Advances in the Keystroke Dynamics: the Practical Impact of Database Quality

Mariusz Rybnik[1], Piotr Panasiuk[2], Khalid Saeed[2] and Marcin Rogowski[3]

[1] University of Bialystok, Bialystok, Poland
[2] AGH University of Science and Technology, Cracow, Poland
[3] King Abdullah University of Science and Technology, Thuwal, Kingdom of Saudi Arabia
mariuszrybnik@wp.pl, panasiuk@agh.edu.pl,
saeed@agh.edu.pl, marcin.rogowski@kaust.edu.sa

Abstract. This paper concerns database quality in the Keystroke Dynamics domain. The authors present their own algorithm and test it using two databases: the authors' own *KDS* database and *Keystroke Dynamics - Benchmark Data Set* online database. Following problems are studied theoretically and experimentally: classification accuracy, database representativeness, increase in typing proficiency and finally: time precision in samples acquisition. Results show that the impact of the database uniqueness on the experimental results is substantial and should not be disregarded in classification algorithm evaluation.

Keywords: keystroke dynamics, identification, authentication, behavioral science, biometrics, computer security.

1 Introduction

With the recent expansion of Internet and the constant development of social networks, a lot of sensitive personal data circulate in the worldwide web. Frequently it is important to maintain *data security*, by limiting the access to a specific trusted group of individuals. It is therefore essential to determine or confirm person identity. The task is known as *authentication* - determining if the specific person's identity conforms to its claim. *Reference authentication data* has to be stored inside an *authentication database*, in order to be compared with *authentication data* provided by the user. After positive *authentication* the user is granted access to the *sensitive data* or *services*. The difference between *authentication* and *identification* is that *identification* is to determine the user's identity, without a claim who it is. In the case the whole *authentication database* has to be searched and the best matching user may be given access (if his *authentication data* is trustworthy enough).

The traditional taxonomy of the human *authentication methods* has been proposed by H. Wood [1] and (after slight modifications) it distinguishes three groups of methods:

- **a proof by knowledge -** something that the user knows and remembers, such as passwords, PIN numbers, lock combinations, answers to secret questions;

- **a proof by possession -** a unique item (token) that the user possesses, e.g. keys, chip cards, magnetic cards, hardware or software tokens;
- **biometrics -** behavior or physical body properties unique for the user, such as fingerprints, signature, keystroke dynamics, eye retina pattern, hand shape, ear shape, etc.

Proof by knowledge is the most popular method of securing digital data, usually referred to as *passwords*. Regarding security it is important to create an efficient *multiple-use password* (as opposed to one-time passwords), which should follow three properties listed by Burnett and Kleiman [2]: *complexity*, *uniqueness* and *secrecy*. In practice, unfortunately, most users ignore at least one of the rules, e.g.: (*i*) a password is unique and complex, but written on an easily-accessed memo; (*ii*) a password is complex and secret but the same for every service; (*iii*) a password is unique and secret, but very simple to guess. As reported by Bruce Schneier [3] about 25% of the passwords can be guessed using a 1000-word dictionary with 100 common suffixes. Larger dictionary along with biographical data brings success rate to 55-65%.

Techniques that use *proof by possession* guarantee neither high security nor availability. As tokens are physical objects, they can be possibly handed over, stolen, misplaced or broken. If they are not secured by an additional password or a PIN code one can assume that the thief will easily access the sensitive data.

The *biometric methods* can be used for both *authentication* and *identification*. *Biometrics* is a science concerning measurements of living organism features. In the past few decades there was a noticeable increase in biometrics popularity, especially in the domain of *data security*. *Biometric methods* vary greatly in terms of uniqueness, classification accuracy and acceptability. Measured *features* can be classified on the basis of their origin as physical or behavioral features. *Physical features* are those that are derived from the way in which our body is built. The most popular and proved physical feature is fingerprint. Among physical biometric features one can also distinguish: face image, iris or retina scan, hand geometry. *Behavioral features* originate from a way user performs certain activities. The most known and the oldest behavioral biometric feature is handwritten signature. Examples of other behavioral features are: voice, gait and – the main subject of this paper – keystroke dynamics.

