**Keystroke Authentication with Adaptive Dynamics**

**Dr. Annapurna P Patil J Geetha Reddy**

Professor, Assistant Professor,

Department of Computer Science and Engineering, Department of Computer Science and Engineering,

Ramaiah Institute of Technology Ramaiah Institute of Technology

Bangalore - 560054 Bangalore - 560054

**Ayush Kumar Bineet Kumar**

Student, Department of CSE, RIT Student, Department of CSE, RIT

**Aman Mehta Deepak Jayaprakash**

Student, Department of CSE, RIT Student, Department of CSE, RIT

***Abstract--*The demand for modern tools and techniques to restrict access to applications and services which contain delicate data is increasing exponentially each year. People can no longer rely on older methods of computer and network security as they fail in dealing with the challenges posed.**

**But even the most powerful cryptographic systems fail to prevent unauthorized access since they are static and identify the validity based on a phrase that the user must have provided. Therefore an intuitive, reliable and precise way of recognizing the identity of users based on intrinsic parameters is being researched.**

**The advantage of biometrics is that they depend on who the person is and how does he behave.**

**Biometrics, defined as the behavioral attributes and physical characteristics that make each of us unique, and hence they become an optimal choice for identity verification. Biometric attributes become the most optimal and ideal candidates for authentication since they cannot be stolen, lost or impersonated.**

*Keywords: Keystroke dynamics, Adaptive Statistical Modeling, LOOM, k-cross validation, Dynamic Thresholding*

1. INTRODUCTION

Two main causes which pose threat to data, digital network and computer system security are fraud and impersonation. Many web based authentication systems have been proposed to safeguard commercial transactions and to secure data. Ideas such as user ID’s and passwords, IP address filtering, message digest authentication, etc. are the popular ones.

However, these systems are not fully effective. For example, the rightful authentication only depends on the password. So if that gets compromised, then the data integrity of the system is lost. Hence the research is ongoing to make the system recognize the “how“ you type, of the passwords rather than the “what” you type. Systems which are trained to utilize these biometric patterns are the hot topics now in the field of security.

The most promising approach has been Keystroke biometrics. The principle of this method is to identify any pattern that a person displays while typing. Most of these patterns are involuntary and unconscious. Compared to other biometric schemas, keystroke has the primary advantages that:

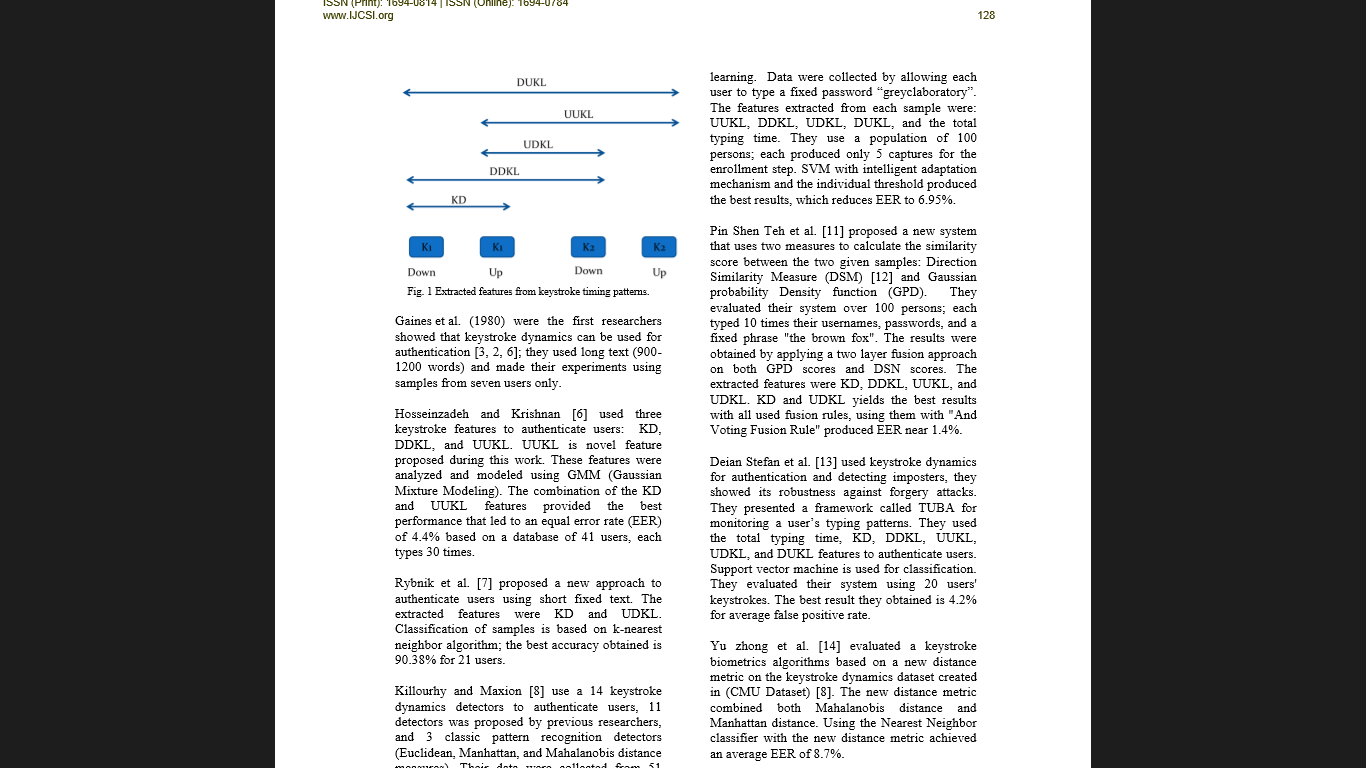
1. No external hardware like scanner or detector is needed. All that is wanted is a keyboard.

