Keystroke Dynamics and Finger Knuckle Imaging Fusion for Continuous User Verification

Tomasz Emanuel Wesolowski, Rafal Doroz, Krzysztof Wrobel, Hossein Safaverdi

Keystroke Dynamics and Finger Knuckle Imaging Fusion for Continuous User Verification

Tomasz Emanuel Wesołowski[1], Rafal Doroz[1], Krzysztof Wrobel[1], and Hossein Safaverdi[1]

[1]Institute of Computer Science, University of Silesia, Katowice, Poland,
{tomasz.wesolowski, rafal.doroz, krzysztof.wrobel,
hossein.safaverdi}@us.edu.pl,

Abstract. The paper presents a novel user identity verification method based on fusion of keystroke dynamics and knuckle images analysis. In our solution the verification is performed by an ensemble of classifiers used to verify the identity of an active user. A proposed verification module works on a database which comprises of data representing keystroke dynamics and knuckle images. The usability of the introduced approach was tested experimentally. The obtained results confirm that the proposed fusion method gives better results than the use of a single biometric feature only. For this reason our method can be used for increasing a protection level of computer resources against impostors. The paper presents preliminary research conducted to assess the potential of biometric methods fusion.

Keywords: biometrics, keystroke dynamics, finger knuckle, user verification

1 Introduction

Increasing computer systems security is a crucial task in the world dominated by electronically stored personal data and sensitive information. The number of attacks is increasing year by year. Only within the years 2014 and 2015 the amount of individuals affected by security breaches, where sensitive personal data such as electronic health records were stolen, increased hundred times [17]. The attacks themselves are becoming more and more sophisticated. There is various kinds of cyber attacks therefore different cyber attack detection strategies have to be developed [11]. Attacks can come from outside of the computer system but a big part of intrusions consists of insider attacks [13]. Hence the requirement for novel security measures is very high. As the main goal of biometrics is the automatic recognition of individuals based on the knowledge of their physical or behavioral characteristics, biometric methods are commonly used in IT security systems because of their high effectiveness.

Behavioral biometric methods use, among other things, an analysis of the movements of various manipulators (eg. a computer mouse [16]) or the dynamics of typing on a computer keyboard (keystroke dynamics) [1,2,12,15]. An analysis

of keystroke dynamics involves detection of a rhythm and habits of a computer user while typing on a keyboard [18]. As the result of such an analysis, a user profile is obtained that can then be used in the access authorization systems. In our approach the registration of the user activity while working with a keyboard is performed automatically and continuously in the background, without additionally involving a user. The data are captured on the fly and saved in text files on the ongoing basis. Based on these text logs keystroke dynamics analysis is performed in order to verify an active user's access permissions. The big advantage of the proposed method is that user verification can be performed continuously on the fly. To increase a protection level the proposed in this paper approach combines the keystroke dynamics with another biometric method based on finger knuckle pattern recognition. Image acquisition is performed using a dedicated device especially designed for this purpose [3].

An intrusion detection can be performed in various ways. Literature sources indicate among others methods based on a fuzzy approach [7]. However more frequently proposed solutions are based on classifiers. The in this paper proposed approach is based on classification. Ensembles of classifiers are used to classify features derived from keystroke dynamics analysis and a single classifier approach is used with knuckle patterns.

The goal of the described research is to develop a real time user verification system based on fusion of keystroke dynamics and finger knuckle images analysis. However a method of analyzing the finger knuckle on the fly has not been developed yet as it needs a lot of resources to analyze the images in a real time. Before the decision was made it was necessary to verify, if it is worth investing time and resources in developing such a method and, if the fusion of keystroke dynamics and finger knuckle analysis is potentially interesting. Therefore the preliminary research was conducted to assess the potential of the above mentioned biometric methods fusion. This paper presents the preliminary results of an intruder detection system based on the introduced novel approach.

