*International Journal of Internet, Broadcasting and Communication Vol.8 No.4 19-25 (2016)*
*http://dx.doi.org/10.7236/IJIBC.2016.8.4.19*

**IJIBC 16-4-3**

# Ensemble-By-Session Method on Keystroke Dynamics based User Authentication

Jiacang Ho[1], Dae-Ki Kang[2*]

*[1]Department of Ubiquitous IT, Graduate School, Dongseo University*
*[*2]Department of Computer & Information Engineering, Dongseo University*
*[1]ho_jiacang@hotmail.com, [*2]dkkang@dongseo.ac.kr*

## *Abstract*

*There are many free applications that need users to sign up before they can use the applications nowadays. It is difficult to choose a suitable password for your account. If the password is too complicated, then it is hard to remember it. However, it is easy to be intruded by other users if we use a very simple password. Therefore, biometric-based approach is one of the solutions to solve the issue. The biometric-based approach includes keystroke dynamics on keyboard, mice, or mobile devices, gait analysis and many more. The approach can integrate with any appropriate machine learning algorithm to learn a user typing behavior for authentication system. Preprocessing phase is one the important role to increase the performance of the algorithm. In this paper, we have proposed ensemble-by-session (EBS) method which to operate the preprocessing phase before the training phase. EBS distributes the dataset into multiple sub-datasets based on the session. In other words, we split the dataset into session by session instead of assemble them all into one dataset. If a session is considered as one day, then the sub-dataset has all the information on the particular day. Each sub-dataset will have different information for different day. The sub-datasets are then trained by a machine learning algorithm. From the experimental result, we have shown the improvement of the performance for each base algorithm after the preprocessing phase.*

***Keywords:*** *Keystroke dynamics, user authentication, ensemble-by-session method*

## 1. Introduction

Most of the free applications on the web require a user to register an account in order to use their service. It is a difficult decision for a user to choose an easy to remember for the user but difficult for an imposter to hack the password. If we choose a simple password, the imposter can intrude the account easily. If we choose a complex password, it is difficult to be remembered. If we write down the password on a paper, the paper may be disappeared on one day or we might forget where the password we have written or placed.

Manuscript Received: Sep. 12, 2016 / Revised: Sep.20, 2016 / Accepted: Sep. 27, 2016
Corresponding Author: dkkang@dongseo.ac.kr
Tel: +82-51-320-1724, Fax: +82-51-327-8955
Department of Computer & Information Engineering, Dongseo University

Therefore, to solve this problem, we can integrate the login system with a behavior-based approach. The behavior-based approach includes keystroke dynamics on the mouse, keyboard or mobile device, gait analysis, etc. The behavior-based approach provides inexpensive cost, implement with no extra hardware and it is easy to be implemented. In the study, we focus on the keystroke dynamics on the keyboard. Due to these advantages, there has been a considerable amount of researchers performing behaviour-based research because they believe that keystroke dynamics can increase the level of the security system and it can be a common safety feature in the future [1 – 10].

Keystroke dynamics is the automated method of identifying the personality of an individual based on the rhythm of typing on a keyboard [11]. The two common keystroke dynamics that researchers used in their papers are dwell time and flight time. The dwell time is the time interval between a key being pressed and released. The flight time, on the other hand, is the time interval between a key being released and a next key being pressed. We show the possible keystroke dynamics as following:

- Hold (H): time interval (or a dwell time) of pressing a key
- Up-Down (UD): time interval (or a flight time) between key-up of the first key and key-down of the second key.
- Down-Down (DD): time interval between key-down of the first key and key-down of the second key.
- Up-Up (UU): time interval between key-up of the first key and key-up of the second key.
- Down-Up (DU): time interval between key-down of the first key and key-up of the second key.

A machine can learn a user's typing behavior [8], emotion [12], gender [13], dominant hand [4], etc. with the keystroke data and a machine learning algorithm. Since the fact that every user has a different style of typing pattern, it is difficult for an imposter to intrude the user account.

We explain our proposed method in next section (Section 2). Section 3 describes the experimental method. We show the experimental result in Section 4. We present the related work in Section 5. Last but not least, we conclude the paper in Section 6.

## 2. Ensemble-By-Session Method

Ensemble methods are machine learning algorithms that produce multiple classifiers to classify new data by selecting the majority vote of their predictions [14]. In the paper, we have proposed ensemble-by-session method (EBS). The EBS is a preprocessing method before the training phase and testing phase. It will generate multiple sub-datasets from the original dataset. Later, we train each sub-dataset with a machine learning algorithm and achieve a model. We explain the detail of the distribution of sub-datasets in the following sub-sections.

