International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 14, Number 12 (2018), pp. 1061-1067

http://www.ripublication.com

Comparison of learning models in Behavioural Biometrics using Keystroke Dynamics

Amita Yadav

Maharaja Surajmal Institute of Technology, New Delhi, India.

E-mail: amitaay@gmail.com

Abstract

Keystroke Dynamics is a type of behavioral biometrics, which identifies an individual on the basis of his rhythm and style as he types characters on a keyboard or a keypad. User's typing pattern is used to create a biometric template using the Keystroke rhythm of the user. This kind of biometric is a highly economical extension to the existing security systems and no special equipment is required for its implementation. we have identified various parameters that can be used for uniquely identifying the user and the various methods to train the model to achieve maximum accuracy.

Keywords: Keystroke Dynamics, Behavioural Biometrics, Parameters for Keystroke Dynamics

I. INTRODUCTION

Biometrics refers to human characteristics which are uniquely defined for every individual. For years, physiological characteristics like fingerprints, retina and iris recognition, face recognition, palm, etc., have been used and are currently being as used for user identification an authentication. However, these require special equipment to capture these minute details of any human. The other kind of biometrics, Behavioural biometrics have opened up a lot of possibilities and higher security. Physiological characterized systems can be penetrated, and have weaknesses, for example, the fingerprint scanner can be cheated using a gelatin made fake fingerprint, due to the static nature and minimal changes over time, the physiological traits can be copied. But, in case of behavioral characteristics, thetraits are acquired by the individuals after a thorough practice and routine reenactment, which allows to form their own rhythm for that particular characteristic, like, signature, walking, Keystrokes, voice, etc.

However, the security is not the only parameter to judge, but the feasibility too, for example, DNA is a kind of physiological trait and cannot be copied, but to use it in the everyday life as a measure for security isn't feasible. The behavioural biometrics pose as a highly feasible security solution, the existing system can be used to implement these without the requirement of any special equipment.

In this work, our focus is limited to Keystrokes only. Keystroke Dynamics allow us to record a user's typing pattern and create a biometric template based on the manner and the rhythm of typing on a keyboard. As, keystroke dynamics is a behavioural biometric, it is 'something that you do'.

The concept of Keystroke dynamics came from the second world war where the military used a technique called 'The Fist of the Sender' to identify the sender and trace messages.

In this work, we will first study and identify the various parameters that can be used to uniquely identify a user, and also study the parameters which can be useful in implementing keystroke dynamics in touch-based/smartphoneskeypads.

Section I contains the introduction, need and the scope ofKeystroke Dynamics, Section II contains work, starting withthe identification of the parameters required for keystrokebiometrics, development of Keylogger for data acquisitionfollowed by the extraction of the features from the acquireddata set, Section III involves the comparison of variousapproaches followed to achieve biometrics using keystrokesand finally concluding with best(most efficient) approachwhich helped us in achieving our goal.

II. RELATED WORK

Originally designed to help distinguish ally from enemy during the time of World War II, a lot of work has been done to improve the efficiency of user authentication.

In 1999, Daniele Gunetti and Giancarlo Ruffo performed experiments based on the user's behavioral work. Their experiments were designed in a manner which would allow the application to distinguish between a genuine user and unauthorized user depending upon the typing speed and the commands that are used frequently by the user. The data was collected for 10 users over a period of three months and parameters like the number of characters typed per minute was calculated after a time interval of 10 minutes had elapsed (In case no input was recorded for a time period of 600 milliseconds then that time period was not taken into consideration). The drawback of this approach was that authentication based on commands history was not very reliable as these commands can become redundant due to software updates or if the user decides to use a different software.

A major advancement happened after Charles C. Tappert in 2006, took long text input to improve user authentication. Charles and his team designed a Java applet to collect unprocessed keystroke data, from which feature vector was extracted. This feature vector contained the measurements for average and standard deviation of key press

duration for most commonly used keys, along with average and standard deviations for transition time for most frequent pair of keys pressed together. However, an input of minimum 600 characters was required for the software to effectively carry out these data extractions.

K-Nearest Neighbor Approach was adopted by J. Hu, D. Gingrich and A. Sentosa in 2008. The approaches taken prior to this either had a very high False Acceptance Rate (FAR) and False Reject Rate (FRR) or had a slow authentication speed. The reason for slow authentication speed was that each input had to be compared and verified against the training set for every user in the entire database. This resulted in very high search time which was done in a redundant manner. To overcome these issues, K-Nearest Neighbor Approach was introduced. User is required to provide a number of training data sets on which K-Nearest Neighbor method is applied for clustering of representative profiles. This increases the authentication speed as for successful recognition, the input must lie close to the cluster to which the profile of that particular user belongs. So, instead of comparing the input value to the training set of each user, the input is compared to the closest cluster and the user is verified as authorized user if his/her profile belongs to that cluster.