2 Proposed biometric user verification system

The proposed computer security approach involves two phases: legitimate user profiling and active user verification. In the first stage user profiling is performed that consists of recording a legitimate user activity while working with a keyboard and acquiring this user's finger knuckle images. Based on the acquired data user's profile is being established according to the procedures described in the following sections of this paper.

After establishing user's profile in the first stage the profile can be used for verification of an active user in the second stage. The proposed user verification model shown in Fig. 1 connects two methods of user verification. The introduced approach is based on the fusion of keystroke dynamics and knuckle analysis. For the purpose of the fusion the biometric user verification methods were chosen that according to the [17] for keystroke based approach and [3] for finger knuckle pattern analysis perform better than other methods described in literature.

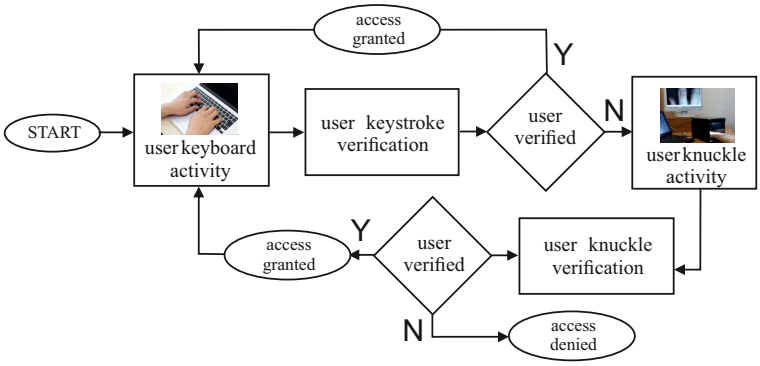

Fig. 1: The proposed biometric user verification algorithm

User activity should be verified continuously, in the background, while a user is performing his everyday tasks. To allow this, all keyboard events generated by the user are recorded and time dependencies between them are analyzed. The keystroke verification unit basing on the profiles of legitimate users and recorded activity of an at the moment active user establishes a decision if the activity belongs to a legitimate user. If the user has been successfully verified an access is granted and the verification procedure continues. In case the verification was not successful there is a suspicion that an active user is an intruder. Therefore an additional verification is made by means of knuckle pattern recognition unit after taking a picture of the user's finger knuckles. If the user has been successfully verified an access is granted and the verification procedure continues with the keystroke based verification. If, again, the verification has not been successful an access to resources is denied and the security breach alert is generated.

2.1 Keystroke based subsystem

Keystroke dynamics refers to a typing pattern of an individual which in practice constitutes a so called profile. In the proposed method a profile of a user contains information on a sequence of key events and time dependencies that occur between the key events. The advantage of the proposed profiling method is that activity data collecting and analyzing is performed continuously in the background which makes it practically transparent for a user. For this reason the profiling method can be used in Host-based Intrusion Detection Systems (HIDS) that analyze the logs with registered user activity in real time to detect an unauthorized access. For the purpose of the research the dedicated software for data acquisition was implemented. The software was designed to collect events generated by individuals (operators of computer systems) while working with a protected system. Proposed software works continuously in a background and records the user activity. The events are captured on the fly and saved in text files. The consecutive lines of the data file contain a sequence of events related to a user activity. Each line starts with the prefix describing a type of an event,

followed by the timestamp of this event and an identifier of a key that generated the event. Possible values of prefix are: *keyDown* representing pressing of a key and *keyUp* for key release event. An example of a raw input data is presented in Fig. 2. Such a recorded raw data can be presented in a data vector form (1).

```
keyDown, 1396968151226, ID15
keyUp,   1396968151376, ID15
keyDown, 1396968152306, L3
keyDown, 1396968153376, ID34
keyUp,   1396968152446, L3
keyUp,   1396968153576, ID34
```

Fig. 2: A fragment of an input dataset with recorded keyboard events

Data of a single j-th keyboard event for a given user constitute a vector $\mathbf{e}_j$:

$$\mathbf{e}_j = [type_j, t_j, \omega_j], \tag{1}$$

where $type \in \{keyDown, keyUp\}$ describes a type of a j-th event; t_j is a timestamp of a given event; ω_j is an identifier of a used key.
Activity data analysis is carried out separately for each user identified in the system by user identifier uid. All vectors $\mathbf{e}_j$ of the same user constitute this user activity dataset E^{uid}. In practice a number of vectors $\mathbf{e}_j$ is limited by period of time when the user activity was recorded.

