International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

User Authentication using Keystroke Analysis

Yashashree Amankar[1], Sakshi Gangurde[1], Achal Khachane[1], Prof. Prashant Y. Itankar[2]

[1]BE Scholar, Department of Computer Engineering, Datta Meghe College of Engineering, New Mumbai, Maharashtra, India
[2]Professor , Department of Computer Engineering, Datta Meghe College of Engineering, New Mumbai, Maharashtra, India

---

Abstract - *The security of devices, applications and data has became an important issue due to drastic increase in cyber crime rate. Hence it is necessary to protect the private or confidential data from getting accessed by unauthorized user. There are several traditional methods like PINs, Patterns, passwords, tokens, etc.; but they are not that efficient as they can lost or get stolen, therefore fails to fulfill the security challenges, which ultimately compromises the system security. Even the powerful cryptographic techniques also fail to prevent unauthorized access since they are static. As far as biometrics are concern ,they have proven to be efficient to satisfy security challenges .Biometrics, defined as intrinsic physical traits and behavioral characteristics that make each of us unique ,for identity verification. Biometrics of each individual is unique so they can't be stolen or impersonated, it comes out to be optimal and safest option for authentication purpose.*

Key Words: Keystroke dynamics, Distance based measures, Dynamic model, Leave-One-Out-Method

1. INTRODUCTION

Biometric system is mainly classified into two categories for authentication purposes viz. the physiological traits and behavioral traits of individuals. Physiological traits include fingerprints, voice, hand-geometry, face, iris, retina, palm-print, etc., and behavioral traits include signature, keystroke dynamics, gaits and voice [1].

The main threats to the data, computer systems and digital networks are frauds, intruders and impersonation. Several applications created the login ID and password system for authentication purpose , but these systems are not perfect as if the password get stolen then it is no longer secured with respect to the legitimate user. To avoid such problem and to overcome flaws of static verification system we are designing the system which is combination of biometrics system and dynamic updating model .The most propitious approach for such system is keystroke biometrics.

Keystroke dynamics is the process of authenticating individuals based on their typing style. It is not about what you type, but simply about how you type [2, 3]. It refers to the habitual patterns or rhythms an individual exhibits while typing on a keyboard .It is unique due to the similar neuro - physiological factors which makes hand written signature of an individual unique.

The keystroke analysis have various merits over other biometric systems:
1. The "pattern" or "rhythm" of user's typing is considered as reliable statistics.
2. No external hardware like detector, scanner, etc. is required. Only basic input device i.e. keyboard is required [4, 5].
3. With the existing authenticating systems, it can be easily deployed.

The Keystroke authentication approach is classified into two: static and dynamic. Most of the existing approaches focus on static verification, where user enters the string i.e. predefined password. The second one is dynamic, it is also known as "free text dynamics"which does not demand any strings like predefined passwords. It is adaptive in nature as it updates the keystroke records dynamically. The only problem with the keystroke analysis is that, it is behavioral parameter so it can fluctuate according to user's emotional state, type of the keyboard used by user and the position of the keyboard with respect to the user. These variations lead to create error in the static system, But are update in dynamic system as adaptive feature.

2. LITERATURE SURVEY

Keystroke Dynamics has become a widely researched and active area due to the increasing importance of cyber security and computer or network access control. Most of the existing approaches focus on static verification, where a user types specific pre-enrolled string, e.g., a password during a login process, and then their keystroke features are analyzed for authentication purposes[5]. Only a few research studies address the more challenging problem of keystroke biometrics using "free text", where the users can type arbitrary text as input.

Keystroke dynamics features are usually extracted using the timing information of the key down/hold/up events. The hold time or dwell time of individual keys, and the latency between two keys, i.e., the time interval between the release of a key and the pressing of the next key are typically exploited.

