Modified Distance Metric That Generates Better Performance for the Authentication Algorithm Based on Free-Text Keystroke Dynamics

Augustin Catalin Iapa and Vladimir-Ioan Cretu

July 16, 2021

Modified Distance Metric That Generates Better Performance For The Authentication Algorithm Based On Free-Text Keystroke Dynamics

Augustin-Catalin Iapa
Department of Computers and Information Technology
Politehnica University Timisoara
Timisoara, Romania
catalin.iapa@gmail.com

Vladimir-Ioan Cretu
Department of Computers and Information Technology
Politehnica University Timisoara
Timisoara, Romania
vladimir.cretu@cs.upt.ro

***Abstract*— The authentication process can be categorized by the number of incorporated factors: something you know, like a username and a password, something you have, like card, token or something you are, like biometrics. Keystroke dynamics has been pointed out as a practical behavioral biometric feature that does not require any additional device for scale up user authentication. The input data of an authentication system based on keystroke dynamics are the typing times on the keyboard. Given that typing times result in time vectors, and these must be compared to see the similarities between them to validate the user, the convenient method that is also used frequently is to calculate the distance of two vectors. The paper aims to analyze the possibilities of increasing the efficiency of an authentication algorithm based on keystroke dynamics, in the sense of reducing the value of the Equal Error Rate (EER). The distance method is used to calculate the similarity between users. The paper (1) analyzes the optimal number of di-graphs, (2) analyzes the optimal time combinations generated by a di-graph to be used and, finally, analyzes the possibility of modify the distance calculation metric. These analyzes aim to reduce the error rate generated by an authentication system based on free-text keystroke dynamics. The authors propose a modification of the Manhattan distance calculation formula that generates better performances.**

Keywords—keystroke dynamics, typing pattern, authentication algorithm, distance metric, di-graphs

I. Introduction

Keystroke dynamics use for users identification was researched for the first time in the 1970`s [1]. Spillane wrote his conclusions about the first investigation in 1975 [2] and Forsen, Nelson and Staron in 1977 [3]. ”Fist of the Sender” was a methodology in World War II that was used to identify, by using the rhythm, the sender of the telegraph [4] [5] [6].

The method of authentication using keystroke dynamics has been exhaustively researched lately. This practice has several fields in which it can be successfully applied, for example, as an additional security method when a user accesses his bank account on the internet or when making a payment in a similar way [4].

The authentication based on keystroke dynamics can be applied for e-mail accounts, or any other online platform that requires a lot of typing. The authentication process can be categorized by the number of incorporated factors: something you know like a username and a password, something you have, like card, token or something you are, like biometrics. [7] A combination of these processes is a strong authentication. [4]

Two-factor authentication is a large scale used approach, in some systems even mandatory, for online services [8]. The traditional password is the first factor and the second factor can be a SMS access code or a PIN generated randomly at the time of authentication [9]. The keystroke dynamics can also be the second factor authentication.

Keyboard analysis can be done without the help of special tools, the classic computer keyboard is enough [10]. For institutions of higher education, “typing signature” is the most cost-effective and reasonable approach to improve online assessment security [11][12].

Keystroke dynamics have been studied mostly in connection to authentication, but some studies, such as [13], have also studied the detection of emotional states of the user who uses the keyboard. Other studies focus on predict users age and gender from unintentional traces, that left behind by use of keyboard and mouse [14]. In [15], the authors explored the relevance of individual a general keyboard and mouse interaction patterns and they had modeled user`s keystroke dynamics and mouse movements with data mining techniques to detect the emotion of users in real-world learning scenarios. In [16], the authors indicates that automatic analysis of human stress from mouse input and keyboard input is potentially useful for providing adaptation in e-learning systems.

If most studies use only data retrieved from the keyboard, there are studies that use a mixed method of user identification, based on data retrieved from the keyboard, but also on data retrieved from the mouse [17]. Additional features, like pressure, are used in addition to time-based features, but to capture this data you need touch screens or other special devices [18]. The stages that a research in this field goes through are: extracting the keyboard features, creating user profiles and updating them and identifying the efficiency criteria [19].