Data in this form are difficult to interpret because they do not provide directly in-formation on how a user interacts with a computer system. Therefore it is necessary to process the data to obtain characteristics of a user by extracting time dependencies between keyboard events generated while the user was working. It should be noted that during the user activity analysis not only single characters are taken into account but pairs of keys are analyzed as well (for example when writing capital letters). In the proposed method there are 113 separate keys and key pairs considered.

Time dependencies were depicted as a difference of time between two keyboard events and were calculated according to the following rules. Time dependencies for single keys are represented by dwell times (time when a key stays pressed) and for pairs of keys by a delay time between two consecutive key down events of the overlapping keys (as shown in Fig. 3). In the next step time dependencies representing a use of the same key or key pair are grouped together. As, in total, there are 113 different keys and key pairs, there are also 113 separate time dependency groups G^k, $k = 1, ..., 113$ considered. The allowed number of time dependencies stored in a single group is limited by the parameter g. Each time a number of time dependencies in any of the groups G^k reaches g a feature vector $\mathbf{F}$ is created and this group that reached the limit is cleared. The value of parameter $g = 15$ has been determined experimentally.

Based on time dependencies stored in all previously formed groups G^k a feature vector $\mathbf{F} = [f_1, ..., f_{113}]$ is constructed as follows. The k-th element f_k of the

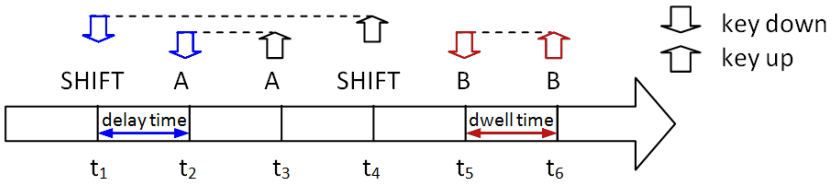

Fig. 3: Time dependencies between keyboard events

vector $\mathbf{F}$ is calculated as the standard deviation of time dependencies stored in a k-th group G^k. For a given user identified by uid (based on this user's input data set) more feature vectors $\mathbf{F}$ are created and a profile $\Phi^{uid} = \{\mathbf{F}_1^{uid}, \mathbf{F}_2^{uid}, ..., \mathbf{F}_z^{uid}\}$ describing the activity of a user in a computer system is constituted. The value of the parameter $z = 100$ has been determined experimentally. User's profile Φ^{uid} is stored in the database to be used by a classification based intrusion detection module. User profiling method based on keystroke analysis used in this proposed approach is described in details in [17].

The keystroke verification system is based on three ensembles of classifiers EC_a, $a = 1, ..., 3$. Each of them consists of four heterogeneous classifiers: $\Psi^{(1)}$, $\Psi^{(2)}$, $\Psi^{(3)}$ and $\Psi^{(4)}$. The ensembles of classifiers EC_a work simultaneously and each one of them is trained using a separate training set TS_a (see Fig. 4) established by means of the Algorithm 1. The general structure of the proposed classification module is presented in Fig. 4.

