Continuous Authentication Based on User Interaction Behavior

Long Chen
Institute of Computer Forensics
Chongqing University of Posts and Telecommunications
Chongqing, China
chenlong@cqupt.edu.cn

Yi Zhong
College of Computer Science and Technology
Chongqing university of Posts and Telecommunications
Chongqing, China
2273197134@qq.com

Weidong Ai
College of Computer Science and Technology
Chongqing university of Posts and Telecommunications
Chongqing, China
452926710@qq.com

Difang Zhang
College of Computer Science and Technology
Chongqing university of Posts and Telecommunications
Chongqing, China
991146797@qq.com

***Abstract*—Continuous authentication (CA) is the process which continuously verifying a user based on their on-going interaction with a computer system. In this paper, we propose an adaptive continuous authentication method based on the changes of context, in which providing protection for the user's on-going interaction with computer in different contexts. In order to prevent a situation where an attacker tries to avoid detection by limiting to one input device, we considered both keystroke and mouse usage behavior patterns. In this research, collecting 30 users' data in an uncontrolled environment, extracting the user behavior feature from data by a new feature extraction method, using fusion technology to identify users, and then, according to the recognition result we can judge whether the current user is a real user or not. The experiment result shows that our scheme has a false acceptance rate (FAR) of 0%, a false rejection rate (FRR) of 2.04%, and the authentication time that between 10 seconds and 60 seconds for authentication.**

Keywords—authentication, context, adaptive, mouse dynamics, keystroke dynamics

I. INTRODUCTION

For most existing computer systems, the internal resources of the system are available to the user when he/she successfully logins. The attacker may pretend to be a real user into the system when the user logins and leaves the computer, and that will cause a security crisis of internal system. Therefore, the continuous check of user's identity is extremely important. We will select two kinds of user's biological behaviors, mouse dynamics(MD) and keystroke dynamics(KD), to authenticate the user because there are non-replicable, anti-lose-easily, stable, unique, ubiquitous and they do not need additional equipment.

Many of the articles study of KD or MD, they have restricted user's environment or assigned tasks to users. However, in real life, user can login system by different personal computer (PC) at anytime and anywhere. Sometimes, the user may login system by a laptop or desktop computer, or he/she can use the PC in different places (i.e. homes, offices, cafes, etc.). For example, the user uses the desktop computer in the office during the day, and the user uses the laptop at home in the evening, his/her behavior of logging in to the system at night may be different from the daytime. So, the changes in context will affect the user's behavior, and differences in these behaviors may cause that the authentication system to consider the real user to be an imposter.

For a CA method, the method should not only detect the imposter, but avoid treating real users as imposters when the user has absolute freedom in different contexts. This is more in line with the realization of life scenarios, and achieving personalized continuous authentication.

The contributions made in this paper are as follows:

- In this research, we have explored the possibility to perform CA without constrains. We proposed an adaptive CA method based on the change of context and quantified the user's context information.
- We have come up with a novel KD feature selection technique during our research.

II. RELATED WORK

KD is the way that a user type on his/her keyboard. A KD based on the authentication system is low cost and easy to implement. In such a system, the typing rhythm(i.e. keystroke timing information) has to be captured for authentication. Most of the KD-based research only pay attention to static authentication and neglect the CA. We can divide CA research using KD into three groups based on the keystroke timing feature, n-graph duration was used as a feature [1],while [2] have used n-graph duration along with time features and [3] have used word specific n-graph features. Furthermore, a large number of classification method has been studied for mapping these features into authentication decisions. Broadly, these approaches fall in one of two categories: clustering algorithm [4] and classification algorithm [5], and the latter generally showing higher equal error rate (EER).

MD has been defined as the way users are interacting with their system through the mouse. For MD based biometric authentication, we need to capture the mouse trajectory and mouse click data while users interacting with their system. [6]

This work is supported by the National Key Research & Development Program of China. No. 2017YFB0802300.

978-1-7281-2827-6/19/$31.00 ©2019 IEEE

collected mouse movements from users playing a memory game, and attempted to identify the users from data. [7] calculated angle based features on the points that the user clicked or hovered at with the mouse and identified the users from data. Most of MD-based research considers the given environment, because the collecting data of mouse is more dependent on the type of mouse and the environment in which the mouse is used.