Keystroke dynamics, like *gait analysis*, has significant advantages over other biometric features. It is non-invasive, highly acceptable and it does not need specialized hardware (in its basic form). There are also some disadvantages of *keystroke biometrics*: (*i*) efficient features interpretation can be problematic; (*ii*) limitations of present Operating Systems can affect the data quality. Reference [4] correctly points out that the researchers often overlook an important disadvantage of many biometric methods – acceptability. Obtaining fingerprints or an iris scan may be considered insulting by some people.

Sometimes a particular biometric feature cannot be obtained from the user (e.g. a finger blessing altering the fingerprint), thus systems based on more than one feature are desirable. In [5] voice, hand geometry and face image are used together.

This paper is organized as follows: section 2 describes state of the art in *keystroke dynamics*, section 3 presents two databases: database *KDS* created by the authors and

Keystroke Dynamics - Benchmark Data Set [6] database available online, section 4 presents briefly the fundamentals of the authors' classification approach to *keystroke dynamics* (for details please refer to papers [7]-[9]), section 5 presents classification results and problems related to database quality regarding *keystroke dynamics*, finally section 6 concludes the paper.

2 State of the art

Keystroke dynamics is a *behavioral biometric feature* that describes human *keyboard typing pattern*. This method is dated as far as the invention of the telegraph and its popularization in the 1860s [10]. *Keystroke dynamics* is not as accurate method as *fingerprint* pattern so it cannot be used for forensic purposes [11], as the method does not meet the European access control standards such as EN-50133-1. It specifies that FRR should be less than 1% and FAR should be no more than 0.001%. However, if one includes other features of typing, the keystroke dynamics results will definitely be improved. Similarly, as it is with handwritten signature, when existing (off-line) signature is analyzed, only two features are considered – its dimensions. On the other hand, when on-line signature is analyzed, additional features such as the pressure of the pen, its angle and the position in time can be extracted. This gives five features to analyze and improves accuracy significantly. A good idea is also to analyze the pressure of the keystroke.

Keystroke dynamics itself is not likely to give satisfying results, unless merged with some other biometric features, preferably non-invasive physiological ones in a multimodal system. An example of multifactor systems could be keystroke dynamics merged with face image recognition used to verify user identity while inserting PIN number at the ATM.

2.1 Latest achievements and other possible directions

Latest research focuses in general on the *user authentication* in order to secure personal computers. There are only a few works on the topic of *user identification*. Artificial Neural Networks are one of the most common tools for classification. The main disadvantage of ANN is the high dependence on the training database and high cost of retraining. Also, it is a *black-box model*, so no information about the specific attributes is available. Researchers mainly focus on the algorithms that are ready and known to work well, but in general the number of untested approaches is constantly decreasing.

With many of the algorithm ideas tested, researches started looking for new features that would improve the classification accuracy. One of the ideas is to use pressure sensitive keyboards. Microsoft is working on the hardware [12] and a student team contest was organized using the prototypes, searching for new ideas [13]. It is shown that *pressure* is even more important characteristics than the dynamics itself [14]. In [15] the authors constructed their own keyboard and used pressure as an additional feature, which turned out to be very helpful for the *user authentication*. This

should not surprise anyone since i.e. *on-line signature recognition* is generally more reliable than *off-line*. The results suggest that the use of *pressure information* would greatly help in *user identification*. The main problem with this approach is very low availability of pressure sensitive keyboards.