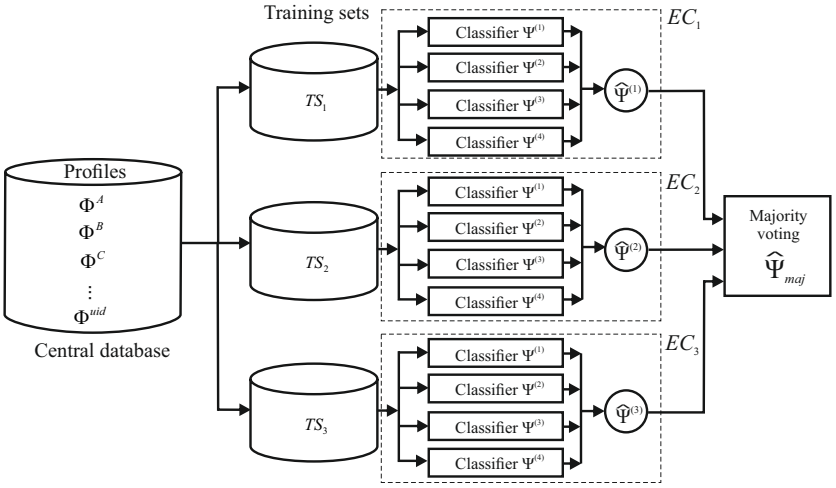

Fig. 4: The general scheme of the proposed classification module

Algorithm 1: Creating the training data sets TS_a, $a = 1, ..., 3$

Data: users' profiles $\Phi^{uid} = \left\{\mathbf{F}_1^{uid}, \mathbf{F}_2^{uid}, ..., \mathbf{F}_z^{uid}\right\}$ comprising feature vectors $\mathbf{F} = [f_1, f_2, ..., f_{113}]$

Result: training sets TS_1, TS_2 and TS_3 of a user *uid*

$O^{uid} :=$ n randomly chosen vectors $\mathbf{F}$ from user's profile Φ^{uid} and $n \leq z$;

for $a = 1$ ***to*** 3 **do**

  $O^a :=$ n randomly chosen vectors $\mathbf{F}$ from another (randomly chosen) user's profile, and $n \leq z$;

  $TS_a := O^{uid} \cup O^a$;

return TS_1,TS_2,TS_3;

User verification consists in assigning a user to one of two possible classes: legitimate user or an intruder. A classifier Ψ maps the vector $\mathbf{F}$ of a given user to a class label c_j, where $j \in \{1, 2\}$:

$$\Psi(\mathbf{F}) \rightarrow c_j \in C. \tag{2}$$

In the proposed approach, the classifiers $\Psi^{(i)}$return a probability $\hat{p}_i(c_j|\mathbf{F})$, $j \in \{1, 2\}$ that a given object $\mathbf{F}$ belongs to a class c_j. At the input of the node $\widehat{\Psi}^{(a)}$ (Fig. 4) the following data matrix is introduced:

$$\mathrm{I}^{(a)}(\mathbf{F}) = \begin{bmatrix} \hat{p}_1(c_1|\mathbf{F}) \; \hat{p}_1(c_2|\mathbf{F}) \\ \hat{p}_2(c_1|\mathbf{F}) \; \hat{p}_2(c_2|\mathbf{F}) \\ \hat{p}_3(c_1|\mathbf{F}) \; \hat{p}_3(c_2|\mathbf{F}) \\ \hat{p}_4(c_1|\mathbf{F}) \; \hat{p}_4(c_2|\mathbf{F}) \end{bmatrix}. \tag{3}$$

Following the classification, each ensemble of classifiers EC_a, $a = 1, .., 3$ generates a local decision $\widehat{\Psi}^{(a)}$ according to the soft voting:

$$\widehat{\Psi}^{(a)}(\mathbf{F}) = \arg\max_{c_j \in C} \sum_{i=1}^{4} \hat{p}_i(c_j|\mathbf{F}), \quad a = 1, .., 3. \tag{4}$$

