International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

On keystrokes as Continuous User Biometric Authentication

Suhail Javed Quraishi, Sarabjeet Singh Bedi

Abstract*: Authentication of a user through an ID and password is generally done at the start of a session. But the continuous authentication system observe the genuineness of the user throughout the entire session, and not at login only. In this paper, we propose the usage of keystroke dynamics as biometric trait for continuous user authentication in desktop platform. Biometric Authentication involves mainly three phases named as enrollment phase, verification phase and identification phase. The identification phase marks the accessed user as an authenticated only if the input pattern matches with the profile pattern otherwise the system is logout. The proposed Continuous User Biometric Authentication (CUBA) System is based on free text input from keyboard. There is no restriction on input data during Enrolment, Verification, and Identification phase. Unsupervised One-class Support Vector Machine is used to classify the authenticated user's input from all the other inputs. This continuous authentication system can be used in many areas like in Un-proctored online examination systems, Intrusion & Fraud Detection Systems, Areas where user alertness is required for entire period e.g. Controlling Air Traffic etc.

Index Terms*: Continuous Authentication, Free Text, Keystroke Dynamics, Unsupervised, Support Vector Machine.

I. INTRODUCTION

Biometric verification is the mean of identifying an individual by evaluating one or more biological traits of that individual. In other words, biometrics is the study of physiological or behavioral characteristics of an individual to identify him. Features like Face, Iris, Fingerprint, DNA pattern, Hand geometry, Signature flow, Voice, Keystroke, Mouse movement, Gait are few of the biometric traits that are unique to an individual. These features are mainly used for identification and access control in computer-based security systems.

There are many advantages to using biometric features for authentication. It is more secure, as the person to be authenticated should present at the point of authentication. This technology is capable of identifying people consistently, swiftly, and reliably. Biometrics characteristics cannot be stolen or conjectured. Another advantage is that it is less exposed to damage and sudden changes. It has the capability of high individual identification accuracy. Yet another advantage is that it is user-friendly, less time consuming, hard to falsify, require negligible training, and less expensive in the long run.

Whereas typical methods like PINs and Passwords are easy to forget, causing people to write them down and consequently can be stolen, and can at times be hacked. Biometric authentication has already gained widespread acceptance in the form of fingerprint scanners being used in organization for employee attendance. Biometric authentication is the next step in security systems.

Basically biometric is classified into two types: physiological and behavioral. Physiological type includes Face, Fingerprint, Iris, Hand Geometry and DNA pattern [1]. The physiological biometrics are more accurate in comparison to behavioral biometrics. But they require the use of special dedicated devices for feature extraction and this lead to a high cost of implementation. On the other hand, behavioral traits are User speech, handwritten signature, keystrokes dynamics, mouse dynamics, gait, etc. For using behavioral biometric, we do not require extra hardware and it is less disturbing to the user, so it can work in the background.

A. *Need for Continuous Authentication*

Most of the existing systems authenticate a user only at the initial login and no authentication is performed afterward. That is, once a user has logged-in, attacks may occur when the user leaves the system open and unattended, motivating an intruder to access the system. As a result, it is possible for an unauthorized user to access the system resources, without the permission of the authorized user. So, a system that can continuously check the identity of the user throughout the session is much needed. Such a system is called a continuous authentication system. Continuous authentication represents a new generation of security mechanisms that continuously monitor user behavior, like a guard, constantly watching over who is using a computer, using biometrics. This continuous biometrics authentication system works on the user's characteristics and traits. In computing scenario, Keystroke dynamics become more popular as one of the main sources of behavioral biometrics for providing continuous user authentication.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Suhail Javed Quraishi, Department of Computer Science & Engineering, Invertis University, Bareilly, UP, INDIA.
Dr. Sarabjeet Singh Bedi, Department of Computer Science & Information Technology, MJP Rohilkhand University, Bareilly, UP, INDIA.