2. The “rhythm” or the pattern of the users is a very reliable statistic.

3. It can easily be deployed in conjunction with existing authentication systems.

The keystroke authentication approach has been divided into two. Majority of the approaches that are in use concentrate on “**static”** verification. It’s called static since the pattern recognizer is trained to identify a fixed-length string which acts as the password which is entered during a login process, and then their keystroke features are analyzed for authentication purpose. The second one is called as **“free-text” dynamics** which does not have a pre-determined strong. It adapts to the typing pattern on the go. The free text is still in research phase and has not been in production since the accuracy is still quite low.

1. LITERATURE SURVEY

Keystroke Dynamics has become a widely researched and active topic. This is because of the rising significance in the area of digital signatures, data integrity and security. Many existing approaches implement the static text method of keystroke dynamics [5]. The existing live implementations for “free text” are very hard to find, which can be attributed to the low accuracy rate and other problems like bias, over-fitting, false recognition rates and non-significance.****

**Fig 2.1: Keystroke timing features**

The features/ statistics used for keystroke threshold calculation are the timing recorded at the event of pressing/releasing the keys.

Fig shows the options for feature extraction:

1. The delay produced between key press and release: Delay of Key Hold
2. The delay in between two consecutive key presses:Latency of key down to key down
3. Thetime between current key release and next key release: Latency of key up to key up
4. The time spent in key release of current stroke and press of next stroke: Latency of key up to key down
5. The time spent in key press of current stroke and release of next stroke: Latency of key down to key up

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.

First true work on keystroke dynamics originated from Monrose and Rubin [23]. Their learning models were implemented by using statistics like digraph/ trigraph mean and variance

Then there were statistical Euclidean distance metrics was able to an accuracy of ~93%. Even though the data set used for the research was only around 60-70 tuples, the methods were intrinsic and valuable. In the past, keystroke biometrics has been implemented in traditional algorithms like classification and clustering algorithms. But now other Big Data technologies like Bloom Filter, Indexing, and Hashing too are gaining prominence.

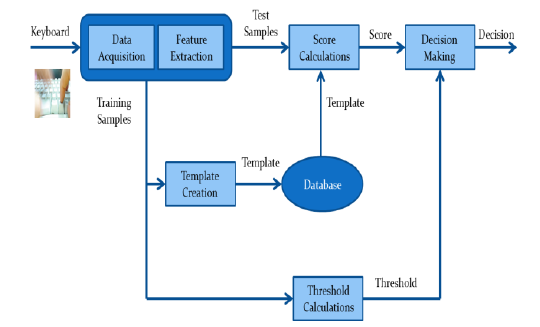
Researchers have presented their work on choosing different distance metrics ranging from Manhattan to Euclidian and combined them with mean, median and tested their suitability on the biometric authentication. For the implementation both traditional and modern classifiers have been used, including knn[5], Bayes Point Machines, K-mediods/means methods [11], genetic algorithms, neural networks , and SVMs if the attributes are non linearly dependant.