Algorithms of dynamic authentication can be divided into three major groups: estimation of metric distances, statistical methods and machine learning. Methods of keyboard recognition used in the literature are: distance, neural networks, statistical, probabilistic, machine learning, clustering, decision tree, evolutionary computing, fuzzy logic or other [19].

Fixed text keystroke dynamics is applied to the exactly same text typing, both in the user data retrieval phase and in the user identification or verification phase. Being the same text, with the same sequences it is much easier to analyze how it is typed. For example, when a user enters their username and password it is always the same text sequence. In this case you can analyze the similarities or the differences with greater accuracy, remaining at the same typing mode. Difference occurs if every time there can be another text typed from the keyboard, as is the case with free text keystroke dynamics.

„While static text keystroke dynamics biometrics are often used during the logon process to provide a onetime authentication, free text keystroke biometric systems enable continuously authentication of a user during the entire session for increased security” [20].

II. Keystroke dynamics features

The analysis of the keyboard typing pattern can be done by analyzing the times generated by the di-graphs captured from the text typed on the keyboard by a user. A di-graph is a sequence of two consecutive keys. The time for which each keystroke was pressed is named as key hold time or dwell time. [4] The dwell time is the Down-Up time for one single key. In the Fig. 1, the graph shows the distribution of dwell times (DU) for one user.

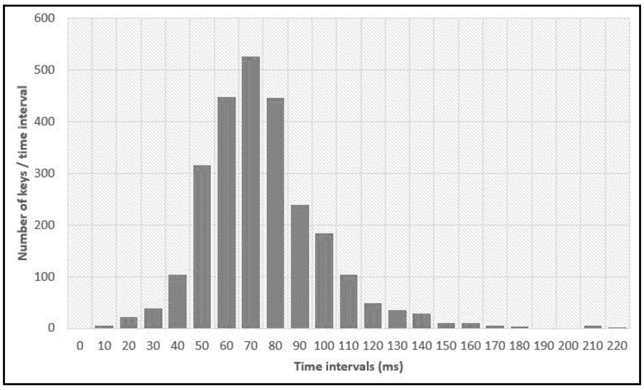

Fig. 1. Distribution of dwell times (DU) from a user

The Release - Press time or Up-Down time between two consecutive keys was called Flight Time [21].

Fig. 2 shows an event required to retrieve the data for a di-graph, a sequence of two consecutive keys pressed by the user.

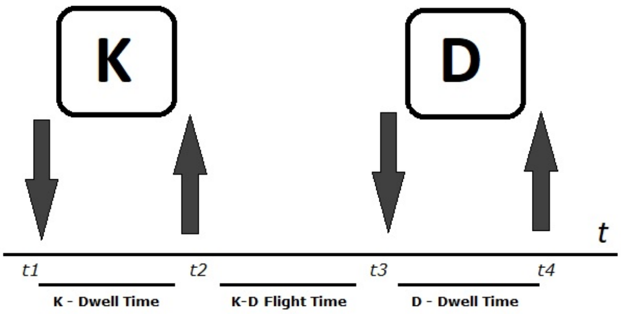

Fig. 2. Key events and time intervals for a di-graph

When pressing two consecutive keys we will take 4 times, noted on the image with t1, t2, t3 and t4. These time periods are captured when the K key (t1) is pressed, when the K key (t2) is raised, when the D key (t3) is pressed and when the D key (t4) is raised. The time the K key is pressed is dwell time and is calculated as the difference between t2 and t1:

$$DU(K) = t2 - t1 \tag{1}$$

Flight Time represents the time period between the 2 keys, or more precisely the time from which the first key is left until the second key is pressed. It is calculated as the difference between t3 and t2 in the image:

$$UD(K - D) = t3 - t2 \tag{2}$$

In the same way as the calculation method for (4.1) Dwell Time is calculated for the second key (in our example, the D key). The time the second key was pressed is calculated as the difference between t4 and t3:

$$DU(D) = t4 - t3 \tag{3}$$

In the Fig. 3 the graph shows the distribution of flight times (UD) for one of the users from data set.

