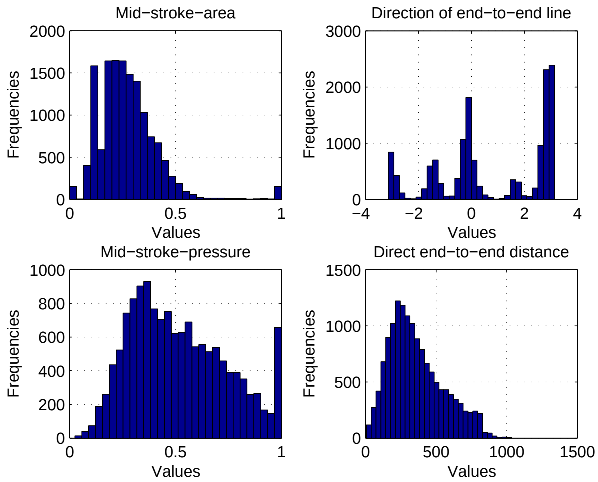


FIGURE 4. Histograms for the best features

to the feature ranking obtained by Frank et. al. in their study [7]. Based on the classification results we can state that this stroke based method can be used for continuous authentication in mobile applications where the basic navigation mechanism is similar. However, other types of touch interactions should be investigated in order to offer a general continuous authentication method for mobile devices. The fact that we investigated only vertical and horizontal scrolling can be considered a major limitation of our study.

5. Acknowledgment

This research has been supported by Sapientia-Institute for Research Programmes.

References

[1] J. Angulo, E. Wastlund, *Exploring Touch-Screen Biometrics for User Identification on Smart Phones*, Privacy and Identity Management for Life. IFIP Advances in Information and Communication Technology Volume 375, 2012, pp. 130–143.
[2] F. Bergadano, D. Gunetti, and C. Picardi, *User authentication through keystroke dynamics*, ACM Trans. Inf. Syst. Secur. 5, 4 (2002), pp. 367-397.
[3] P. Bours, *Continuous keystroke dynamics: A different perspective towards biometric evaluation* Inf. Secur. Tech. Rep. 17, 1-2 (2012), pp. 36–43.
[4] L. Breiman, *Random forests*, Machine Learning 45(1) (2001), pp. 5-32.
[5] D. Damopoulos, G. Kambourakis, S. Gritzalis, *From keyloggers to touchloggers: Take the rough with the smooth*, Computers & Security, 32 (2013), pp. 102–114.

[6] A. De Luca, A. Hang, F. Brudy, C. Lindner, H. Hussmann. *Touch me once and I know it's you!: implicit authentication based on touch screen patterns*, in Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems, ser. CHI'12. New York, NY, USA: ACM, 2012, pp. 987–996.

[7] M. Frank, R. Biedert, I. Martinovic, D. Song, *Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication*, IEEE Transactions on Information Forensics and Security, 8(1) 2013), pp. 136–148.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H., Witten, *The WEKA data mining software: an update*, SIGKDD Explorations Newsletter, 11(1) (2009), pp. 10–18.

[9] J. Han, B. Bhanu, *Individual Recognition Using Gait Energy Image*, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2) (2006), pp. 316-322.

[10] K. S. Killourhy, R. A. Maxion, *Comparing anomaly-detection algorithms for keystroke dynamics*, IEEE/IFIP International Conference on Dependable Systems and Networks, Estoril, Lisbon, Portugal (2009),pp. 125134.

[11] T. Kinnunen, H. Li, textitAn overview of text-independent speaker recognition: From features to supervectors, Speech Communication, 52(1) (2010), pp. 12–40.

[12] A. Kong, D. Zhang, and M. Kamel. *A survey of palmprint recognition*, Pattern Recognition, 42, 7,(2009), pp. 1408-1418.

[13] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer-Verlag London Limited, 2009.

[14] F. Monrose, A. Rubin, *Authentication via keystroke dynamics*, Proceedings of the 4th ACM conference on Computer and communications security, Zurich, Switzerland (1997), pp. 48–56.

[15] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, *Face recognition: A literature survey*, ACM Comput. Surv. 35 (4) (2003), pp. 399-458.

[16] B. Uestuen, W. J. Melssen, L.M.C. Buydens, *Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel*, Chemometrics and Intelligent Laboratory Systems, 81 (2006), pp. 29-40.

[17] I. H. Witten, E. Frank, M. Hall, *Data Mining: Practical machine learning tools and techniques*,Morgan Kaufmann Publishers , 2011.

Sapientia – Hungarian University of Transylvania, Faculty of Technical and Human Sciences, 540053 Tîrgu-Mureş, Romania

E-mail address: manyi@ms.sapientia.ro

E-mail address: lszabo@ms.sapientia.ro

E-mail address: zsoly91@gmail.com