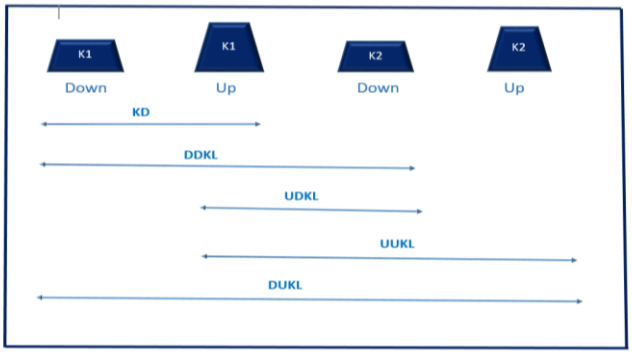

Fig.1:Timing features extracted from keystroke patterns

The features extracted from keystroke dynamics pattern in most of researches are timing features. Fig shows the extracted timing features:

1.**Key Hold(KD)**:key delay between pressed and key released.
2.**Down-Down Key Latency (DDKL):** time in between two consecutive presses.
3.**Up-Up Key Latency (UUKL):** time between two successive releases.
4. **Up-Down Key Latency (UDKL):** time in between the current key release and the next key press.
5. **Down-Up Key Latency (DUKL):** time between the current key press and the next key release.

Research work on keystroke dynamics all originated from Gaines et al. [8] who did a preliminary study authentication using the T-test on digraph features.

Monrose and Rubin [22] few years later extracted keystroke features using the mean and variance of both digraphs and trigraphs.

Then there were statistical Euclidean distance metrics with Bayesian-like classifiers identified 92%fortheir small dataset containing 63 users correctly. Over the years, keystroke biometrics research has been implemented in many existing machine learning algorithms and classification techniques .

Researchers have presented their work on choosing different distance metrics, such as the Euclidean distance , the Mahalanobis distance and the Manhattan distance and have been explored their suitability on the biometric authentication. For the implementation both classical and advanced classifiers have been used, including K-Nearest Neighbours (KNN) classifiers [4], Bayesian classifiers, K-means methods [12], Fuzzy logic, neural networks , and support vector machines (SVMs).

A promising research effort in applying keystroke dynamics as a static authentication method originated from the work of Joyce and Gupta [14]. Their approach is relatively simple and yields impressive results.

The main idea of our work is to allow users to access different systems by typing their own usernames and passwords as usual. Then, the users' typing styles features are extracted from their passwords, so there is no additional text required for authentication.

3. PROPOSED SYSTEM

The block diagram of proposed system is as shown in Fig.(2)

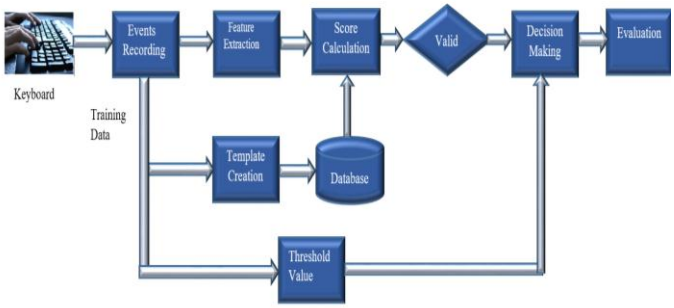

Fig.2:Block diagram of the model proposed

The problem with the existing work and implementations related to keystroke authentication based on static text is that the statistic chosen and the model built are not very accessible and compatible with each other. Therefore we propose an easier and a much simpler model and metric to achieve the desired classification which has better interpretability as shown in Fig.(2).

The whole model can be divided four distinct steps. These are listed as follows:

1. The individual register their name and password with the database. Then the user has to type his username and train the machine for six times.
2. Features are extracted when individuals press and release keys. More specifically the delay between the key-down and key-up time.
3. The algorithm is applied and the threshold is generated based on the variations that the user has done while typing the 6 training set. Hence, the adaptiveness.
4. Calculate the Euclidean distance between training and the test samples to get the user's score.
5. Finally, the user's score is compared against its threshold to make the decision. If the Euclidean measure generated from the test sample is too high when compared to the training set then the user is classified to be an imposter.

3.1 Key points of proposed model (system)

3.1.1 Data and feature extraction

A dataset is created to evaluate the proposed system. A software application validates the entered data at the time of

registration and the credentials are implemented to acquire samples from individuals and extract their features. The user has to simply type his username and passwords that they can comfortably type and the rhythm of which they can easily remember.