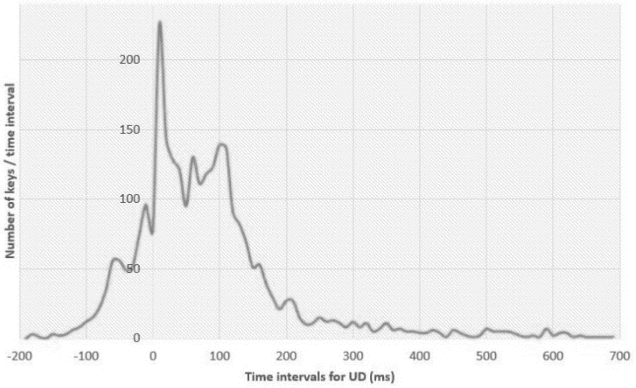

Fig. 3. Distribution of flight times (UD) from a user

Other time periods that can be calculated and used in algorithms are:

- the time period between pressing the two DD (K-D) keys that we calculate, according to the notations in the figure as the difference between t3 and t1:

$$DD(K - D) = t3 - t1 \tag{4}$$

- the time between raising the first key and raising the second key, in our drawing it is about the difference between times t4 and t2:

$$UU(K - D) = t4 - t2 \tag{5}$$

- the total time required to press the 2 keys, in our example, is calculated as the difference between t4 and t1:

$$D1U2(K - D) = t4 - t1 \tag{6}$$

III. Metric distances used for users similarity

The typing times result in time vectors, and these must be compared to see the similarities between them to identify or validate the user, the convenient method that is also used frequently is to calculate the distance of two vectors. In this way we can say that whether some vectors are similar or not similar. To calculate the distance, several types of distances between two vectors are used in the literature. Each distance can be effective in given cases, in certain circumstances. Given two typing samples of the same letters is necessary to approximate their similarity or their difference. It is necessary to choose a measure of the distance of the two samples [10].

A. *Euclidian distance*

Euclidian distance is the most used distance between two points. For points given by Cartesian coordinates in n-dimensional Euclidean space, the distance between the vectors x any y is [TAB14]:

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \qquad (7)$$

Where n is the dimmension of the vectors x and y.

In [20], the authors conclude that despite its intuitiveness and simplicity, Euclidean distance has two limitations: It is highly sensitive to scale variations in the feature variables and it has no means to deal with the correlation between feature variables.

B. *Manhattan distance*

For points given by Cartesian coordinates in n-dimensional space, the Manhattan distance between the vectors x and y is:

$$d(x,y)=\sum_{i=1}^{n}|x_i-y_i| \qquad (8)$$

Where n is the dimmension of the vectors x and y.

The Manhattan distance has the advantages of easy decomposition into contributions made by each variable and simple computation [20].

C. *Bhattacharyya distance*

The Bhattacharyya distance between two vectors, x and y, is defined as:

$$d(x,y)=-\ln(BC(x,y)) \qquad (9)$$

where

$$BC(x,y)=\sum_{i=1}^{n}\sqrt{x_i.y_i} \qquad (10)$$

Where n is the dimmension of the vectors x and y.

D. *Mahalanobis distance*

Mahalanobis Distance has been popularly used to match keystroke features because it handles the correlated data well [20]. The squared Mahalanobis distance between two vectors, x and y, is defined as:

$$(x-y)^2=(x-y)^T S^{-1}(x-y) \qquad (11)$$

where S is the covariance matrix of the data.

Mahalanobis distance is related to the logarithmic likelihood under the assumption that the data follows a multivariate Gaussian distribution, which is a reasonable approximation for most practical data. [20]

IV. Data set used to test metric performance

To research in the field of keystroke dynamics biometrics the researchers need input data obtained from computer users in different real situations. The necessary data are represented by the keys typed on the keyboard but also by the times at which they are pressed. The difference between these times is the keystroke time. Another important piece of information is the time between two keys. The difference between the time a key was released and the time a next key was pressed.