Some research has been done using mobile phone keyboards as input devices [16]-[18]. The motivation behind is the rising popularity of mobile phones and the fact that many users do not even use PIN to protect their devices. The proposed solution is to use *keystroke dynamics* to *authenticate* users as they type text messages and phone numbers. For the standard 9-key keyboard, both numerical and alphabetical inputs have been tested and the error rates are reported to be about 12.8% for 11-digit phone number [16] and 13% using fixed 10-character long alphabetical input [17]. Interestingly, for mobile version of QWERTY keyboard, dwell time for each key did not prove to be a reliable feature and only latency between keys was used [18]. Results were similar as for 9-key keyboard and the error rate was 12.2%.

ATM hardware was also considered [19], but rather than *keystroke dynamics*, keystroke motion and hand shape at different time points were analyzed and the results proved to be very good. Error rate achieved was as low as 1.1% to 5.7% depending on the PIN and exact features used. This approach requires a camera which records hands movements as the PIN is typed. It raises safety issues, as it is generally advised to hide hand movements while typing PIN.

2.2 Database and sample validity

The work [20] summarizes all major efforts in *keystroke dynamics* with attention put on database issues. The algorithms in the field are mostly developed using dedicated databases. The main problem is that all those various and 'specialized' databases are very difficult to compare. Some of them were collected in *supervised conditions*. In this case certain samples may be disregarded, i.e., the users who make a lot of mistakes or users that want to sabotage the experiment (by intentionally inserting unnaturally different samples). Samples are gathered with various amounts of characters. One cannot tell if a phrase is as good in discriminating user's identity as the other with the same length [8]. Some of the phrases also need pressing additional special keys in case of typing capital letters or diacritic characters. The size of the users' population matters greatly, especially with *identification* algorithms. Another issue is incomplete or corrected data. That leads to sample inconsistencies that may render the results unreliable. The event timing may be affected by OS clock process queuing. It was examined using arbitrary waveform generator [21] and reported that 18.7% of the keyboard events are registered with 200μs latency. However, while using typical PC, samples are limited in precision with OS event clock, which is limited to accuracy of 15.625 ms (64 Hz) using MS Windows and 10 ms using most Linux distributions.

Considering the constraints described above, the authors of [20] released their database online: *Keystroke Dynamics - Benchmark Data Set* that is very accurate and has many samples. The database is available online free of charge [6] and was used by the authors along their own *KDS database* to experimentally test database-related issues.

3 Database classification

It has been shown that *keystroke dynamics authentication* results highly depend on the database quality [20], [22]. Viable algorithms should deal with noisy samples: the ones with typos or random pauses in user typing. Among the databases the authors can distinguish ones collected in a supervised way, meaning every test subject was individually instructed by a supervisor before the start of the samples acquisition process. The supervisor can also make notes on how the subject types and what influences him. It guarantees samples of good quality. This type of database, however, usually does not reflect real world situations. Databases may have accounts duplicated, for example if the user forgets his password or just wants to have multiple accounts. The typing pattern may be duplicated for two different classes, which may decrease the identification accuracy and in hybrid (rank-threshold) based verification methods it may even increase the FRR. Typing with unnatural manner can also increase FAR.

Another factor is the purpose for which the database is gathered. Authentication requires user ID attached to *keystroke data*. Simulation of hacking requires the same text typed by many users. Passwords are usually short phrases often consisting additional characters like capital letters (that involve *shift* key), dots, semicolons, numbers and symbols. For *identification* samples should be preferably longer, as this application is more complex.

There can be two additional approaches to keystroke *data acquisition*. The first is based on *a fixed text*. The second way is to use *free-text authorization* [22] to continuously monitor user's workstation while trying to *authorize* him/her. There are the following problems with *free-text authorization*: (*i*) how often *user authentication algorithm* should be run, (*ii*) more difficulty with data collection, (*iii*) more samples are needed for learning of the recognition algorithm. Potential noise can be a unique feature that helps to recognize users, so removing it completely – without deeper analysis – would be a loss of valuable information.