Time stamps of each key press (Down) and release (Up) are stored in a logout file and used to calculate KD, DDKL, UUKL, UDKL, and DUKL. For our model we take the key delay between the key up of the current stroke and the key up of the next stroke.These differences become the attributes of our data set and determine the class labels of our machine learning algorithm. Typically these key delays are stored in a comma separated value fashion.

3.2 The metric or the statistic chosen for comparison:

For testing the efficiency and the correctness the two statistics that we chose were Manhattan distance and the Euclidean distance.

3.2.1 Manhattan distance:

The score is calculated as in Eq. (1) represents Manhattan Distance.

$$M = \sum_{i}^{n} (x_i - y_i)$$

----(1)

Where x = (x1 , x2 ,..., xn) represents test vector and
y = (y1,y2,...,yn) represents the mean vector of the training samples

3.2.2 Euclidean distance

The score is calculated as the squared Euclidean distance between the test vector and the mean vector as in the following Eq.(2)

$$E = \sqrt[2]{\sum_{i}^{n} (x_i - y_i)^2}$$

---(2)

Other optimal choices for the distance measures could also be as follows .

3.2.3 Manhattan with Standard Deviation Distance

The standard deviation of each feature is calculated as well. Eq. (3) will be in the form

$$Ms = \sum_{i}^{n} (x_i - y_i)/\alpha_i$$

---(3)

3.2.4 Mahanabolis Distance

The standard deviation of each feature is calculated, where the Mahanabolis distance is presented by Eq.(4)

$$Mh = \sqrt[2]{\sum_{i}^{n} ((x_i - y_i)/\alpha_i)^2}$$

---(4)

4. THREDHOLD CALCULATION

The threshold calculation is what makes the model adaptive and different than other existing models and algorithms. The window for error is the space in which he is permitted to cause any errors. This is decided by a method called Leave One-Out-Method (LOOM).

This method is explained below in some detail in steps:

1. Out of the n samples, divide the training space of (n) samples to one sample used as test sample, and (n-1) samples used to create the training sample.
2. Apply a distance measure (Euclidean in our model) to calculate the distance between the selected test sample and the mean vector of the (n-1) training samples.
3. Iterate the step 2 for (n) times to produce (n) different thresholds for each feature vector.
4. The average of these (n) thresholds is calculated to produce the one single threshold that would represent the effective measure of all the thresholds in total.
5. These steps are repeated to calculate the individual thresholds for the other three distance measures.

4.1 PERCEPTIBLE SIMILARITY

This threshold calculation and also the working of the algorithm can be clearly stated as a visual representation as follows. Assume that the space is made up of 6 data points each of them which would stand for a set of attribute array in out database. Now pick a data point a random and calculate the distance of this particular point from each of the rest of data points.

That would give the threshold that must exist for this data point. Now choose another data point and repeat this calculation. At the end we would end up with six different difference vectors. Now take the average of the vectors and conclude to a single point in space. This point is cumulative distance equivalent of all the vectors combined.

Imagine a sphere centered at this point. The threshold calculated would be the radius of this sphere. If a test data point arrives, we plot this point in space. We then check if this point inside the so formed sphere. If it does then it is equivalent to a data set which is of a valid user and it's delay

array is within bounds of error. If it doesn't, then it would mean that the data set belongs to that of an imposter and the delay discrepancy is beyond the margin allocated.