The method verifies identity by KD or MD has been around for many decades [8]. While both methods have seen a large amount of research, there has been less work done on combining these two techniques into one system. Only very few studies exist where researchers have used a combination of KD and MD for CA [9, 10, 11], but all of these studies were conducted in a controlled environment.

III. Adaptive Continuous Authentication Method

On the one hand, no one can always behave in exactly the same manner, and for a real user, sometimes his/her behavior deviates from normal behavior. On the other hand, in our daily life, the user can use the different PC login the system at any time or any place, and the user can use this system in an uncontrolled environment. According to these cases, a good CA method needs to consider the different contexts of users and the behavioral deviations of users in the system.

This paper proposes an adaptive CA method, quantifying the user's context information when the user login, and getting the trust(T) of the system in the genuineness of the current user that the deviations from the way this user performs various actions on the system. If the user's behavior in a certain period of time is judged to be a real user, then the system's trust in the genuineness of this user will increase. If the user's behavior in a certain period of time is judged to be an imposter, then the trust of the user will decrease. Finally, the quantitative context information is linked to the trust of the current user.

This method consists of three parts: context acquisition module(CAM), trusted evaluation module(TEM) and dynamic evolution module(DEM). The method is shown in Fig.1.

A. Context Acquisition Module

After the user successfully login the system, the system will judge whether the current user is a newcomer or not. If he/she is a newcomer, the procedure goes to trains the training model. Otherwise, we will enter to the CAM. The main function of the module is to continuously collect the real-time biological behavior and context information of the user, quantify the user's context information and extract the real-time feature at a fixed time according to the acquired biological behavior (including MD and KD) dataset.

Different context information reflects the user's environment when he/she login the system, and the environment reflects the daily habits of user. We find a way to quantify these context information into a threshold U.

B. Trusted Evaluation Module

The main function of the module is to match the extracted feature with the template library, and adjusts T of the current user according to the matching result. Next, comparing the value of T with U, if T<U, the current user will be logged out of the system, otherwise, we will enter to the DEM.

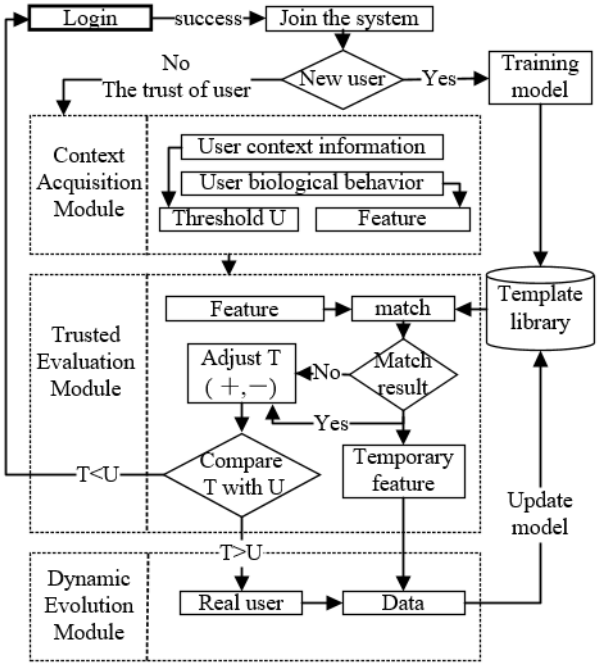

Fig. 1. Adaptive continuous authentication method

It is necessary to extract the feature value in a fixed time from the KD and MD data when authenticating a user, because the same amount of data can ensure the accuracy of the authentication. Data in a fixed time may contain four combinations: KD data and MD data; MD data and KD data; KD data and KD data; MD data and MD data.

Then, we need to choose the right classifier to match feature. Fusion was proved to reduce error rates in different data source compared to a single classifier [12]. In this paper, there are two different data source(KD data and MD data). For this point, we use feature-level fusion method and decision-level fusion method to deal with two different data source, and we use single classifier to deal with features of the same data source.