The class labels returned as a result of (4) are converted to numerical values according to the formula:

$$\widehat{\Psi}^{(a)}(\mathbf{F}) = \begin{cases} -1 \text{ if } \widehat{\Psi}^{(a)}(\mathbf{F}) = c_1 \\ +1 \text{ if } \widehat{\Psi}^{(a)}(\mathbf{F}) = c_2 \end{cases} \tag{5}$$

The results of each ensemble of classifiers EC_a are stored in the set $L = \{\widehat{\Psi}^{(1)}(\mathbf{F}), \widehat{\Psi}^{(2)}(\mathbf{F}), \widehat{\Psi}^{(3)}(\mathbf{F})\}$. On the basis of the set L the value of LS is determined (6).

$$LS(\mathbf{F}) = \sum_{a=1}^{3} \widehat{\Psi}^{(a)}(\mathbf{F}) \in L, a = 1, .., 3. \tag{6}$$

If the value of $LS(\mathbf{F})$ is greater than a threshold τ than the user is allowed to keep working, and the process of keystroke verification is repeated continuously. Otherwise the user must proceed to knuckle verification stage. The influence of the threshold τ value on the keystroke verification accuracy is presented in the section concerning experiments.

2.2 Knuckle analysis subsystem

The aim of knuckle image analysis is to compare and find the similarity between the knuckle image of a person being verified and the reference knuckle images. The reference knuckle images are the images that have been acquired from a user and stored in the database during the profiling phase. In our method a special device was used for knuckle image acquisition. This device consists of a box that has a camera and three built in, white LED-lights. The purpose of the LED-lights is to illuminate the fingers equally from different directions. When taking a picture the camera is focused on the index finger. The example of finger knuckle image acquisition is shown in Fig 5.

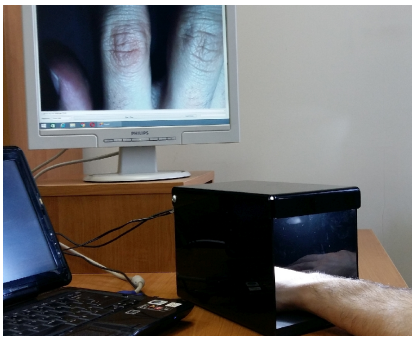

Fig. 5: Finger knuckle image acquisition

After a knuckle image acquisition an analysis is carried out which consists in extracting finger ridges. At first, the Hessian filtering was applied. The reason to choose this filtering method is, that it can detect the local strength of lines, ridges and direction of edges [4,5,9]. In the next step to the analyzed image a binarization is applied using the Otsu method - a well known binarization technique. This method assumes that there are two different classes in the image, foreground (object) and background. Classes are separated from each other by an intensity factor. The Otsu method automatically seeks for the optimum threshold that can maximize the distance between these two classes [10]. On the binarized image a skeletonization is performed which allows to reduce the thickness of lines in the image to one pixel. In the presented method the Pavlidis thinning algorithm was applied. In Fig. 6 all stages of line extraction are shown. The detailed description of the acquisition procedure is presented in [3].

Fig. 6: Stages of finger knuckle image analysis

Upon completion of image analysis, for a verified person (let it be denoted as A), two sets X and Y are formed. The set X contains values of similarity coefficients calculated between all possible pairs of images taken from the person A. All elements of the set X are assigned to the class c_1. The construction of the set X is shown below:

$$X = \{sim(ImR_i^A,\ ImR_j^A)\}, \quad i, j = 1, ..., r, \quad i \neq j, \tag{7}$$

where ImR_i^A is an i-th reference knuckle image of the person A being verified, r is a number of all reference knuckle images of the person A.

The set Y contains values of similarity coefficients calculated between knuckle images of person A and the knuckle images of another user B, where $B \neq A$, randomly selected from a database. The elements of the set Y are assigned to the class c_2. The construction of the set Y is shown below:

$$Y = \{sim(ImR_i^A,\ ImR_j^B)\}, \quad i = 1, , r, \quad j = 1, , s, \tag{8}$$

where ImR_i^A is the i-th reference knuckle image belonging to the person A being verified, r is the number of all reference knuckle images of person A, ImR_j^B is the j-th reference knuckle image of the person B, s is the number of analyzed reference knuckle images of person B.