Analyzing typing behavior has proved very useful in detecting whether the user is a valid user or not as the keyboard is an integral input device of the computer system, is cheap, requires no extra hardware, will be used by any individual trying to access the system.

Dynamic or continuous observance of the interaction of users whereas accessing extremely restricted documents or executing tasks in environments wherever the user should be alert at entire times (for example Air traffic control), is an ideal scenario for the application of a keystroke authentication system. Keystroke dynamics could also be used to observe or detect a typical writing rhythm (brought on by drowsiness, fatigue, etc.) within the user and inform to third parties.

II. RELATED LITERATURE

A keystroke biometric system receives keyboard typing data stream as input containing details such as key pressed or key released, key id, time, etc. The general extracted features are, Flight time and Dwell time:

- The Flight time is referred to as time interval between the time of key release between the current key and time of keypress of the next key. Also known as Inter keystroke interval.
- The Dwell time is referred to as time interval between key press and key release of the current key. Also known as keystroke duration.

Keystroke analysis biometric system can be classified into Static Keystroke System and Continuous Keystroke System. Static system is characterized by the use of fixed data sample for profile building of any user by the biometric system. Whereas Continuous System works on the free text that is being entered by the user in real-time for the building of profile of the user for later verification/identification.

Authentication Systems are measured generally by two metrics: FRR (False Reject Rate), when the system incorrectly rejects an access attempt made by an authorized user and FAR (False Accept Rate), when the system incorrectly accepts an access attempt made by an unauthorized user.

Research work in the field of user authentication using keyboard biometric date back to 1990.

Joyce & Gupta [2] took 33 users and recorded their username, password, & last name for eight times. They build a classifier on a statistical method using mean & standard Deviation. They were able to report a FAR of 16.36% & FRR of 0.25%.

Bleha et al. [3] employed a Bayes classifier and a minimum distance classifier using inter-key latency values. Fourteen valid and twenty five invalid users were instructed to type their first and last names and a password. To create each valid user's profile, the time duration between successive keystrokes was collected. For each classifier, value normalization was applied to accommodate the variation in the name lengths and a different threshold value was used. An entry was rejected if the threshold values were exceeded for both of the classifiers with FAR of 2.8% and FRR of 8.1%.

Monrose & Rubin [4], in 2000, took profile data collected over 11 months by 63 users and employed the method of factor analysis to reduce the data to lower dimension and used Bayesian like classifier to find the distance of features with reported FAR of 7.86%.

Bergadano et al. [5] in 2002 published their work done on 154 individuals. It took samples of length 683 from all user and by use of disorder of duration of an array of trigraph of input data for distance to profile. The results published were a FAR of 0.01% and FRR of 4%. This method was largely fixed text-based.

Gunetti & Picardi [7] extended above work in 2005 to free text and obtained results of FAR 0.005% and FRR of 5% based on same technique of disorder of duration of an array of trigraph modified to work over free text input by considering only trigraph arrays common between incoming input and profile.

Fong, Warren et al. [8] in 2005, collected free-text data from 27 individuals and used the Chi-square goodness of fit test to see how well the individual diagrams and durations fit the distribution from the user profile, taking 100 Keystrokes from user as current data, and then comparing against 100 keystroke subsets of profile. Reported FAR of 0.8% taking 100 Keystrokes & 0.5% taking 150 keystrokes as current data.

Bours [10] in 2012, collected free-text data from 25 individuals and suggested use of Trust metric for with varying confidence measure of user with each key action employing statistical techniques for profile & distance calculation and obtained results of detecting intruders in 79 to 348 keystrokes with an average of 98 keystrokes.

III. PROPOSED SYSTEM

The proposed system is outlined in this section. Our proposed system has two stages to distinguish between genuine and impostor user:

- **Enrolment Stage:** At the enrollment, stage user checks in or sign up their login details like user name and passwords. The system will run in the background, recording user dynamic keystrokes, extracts the features, and then create a profile for evaluation in the next stage. The reference template is stored in a CSV file.
- **Authentication Stage:** At the authentication stage, the user's activity is recorded in background, is predicted against user's reference template which is already stored. This phase consists of collecting user dynamic keystrokes, feature extraction, and feature matching with reference template. At last, the process of verification yields two types of action: accepted or rejected user access. The characteristics are extracted from the user's keystroke for the formation of the template and later for verification.