1) Feature-level fusion

We use feature-level fusion method to form a new feature which combines the mouse feature and the keystroke feature.

2) Decision-level fusion

When the data sources of the two nodes are different, finding the classification algorithm that is most suitable for its data source according to the experiment, and classify them separately. Finally, the results are sent to the final module that produces the decision, and the results are obtained by using the majority voting mechanism.

3) Single classifier

When the data sources of the two nodes are the same, finding the classification algorithm that the most suitable for its data source according to the experiment.

The trust(T) of the current user depends on the user's various operation behavior in the system. There are four combinations of data in a fixed time, and each case has four results. We have separately analyzed these 16 cases, as shown in Table I. The left column in Table I indicates the four cases of a node, and the beginning of the line indicates four kinds of results. The values in the table indicate the addition and subtraction of the current T of the user. In the table, "T" indicates that the current user is a real user, "F" indicates that the current user is an imposter.

TABLE I. ADJUST VALUE OF TRUST

Node/Result	T-T	F-F	T-F	F-T
Mouse-Mouse	+1	-1	-0.5	-0.5
Mouse-Keystroke	+1	-2	-1	-1
Keystroke-Mouse	+1	-2	-1	-1
Keystroke-Keystroke	+1	-1	-0.5	-0.5

C. Dynamic Evolution Module

When user first uses the system, the proficiency is not strong, but the proficiency will become stronger with time goes by, so the user's operation is not always the same and slowly changes over time. We propose a dynamic evolution module. When the T of the user's authenticity is increase, the feature in the fixed time is dynamically updated to the template library. The user context information will be updated to the database when the user exits the system.

IV. DATASET DESCRIPTION AND FEATURE EXTRACTION

In order to obtain the user's context information and biological behavior, we have developed a Web site. This paper invited 30 volunteers to participate in our experiments and they used our system to collect data continuously for 30 days. From previous studies, we learned that there are many disadvantages of collecting data in a controlled environment and performing specific tasks on a PC. In this case, the collected data represents behavior that is not their normal behavior. In this research, we consider user behavior in real life, so, collecting data in conditions as the follows:

- Do not assign tasks to users.
- We collect the user's context information and biological behavior in an uncontrolled environment.
- All volunteers are free to choose an operating system to eliminate the impact of fixed hardware on the user's personalized operation.

A. Context Information

In this research, context information consists of the user's browser information, login location, online time and login time. The user's browser information includes the browser name and version number. The login location is obtained by the user's IP address.

The online time is the time difference between when the user login time with logout time. The online time is not static, but the user online time in a period of time. Therefore, according to the results of user experiments, this paper consider that 5 minutes as the time period, each time period is represented by a number (i.e. the online time of the user is 6 minutes and 20 seconds, the online time is 2 in this paper).

The user's login time is not static, but the user login system in a period of time. Therefore, according to the results of user experiments, this paper divides the day into 12 time periods, each time period is represented by a number. As shown in the Table II.

TABLE II. 12 TIME PERIODS

Type	1	2	3	…	11	12
Time period	00:00-02:00	02:00-04:00	04:00-06:00	…	20:00-22:00	22:00-24:00

In this paper, we assume that each context information has a security factor that is calculated from the context information of user history, and the security factor(range [0, 1]) is adaptive by the daily habits of user. The algorithm of the security factor is as follows:

$$P_i = \frac{sum_{ij}}{sum_j} \tag{1}$$

sum_j represents the sum of times in a category of context information(i.e. j represents the category of the login location); sum_{ij} the total of the specific cases in this category(i.e. the total number of times this user logged in at the i location).