To avoid imbalanced data [6] the number of elements in the set X should be close to the number of elements in set Y. This assumption is fulfilled if the number of knuckle images used for creating the set Y is equal to $s = r - 1$.

In the presented method, the similarity between any two knuckle images is estimated based on the shape and localization of the knuckle ridges. The comparison of images is carried out by means of the Normal Cross Correlation (NCC) technique. The NCC has been widely used as a metric to evaluate the degree of a similarity (or dissimilarity) between two compared images [8]. In our method all images must have the same size and the shape of a square.

To find the similarity $sim(Im1, Im2)$ between two images denoted as $Im1$ and $Im2$ the image $Im1$ is divided into the square shaped sub-images. A length of a side of these sub-images is a parameter of the method. Each k-th sub-image

in $Im1$ is treated as a template and is noted as T_k. The task is to find a fragment in the tested image $Im2$ which has the most similarity to the template T_k. An example of searching for the sub-image T_1 in the image $Im2$ is shown in Fig. 7.

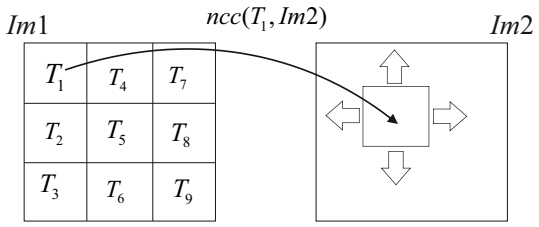

Fig. 7: Searching for the template T_1 within the image $Im2$.

By means of the following formula the similarity between a template T_k and tested image $Im2$ is calculated:

$$ncc(T_k, Im2) = \max\left(\frac{\sum_{(i,j)\in T_k} Im2(x+i, y+j)\cdot T_k(i,j)}{\sqrt{\sum_{(i,j)\in T_k} Im2^2(x+i, y+j)}\cdot\sqrt{\sum_{(i,j)\in T_k} T_k^2(i,j)}}\right). \tag{9}$$

The final similarity between the images $Im1$ and $Im2$ is calculated using (10).

$$sim(Im1,\ Im2) = mean\{ncc(T_1,\ Im2), ncc(T_2,\ Im2), ..., ncc(T_b,\ Im2)\}, \tag{10}$$

where b is the number of all sub-images T_k created in the image $Im1$.

After the creation the sets X and Y the tested knuckle image Im^* is compared with a random image in the database taken from the person who claimed the identity (let it be A). The result of comparison is an object d^*:

$$d^* = sim(Im^*,\ Im_i^A), \tag{11}$$

where, Im^* is the knuckle image to be verified and Im_i^A is a randomly selected original knuckle image of the person A.

Next, the verified knuckle image Im^* is used in the classification stage, where the k-NN classifier is applied [14]. In this approach for k-NN classifier the commonly used Euclidean distance metric is used. In the classification stage first, the distances between classified object d^* and objects from X and Y sets are determined. The selection of the value of parameter k for the k-NN classifier is presented in section concerning experiments. Based on majority, the k-NN gives a decision to which class (c_1 or c_2) the classified object d^* belongs to.

$$\widehat{\Psi}^{knu}(d^*) \to c_j, j \in \{1, 2\}. \tag{12}$$

If the object d^* belongs to the class c_1 it means that the verified knuckle image Im^* comes from the legitimate user.