4.2 DECISION MAKING

The proposed system's is evaluated using two statistics. These metrics are listed as follows:
1. **False Rejection Rate (FRR) :** which is the refused fraction of genuine individuals, and
2. **False Acceptance Rate (FAR) :** which is the accepted fraction of impostor individuals.
Eq. (5) and (6) shows FRR and FAR respectively.

$$FRR = \frac{Number\ of\ refused\ genuines}{Total\ number\ of\ genuines} \quad ---(5)$$

$$FAR = \frac{Number\ of\ accepted\ imposters}{Total\ number\ of\ imposters} \quad ---(6)$$

The biometric system performance could be measured using Equal Error Rate (EER) which refers to the point on the ROC(Receiver Operating Characteristic) curve where the FAR and the FRR are equal. It is shown in the Fig.(3)

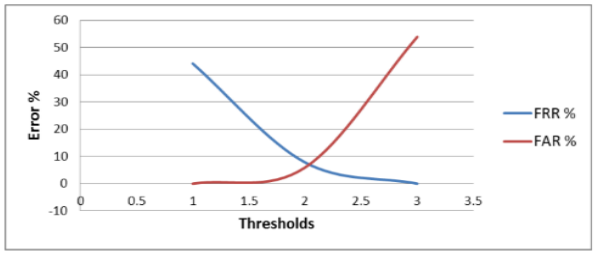

Chart.1 : Equal Error Rate (EER)

5. IMPLEMENTATION

5.1 Design and Technology

Using the design described in the above section the experiment software was successfully built, tested and used to gather data. Rather than give an in-depth analysis of the code, we shall provide an more informative overview of the technologies we used, and describe the interaction with the software that volunteers experienced.

We built the experiment software as a web application using the following technologies:

1. **HTML 5, jQuery, Bootstrap**: For the front end and creation of forms to accept the data from the user
2. **JavaScript**: To perform the front end validations and also to gather the key up time from the user
3. **Ajax**: To parse the timing data from the front-end JavaScript to the PHP
4. **PHP**: To perform all the file operations and IO (input output operations)
5. **R Language**: To build the model and to calculate the threshold.
6. **MySql**: To store the user id's and the passwords and to maintain session information

The software was designed to be very modular. This means that if similar experiments are required, the software to very easily be re configured to with different groups,passphrase and schedules. The volunteer was authenticated with the site using their username and a password. (This is same phrase that we use in learning).

Once authenticated they were directed to a page containing a javascript client which allowed them to perform the experiment.

The directory structure of the proposed system is shown in the Fig.(4)

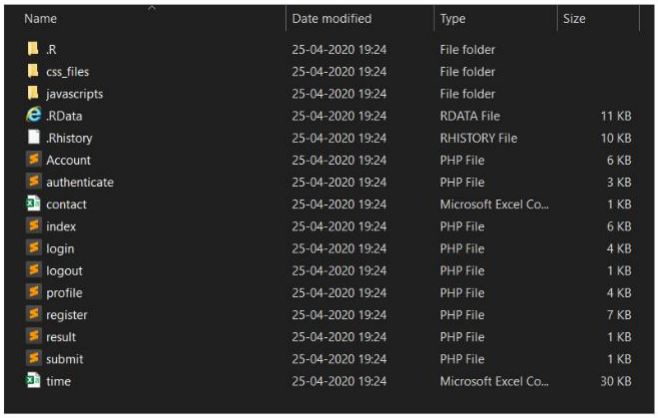

Fig.3: Directory structure of designed model

5.2 Experimental Results

The user first chooses a user name and a pass phrase as shown in Fig. (3) which would the same password he would be using to train the machine. The user is then logged into a page and is asked to go to the logistics page where the training phase happens. Then once he logs out. The next time the user tries to login the algorithm runs and the test data that is the current attempt is recorded and tested with the database as shown Fig.(4) , Fig.(5) & Fig.(6)