This user in the case of i, the adaptive threshold U_i is calculated according to the security factor, n represents the total number of context information categories, h_j is the weight of the context information category j, P_i represents the current safety factor in the case of i, and the calculation formula is:

$$U_i = \sum_{j=1}^{n} (1 - P_i) * h_j \tag{2}$$

The value of U will be smaller when a user login at a location for a long time, and will be larger when the user login at a new location. In order to prevent the real user from login the system in a strange context and the system makes a wrong judgment on the real user, so, we limit U to less than 9. At the same time, the imposter may imitate the real user's daily context information, causing the system cannot exclude the imposter, so we limit U to be more than 6, $U \in [6,9]$. Therefore, the expression of U is as follows:

$$U_i = 6 + \sum_{j=1}^{n} (1 - P_i) * h_j \tag{3}$$

Considering the importance of various contexts, the weight of the login location to $h_1 = 1$, the weight of the login time to $h_2 = 1$, the weight of the browser to $h_3 = 0.8$, and the online time to $h_4 = 0.2$.

B. Keystroke and Mouse Event

Table III shows the data format of the collected keystroke and mouse events. The " Seq." is a sequence that indicates the order in which events occur. The "Tool" is a type that indicates whether the tool is mouse or keyboard. The "action" is the

behavior. The “Value” represents the key value in the KD and represents the screen position in MD. The “Time” indicates the time of the current behavior, unit: ms.

TABLE III. DATE STRUCTURE FOR KEYSTROKE AND MOUSE ENENT

Seq.	Tool	Action	Value	Time
n	Keystroke	KeyDown KeyUp	string	ms
	Mouse	MouseMove MouseLDown MouseLUp MouseRDown MouseRUp	x_y	ms

C. Keystroke Dynamics Features

The traditional KD features are divided into the following five types[13], as shown in Fig.2, which detail and visually describes the relationship and feature of five types. However, the amount of training required for this type data sets is too large and does not meet the requirements of this article. We need to re-extract the feature on this basis.

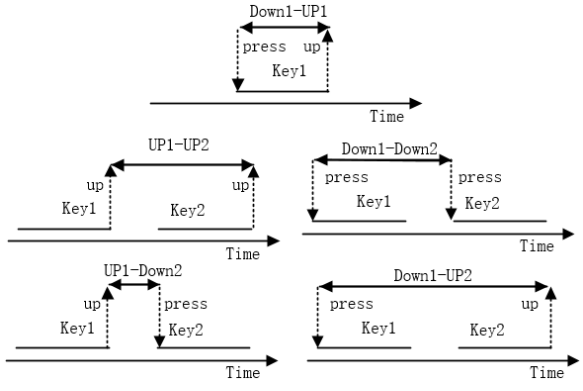

Fig. 2. keystroke dynamic features

According to the different keystroke styles and the most important feature in the KD is the keystroke speed. We propose a novel feature extraction method, which makes the feature extracted by each user have obvious differences, can clearly distinguish the feature of each user.

We extract five types of features from the KD of a user in a fixed time. All the time of each type of feature is sorted in ascending order, and all the time in each feature is divided into n time segments, calculated the proportion of each time segments, as shown in Fig.3.

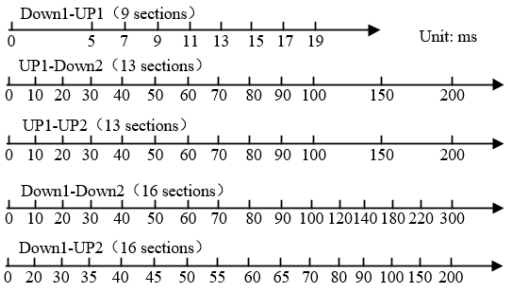

Fig. 3. Keystroke dynamics dimension division

The Down1-UP1 is divided into 9 section (unit: ms): (0, 5], (5, 7], (7, 9], (9, 11], (11, 13], (13, 15], (15, 17], (17, 19], (19, ∞).

The UP1-UP2 is divided into 13 section (unit: ms): (0,10], (10,20], (20,30], (30,40], (40,50], (50,60], (60, 70], (70, 80], (80, 90], (90, 100], (100, 150], (150, 200], (200, ∞).

The Down1-Down2 is divided into 16 section (unit: ms):(0,10], (10,20], (20,30], (30,40], (40,50], (50,60], (60 , 70], (70, 80], (80, 90], (90, 100], (100, 120], (120, 140], (140, 180], (180, 220], (220, 300], (300, ∞).