3.1 KDS database description

The authors' keystroke dynamics database (*KDS database*) was gathered in non-supervised conditions using JavaScript web-browser platform [23]. It is therefore OS independent and globally available, however at a cost of unpredictable latency. Data from over 400 users and total of over 1500 samples is stored in the database.

KDS database is unique, as it stores additional meta-information like *user's name*, *age*, *sex*, *hand used while writing* and *estimated proficiency with keyboard*. This additional information could serve for other purposes than authentication. The samples consist of five phrases, different among language versions (Polish and English). Uppercase and lowercase letters, special characters and key modifiers (Shift, Alt) are registered. The first phrase is a popular sentence, in English it is "To be, or not to be, that is the question. The second phrase is a tongue twister; in English it is "Shy Shelly says she shall sew sheets." The third phrase is an example of simple password: short Polish word: "kaloryfer". The fourth phrase is a user-chosen sentence. The fifth

phrase is a Psylock (commercial keystroke dynamics solution) password “After some consideration, I think the right answer is:” [24].

3.2 *Keystroke Dynamics – Benchmark Data Set* database description

Keystroke Dynamics - Benchmark Data Set database [6] was used for the reference. It was gathered in supervised conditions from 51 subjects, using an external high precision clock. Sample acquisition was divided into eight sessions, 50 samples each. Each user had to type a phrase “.tie5Roanl” 400 times. The data acquisition sessions were separated by at least 24 hours. The database was used to test 14 published classifiers [20]. The database is especially useful for testing *fixed-text* algorithms. It is time-accurate, has a reasonable number of users and many samples per user.

4 The authors’ approach

In this section, the authors describe their approach to identification, operating on *fixed-text* samples. Main goal is to compare the results obtained with the two above-mentioned databases. The authors use *k-Nearest Neighbor classifier*, so *k* value is chosen and a *training dataset* is built, where the amount of samples per user cannot be less than *k*. The remaining user samples are assigned to the *testing dataset.* The authors' latest approach [8] was to calculate initial weights for all expected key events. However, during tests with *Keystroke Dynamics - Benchmark Data Set* it turned out that the classification results are better with the use of the former algorithm [7]. The possible explanation of this phenomenon is given in section 5.

Absolute times are processed into *flight times* and *dwell times*. *Flight times* are the times between releasing one key and pressing another. *Dwell time* is the time when a key is in the pressed state. The reason the authors convert *event times* into those two characteristics is because they are more stable. When the user makes a mistake or hesitates on some key, this would only affect the next two keys and not all the remaining times. The distances between samples are calculated using Manhattan metrics between corresponding *keyboard event* times.

Partial distances for two given samples were calculated using Manhattan distance (for corresponding *dwells* and *flights*), as specified in (1) and (2), where d_d is the partial dwell distance, d_f is the partial flight distance, d_{1i} and d_{2i} are the *i*-th dwells for *1st* and *2nd* samples, respectively, d_d is the partial dwell distance, f_{1i} and f_{2i} are the *i*-th flights for *1st* and *2nd* samples, respectively.

$$d_d = \sum_{i=1}^{n} |d_{1i} - d_{2i}| \quad (1)$$

$$d_f = \sum_{i=1}^{m} |f_{1i} - f_{2i}| \quad (2)$$

The total distance *d* between the two samples is calculated as in (3), where *p* is the ratio of importance of the *flight time* compared to the *dwell time*.

$$d = p * d_f + (1 - p) * d_d \quad (3)$$

Both *flight* and *dwell* are important as the authors presented in [7]. In the previous experiments the authors had determined the best *p* ratio value as 0.7. However, due to use of the different database, the authors decided to use the arbitrary value of 0.5.

After the calculation of all distances *k* samples are labeled with the training author ID and assigned a *rank*. The authors evaluate only *closed-world case*. Among all the results the authors take the *k* best ones and then conduct *voting* procedure on users (as described in detail in [7]. The shortest distance gets the highest score of *k*, the longest distance gets the lowest score of 1. The *winner* is the user with the greatest sum of scores.