For the purpose of the research the authors developed their own environment to obtain data from 80 volunteers. The authors created a web environment for taking over keys and typing times in JavaScript. A form is created that takes over the keys and typing times while completing a form on a web page. The text written by users is in Romanian language. Most datasets in the literature are texts captured from users who have written in English. The form created to purchase data sets for research purposes was completed by a number of 80 users. They handed over data for 410,633 key-events. Using information obtained from the 410.633 of key events the author rebuilt the characters typed by each user at the keyboard. A total of 200,299 keys were typed on the keyboard.

Each user has his own unique way to type text on the keyboard. This pattern is specific and does not change during a writing session or short term. The typing pattern may change over time or may differ if the same user uses different keyboards. The differences between different users, on the other hand, can be analyzed even visually, as for example in Fig. 4. The graph shows the typing times for user0001 and user0002 from the database. The graph shows how the differences between the typing times for user0001 are larger, both the average of the times and the standard deviation. Most of the time intervals for user0001 are between 50 and 150 milliseconds. Instead, user0002 has a smaller difference between keystrokes. At user0002 most of the time intervals are in the range of 50-75 milliseconds.

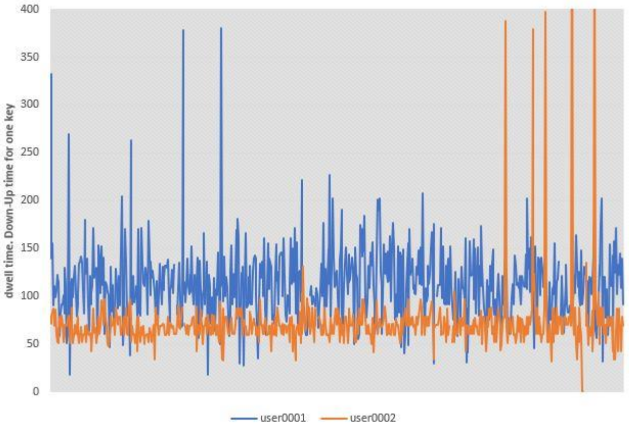

Fig. 4. Typing pattern from two different users

V. The arhitecture of the authentication algorithm

The architecture of the keystroke dynamic authentication system has two important parts. The first is the system training phase part, part in which users enroll in the system providing data on how to type. In this phase a pattern is created for each user and is stored in the database to be used in the continuous authentication phase. The second part is the continuous authentication phase. In this phase the system continuously verifies the users connected with a valid username and password. Throughout the time a user is logged in to the account, the system takes data from it on the typing mode and continuously compares the resulting pattern with the pattern in the database. As long as there is acceptable similarity between the two patterns the user remains logged in to the system. When the system finds that the two patterns are no longer similar, the one taken from the user logged in to the account and the one from the database, the system generates an alarm signal and the user is removed from the account. He can re-enter the account by re-entering the username and password. The architecture of the keystroke dynamic authentication system described in this phrase is visually represented in Fig. 5, the scheme adapted by the author starting from the figure made in the paper [22].

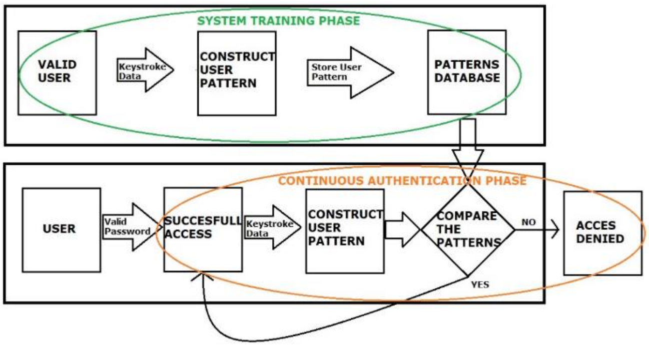

Fig. 5. The architecture of the keystroke dynamics authentication system

VI. Experiments and Results

The data obtained from the 80 users was divided into sets of 1000 keys, obtaining 160 data sequences. From each set it was made a pattern that indicates the calculations of the average keystroke times, made for each di-graphs separately.