The proposed system is given as Figure 1. Two features were extracted during the keystroke: Keystroke Duration (Dwell Time) and Keystroke Latency (Flight Time). The Keystroke Duration is just composed by positive whole values, however, the Keystroke Latency can contain positive values as negative values. The negative value happens when the user before releasing the current key, presses the key successor.

This usually happens with users that possess a practice of typing. The Classifier is responsible for deciding the authentication. Upon receiving an input, the user gets accepted or rejected, based on Criterion of Separation (Threshold) by the Classifier which predicts the similarity between the pattern to be verified and the template of the prototype.

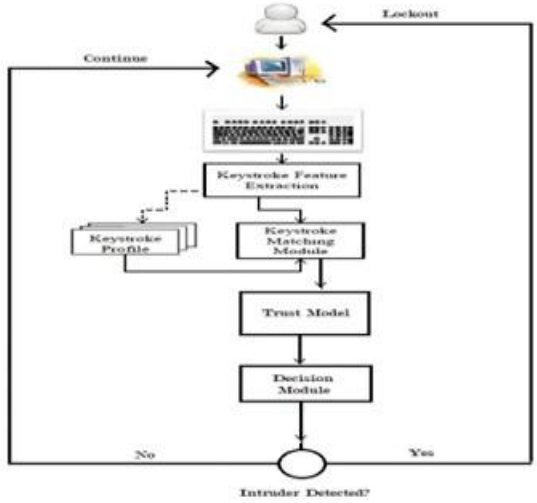

Figure 1. Proposed System

IV. APPROACH USED

A continuous authentication system may operate in an open-setting environment or a closed setting environment. For these two environments, we can use Unsupervised Learning & Supervised Learning respectively.

The closed-setting environment requires that user profile & intruder profile are known in advanced, and the system verify the active user to either the indicated profile or in remaining intruder profiles.

The open-setting environment requires no profile in advance and works with building a profile with the use of the system and verifying when the behavior deviates to recognize intruders. Our system works in an open-setting environment scenario. It has applications in cluster-based unlabeled data classification systems.

Anomaly Detection refers to detecting whether the sample received conforms to the profile, is an inlier, or is an anomaly to the profile data, is an outlier. Distinguishing between noise & anomaly is one of the issues in this technique when we are actively trying to recognize anomalies. It has application in Intrusion & Fraud Detection Systems.

Our Approach for the proposed system is an open-setting environment, wherewith the use of unsupervised learning & anomaly detection, we try to classify incoming data as part of a user profile or an anomaly, i.e. intruder.

Our Approach to build the system & obtain the result can be divided into workflow phases as follows, shown in Figure 2:

- Global Key Logging Tool
- Data Collection
- Feature Extraction
- Profile Creation
- Intruder Attack Simulation
- FAR & FRR Calculation

A. *Global Key Logging Tool*

We created a tool in Python to capture any keystroke raised in the system. The tool could capture the keystrokes when it was minimized or not in focus.

B. *Data Collection*

We distributed the tool to 23 participants for data collection over a period of 4 Weeks. Out of the collected data, data of the top 4 participants were considered for the results.