5 Experimental results

5.1 On classification accuracy

The authors have tried many varieties of combinations of their algorithm, while using the same amount of users in both databases, number of characters in a phrase and amount of training samples. In both experiments the training data sets were created using random samples, *k*=2, training set containing 6 samples for each user and 51 classes. Fig. 1 shows the results of this comparison where the *flight-to-dwell importance* is presented in horizontal axis. As can be seen, the classification results are significantly different. It leads to the claim that the results are incomparable even if the authors test the same algorithm in similar conditions. This supports the conclusion from [20] that the results obtained by research teams on their own databases may be incomparable.

5.2 On database representativeness

The main issue with databases collected in the supervised conditions is that they do not refer to the real-world conditions and therefore may lead to false results. Watching a user may be frustrating and lead to the acquisition of corrupted samples. Supervised acquisition, however, eliminates samples intentionally counterfeited. Real-world samples are sometimes corrupted. In *Keystroke Dynamics - Benchmark Data Set* there seem to be no corrupted samples. When using this database samples written with mistakes should be therefore rejected, as they probably cannot be classified.

Keystroke Dynamics - Benchmark Data Set is accurate and has a large amount of samples. It is perfect for testing algorithms for *user authentication*. However, when it comes to *user identification*, samples are too short to obtain satisfactory accuracy. Obtained accuracy of about 67% is satisfactory for such a short phrase and 51 classes.

5.3 On increase of user's typing proficiency

In the *Keystroke Dynamics - Benchmark Data Set* users were asked to type 50 samples each time in 8 sessions separated one from another by at least 24 hours. The authors wondered how the learning process influences the results, so the authors tried to

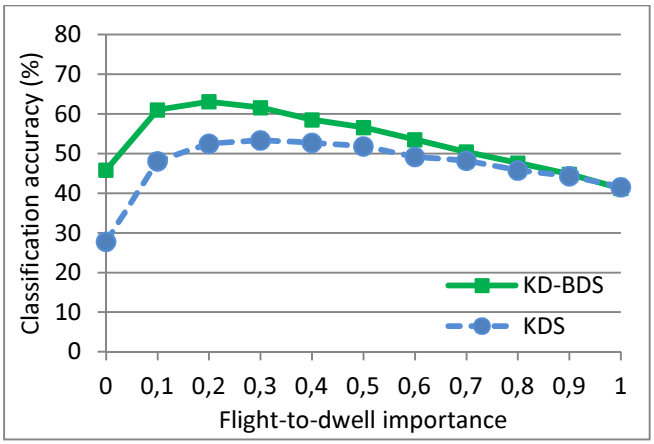

Fig. 1. The authors' approach classification rates, using two various databases, versus *flight-to-dwell importance* factor

test few *learning set* building algorithms. For the first experiment the authors took 8 random samples from all sessions per each user. Those samples should contain the best characteristics of the typing style of each user. In the second experiment the authors selected the first sample from each session. It means that there was at least 24 hour time span in acquisition between any of the samples from any single user used in the experiment. In the third experiment there were only the first 8 samples from the first session. This means that the user was not familiar with the password and has not developed the typing pattern yet. In the fourth experiment the last 8 samples from the last session were selected for training. It means that the users were well trained in typing the password. However, those are the last samples in the 50-sample session, so the users could be already tired. The authors always used 8 training samples per user profile and the authors set *k* value to 8 in our algorithm. In Fig. 2 one can see that the results vary a lot.

As one can conclude, using the samples collected *early* does not result in satisfactory accuracy. The characteristics obtained from them are differentiated, distorted by the fact that the user was still unfamiliar with the password. The first samples from each session also are not very good training dataset because the user had a long break between inserting them and they differ from the average user's characteristics. The last inserted samples are better, however, the user seemed to be tired typing so many samples and they may be not as stable as the samples from the middle of the session.

