Keyboard Dynamics, A Novel Approach for Continuous User Authentication

Suhail Javed Quraishi

Invertis University, Bareilly

Dr. S. S. Bedi

IET, MJP Rohilkhand University

ABSTRACT

The key factor in the continuous authentication system is monitoring the genuineness of the user during the whole session, and not only at log-in. In this paper, we propose the application of keystroke dynamics for continuous user authentication in desktop platform. Keystroke dynamics involves three phases namely enrollment phase, verification phase and identification phase. The identification phase makes the accessed user as an authenticated user only if the template matches with the reference template otherwise the user gets exit. The proposed Continuous Authentication System is based on free text approach. There is no restriction on input during Enrolment, Verification, and Identification phase. We have used Unsupervised Machine Learning Algorithm in conjunction with Anomaly Detection for intrusion detection at every input.

Keywords

Authentication, Keystroke Dynamics, Template, Free Text, Unsupervised, Anomaly Detection, Intrusion Detection.

1. INTRODUCTION

The biometric is the study of physiological or behavioral characteristics that can be used for the identification of a human. These traits of a human include the features like DNA, Iris, Fingerprints, Face, Hand geometry, Voice, Signature, Keystroke, Gait and various other biometric features. We may use biometrics features for authentication purpose in computer based security systems.

The advantage of it is, it requires the person being authenticated to be present at the point of authentication. Thus biometric-based authentication method is most secure.

Biometric authentication is the next step in security systems. It has gain widespread acceptance already in terms of fingerprint scanners being used in organization for employee attendance.

Basically there are two types of biometrics: behavioral and physiological. Behavioral type includes User speech, handwritten signature, keyboard dynamics, mouse dynamics, gait, etc. Physiological type includes Fingerprint, Hand, Eyes, and Face. The physical biometrics has advantage of higher accuracy in comparison to behavioral biometric. But, Physical Biometric requires use of special devices for feature extraction of the biometrics and this lead to high cost of implementation. And, for using behavioral biometric, we do not require extra hardware and it is less noticeable. But it produces insufficient accuracy because behavior may change over time.

1.1 Need for Continuous Authentication: The desired quality level of any authentication system is that it be invisible to the valid user, while being impenetrable to an intruder.

Most existing computer systems authenticate a user only at the initial log-in session and do not detect whether the current user is authorized or not. That is, once a user has logged-in, open workstation attacks may occur when the user leaves and an intruder access the system. As a result, it is possible for an unauthorized user to access the system resources, with or without the permission of the authorized user, behaving like a valid user. So, a system that will continuously check the identity of the user throughout the session, i.e. a continuous authentication system is needed.

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. These continuous biometrics authentication system works on the user's characteristics and traits. In computing scenario, Keyboard dynamics has gained popularity in the development in continuous user authentication system development. Analyzing typing behavior has proved very useful in detecting whether the user is a valid user or not as 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 monitoring of the interaction of users while accessing highly restricted documents or executing tasks in environments where the user must be alert at all times (for example: Air traffic control), is an ideal scenario for the application of a keystroke authentication system. Keystroke dynamics may be used to detect uncharacteristic typing rhythm (brought on by drowsiness, fatigue etc.) in the user and notify third parties.

2. 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, digraph time and keystroke duration:

- The Digraph is referred to as time interval between the time of key release between current key and time of key press of next key. Also known as flight time, or inter keystroke interval.
- The keystroke duration is referred to as time interval between key press and key release of current key. Also known as dwell time.

There are two types of keystroke analysis biometric system: Static Keystroke System and Continuous Keystroke System. Static keystroke system is characterized by use of fixed data sample for profile building of any user by the biometric system. Continuous Keystroke System work on free text that is being entered by the user in real time for building of profile of the user for later verification/identification.

Authentication Systems are measured 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.

Work in 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 statistical approach using mean & standard Deviation. They were able to report an 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 25 invalid users were instructed to type their first and last names and a password. To create each valid user's profile, 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 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 array of trigraph of input data for distance to profile. The results published were an FAR of 0.01% and FRR of 4%. This method was largely fixed text based.

Gunetti & Picardi [6] 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 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. [7] in 2005, collected free text data from 27 individuals and used the Chi-square goodness of fit test to see how well the individual digrams 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 [8] 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 average of 98 keystrokes.