But the first application which lead to huge progress was due to the research of Joyce and Gupta [13]. Their work comprised of static keystroke dynamics. The algorithms used were simple and effective but yielded remarkably impressive results.

The design pattern and flow process of out implementation is also simple. The users are authenticated by typing in their usernames and passwords just like any other authorization system. Then instead of use comparing passwords, we run a series of ML scripts which contain 3 different algorithms explained in the later sections. The ensemble results are used to classify the user as valid or imposter.

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.

1. PROPOSED SYSTEM



**Fig 3.1: 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 3.1.**

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. Parameters are derived when test subjects enter 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. The distance (Euclidian) between trained and test are calculated to extract the score
5. In the end, the user's score and the threshold generated are compared and based on this difference the classification is decided. 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 Some key steps and their brief description:*

*3.1.1 Data extraction and feature selection:*

A test data is created to check the accuracy of the designed system. A web portal is created to record the keystroke and check for authentication. The user has to simply type his username and password that they can comfortably type and the rhythm of which they can easily remember.

The time of each keystroke is recorded in a temporary file. After some front end validation, the difference of 2 successive strokes is entered into a data frame. For our model we take the key delay between the key up of the current stroke and the key up of the next stroke.

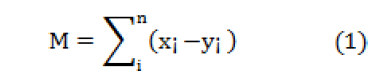
These differences become the attributes of our data set and determine the class labels of our machine learning algorithm. Typically these key delays are stored in a comma separated value fashion.

*3.1.2 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.

*Manhattan distance:*

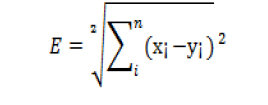
The score is calculated as in **Eq. (1)** represents Manhattan distance



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

*Euclidean threshold*

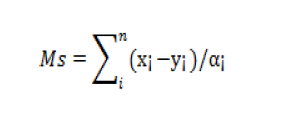
The threshold is nothing but the RMS value of user’s entered test query and the pattern that was trained at the time of ML training. The equation is as shown in **Eq (2).**

(2)

Other optimal choices for the distance measures could also be

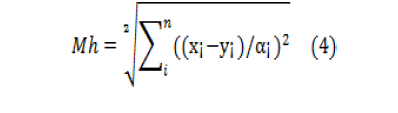
*Manhattan threshold:*

The metric used to calculate the Manhattan threshold is the standard Eq. (3) will be in the form:

 (3)

*Mahanabolis Distance:*

The standard deviation of all features with the presence of Mahanabolis distance is presented by Eq. (4)



*3.2 Threshold 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).** The method is quite comparable to the k cross validation. This method is explained below in some detail in steps:

1. Let there be x samples, split the total sample space of (x) samples itno one set used as test subset, and (x-1) samples used to as the training set.
2. Choose a distance metric (Euclidean in our adaptive model) to calculate the difference between the subset labelled as test data and the cumulative array of the means of the (x-1) training samples.
3. Iterate the step 2 for (x) times to produce (x) combinations of thresholds for each pair of coordinates.
4. The cumulative mean of these (x) combinations of thresholds is calculated to produce the one single value that would represent the effective measure of all the thresholds in total.

*3.3* ***Visual Equivalence*:**

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.

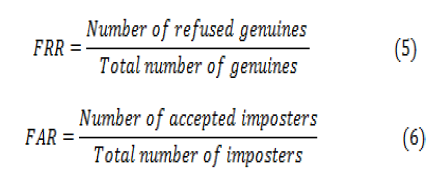
Imagine a sphere centered at this point. The threshold calculated would be the radius of this sphere. If a test data point arrives, we plot this point in space. We then check if this point inside the so formed sphere. If it does then it is equivalent to a data set which is of a valid user and it’s delay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

*3.4 Classifier metrics:*

The performance of classifier is judged using some statistics. These metrics are listed as follows:

1. **False Rejection Rate (FRR),** the ratio of valid test cases which were rejected, and
2. **False Acceptance Rate (FAR),** ratio of false test cases which were accepted.