Many of *keystroke dynamics* methods are based on Artificial Neural Network (ANN) algorithms. The authors' experiments show that the first samples of the user are the noisiest ones, and using them to train the ANN yields poor results similar to those shown in [20].

The next problem the authors examined was the time of the sample typing. Fig. 3 presents the average time of phrase acquisition of 5 randomly selected users in each session. One can observe the gradual decrease in the mean time values.

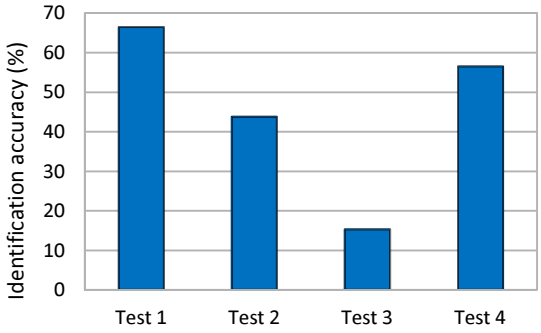

Fig. 2. Sample data selection using different methods. Test 1 – random samples. Test 2 – first sample from each session, Test 3 – first 8 samples from the first session, Test 4 – last 8 samples from the last session

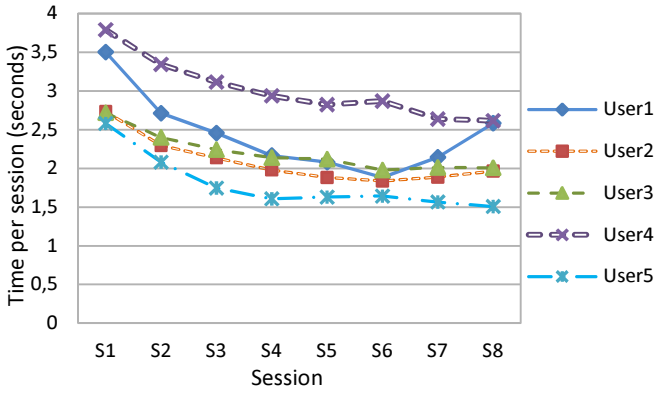

Fig. 3. The mean sample total typing time for five randomly selected users, versus sessions

The standard deviation of each keystroke decreases over time, as could be seen in Fig. 4 Gradually the users insert samples in a more consistent manner. Inner-class differences decrease, which helps in classification. It reduces FAR in verification systems.

5.4 On time precision in samples acquisition

In [21] keyboard functioning using 15MHz function and arbitrary waveform generator was examined. It was noticed that 18.7% of keystrokes were acquired with a *200μs* error. Therefore, the keyboard was calibrated and the database was collected using higher precision. Data have been gathered with an accuracy of *100μs*. This experiment has shown that databases gathered using different machines may not be comparable because of *the lack of the main bus clock calibration*. Moreover, when

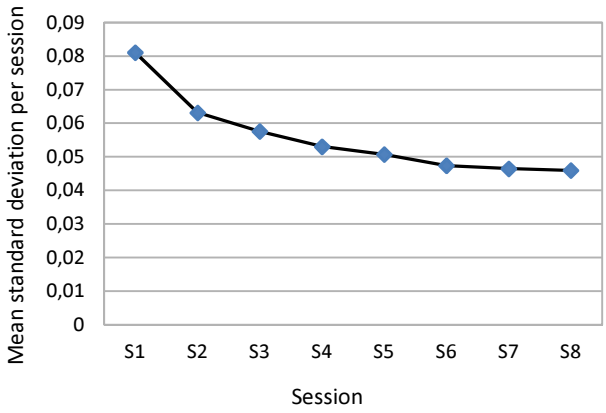

Fig. 4. The average of the keystrokes *standard deviation* for all samples, versus session number

CPU(s) is under load, delays in keystroke acquisition occur, as they are usually handled by *the message queue*. There is also a difference between the lengths of keyboard clock frames of Linux/Unix (*10ms* frame) and Windows (*15ms* frame, 64 ticks per second) operating systems. The influence of time resolution on the algorithm classification accuracy was tested using the authors' approach.