The UP1-Down2 is divided into 13 section (unit: ms): (0,10], (10,20], (20,30], (30,40], (40,50], (50,60], (60 , 70], (70, 80], (80, 90], (90, 100], (100, 120], (120, 140], (140, 180], (180, 220], (220, 300], (300, ∞).

The Down1-UP2 is divided into 16 section (unit: ms): (0,20], (20,30], (30,35], (35,40], (40,45], (45,50], (50, 55], (55, 60], (60, 65], (65, 70], (70, 80], (80, 90], (90, 100], (100, 150], (150, 200], (200, ∞).

As shown in Table IV, for the proportion of the three users in the 9 sections of the Down1-UP1, the feature of each user is very obvious.

TABLE IV. COMPARISON OF USER KEYSTROKE DYNAMICS FEATURES IN DOWN1-UP1

Period Unit：ms	User A	User B	User C
0-5	0.0818	0.0189	0.0050
6-7	0.3329	0.0844	0.0136
8-9	0.2178	0.3396	0.0408
10-11	0.2136	0.2761	0.1237
12-13	0.1262	0.1321	0.1738
14-15	0.0250	0.0506	0.2031
16-17	0.0014	0.0209	0.1588
18-19	0.0014	0.0149	0.1109
20+	0	0.0626	0.1702

D. Mouse Dynamics Feature

We use four MD feature extraction methods:

1) Average speed per direction of motion: The average speed per movement direction records the user's mouse movement speed in eight different directions along the screen (Ahmed and Traore [14]). The direction of motion is determined by calculating the angle between the coordinates of motion starting point and the moving point.

2) Regional distribution: Based on the idea that Shen [15] divided the screen into nine different regions on a fixed screen, we adaptively divide the screen size into 9 regions according to the user's screen, and calculate the percentage of movements that nine different regions of the screen.

3) L / R click duration: The left mouse button and right mouse button duration is the time required for the user to click the mouse button after stopping the cursor. Zheng [16] proves that this is a distinguishing feature.

4) area average moving speeds: 9 area average moving speeds contains the average moving speed of the user in the nine areas of the adaptive screen division.

V. DATA PROCESSING

We need to select the appropriate time period, extract the user's behavior feature according to data of the time period, and select the appropriate classifier according to the feature.

Compared with previous studies, we use SVM, Bayes and ensemble classifier (ensemble classifier based on SVM and Bayes) as classifier in our study, because it can provide better classification accuracy when the training set is small and faster classification speed than others algorithm.

In KD, we select 3 seconds, 5 seconds and 10 seconds as a time period, extract the feature from the data, and test it by 10-fold cross-validation [17]. The results are shown in Table V.

TABLE V. EER IN EACH CLASSIFIER OF KD

Time/Classfier	Bayes	SVM	ensemble classifier
3 seconds	0.0988	0.0371	0.0638
5 seconds	0.0523	0.0337	0.0511
10 seconds	0.0224	0.0342	0.4160

In order to achieve the fastest speed, good accuracy and the least amount of data when the method authenticates the current user identity. So, in MD, we select 3 seconds and 5 seconds as a time period, extract the feature from the data, and test it by 10-fold cross-validation, the results are shown in Table VI.

TABLE VI. EER IN EACH CLASSIFIER OF MD

Time/Classfier	Bayes	SVM	ensemble classifier
3 seconds	0.1000	0.3829	0.1343
5 seconds	0.0301	0.2650	0.0787

Based on the experimental results, we choose 5 seconds as the time period of KD and MD. The EER is the smallest when the SVM classifier match KD feature or the Bayes classifier match MD feature.

In the feature-level fusion, MD feature and KD feature constitute a new feature, we use SVM, Bayes, and ensemble classifier to test the new feature when the node's time is 5 seconds, and the results are shown in Table VII. So, we chose ensemble classifier as the classifier in feature-level fusion of KD feature and MD feature.

TABLE VII. EER OF EACH CLASSIFIER IN FEATURE-LEVEL FUSION

Type/ Classfier	Bayes	SVM	ensemble classifier
Keystroke-Mouse	0.0301	0.0197	0.0104

In the decision-level fusion, according to Table V and Table VI, the Bayes classifier is the most suitable for MD and the SVM classifier is the most suitable for KD when the node's time is 5 seconds. The results of decision-level fusion are shown in Table VIII.