The following equations show the formulae for the above statistics.



The accuracy of the each system/ metric can be measured using Equal Error Rate (EER).

1. IMPLEMENTATION

*4.1 Design and technology*

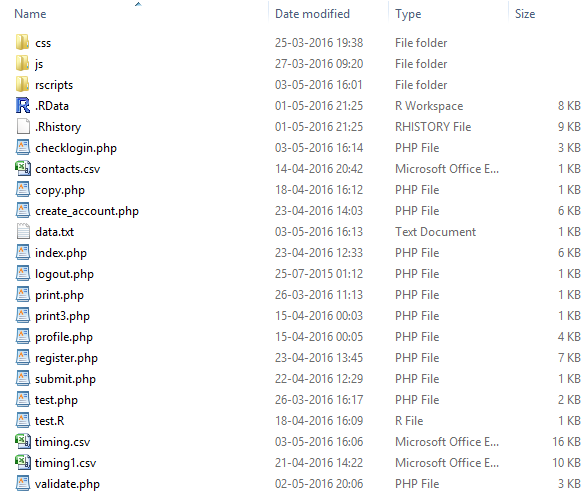
Using the design described in the above section the experiment software was successfully built, tested and used to gather data. Rather than give an in-depth analysis of the code, we shall provide an more informative overview of the technologies we used, and describe the interaction with the software 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
5. R: 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 can be very easily 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.

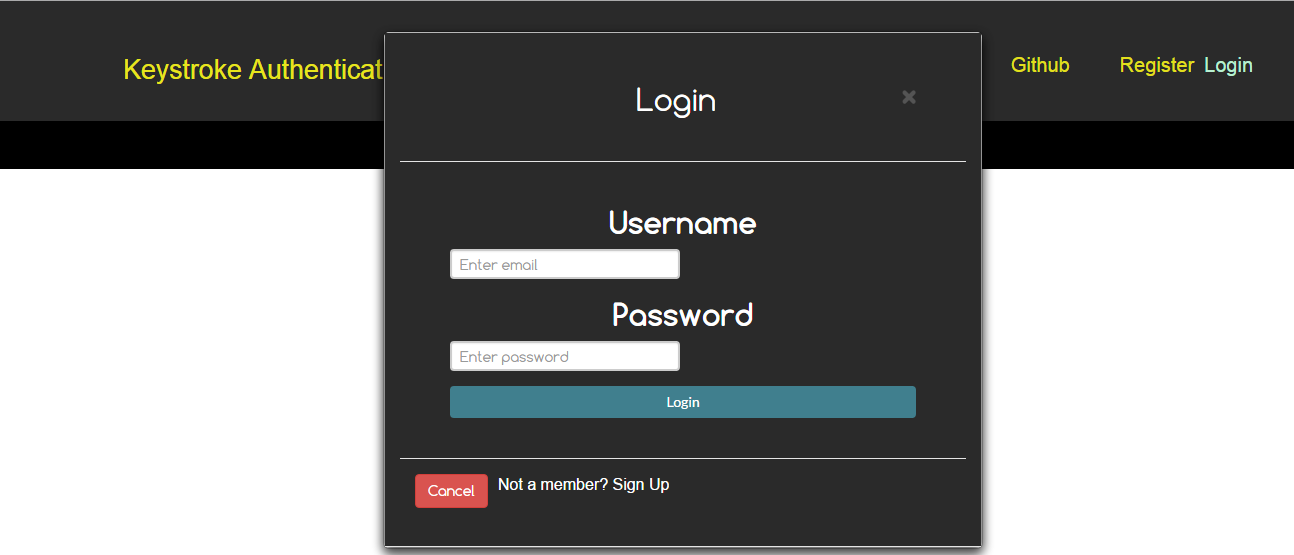
The **directory structure** used for the project is as shown in **Fig 4.1.**