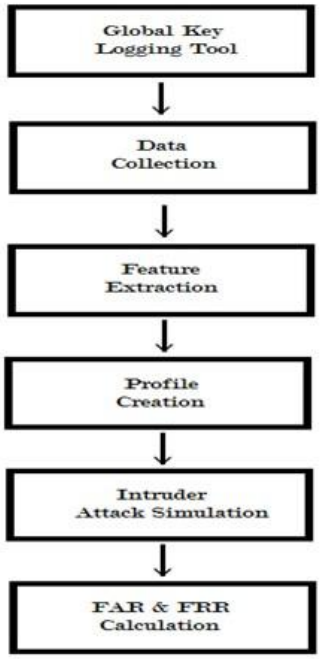

Figure 2. Workflow for Development of System

C. *Feature Extraction*

After collection of data from the participants, we extracted Dwell Time & Flight time from the raw files in .csv format, given in Table I:

Table I. CSV File values after Feature Extraction

Dwell Time	Flight Time
key, dwell	key1, key2, flight
70, 140	70, 68, 0
68, 32	68, 68, 670
68, 188	68, 68, 592
68, 219	68, 68, 1170
68, 203	72, 65, 203
80, 203	65, 80, -63
86, 171	80, 80, 62
72, 109	80, 89, 296
……	…..

In the Extraction of Dwell Time & Flight Time from the raw log files, we needed to take care of noise values. We considered only Dwell Times shorter than 500ms & Flight Time shorter than 2500ms.

D. *Profile Creation:*

Support vector machine (SVM) is a supervised learning model that analyzes information and establish the patterns, which can be used for classification as well as regression tasks. Generally, a set of training examples is given to SVM algorithm to be labeled as belonging to one of two classes.

The SVM algorithm maps the points in space so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then predicted to belong to one category or another, based on which side of the gap they fall on.

But, in one-class SVM, the support vector model is trained on data that has only one class, which is the "normal" class. It extracts the properties of "normal" class and from these properties can guess which examples are like the normal class or different from it. This is helpful for anomaly detection as a result of the scarceness of training examples is what defines anomalies.

Procedure for Profile Creation: First, we imported the .csv format into a Data Frame. Then we normalized the data, so that all values lie in the range of -1 to +1. This is done to improve the performance of SVM. Then using the scikit-learn library's One-Class SVM, we created the profile using RBF Kernel with nu=0.5 & gamma=0.00005.

E. *Intruder Attack Simulation*

We had selected 4 participants whose collected data were of size large enough to work with. We created the profile for these 4 Users with the help of One-Class SVM. Then for each user, we used the 3 remaining participant's key data, and 1 other participant key data as intruder over their profile, with key input ranges from 50 to 500 with the increment of 50.

F. *FAR & FRR Calculation*

For the FAR of a user, we simulated key input from the other profiles as defined in the previous para, over its profile. Then the percentage of Dwell Time & Flight Time that were accepted by the profile for an intruder was recorded as FAR of that particular User.

For the FRR of a user, we simulated key input of the user over its own profile. Then the percentage of Dwell Time & Flight Time that were rejected by the profile for its own input was recorded as FRR of that particular User.

V. RESULTS & DISCUSSIONS

Out of the 23 participants, the sizes of 4 used profile data are reported in table below. The other extracted profile sizes were smaller than this.

Table II. Profile Size for Participating Users

User	Dwell Time Key Logs	Flight Time Key Logs
User 1	1443	1360
User 2	1020	985
User 3	927	775
User 4	3227	2650

A. *FAR Results*

The FAR of our System for the four User Profile is given as:

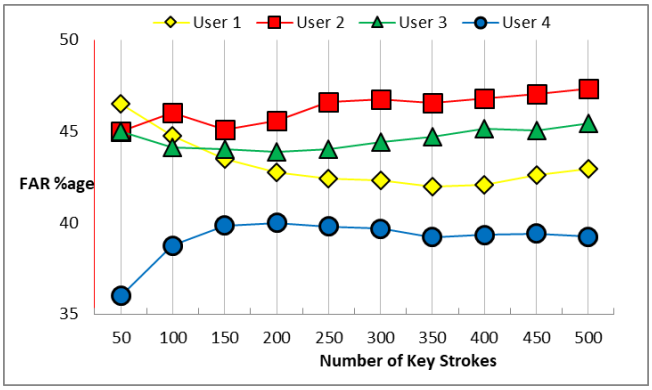

Figure 3. Graph depicting FAR Value

	User 1	User 2	User 3	User 4
FAR	42% - 47%	45% - 47%	44% - 45%	36% - 40%

The FAR for all the user profiles can be seen touching stability after 250 keystrokes.