TABLE VIII. EER OF EACH CLASSIFIER IN DECISION-LEVEL FUSION

Type/ Classfier	Bayes	SVM	decision-level fusion
Mouse	0.0301		0.0122
Keystroke		0.0337	

It is found that the fusion technology could improve the accuracy. Therefore, the authentication time period for the data that we obtained in this paper is set to 10 seconds, that is, the time period of two KD or MD. And there is four combinations in this 10-second period: 5 seconds KD data+5 seconds KD data; 5 seconds KD data+5 seconds MD data; 5 seconds MD data+5 seconds MD data; 5 seconds MD data+5 seconds KD data. When the user uses the system, the user's trust(T) is increased or decreased according to the matching result within 10 seconds, and the specific method is shown in Table I.

VI. RESULT ANALYSIS

We invited 30 volunteers to use our data collection website for 30 days, the volunteer login 1 to 4 times at one day, each time is about 7 minutes. Assuming that one of the volunteers is used as a real user, the remaining 29 volunteers can be used as imposter. We will build a model and a template library from the first ten days of real user's data, use the remaining 20 days of data as a test set. This model and template library are constantly updated as data increases. Table IX shows the number of actions that we have on average available for testing for each user to measure our method performance.

TABLE IX. AVERAGE NUMBER OF ACTION TESTED FOR EACH USERS

Action	Genuine User（1）	Counterfeiter（29）
Mouse Action	1.6×10^4	35.4×10^4
Key Action	8.9×10^4	198.1×10^4
Total	10.5×10^4	233.5×10^4

After the model is built, when the user logs in to the system again, step-1 is to obtain user context information and calculate U. Step-2, we obtain the mouse or keystroke dynamics data for consecutive 10 seconds. Step-3, the feature is extracted separately. Step-4 is to match the extracted features. Step-5, T is adjusted according to the matching result. Step-6, the size of T and U is compared to determine the current user identity. The above steps are repeated until the user exits the system.

TABLE X. RESULTS OF USER AUTHENTICATION

User	FAR	FRR	User	FAR	FRR
User 1	0%	1.67%	User 16	0%	0%
User 2	0%	2.44%	User 17	0%	1.96%
User 3	0%	0%	User 18	0%	3.77%
User 4	0%	1.25%	User 19	0%	4.35%
User 5	0%	3.77%	User 20	0%	1.37%
User 6	0%	3.33%	User 21	0%	0%
User 7	0%	0%	User 22	0%	2.17%
User 8	0%	5%	User 23	0%	2.13%
User 9	0%	4%	User 24	0%	1.64%
User 10	0%	3.03%	User 25	0%	2.56%
User 11	0%	3.23%	User 26	0%	0%
User 12	0%	0%	User 27	0%	0%
User 13	0%	0%	User 28	0%	1.61%
User 14	0%	2.38%	User 29	0%	1.61%
User 15	0%	5.88%	User 30	0%	2.04%
Average		**FAR=0%**		**FRR=2.04%**	

Finally, the FAR and FRR are calculated based on the test results. Table X shows our test results. It can be seen that the average probability of authenticating a real user as an imposter is FRR=2.04%, and the average probability of authenticating an imposter as a real user is FAR=0%.

TABLE XI. RESULT COMPARISON WITH PREVIOUS RESEARCH

Ref.		[9]	[10]		[11]	Us
Data	mouse	1.4×10^4	34.6×10^6		8.5×10^5	37×10^4
	keystroke	4.5×10^5	12.4×10^5		13.8×10^5	21×10^5
	other	GUI:2635	11 sensor		No	No
Restrict		3 tasks	Fixed equipment; 1 task		Same equipment; installation	No
Users		31	67		53	30
Time		10min	30s	5min	Not stated	10s-60s
Result	FAR	2.24%	0.4%	0.1%	5.7%	0%
	FRR	2.10%	1%	0.2%	0.1%	2.04%

Table XI shows the result comparison between our studies with the previous research on CA. In this table, the "Data" shows the data source and the amount of data in each study, the "Restrict" shows the specified action for the user, the "Users" shows the number of testers, the “Time” shows the length of the authentication time, and the "Result" shows the result of FAR and FRR.