A. The optimal number of di-graphs

To calculate the similarities between two vectors (two distinct users or two vectors obtained from the text of the same user) it was applied the calculation of the Manhattan distance at di-graphs. It was considered that the user has successfully accessed the account if the Manhattan distance (calculated between the vector resulting from the user's key events and the vector resulting from the key events of the account to be accessed) was less than a certain threshold. In case it was higher than the certain threshold, it was considered that he failed to access the account.

In order to be able to compare the performances of the different types of tests, tests were performed to work only with the most used di-graphs. For the algorithm that uses Manhattan distance, accesses of the accounts were simulated and took into account one by one, first the most used di-graph, ie the one consisting of the letters IN, then the first two (IN and RE), then the first 3, 4 etc. The performances obtained for the first 200 tests are represented graphically in Fig. 5. The best result obtained in this series of tests is in the case of EER for the analysis of the first 12 di-graphs as well as the frequency of use. The EER value is 13.89%.

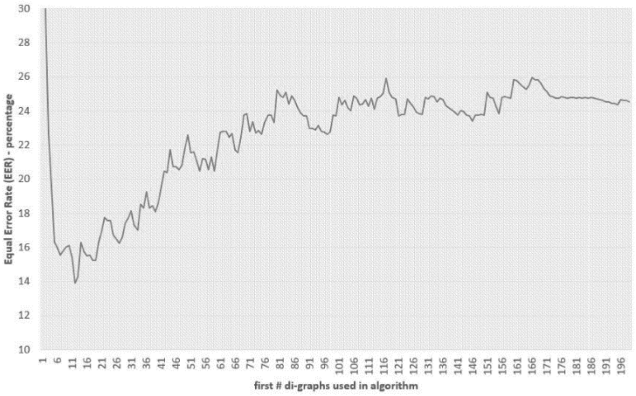

Fig. 6. EER values for Manhattan distance at different numbers of di-graphs

From the analysis of the graph in Fig. 5 it is observed that better values for EER are obtained when analyzing a small number of di-graphs but with high frequency in the text in the database. This also helps in the analysis, being faster to analyze a smaller number of elements. The optimal number of digraphs for calculating the distance, resulting from the experiment, is 12 di-graphs.

B. Choosing the optimal combination of time intervals

In the Table I are the EER values in different scenarios, when using, in certain combinations, the six time intervals generated by a di-graph The time intervals are presented in the paper at Keystroke dynamics features chapter.

TABLE I. The most efficient combinations of times for calculating the distance

	Equal Error Rate	
	Components	***EER (%)***
1	DUtotal, DU1, DU2	5,23
2	DU1, DU2, UD	5,42
3	DU1, DU2	5,69
4	DUtotal, DU1, DU2, UD	6,47
5	All without UU	6,52
6	DU1, DU2, UU, DD	6,61
7	DU1, DU2, UU, DD, UD	7,11
8	All 6 intervals	7,53
9	All without DD	7,58
10	All without UD	7,68

In Fig. 7 are represented graphically FAR (False Acceptance Rate) and FRR (False Rejection Rate) for the case where the best performance was obtained, calculating the distance using the three time intervals of the 6: DU1, DU2 and DUtotal. The EER (Equal Error Rate) value obtained in this case is 5.32%. The simulation was performed using only the first 12 letters, the most common.

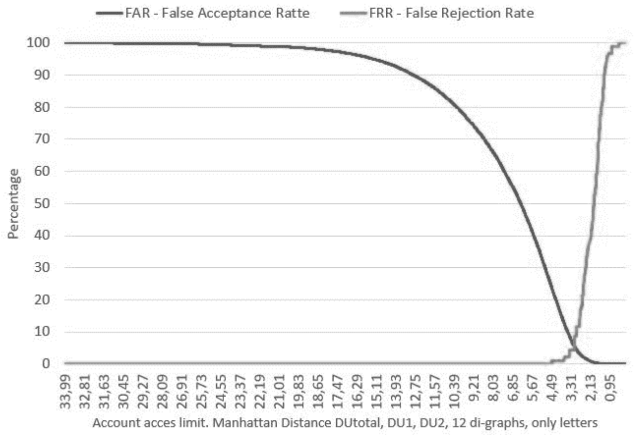

Fig. 7. FAR and FRR for Manhattan Distance DU1, DU2,UD , first 12 di-graphs, only letters

The graph in Fig. 8 shows the Receiver Operating Characteristic (ROC) curve for the best performing case in this till here. EER value = 5.32%.