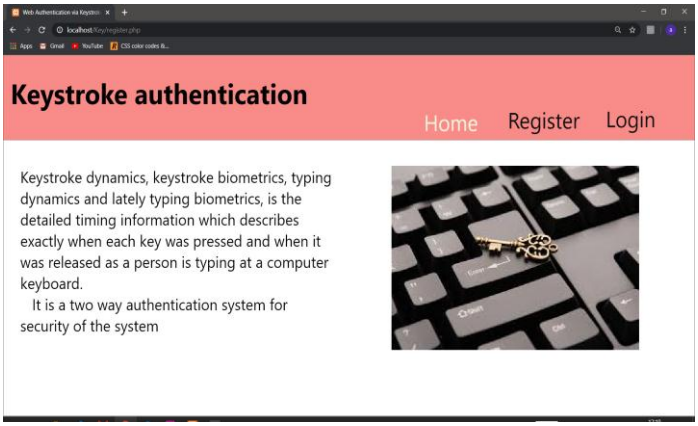

Fig.4: Home page of the designed model

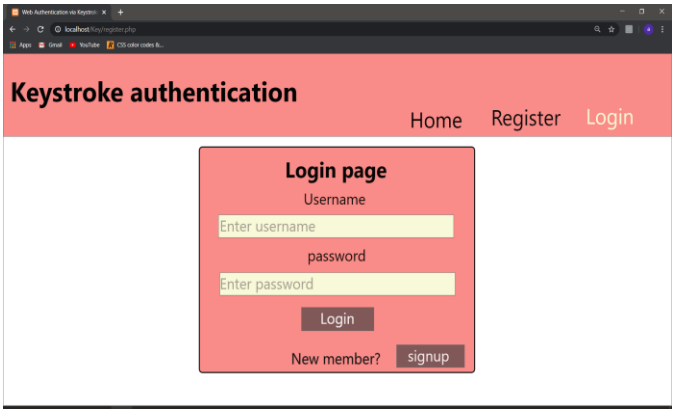

Fig.5: Login page of the designed model

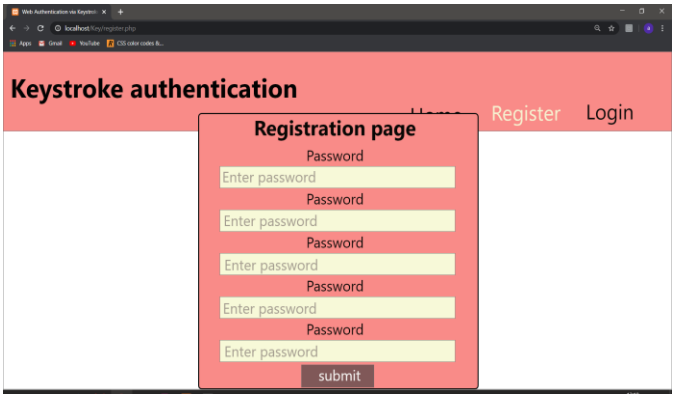

Fig.6: Password Registration page of designed system

5.3 Comparison with Existing work

Four datasets where used evaluation of the comparative efficiency of our system; the first one is by Yu Zhong (2012) [6]; the second one being CMU by Kevin S. Killourhy (2009)[7] ; the third being Shimaa I. Hassan(2013) and the fourth one being our model.

Killourhy and Maxion [7] used 14 keystroke dynamics anomaly detector to authenticate users, 11 of which where previously proposed, and 3 where classic recognition patterns using various distance statistic models. (Euclidean, Manhattan, and Mahalanobis distance measures). Their dataset composed of 51 users, who typed the password for 400 times along 8 sessions ,i.e., 50 times per session, out of which 200 samples where taken for training the model, and the rest where used for testing the model built. The features from each sample included DDKL, UDKL and KD and achieved and EER of 9.6%.

Yu Zhong et al. [6] evaluated a keystroke authentication based on a new distance metric, i.e., by combining Mahalanobis distance and Manhattan distance on the keystroke dynamics dataset created in (CMU Dataset). They used Nearest Neighbor classifier with their new distance metric to authenticate the user to achieve an EER of 8.4%. Shimaa I. Hassan, Mazen M. Selim, and Hala H. Zayed used the concept of majority voting. If there are n features, the input sample is assigned an identity when at least k of the features agree on that identity, where k = (n/2)+1 if n is even and k = (n+1) / 2 if n is odd. It first Authenticated based on feature separately, and UDKL produced 8.8% for EER using Manhattan with standard deviation.

Finally, individuals are authenticated based on majority voting (MV), the best results is 7.0 % for EER using Manhattan with standard deviation and MV.

Our model used two distance statistics viz. Manhattan distance and Euclidean distance. Our dataset composed of 40, who typed the password for 5 times and authenticated using the model for 28 times during 4 sessions.