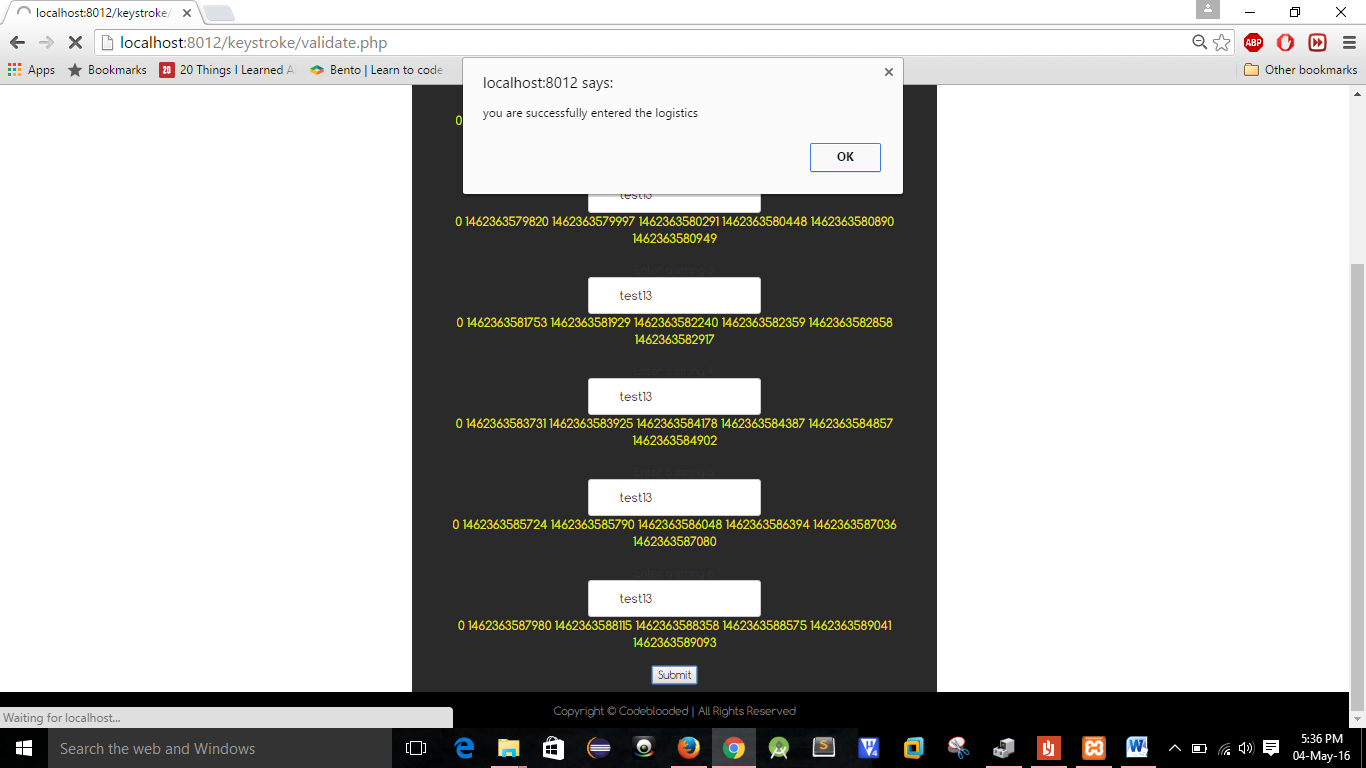
**Fig. 4.1: Directory Structure**

*4.2 Experiment*

The user first chooses a user name and a pass phrase as shown in **Fig 4.2.1** 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 4.2.2**