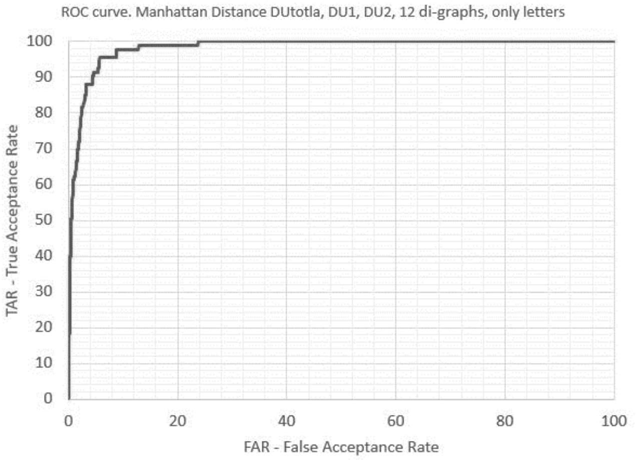

Fig. 8. ROC curve. Manhattan Distance DU1, DU2,UD , first 12 di-graphs, only letters

C. *Modifying the distance calculation formula to generate better performance*

For the third experiment from this paper, we used only the first 12 di-graphs, the most frequently used di-graphs and only three of the six time intervals generated by a di-graph: first key dwell time, second key dwell time and di-graph total time (DU1, DU2 and total DU). These criteria were chosen because they generated the best performance in the experiments performed previously.

Starting from the elements stated above, tests were performed by reducing the weight of the total time of the di-graph from the value of the distance. While first key dwell time and second key dwell time remained with the same weight, the total time value of the di-graph was decreased by multiplying by the coefficient 1 / C like in the (12) formula:

$$d(x,y)=\sum_{i=1}^{n}|xDU1_i-yDU1_i|+\sum_{i=1}^{n}|xDU2_i-yDU2_i|+$$

$$+\sum_{i=1}^{n}\left|\frac{xDUtotal_i-yDUtotal_i}{C}\right| \tag{12}$$

In the graph in Fig. 9 are the values obtained for EER, with the coefficient C in the range [1:15]. The best performance was obtained for C = 3. At this point the performance obtained by the authentication algorithm is 3.27%.

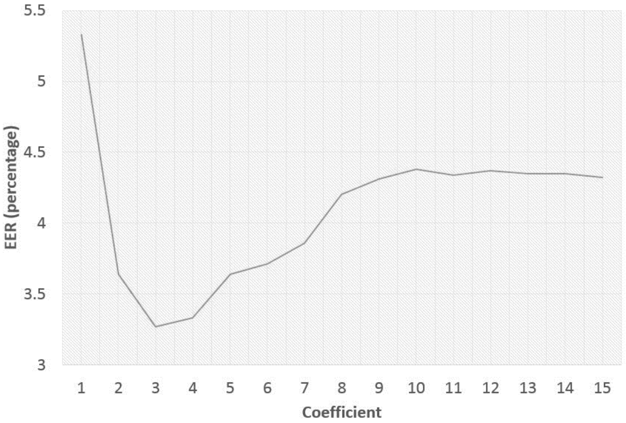

Fig. 9. EER values for different value of coefficient C

The proposed metric modified in this paper, which improves the success rate of the algorithm, is at (13), but only the first 12 di-graphs, the most common times, are used:

$$d(x,y)=\sum_{i=1}^{n}|xDU1_i-yDU1_i|+\sum_{i=1}^{n}|xDU2_i-yDU2_i|+$$

$$+\sum_{i=1}^{n}\left|\frac{xDUtotal_i-yDUtotal_i}{3}\right| \tag{13}$$

Fig. 10 graphically represents the values of False Acceptance Rate (FAR) and False Rejection Rate (FRR) for the best performance obtained in the present research. The intersection, on the graph, of FAR and FRR is at the point of EER = 3.27%.