For the development of an application that would allow userauthentication based on Keystroke Dynamics, we need a dataset of how users press the keystrokes, that is, factors likeduration for which a key is held, error/accuracy of a user etc.to compare it to the dynamic data that is being stored,processed and analyzed in real time, so as to ensure minimalFAR and FRR.Parameter Identification*Keystroke Latency(or the DiagraphLatency)*is the time interval or the delay between twoconsecutive keystroke presses.

Dwell Time is another parameter that isused for comparison is the time duration for which a key ispressed. In other words, the time elapsed between press and release of a single key. The time period for which a key is pressed can vary drastically depending upon the skills of an individual.

CPM is a parameter that can help achieve better accuracy in useridentification. CPM (Characters Per Minute) is the numberof keystrokes pressed per minute that vary from individual toindividual.

*Accuracy*is based on taking into account theerror/accuracy of each user. The number of times and thefrequency of backspace keystroke presses can also help inuser authentication.

*Hold Time*is the duration for which a key is pressed, from key-down to key-up events. Hold time varies from person to person, vast difference in holdtime is seen from beginner and expert keyboard users.

*Flight Time*is a parameter which refers to the time elapsed between pressing and releasing of two consecutive keys.

Irrelevant Keystrokes are those which have no effect on a certain input or those which are irrelevant to data acquisition.

These parameters were used to implement behavioralbiometric using keystroke dynamics on a physical keyboard.To implement it on keypads or touch devices, we can useaccelerometer data, hand positions and the inclination atwhich the device is held while using the device asparameters.

Data Acquisition

A Keylogger was used to record the keystrokes whichcomprises as the Data Acquisition phase of the research.

Parameter Extraction

The data acquired in the above phase was used to extract theparameters (or the features) which were passed onto as input to various approaches which are compared in the research.

III. METHODOLOGY

The following approaches were considered for the progress of this work, so as to identify the user more accurately, by not only his typing mechanism but also his behavior while typing

Using long-text input for biometric recognition, where a python script integrated with a batch file was used to collect keystroke data in raw form andusing the extracted long-text input features, recognitiondecisions were made by pattern classifier.

Using behavioral data for intrusion detection, where datasuch as the keystrokes or commands that are used by the userwhen he/she logs in to a computer were used forauthentication.

Using weighted probability, where weight was attached tomore reliable features such as characters which had arelatively greater occurrence rate; example in, th, ti, on, an,he, al, er etc.

We used four learning models which could capture the typing behavior of various users and could also be used for feature vector.

Manhattan Filtered Detector

This model takes into account any deviation or oddity in the user's typing behavior. These exceptions could be due to multiple factors, for instance – the user might be fatigued and could be typing at an unusually slower pace. So MFD filters out such anomalies.

The training() function of the MFD takes into account any outliers in a subject's typing habits. Such deviations from his/her usual typing habits may occur due to a variety of reasons, like the user being tired or bored and hence typing exceptionally slower than normal, etc. MFD simply filters (removes) such outlier.

Manhattan Scaled Detector

This model provides the lowest equal error rate out of all the detectors based on Manhattan distance. The reason being that it calculates the mean absolute deviation for each feature of the sample data along with the mean vector but mean absolute deviation improves upon the equal error rate by scaling the mean vector.

While training, we calculate the mean_vector()as well as the mad_vector() which has the mean absolute deviation (MAD) of each feature of the training data. In testing(), score for a test sample is being calculated as \sum_{i=1}^p \frac{|x_i - y_i|}{\alpha_i} , where x_i and y_i are the i^{th} feature in the test sample and mean_vector()respectively, and \alpha_i is that feature's MAD , taken from the mad_vector(). Thus, we are essentially calculating the city-block distance but each feature is getting scaled by its MAD.

One-Class SVM

The approach of One-Class SVM is rather different as it uses the data of only a single class to learn a decision function and this decision function is used to test a new input to check whether it is in close approximation to the training data or not. The equal error rate, in case of One-Class SVM is close to 0.1206.

Gaussian Mixture Models

It is a probabilistic model which is based on clustering. The clustering is done in a manner similar to K-Means algorithm, that is, the data point either is completely included in a cluster or not at all. However, K-Means algorithm fails in case of round shaped clusters. So, to overcome this, GMM uses the centroid, size of cluster and its covariance to describe each cluster.