2.3 Fusion of the methods

The proposed system is based on fusion of two verification methods. The ultimate decision of the user verification depends on both: keystroke and knuckle image verification results. This task is done by analyzing the value of τ, which is the result of the keystroke verification, and the parameter k value in k-NN method. The rules of fusion system are presented below:

$$\text{decision} = \begin{cases} \text{access granted} & if \quad (LS > \tau^{'}) \; or \; (LS > \tau^{''} \; and \; k > k^{*}) \\ \text{access denied} & otherwise \end{cases}. \tag{13}$$

Values of parameters $\tau^{'}$, $\tau^{''}$ and k^{*} have been described in experiments section.

3 Experimental results

The efficiency of the proposed method has been investigated experimentally. The researches have been conducted on a database which consists of 4000 vectors **F** and 150 knuckle images acquired from 30 persons. All experiments were repeated 10 times to provide better statistical accuracy and then the average values of evaluation metrics for all trials were calculated.

The proposed architecture of the classification module for keystroke dynamics based verification assumes the use of four single classifiers in an ensemble: C4.5, Bayesian Network, Support Vector Machine, Random Forest. Those classifiers were chosen because of their high accuracy confirmed in [17]. The aim of the first stage of this research was to determine an optimal values of parameters $\tau^{'}$, $\tau^{''}$ and k^{*}. The values of mentioned parameters were determined by use of grid search procedure from the following sets $\tau^{'}, \tau^{''} \in \{1, 2, 3\}$, $k^{*} \in \{1, 3, 5, 7\}$.

The experiments were conducted several times. Each time we used different numbers of knuckle images n and r to determine the sets X and Y. Table 1 shows the best obtained results and the values of the parameters $\tau^{'}$, $\tau^{''}$ and k^{*} for which these results were obtained.

Table 1: The best results and the values of parameters used in experiments.

Number of elements in sets X & Y	**FAR** [%]	**FRR** [%]	**AER** [%]	**ACC** [%]	$\tau^{'}$	$\tau^{''}$	k^{*}
4 and 4	7.72 ± 1.36	19.38 ± 2.32	13.55 ± 1.73	93.36 ± 1.89	3	2	5
6 and 6	6.38 ± 0.51	12.56 ± 1.45	9.47 ± 2.12	95.17 ± 2.54	3	2	5
8 and 8	1.07 ± 1.47	3.35 ± 2.04	2.21 ± 1.76	98.50 ± 1.33	3	2	5
10 and 10	1.11 ± 1.13	3.33 ± 2.26	2.22 ± 1.64	98.49 ± 1.22	3	2	7
12 and 12	1.09 ± 1.45	3.36 ± 2.35	2.22 ± 1.32	98.51 ± 1.53	3	2	5

Based on the obtained results we can state that the optimal values of the parameters for the proposed method are $\tau^{'} = 3$, $\tau^{''} = 2$ and $k^{*} = 5$. When

we analyze the influence of a number of knuckle images on the results of an investigation, we can observe that the efficiency of the classification is the best when we analyze only 8 knuckle images from each person.

In order to fully assess the effectiveness of the proposed fusion-based approach, its efficiency has been compared with the efficiencies obtained by each of the biometrical verification methods separately. For this purpose the optimal values of parameters τ and k have been selected once again but this time the efficiency of each verification method (keystroke verification and knuckle verification) has been assessed independently. The comparison of efficiency obtained by the fusion-based method and each individual verification method is presented in Table 2.

Table 2: The comparison of the performance of various verification methods.

Method	**FAR** [%]	**FRR** [%]	**AER** [%]	**ACC** [%]
Keystroke	5.59 ± 1.16	7.94 ± 1.46	5.41 ± 1.72	97.83 ± 3.28
Knuckle	4.30 ± 0.23	8.19 ± 1.05	6.24 ± 1.09	95.96 ± 7.53
Keystroke + Knuckle	1.07 ± 1.47	3.35 ± 2.04	2.21 ± 1.76	98.50 ± 1.33

By analyzing Table 2, we can notice that the fusion of two methods allows to obtain better efficiency in classification than using only one of these methods separately.