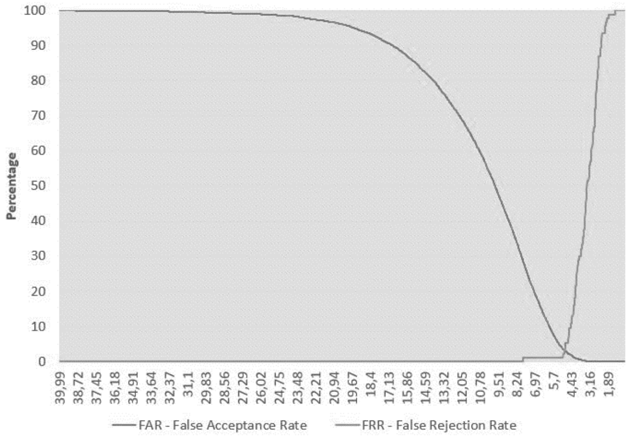

Fig. 10. FAR and FRR graph for the best performance of this research

Fig. 11 shows several ROC curves generated from the experiments performed for the present paper. The best performance obtained during the research, with the proposed new metric is with red on the graph. It can be seen that it is the best performance from the graph.

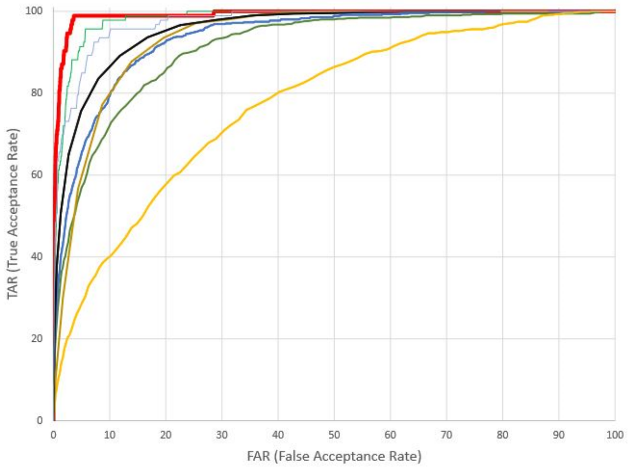

Fig. 11. ROC curve for the best performance of this research - the red one

VII. Conclusions

Two-factor authentication is a large scale used approach, in some systems even mandatory, for online services. The traditional password is the first factor and the second factor can be the keystroke dynamics. Keystroke dynamics is a practical behavioral biometric feature that does not require any additional devices to authenticate a user. Typing pattern analysis using di-graph is a method that generates good results, which achieves performance and less than 10% in terms of Equal Error Rate (EER).

First, this paper analyzed the optimal number of di-graphs, and the best performance was obtained when calculating the distance between vectors only with the first 12 di-graphs, the most frequently used di-graphs by users. On the other hand, the best performance was obtained when calculating the distance using only three of the six time intervals that can be generated from a di-graph: first key dwell time, second key dwell time and total di-graph time (DU1, DU2 and DU total).

Considering the conclusions of the experiments, the present paper proposed a modification of the Manhattan metric for calculating the distances. By modifying it, the performance of the authentication algorithm was improved by 38.53%. The EER value obtained from the metric change is 3.27%, compared to 5.32% obtained with the classic Manhatten formula in our experiments.

References

[1] Y. Zhong, Y. Deng and A. K. Jain, "Keystroke dynamics for user authentication," 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, 2012, pp. 117-123, doi: 10.1109/CVPRW.2012.6239225.

[2] G. Forsen, M. Nelson, and R. Staron, Jr. "Personal attributes authentication techniques", Technical Report RADC-TR-77-333, Rome Air Development Center, October 1977.

[3] R. Spillane, "Keyboard Apparatus for Personal Identification", IBM Technical Disclosure Bulletin, vol. 17, no. 3346, 1975.

[4] Banerjee, Salil & Woodard, D.L.. (2012). Biometric Authentication and Identification Using Keystroke Dynamics: A Survey. Journal of Pattern Recognition Research. 7. 116-139. 10.13176/11.427.