**Fig 4.2.1: Login Page**



**Fig 4.2.2: Entry of Statistics**

1. COMPARISON WITH EXISTING WORK

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

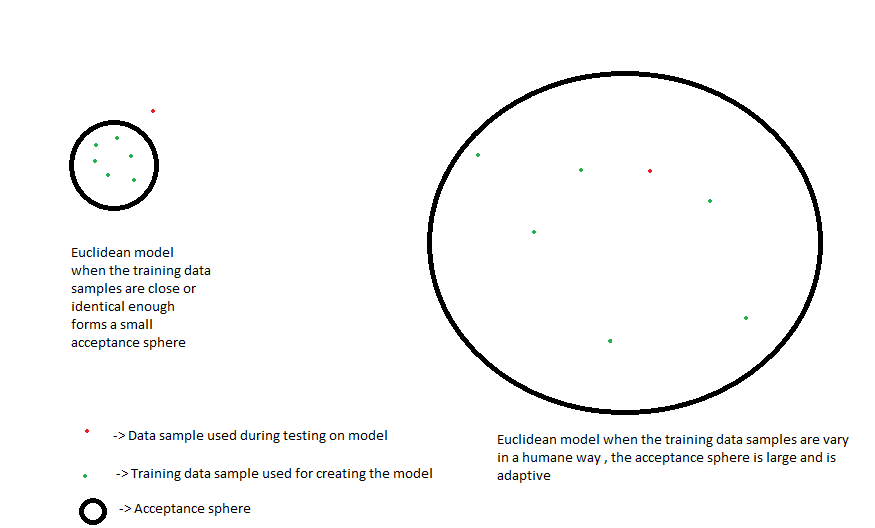
Killourhy and Maxion [2] used 14 keystroke dynamics anomaly detector to authenticate users, 11 of which were previously proposed, and 3 where classic recognition patterns using various distance statistic models. (Euclidean, Manhattan, and Mahalanobis distance measures). Their dataset composed of 51 users, who typed the password for 400 times along 8 sessions ,i.e., 50 times per session, out of which 200 samples were taken for training the model, and the rest where used for testing the model built. The features from each sample included delays and latencies mentioned above and achieved and EER of 9.6%.

Our model used two distance statistics Manhattan distance and Euclidean distance. Our dataset composed of 40, who typed the password for 6 times and authenticated using the model for 21 times during 3 sessions.

The model build consists of two distance measures:

1. ***Manhattan distances:*** Takes the average of all the training data set and compares with a threshold of 150 ms.
   * 1. 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.
     2. 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. The acceptance sphere will be small as shown in **Fig 5.1.**
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.
   * 1. 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 is taken and the average of them gave the threshold of our model.
     2. 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.

3. ***Manhattan Median Model:*** Both the models are not robust. If there exists any outlier then the models will not learn properly. The models built will be error full. This is because both the models are constructed with mean as their statistic. The statistic which gives robustness is the median, and then model is the same as used in Manhattan. This adds the robustness to the feature. Now, is the user trains the model and enters an outlier test case by mistake, the algorithm will simply ignore it and take the rest of the cases.



**Fig 5.1: Euclidian Distance Model**

Table 6.1: EER for the two distance models for our proposed system:

|  |  |
| --- | --- |
| **Distance measure** | **EER** |
| Manhattan distance | 8.9 |
| Euclidean distance | 9.8 |

Table 6.2: Comparison of models:

|  |  |
| --- | --- |
| **System** | **EER** |
| Yu Zong (2012) [1] | 8.4 |
| Kevin S. Killourhy (2009) [2] | 9.6 |
| Shimaa I. Hassan(2013)[3] | 7.0 |
| The proposed System | 8.9 |

1. CONCLUSION AND FUTURE WORK

We analysed the attributes and algorithms of keystroke dynamics and came up with a new adaptive model and statistic which would change the threshold according to the user’s dissimilarity in his typing patterns. Since we have eliminated the outliers, made the models adaptive, and shown different algorithms implementations, the models proved to be more accurate than existing models. Even after applying the newly designed metric and ensemble algorithm 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] Araújo, Lívia CF, Luiz HR Sucupira, Miguel Gustavo Lizarraga, Lee Luan Ling, and João Baptista T. Yabu-Uti. "**User authentication through typing biometrics features**." *IEEE transactions on signal processing* 53, no. 2 (2005): 851-855.

[2] Bergadano, Francesco, Daniele Gunetti, and Claudia Picardi. "**User authentication through keystroke dynamics**." *ACM Transactions on Information and System Security (TISSEC)* 5, no. 4 (2002): 367-397.

[3] Cho, Sungzoon, Chigeun Han, Dae Hee Han, and Hyung-Il Kim. "**Web-based keystroke dynamics identity verification using neural network**" *Journal of organizational computing and electronic commerce* 10, no. 4 (2000): 295-307.