4 Conclusions

This paper presents the preliminary results for the biometric user verification system based on the fusion of keystroke and knuckle analysis. The experiments were conducted to assess the potential of the mentioned biometric methods fusion. The obtained results show that the fusion of the two methods performs better than the keystroke and knuckle analysis separate. Therefore, there is a motivation to continue this research and to develop a real time knuckle analysis method allowing o verify finger knuckles of a computer user while typing on the keyboard. This seems to be a complex task due to the constant movement of fingers over the keyboard. Preliminary research and experiments show that basing on images of the user's knuckles taken in various hand positions it is difficult to verify an identity of a user.

What more, all the image processing has to be performed in the background of the computer system while users are performing everyday tasks. Depending on the frequency of taking a picture this can cause some computer system efficiency issues. Solving the issues mentioned above is the next step of our research on developing fusion-based computer user continuous verification method.

References

1. Alsultan, A., Warwick, K.: Keystroke Dynamics Authentication: A Survey of Free-text Methods. International Journal of Computer Science Issues 10, 1–10 (2013)
2. Banerjee, S.P., Woodard, D.L.: Biometric Authentication and Identification Using Keystroke Dynamics: A survey. Journal of Pattern Recognition Research 7, 116–139 (2012)
3. Doroz, R., et al.: A New Personal Verification Technique Using Finger-Knuckle Imaging. Lecture Notes in Computer Science 9876, 515–524 (2016)
4. Iwahori, Y., Hattori, A., Adachi, Y. et al.: Automatic detection of polyp using Hessian Filter and HOG features. Procedia Computer Science 60(1), 730–739 (2015)
5. Jin, J., Yang, L., Zhang, X. et al.: Vascular tree segmentation in medical images using Hessian-based multiscale filtering and level set method. Computational and Mathematical Methods in Medicine, (2013)
6. Krawczyk, B., Wozniak, M., Schaefer, G.: Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing 14, 554–562 (2014)
7. Kudlacik, P., Porwik, P., Wesolowski, T.: Fuzzy approach for intrusion detection based on user's commands. Soft Computing 20, 2705–2719 (2016)
8. Nakhmani, A., Tannenbaum, A.: A new distance measure based on generalized Image Normalized Cross-Correlation for robust video tracking and image recognition. Pattern Recognition Letters 34(3), 315–321 (2013)
9. Nitsch, J., Klein, J., Miller, D. et al.: Automatic Segmentation of the Cerebral Falx and Adjacent Gyri in 2D Ultrasound Images. Bildverarbeitung fr die Medizin, 287–292 (2015)
10. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man and Cybernetics 9, 62–66 (1979)
11. Raiyn, J.: A survey of cyber attack detection strategies. International Journal of Security and its Applications 8(1), 247–256 (2014)
12. Rybnik, M., Tabedzki, M., Adamski, M., Saeed, K.: An Exploration of Keystroke Dynamics Authentication using Non-fixed Text of Various Length. In Proc. of Int. Conference on Biometrics and Kansei Engineering (ICBAKE), 245–250 (2013)
13. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. Advances in Information Security 39, 69–90 (2008)
14. Shakhnarovich, G., Darrell, T., Indyk, P.: Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. Neural Information Processing, (2006)
15. Teh, P.S., Teoh, A.B.J., Yue, S.: A Survey of Keystroke Dynamics Biometrics. The Scientific World Journal 2013, (2013)
16. Wesolowski, T., Palys, M., Kudlacik, P.: Computer User Verification Based on Mouse Activity Analysis. Studies in Computational Intelligence 598, 61–70 (2015)
17. Wesolowski, T. E., Porwik P., Doroz R.: Electronic Health Record Security Based on Ensemble Classification of Keystroke Dynamics. Applied Artificial Intelligence 30, 521–540 (2016)
18. Zhong, Y., Deng, Y., Jain, A.K.: Keystroke Dynamics for User Authentication. Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference, 117–123 (2012)