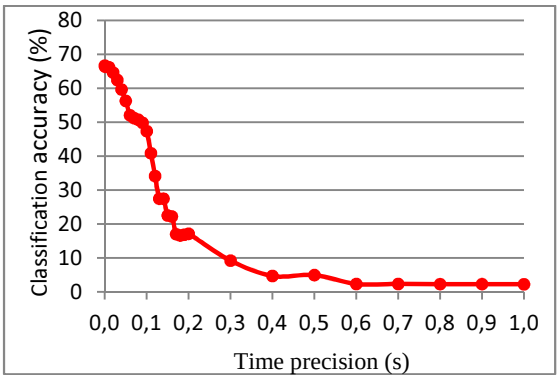

Fig. 5. Influence of time precision on algorithm's classification accuracy

Fig. 5 shows that in the typically used time frames (*10-15 ms* – operating system clock intervals) there is no concern about data precision. As long as the resolution is smaller than *1ms* there is no significant difference in the algorithm classification accuracy. However, it is easily noticeable, with a resolution greater than around *20ms*, that the accuracy drops significantly. With precision of 0.6*s* almost random results are obtained.

6 Conclusions

There has been a lot of research done in the field of keystroke dynamics during past decades. Many classification methods were researched in order to improve *keystroke dynamics* classification accuracy for both *authentication* and *identification* tasks. Obtained results are however hardly comparable due to the use of various database acquiring procedures and non-availability of databases.

The authors have tested two databases. The *KDS database* has worse time precision than *Keystroke Dynamics - Benchmark Data Set* due to the acquiring procedure performed over the Internet with use of JavaScript and web browser. It is however more universal as it could be used remotely (however user identity would not be guaranteed in this case). *KDS database* is more suitable for user *identification* because the samples are longer. It contains users' mistakes and corrections, what could be used in further experiments.

As the authors have indicated experimentally - using the same algorithm and conditions - specifics of *training database* affects *classification accuracy*. The samples obtained later with greater users' proficiency are of better consistency and distinguish users more reliably. Training set should also contain imperfect samples as they increase FRR error margin. An algorithm for updating the training set should be considered, as using only the initial samples would affect the classification accuracy.

The observations lead to the conclusion that the selection of database for the tests has the vital meaning for the reliability of results. The tests of classification algorithms should be run on the same database without any modifications and with a fixed *training dataset* building. If the conditions are not satisfied, the obtained results are hardly comparable with others.

ACKNOWLEDGEMENTS

This work was supported by AGH University of Science and Technology, grant no. 11.11.2010.01.

References

1. H. M. Wood, “The use of passwords for controlling access to remote computer systems and services,” in *Proc. of the National Computer Conf.*, New York, 1977, pp. 27-33
2. M. Burnett and D. Kleiman, *Perfect Passwords*. Rockland, MA: Syngress, 2005.
3. B. Schneier. (2007, January 11). *Secure Passwords Keep You Safer* [Online]. Available: http://www.wired.com/politics/security/commentary/securitymatters/2007/01/72458.
4. X. Li; S. J. Maybank, S. Yan; D. Tao, and D. Xu, "Gait Components and Their Application to Gender Recognition," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 145-155, 2008.
5. K. Veeramachaneni, L. A. Osadciw, and P. K. Varshney, "An adaptive multimodal biometric management algorithm," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 35, no. 3, pp. 344-356, 2005.
6. K. Killourhy and R.A. Maxion. (2009, June 29). *Keystroke Dynamics - Benchmark Data Set* [Online]. Available: http://www.cs.cmu.edu/~keystroke/