The model build consists of two distance measures :

1.**Manhattan distances**

It takes the average of all the training data set and compares with a threshold of 150 ms.

(i) This model is less adaptive as irrespective of the user , the threshold is fixed and the model is not properly built according to the typing pattern of the user.
(ii) As per our results, this model is more flexible when compared to the other model . The major reason behind that is while training the model, the user types the same password six times and hence the training model build using Euclidean distances is very small and precise, this means that during authentication, the user must type with the same pattern without even milliseconds of variation in the pattern which is quite inhuman.

2. **Euclidean distances**

This distance measure is quite adaptive compared to the other model, that is the threshold completely depends upon the training data and is not fixed to some constant value.

(i) It takes the first sample as test and the rest five as training sets and finds the anomaly distance. The same

process is repeated for different test sample (second sample, third sample, etc.) and the rest as training example. The anomaly distance of all these six models are taken and the average of them gave the threshold of our model.
(ii) Major advantage of this model is when the sample given during the training differ, i. e . , in a humane or a natural way, not significantly change as if a different user is typing (that will oppose the objective of our project) ,the acceptance sphere will become large compared to the Manhattan distance model.

The EER rate of the Euclidian model and the Manhattan model of our proposed system is given in Table.(1)

Table-1: EER of distance measures in proposed system

Sr. no.	Distance measure	EER
1	Manhattan distance	8.9
2	Euclidian distance	9.8

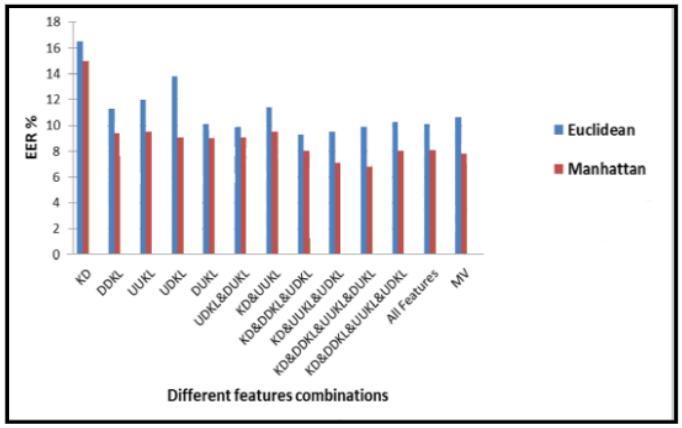

Chart-2: Euclidian and Manhattan distance measure's comparison using combination of data extraction features

The above Chart-2 shows the comparison between distance measures used in our system. This comparison is based on different features extracted from the dataset like DDKL, UDKL,DUKL, and their various combination .

The EER of the various models is as shown in the Table-2.

Table -2: Comparison of EER of the existing systems and our system

Sr.no.	Systems	EER
1	Kevin S. Killourhy(2009) [7]	9.6
2	Yu Zhong and Yu Deng(2012) [6]	8.4
3	Hala H. Zayed (2013)	7.0
4	The Proposed System	8.9

6. CONCLUSION AND FUTURE SCOPE

We studied the characteristics of keystroke dynamics for user authentication and proposed a new adaptive model and statistic which would change the threshold according to the user's dissimilarity in his typing patterns. As outliers and data correlations are typical in keystroke dynamics data, it is not surprising that classifiers using the new distance metric outperform existing top performing keystroke dynamics classifiers which use traditional distance metrics.

Although we applied the new combination of distance metric and adaptive model to the problem of matching keystroke dynamics features, there existed a few anomalies and false predictions. This can be attributed to the problem of over-fitting the data onto the model. If the tying pattern of the user is very much similar in each test case then the acceptance sphere formed has very less radius of threshold due to which the user may be asked to enter the logistics a couple of times. But there was an instance in which an imposter was recognized as a valid user.

All the false predictions attributed the problem of over-fitting. Therefore this problem can be overcome by ensemble learning methods. In our future work, we would present an algorithm which would learn how to assign weights to an average model and the Euclidian model based on the case of over-fitting the threshold.