[4] Zhong, Yu, Yunbin Deng, and Anil K. Jain. "**Keystroke dynamics for user authenticatio**." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 117-123. IEEE, 2012.

[5] Leggett, John, and Glen Williams. "Verifying identity via keystroke characterstics." *International Journal of Man-Machine Studies* 28, no. 1 (1988): 67-76.

[6]Kevin S. Killourhy and Roy A Maxion, **“Comparing Anomaly-Detection Algorithms for Keystroke Dynamics** Proc. of the 3rd Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, pp. 61-64, 2007.

**[7]** Killourhy, Kevin S., and Roy A. Maxion. "**Comparing anomaly-detection algorithms for keystroke dynamics.**" In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pp. 125-134. IEEE, 2009.

[8] Obaidat, Mohammad S., and Balqies Sadoun. "**Verification of computer users using keystroke dynamics."***IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27, no. 2 (1997): 261-269.

[9] Karnan, Marcus, Muthuramalingam Akila, and Nishara Krishnaraj. "**Biometric personal authentication using keystroke dynamics: A review**." *Applied Soft Computing* 11, no. 2 (2011): 1565-1573.

[10] Monrose, Fabian, Michael K. Reiter, and Susanne Wetzel. "**Password hardening based on keystroke dynamics.**" In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 73-82. ACM, 1999.

[11] Li, Yilin, Baochang Zhang, Yao Cao, Sanqiang Zhao, Yongsheng Gao, and Jianzhuang Liu. "**Study on the BeiHang keystroke dynamics database.**" In *Biometrics (IJCB), 2011 International Joint Conference on*, pp. 1-5. IEEE, 2011.

[12] S. Prabhakar, S. Pankanti, and A. K. Jain, “**Biometric Recognition: Security and Privacy Concerns**”, IEEE Security and Privacy Magazine, Vol. 1, No. 2, pp. 33-42, 2003.

[13] T. Sim and R. Janakiraman, “**Are digraphs good for freetext keystroke dynamics?** ”, IEEE CVPR, pp. 17-22, 2007.

[14] Davoudi, Homa, and Ehsanollah Kabir. "**A new distance measure for free text keystroke authentication."** In *Computer Conference, 2009. CSICC 2009. 14th International CSI*, pp. 570-575. IEEE, 2009.

[15] Monrose, Fabian, and Aviel D. Rubin. "**Keystroke dynamics as a biometric for authentication**." *Future Generation computer systems* 16, no. 4 (2000): 351-359.

[16] Singh, Saurabh, and K. V. Arya. "**Key classification: a new approach in free text keystroke authentication system.**" In *Circuits, Communications and System (PACCS), 2011 Third Pacific-Asia Conference on*, pp. 1-5. IEEE, 2011.

[17] Serwadda, Abdul, Zibo Wang, Patrick Koch, Sathya Govindarajan, Raviteja Pokala, Adam Goodkind, David-Guy Brizan, Andrew Rosenberg, Vir V. Phoha, and Kiran Balagani. "**Scan-based evaluation of continuous keystroke authentication systems.**" *IT Professional* 15, no. 4 (2013): 20-23.

[18] Montalvao, Jugurta, Carlos Augusto S. Almeida, and Eduardo O. Freire. "**Equalization of keystroke timing histograms for improved identification performance**." In *Telecommunications Symposium, 2006 International*, pp. 560-565. IEEE, 2006.

[19] PC Tools. Security guide for windows—random password generator, 2008. http://www.pctools.com/ guides/password/.

[20] Peacock, Alen, Xian Ke, and Matthew Wilkerson. "**Typing patterns: A key to user identification**." *IEEE Security & Privacy* 2, no. 5 (2004): 40-47.

[21] Mahar, Doug, Renee Napier, Michael Wagner, William Laverty, R. D. Henderson, and Michael Hiron. "**Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions**." *International journal of human-computer studies* 43, no. 4 (1995): 579-592.

[22] Woodward, John D., Nicholas M. Orlans, and Peter T. Higgins. ***Biometrics:[identity assurance in the information age****]*. New York: McGraw-Hill/Osborne, 2003.