B. *FRR Results*

The FRR of our system for the four User Profile is given as:

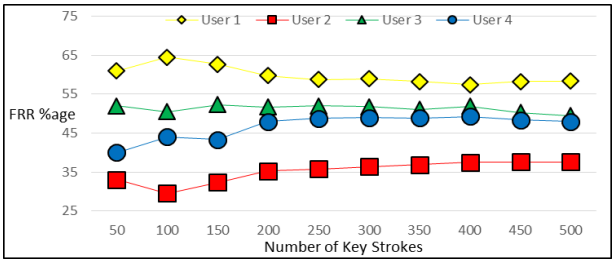

Figure 4. Graph depicting FRR Value

	User 1	User 2	User 3	User 4
FRR	57% - 65%	29% - 38%	49% - 52%	40% - 49%

The FRR for all the user profiles can be seen to touch stability in range after 250 keystrokes.

C. *Overall Average FAR & FRR of System*

The overall average of FAR and FRR of our system for the Four User Profile is given as:

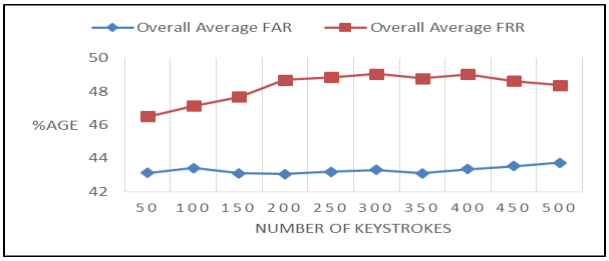

Figure 5. Graph depicting Overall Average FAR & FRR Value

Overall Average FAR	42% - 44%
Overall Average FRR	46% - 49%

This result depicts that our system is capable of detecting 42% to 44% of intruder actions. While for a valid user, it may reject his genuine action about 46% - 49% of the time.

VI. CONCLUSION & FUTURE SCOPE

A. Conclusion

In this paper, we discussed a way to use a biometric feature not just for static authentication, but for continuous authentication. So, we applied the One-Class SVM Anomaly Detection Method. In our experiment, we used participants doing their normal daily business on their own computers, without any restrictions. In this way, we measured their normal typing behavior. From that data, we collect some features like dwell time and flight time and use them to create a user profile. And experiment the system in an open setting scenario and found that our system performance is a bit satisfactory but with a good scope of improvement. The results in the previous section show that intruder will be, on average, caught about 45% of the time. Obviously, this average should be as low as possible to make the system more secure. In our experiment, there were absolutely no restrictions on the environmental or system configuration of users. This implies that the behavior of an intruder on a different system might change as it may depend on the keyboard.

B. Future Scope

In future, we hope to improve the performance or reliability of the continuous authentication system, by taking larger profile size of the user. We can work with a larger test user base and can combine keystroke dynamics with mouse usage. This can help us to prevent against those intruders who will use the mouse and avoid keyboard to try & break the system security. And make the continuous authentication system more efficient and secured.