7. M. Rybnik, P. Panasiuk, and K. Saeed, “User Authentication with Keystroke Dynamics Using Fixed Text,” in *IEEE-ICBAKE’09 – International Conference on Biometrics and Kansei Engineering*, Cieszyn, Poland, 2009, pp. 70-75.
8. P. Panasiuk, K. Saeed, “A Modified Algorithm for User Identification by His Typing on the Keyboard,” in *Image Processing & Communication Challenges 2* (Advances in Intelligent and Soft Computing Series 84), R. S. Choras, Ed., Berlin: Springer-Verlag, 2010, pp. 113-120.
9. M. Rybnik, M. Tabedzki, and K. Saeed, ”A keystroke dynamics based system for user identification,” in *IEEE-CISIM’08 – Computer Information Systems and Industrial Management Applications*, Ostrava, Czech Republic, 2008, pp. 225 – 230.
10. J.C. Checco, “Keystroke Dynamics and Corporate Security,” *WSTA Ticker*, 2003.
11. *CENELEC. European Standard EN 50133-1: Alarm systems. Access control systems for use in security applications. Part 1: System requirements*, Standard Number EN 50133-1:1996/A1:2002, Technical Body CLC/TC 79, European Committee for Electrotechnical Standardization (CENELEC), 2002.
12. P.H. Dietz, B. Eidelson, J. Westhues, and S. Bathiche, “A practical pressure sensitive computer keyboard,” in *Proc. of the 22nd annual ACM symposium on User interface software and technology*, New York, 2009.
13. UIST. (2009, October 6). *Student Innovation Contest Results* [Online]. Available: http://www.acm.org/uist/uist2009/program/sicwinners.html
14. H. Saevanee and P. Bhattarakosol, “Authenticating User Using Keystroke Dynamics and Finger Pressure,” in *Consumer Communications and Networking Conference*, Las Vegas, NV, 2009, pp. 1-2.
15. C.C. Loy, W.K. Lai, and C.P. Lim, ”Keystroke Patterns Classification Using the ARTMAP-FD Neural Network,” in *Intelligent Information Hiding and Multimedia Signal Processing*, Kaohsiung, 2007, pp. 61-64.
16. N. L. Clarke and S. M. Furnell, “Authenticating mobile phone users using keystroke analysis,” *International Journal of Information Security*, vol. 6, no. 1, 2006.
17. P. Campisi, E. Maiorana, M. Lo Bosco, and A. Neri, “User authentication using keystroke dynamics for cellular phones,” *IET Signal Processing*, vol. 3, no. 4, 2009.
18. S. Karatzouni, N. L. Clarke, “Keystroke Analysis for Thumb-based Keyboards on Mobile Devices,” in *Proc. of the 22nd IFIP Int. Information Security Conf.*, Sandton, South Africa, 2007.
19. A. Ogihara, H. Matsumura, and A. Shiozaki, “Biometric Verification Using Keystroke Motion and Key Press Timing for ATM User Authentication,” in *International Symposium on Intelligent Signal Processing and Communications*, Tottori, Japan, 2006, pp. 223-226.
20. K.S. Killourhy and R.A. Maxion, ”Comparing Anomaly-Detection Algorithms for Keystroke Dynamics,” in Dependable Systems & Networks, Lisbon, Portugal, 2009, pp. 125-134.
21. K.S. Killourhy and R.A. Maxion, ”The Effect of Clock Resolution on Keystroke Dynamics.” In *Proc. of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID-08)*, Cambridge, MA, 2008.
22. P. Panasiuk and K. Saeed, “Influence of Database Quality on the Results of Keystroke Dynamics Algorithms,” *Computer Information Systems - Analysis and Technologies (Communications in Computer and Information Science 245)*, Heidelberg: Springer-Verlag, 2011, pp. 105-112.
23. P. Panasiuk. *Keystroke Dynamics System* [Online]. Available (15.03.2012): http://www.kds.miszu.pl
24. Psylock [Online]. Available (08.02.2011): http://www.psylock.com