[5] J. R. Vacca. Biometric Technologies and Verification Systems. Butterworth-Heinemann, 1 edition, 2007.

[6] T. Dunstone and N. Yager. Biometric System and Data Analysis: Design, Evaluation, and Data Mining. Springer, 1 edition, 2008.

[7] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic Authentication Guideline: Recommendations of the National Institute of Standards and Technology. Technical Report 800-63,National Institute of Standards and Technology (NIST), Apr. 2006.

[8] Kang, Jeonil & Nyang, Daehun & Lee, KyungHee. (2014). Two-factor face authentication using matrix permutation transformation and a user password. Information Sciences. 269. 1–20. 10.1016/j.ins.2014.02.011.

[9] Dasgupta, Dipankar & Roy, Arunava & Nag, Abhijit. (2016). Toward the design of adaptive selection strategies for multi-factor authentication. Computers & Security. 63. 10.1016/j.cose.2016.09.004.

[10] BERGADANO, F., GUNETTI, D., AND PICARDI, C. 2002. User authentication through keystroke dynamics. ACM Transactions on Information and System Security 5, 4.

[11] Jay R. Young, Randall S. Davies, Jeffrey L. Jenkins & Isaac Pfleger (2019): Keystroke Dynamics: Establishing Keyprints to Verify Users in Online Courses, Computers in the Schools, DOI: 10.1080/07380569.2019.1565905

[12] Fabian Monrose and Aviel Rubin. 1997. Authentication via keystroke dynamics. In Proceedings of the 4th ACM conference on Computer and communications security (CCS '97). Association for Computing Machinery, New York, NY, USA, 48–56. DOI:https://doi.org/10.1145/266420.266434

[13] A. Messerman, T. Mustafic, S. A. Camtepe and S. Albayrak. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. Proceedings of IEEE International Joint Conference on Biometrics. 1–8, 2011

[14] Avar Pentel. 2017. Predicting Age and Gender by Keystroke Dynamics and Mouse Patterns. In Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17). Association for Computing Machinery, New York, NY, USA, 381–385. DOI:https://doi.org/10.1145/3099023.3099105

[15] S. Salmeron-Majadas, R. S. Baker, O. C. Santos and J. G. Boticario, "A Machine Learning Approach to Leverage Individual Keyboard and Mouse Interaction Behavior From Multiple Users in Real-World Learning Scenarios," in IEEE Access, vol. 6, pp. 39154-39179, 2018, doi: 10.1109/ACCESS.2018.2854966.

[16] Y. M. Lim, A. Ayesh and M. Stacey, "Detecting cognitive stress from keyboard and mouse dynamics during mental arithmetic", Proc. Sci. Inf. Conf. (SAI), pp. 146-152, Aug. 2014.

[17] Lozhnikov, Pavel & Sulavko, Alexey & Ekaterina, Buraya & Viktor, Pisarenko. (2017). Authentication of Computer Users in Real-Time by Generating Bit Sequences Based on Keyboard Handwriting and Face Features. Voprosy kiberbezopasnosti. 24-34. 10.21681/2311-3456-2017-3-24-34.

[18] Teh, Pin Shen & Teoh, Andrew & Yue, Shigang. (2013). A Survey of Keystroke Dynamics Biometrics. TheScientificWorldJournal. 2013. 408280. 10.1155/2013/408280.

[19] Kochegurova, Elena & Luneva, Elena & Gorokhova, Ekaterina. (2019). On Continuous User Authentication via Hidden Free-Text Based Monitoring: Volume 2. 10.1007/978-3-030-01821-4_8.

[20] Zhong, Yu & Deng, Yunbin. (2015). A Survey on Keystroke Dynamics Biometrics: Approaches, Advances, and Evaluations. 10.15579/gcsr.vol2.ch1.

[21] D. Stefan and D. Yao. Keystroke-Dynamics Authentication Against Synthetic Forgeries. In International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010.

[22] Pilsung Kang and Sungzoon Cho. 2015. Keystroke dynamics-based user authentication using long and free text strings from various input devices. Inf. Sci. 308, C (July 2015), 72–93. DOI:https://doi.org/10.1016/j.ins.2014.08.070