REFERENCES

1. R. Bolle, A. Jain, and S. Pankanti. “Biometrics: Personal Identification in a network society”. MA: Kluwer Academic Publisher, Norwell, 1999.
2. R. Joyce, and G. Gupta. "Identity authentication based on keystroke latencies." Communications of the ACM 33.2 (1990): 168-176.
3. S. Bleha, C. Slivinsky, and B. Hussien. "Computer-access security systems using keystroke dynamics." IEEE Transactions on pattern analysis and machine intelligence 12.12 (1990): 1217-1222.
4. F. Monrose, and A. D. Rubin. "Keystroke dynamics as a biometric for authentication." Future Generation computer systems 16.4 (2000): 351-359.
5. F. Bergadano, D. Gunetti, and C. Picardi. "User authentication through keystroke dynamics." ACM Transactions on Information and System Security (TISSEC) 5.4 (2002): 367-397.
6. A. Peacock, X. Ke, and M. Wilkerson, “Typing patterns: a key to user identification,” IEEE Security and Privacy, vol. 2, no. 5, pp. 40–47, 2004.
7. D. Gunetti, and C. Picardi. "Keystroke analysis of free text." ACM Transactions on Information and System Security (TISSEC) 8.3 (2005): 312-347.
8. W. Fong, et al. "Continuous Identity Verification through Keyboard Biometrics." (2005).
9. N. Pavaday and K. M. S. Soyjaudah, “Investigating performance of neural networks in authentication using keystroke dynamics,” in Proceedings of the IEEE AFRICON 2007 Conference, pp. 1–8, September 2007.
10. P. Bours. "Continuous keystroke dynamics: A different perspective towards biometric evaluation." Information Security Technical Report 17.1 (2012): 36-43.
11. M. Karnan, M. Akila, and N. Krishnaraj, “Biometric personal authentication using keystroke dynamics: a review,” Applied Soft Computing Journal, vol. 11, no. 2, pp. 1565–1573, 2011.
12. E. Al Solami, C. Boyd, A. Clark, and I. Ahmed, “User-representative feature selection for keystroke dynamics,” in Proceedings of the 5th International Conference on Network and System Security (NSS ’11), pp. 229–233, September 2011.
13. A. Messerman, T. Mustafi´c, S. A. Camtepe, and S. Albayrak, “Continuous and non-intrusive identity verification in real time environments based on free-text keystroke dynamics,” in Proceedings of the International Joint Conference on Biometrics (IJCB ’11), pp. 1–8, October 2011.
14. P. Bours, and S. Mondal, “Continuous Authentication with Keystroke Dynamics,” chapter in Recent Advances in User Authentication Using Keystroke Dynamics Biometrics, GCSR, Vol. 2, pp. 41-58, 2015.
15. S. Mondal, and P. Bours. "A study on continuous authentication using a combination of keystroke and mouse biometrics." Neurocomputing 230 (2017): 1-22.
16. sklearn.svm.OneClassSVM - scikit-learn 0.18.1 documentation, http://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html
17. Unsupervised Machine Learning with One-class Support Vector Machines, http://thisdata.com/blog/ unsupervised-machine-learning-with-1class-support-vector-machine

AUTHORS PROFILE

Suhail Javed Quraishi, pursued Bachelor of Engineering in Computer Science and Engineering (CSE) from MJP Rohilkhand University, Bareilly in 2003 and Master of Technology in CSE from Aligarh Muslim University in year 2010. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Invertis University, Bareilly since 2010. He is a member of many computer societies like IEEE, CSI, IEI & IAENG. He has published more than 10 research papers in reputed international journals and conferences including IEEE. He is also the coordinator of NPTEL and IIRS-ISRO at Invertis University. His main research work focuses on Biometrics, System Security and IOT based system design. He has 11 years of teaching experience and 3 years of Industry Experience.

Dr. Sarabjeet Singh Bedi is an Associate Professor in the Department of Computer Science and Information Technology, MJP Rohilkhand University, Bareilly. He pursued B.E. (CSE) with distinction, M.E. (CSE) Gold Medalist from NIT-TTR, Chandigarh and Ph.D. (IT) from Indian Institute of Information Technology, Gwalior. He has an experience of 23 years in Academic, Research and Administration. He has already supervised 05 Ph.D. scholars. His research areas are Digital Image Watermarking, Network, Information Management and Security. He has published 58 research papers in reputed International journals and proceedings with 03 chapters in edited books. He introduced online Examination form system for 6.0 Lakh students and designed online admission system for approx. 1.8 Lakh students for MJP Rohilkhand University. He is currently holding many administrative positions in the university.