3. PROPOSED SYSTEM

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

• Enrollment Stage
• Authentication Stage

At the enrollment stage user sign up their login details such as user name and password. The system will run in background, recording user dynamic keystrokes, extracts the features, and then create a profile for evaluation in next stage. The reference template is stored in a CSV file. At the authentication stage, the user’s activity being 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 verification process yields two kinds of action: accepted or rejected user access. The features are extracted from the user's keystroke for formation of template and later for verification. The proposed system is given as Figure 1.

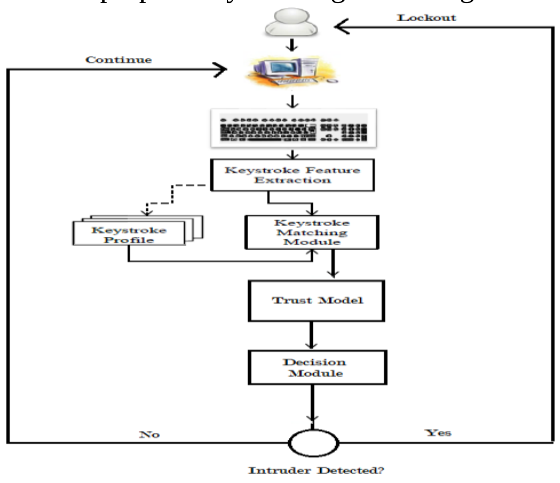

Figure 1. Proposed System

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 current key, presses the key successor. This usually happens with users that possess 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.

4. 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.

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

Open-Setting environment requires no profile in advance and works with building profile with use of system and verifying when the behavior deviates to recognize intruders. Our system works on 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 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 open-setting environment where with use of unsupervised learning & anomaly detection, we try to classify incoming data as part of 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

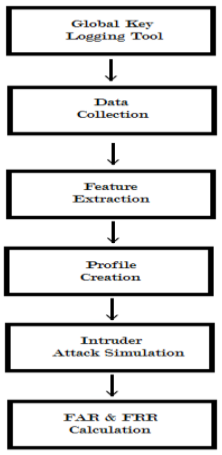

Figure 2. Workflow for Development of System

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

4.2 Data Collection: We distributed the tool to 23 participants for data collection over a period of 2 Week. Out of the collected data, data of top 4 participants were considered for the results.

4.3 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 1:

Table 1. 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 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.

4.4 Profile Creation:

4.4.1 Understanding One-Class SVM: Support vector machines (SVMs) are supervised learning models that analyze data and recognize patterns, and that can be used for both classification and regression tasks.

Typically, the SVM algorithm is given a set of training examples 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 infers the properties of normal cases and from these properties can predict which examples are unlike the normal examples. This is useful for anomaly detection because the scarcity of training examples is what defines anomalies.

4.4.2 Procedure for Profile Creation: First we imported the CSV format into a DataFrame. 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.

4.5 Intruder Attack Simulation: We had 4 participants whose collected data were of size large enough to work with. We created the Profile for these 4 Users with 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 of 50 to 500 and increment of 50.

4.6 FAR & FRR Calculation: For the FAR of a user, we simulated key input from the other profiles as defined in section **4.5** 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.

5 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 2. 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