REFERENCES

[1] A. K. Jain, P. Flynn and A. A. Ross, Handbook of Biometrics, Springer, 2008.

[2] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," Science Publishers B. V. Amsterdam, The Netherlands, Feb, Elsevier , 2000 .

[3] M. Karnan and K. M. Akila, "Biometric personal authentication using keystroke dynamics: A review," Applied Soft Computing, Elsevier,2011.

[4] D. Jamil and M. N. A. Khan, "Keystroke Pattern Recognition Preventing Online Fraud," International Journal of Engineering Science and Technology (IJEST) vol. 3, March, 2011.

[5] E. Lau, X. Liu, C. Xiao and a. X. Yu, "Enhanced User Authentication Through Keystroke Biometrics," Computer and Network Security Final Project Report, Massachusetts Institute of Technology, December 9, 2004.

[6] Yu Zhong and Yu Deng, Anil K Jain, "Keystroke Dynamics for User Authentication" , computer vision and pattern recognition workshop ,IEEE Computer society conference, June ,2012.

[7] K. S. Killourhy and R. A. Maxion, "Comparing Anomaly Detectors for Keystroke Dynamics", in Proc. 39th Annual Int'l Conf. on Dependable Systems and Networks (DSN2009), pp. 125-134, 2009.

[8] S. Hocquet, J. Ramel, and H. Cardot."User Classification for Keystroke Dynamics," Seoul, Korea, Advances in Biometrics, International Conference, ICB, 2007.

[9] S. Bleha, C. Slivinsky, and B. Hussien. Computeraccess security systems using keystroke dynamics. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(12):1217–1222,1990.

[10] H.-j. Lee and S. Cho. Retraining a keystroke dynamics based authenticator with impostor patterns. Computers & Security, 26(4):300–310, 2007.

[11] S. Haider, A. Abbas, and A. K. Zaidi. A multi-technique approach for user identification through keystroke dynamics. IEEE International Conference on Systems, Man and Cybernetics, pages 1336–1341, 2000.

[12] F. Bergadano, D. Gunetti, and C. Picardi, "User Authentication through Keystroke Dynamics", ACM Trans. Information and System Security, 5(4), pp. 367–397, 2002.

[13] S. Cho, C. Han, D. H. Han, and H. Kim. "Web-based keystroke dynamics identity verification using neural network", Journal of Organizational Computing and Eletronic Commerce, 10(4):295–307, 2000.

[14] Y. Li, B. Zhang, Y. Cao, S. Zhao, Y. Gao and J. Liu, "Study on the Beihang Keystroke Dynamics Database", Int'l Joint Conf. on Biometrics (IJCB), pp. 1-5, 2011.

[15] J. Montalvao, C. A. S. Almeida, and E. O. Freire. Equalization of keystroke timing histograms for improved identification performance. In 2006 International Telecommunications Symposium, pages 560–565, September 3–6, 2006, Fortaleza, Brazil, 2006.

[16] J. Leggett and G. Williams, "Verifying Identity via Keystroke Characteristics", Int'l J. Man-Machine Studies, vol. 28, no. 1, pp. 67–76, 1988.

[17] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies", Communications of the ACM , 33(2):168–176, 1990.

[18] P. S. Teh, A. B. J. Teoh, T. S. Ong and H. F. Neo, "Statistical Fusion Approach on Keystroke Dynamics," Third International IEEE Conference on Signal-Image Technologies and Internet-Based System , 2008.

[19] Mideth B. Abisado, Bobby D. Gerardo, and Arnel C. Fajardo , "Towards Keystroke Analysis using Neural Network for Multi-Factor Authentication of Learner Recognition in On-Line Examination" ,Manila International Conference on "Trends in Engineering and Technology" (MTET-17) , 2017 .

[20] Siti Fairuz Nurr Sadikana, Azizul Azhar Ramlib and Mohd Farhan Md. Fudzeec , "A Survey Paper on Keystroke Dynamics Authentication for Current Applications" , AIP Conf. Proc. 2173, 020010-1–020010-11; November , 2019 .