A digraph is a combination of 2 letters, like the password '.tie5Ronal' has these digraphs – .t, ti, ie,…,na, al, . Our typing feature vector is essentially consisting of the various time latencies between the two letters of all the digraphs in the password. Now, GMMs can be used for determining whether a test typing vector belongs to one user or not since it has been proved in many studies that the digraph patterns present in keystroke data are generated by Gaussian distributions. Hence, we can model the user's behaviour by fitting a GMM over the training data. The model is then used to calculate a test vector's score which is its average log-likelihood of belonging to that model.

The other learning model that is considered for comparison was *Random Forest*, it is an ensemble learning model, where various possible decision trees are formed and combined at training time and gives the mode of the classes.Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks.

The comparison of the first four models was obtained as below:

Model Name	Equal Error Rate
MFD	0.1807
MSD	0.1484
One Class SVM	0.1206
GMM	0.1502
Random Forest	0.1222

Fig. 1 Comparison of *Manhattan Filtered Detector* (MFD), *Manhattan Scaled Detector* (MSD), *One Class Support Vector Machine*(One Class SVM), *Gaussian Mixture Models*(GMM), Random Forest

The existing systems are however secure with encryption and safe transport mechanisms, but even the slightest of leakage can potential hackers to infiltrate to the system allow. Keystroke analysis can act as a filter system to separate irregularity in user's behavior toauthenticatehim. However, the usage of physiological biometrics allow exposed security lock, which would alert the hacker beforehand, keystroke can be embedded into the system and run in the background in stealth mode. The best thing about keystroke dynamics is that no new hardware is required and hence it can be integrated in any kind of system that involves keystrokes.

IV. RESULTS AND DISCUSSION

Achieving biometric using Keystroke Dynamics proves to bethe most economic and efficient way of implementingsecurity in Electronic Devices. Keystroke analysis being the most feasible option offers highest security at the cheapest cost. One class SVM and Random Forest are most suitable for the implementation.However, the Random Forest cannot achieve higher accuracy than One-Class SVM but it can allow the identification of the user in a broad spectrum of cases, it is much more consistent with respect to One-Class SVM.

V. CONCLUSION&FUTURE SCOPE

The goal of Keystroke Dynamics is clear and can offer runtime authentication mechanism, but the basis of such system requires constant monitoring, which can also be an issue for privacy. But overall, keystroke dynamics offer high security and guarantee authenticate work at all times at the cheapest.

Keystroke analysis needs to be a constantly evolving security measure which reads user behaviour repeatedly and achieve higher accuracy as the user operates. It is the only security feature that has a lot to offer while being the cheapest of all.

REFERENCES

[1] Curtin, Mary & Tappert, Charles & Villani, Mary & Ngo, Giang & Simone, Justin & Cha, Sung-Hyuk. (2006). Keystroke Biometric Recognition on Long-Text Input: A Feasibility Study. Proceeding International Workshop Scientific Computing and Computational Statistics (IWSCCS 2006).

[2] F. Monrose, M.K. Reiter and S. Wetzel, "Password hardening based on keystroke dynamics, "Proceedings of the 6th ACM Conference on Computer and Communications Security, pp. 73-82, New York, NY,USA, ACM Press, 1999

[3] F. Bergadano, D.Gunetti and C. Picardi, "User authentication through keystroke dynamics," ACM Trans. Info. Syst. Secur., 5(4), 2002, pp.367-397.

[4] Kevin S. Killourhy, Roy A. Maxion, "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics", IEEE conference on Dependable Systems & Networks, July, 2009.

[5] L. C. F. Araujo, L. H. R. Sucupira, M. G. Lizarraga, L. L. Ling and J. B. T. Yabu-Uti, "User authentication through typing biometrics features," in IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 851-855, Feb. 2005. doi: 10.1109/TSP.2004.839903

[6] Robert Moskovitch , Clint Feher , Arik Messerman , Niklas Kirschnick , Tarik Mustafic , Ahmet Camtepe , Bernhard Löhlein , Ulrich Heister , Sebastian Möller , Lior Rokach ,Yuval Elovici (2009). Identity theft, computers and behavioral biometrics (PDF). Proceedings of the IEEE International Conference on Intelligence and Security Informatics. pp. 155–160.

[7] Gunetti, Daniele & Ruffo, Giancarlo. (1999). Intrusion Detection through Behavioral Data. 383-394. 10.1007/3-540-48412-4_32.

[8] Monrose, Fabian & Rubin, Aviel. (2000). Authentication via Keystroke Dynamics. Proceedings of the ACM Conference on Computerand Communications Security. 10.1145/266420.266434.