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

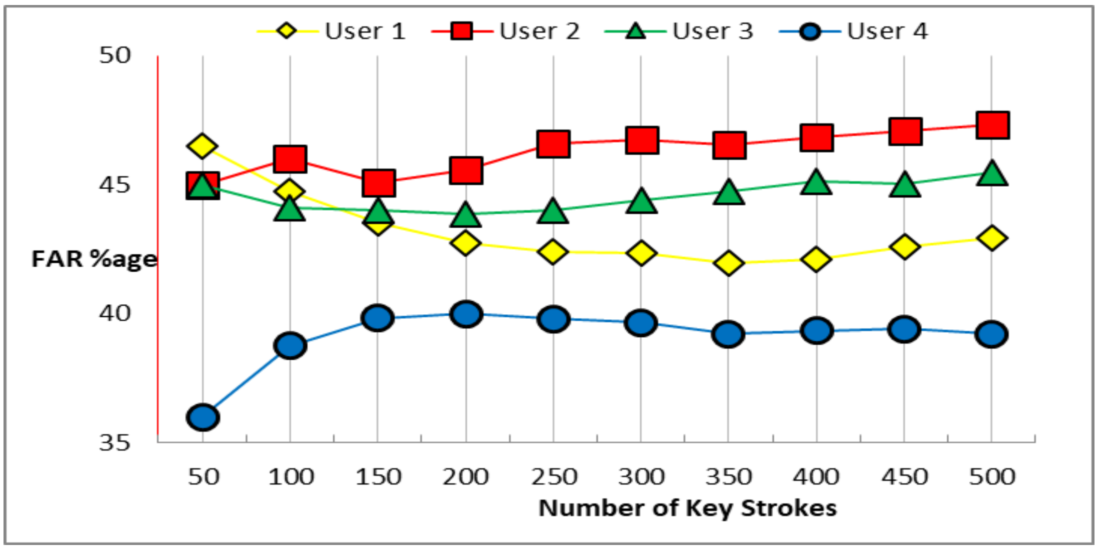

Figure 3. Graph depicting FAR Value

	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.

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

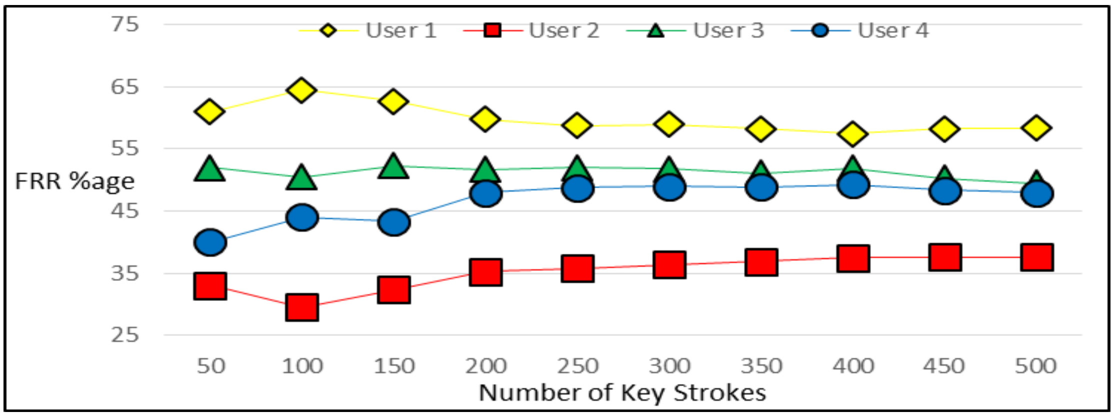

Figure 4. Graph depicting FRR Value

	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.

5.3 Overall Average FAR & FRR of System: The overall average of FAR and FRR of our system for the Four User Profile is given as:

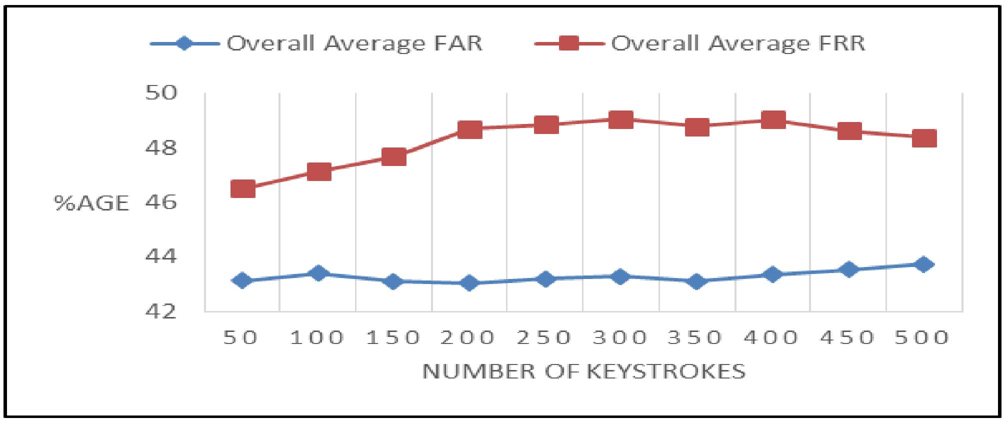

Figure 5. Graph depicting Overall Average FAR & FRR Value

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.

6 CONCLUSION & FUTURE SCOPE

6.1 Conclusion: We have described a way to use a biometric feature not just for static authentication, but for continuous authentication. In order to do so, we applied 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. Using that data, we optimized some parameters and at the end 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. Also, 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 switching system might change.

6.2 Future Scope: We hope to improve the performance of the continuous authentication system in future, by taking larger profile size of the user to work with a larger test user base, combining keystroke dynamics with mouse usage. This we will do to prevent against intruders who will use mouse and avoid keyboard to try & break system security.

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