In these studies, for the data sources, in the study of [10] need to collect data by 11 sensors. For whether to restrict user behavior, in the study of [9] that the user needs to complete three tasks, in the study of [10] that the user needs to complete a task at the same device, and in the study of [11] that all users must operate the system on the same device.

In this paper, we can identify the current user with different PC, at anytime and anywhere. The experiment result is that the FAR of 0%, the FRR of 2.04%, and the authentication time that between 10 seconds and 60 seconds for authentication. Compared with existing research, this paper has higher accuracy and shorter authentication time.

REFERENCES

[1] A. Messerman, T. Mustafić, S. A. Camtepe and S. Albayrak, "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," 2011 International Joint Conference on Biometrics (IJCB). Washington, pp. 1-8, IEEE, December 2011.

[2] J. Ferreira and H. Santos, “Keystroke dynamics for continuous access control enforce-ment,” Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'12). vol. 1, pp. 216–223, December 2012.

[3] R. Janakiraman and T. Sim, "Keystroke dynamics in a general setting." International Conference on Biometrics. Seoul, pp. 584-593, August 2007.

[4] T. Shimshon, R. Moskovitch, L. Rokach and Y. Elovici, "Continuous verification using keystroke dynamics," 2010 International conference on computational intelligence and security. Nanning, pp. 411-415, December 2010.

[5] R. Giot, M. El-Abed, B. Hemery, C. Rosenberger, "Unconstrained keystroke dynamics authentication with shared secret," Computers & security. vol. 30, no.6-7, pp. 427-445, 2011.

[6] H. Gamboa and F. Ana, "A behavioral biometric system based on human-computer interaction," Biometric Technology for Human Identification. vol. 5404, pp. 381-393, August 2004.

[7] N. Zheng, A. Paloski, and H. Wang, “An efficient user verification system via mouse movements,” Proceedings of the 18th ACM Conference onComputer and Communications Security, New York, USA, pp. 139–150, October 2011.

[8] R. Gaines, W. Lisowski, S. Press, and N. Shapiro, “Authentication by keystroke timing: some preliminary results,” Technical Report Rand Rep. R-2560-NSF, Rand Corporation, 1980.

[9] K. O. Bailey, J. S. Okolica, and G. L. Peterson, “User identification and authenticationusing multi-modal behavioral biometrics,” Computers & Security, vol. 43, pp. 77–89, June 2014.

[10] L. Fridman, A. Stolerman, S.Acharya, P. Brennan, P.Juola, R. Greenstadt, et al. “Multi-modal decision fusionfor continuous authentication,” Computers & Electrical Engineering, vol. 41, pp. 142–156, January 2015.

[11] S. Mondal and P. Bours, “A study on continuous authentication usinga combination of keystroke and mouse biometrics,” Neurocomputing,vol. 230, pp. 1–22, 2017.

[12] R. Giot, E. Mohamad El-Abed and R. Christophe, "Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis," 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Piraeus, pp. 1-15, July 2012.

[13] P. Kang and S. Cho, “Keystroke dynamics-based user authentication using long and freetext strings from various input devices,” Information Sciences, vol. 308, pp. 72–93, July 2015.

[14] A. A. E. Ahmed and I. Traore, “A new biometric technology based on mouse dynamics,” IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 3, pp. 165–179, August 2007.

[15] C. Shen, Z. Cai, X. Guan, H. Sha, and J. Du, “Feature analysis of mouse dynamics in identity authentication and monitoring,” IEEE International Conference on Communications, pp. 673–677, August 2009.

[16] N. Zheng, A. Paloski, and H. Wang, “An efficient user verification system via mouse movements,” Proceedings of the 18th ACM Conference onComputer and Communications Security, New York, USA, pp. 139–150, October 2011.

[17] Y. Bengio and Y. Grandvalet, “Bias in estimating the variance of K - Fold Cross-Validation,” Statistical modeling and analysis for complex data problems, New York, pp. 75–95, 2005.