### 2.1 Distribution of sub-dataset

During the preprocessing phase, the EBS method has divided the dataset into multiple sub-datasets based on the session given as illustrated in Fig. 1.

In the Fig. 1, the dataset has three attributes (i.e. x, y, and z) and 12 instances per attribute. From this dataset, every three instances of each attribute are inserted as one session during the enrolment phase. The session refers to one day. The dataset has collected the user data for four sessions. In common case, we will combine them as a single dataset and operate the classification. However, in this paper, we use the dataset based on the session. In other words, we do not assemble them into one dataset, but we use the dataset session by session.

### 2.2 Training phase and testing phase

Theoretically, we have two important phases in the experiment, which are the training phase and the testing phase. In the training phase, we have created a model for each sub-dataset. During the testing phase, we have tested a testing data with the models that we have generated from the training phase. Each model will produce a score for the testing data. In the end, we total up all of the scores as a final score which is then used to generate a receiver operating characteristic (ROC) curve. We can calculate an equal error rate (EER) from the ROC curve.

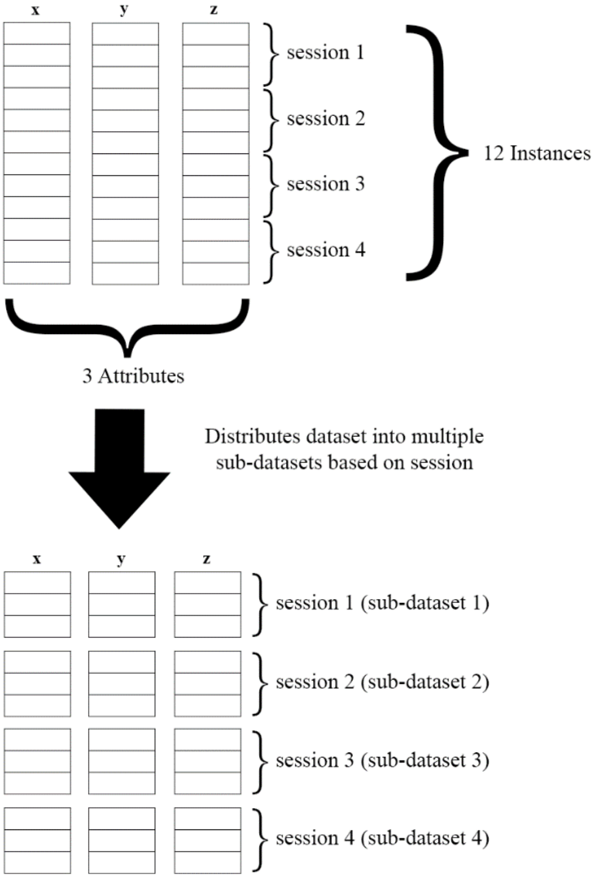
x
y
z
session 1
session 2
session 3
session 4
12 Instances
3 Attributes
Distributes dataset into multiple sub-datasets based on session
x
y
z
session 1 (sub-dataset 1)
session 2 (sub-dataset 2)
session 3 (sub-dataset 3)
session 4 (sub-dataset 4)

**Figure 1. Ensemble-By-Session concept**

## 3. Dataset

### 3.1 CMU benchmark dataset

We have only used CMU benchmark dataset [5] in our experiment. The reason is CMU dataset has collected user keystrokes for eight sessions. Each session has 50 instances. There are 51 subjects have participated during the enrollment phase. Therefore, it consists of total 20,400 instances in the dataset. The password used in the dataset is ".tie5Roanl[enter]" ('[enter]' is an ENTER key on the keyboard). There are 31 different attributes (keystrokes) in this dataset. The attributes are 11H, 10UD, and 10DD. To construct the experiment, we have selected one of the subjects to be a genuine user. The remaining subjects are the imposters. During the training phase, we use the first 200 instances (from first four sessions) of the selected subject as the training data. As we have mentioned in Section II, we have split the dataset into four sub-dataset which each sub-dataset consists of 50 instances. During the testing phase, we use the last 200 instances of the selected subject as the testing data for the genuine user. Meanwhile, we use first five instances from the remaining subjects (except the selected subject) as the testing data for the imposter. The reason we have extracted first five instances from the imposters is because of the assumption that the imposter is unfamiliar with the password in the experiment [5]. We have tested 51 subjects to obtain average equal error rate.

### 3.2 Performance criteria using ROC curves

For the performance evaluation, we have selected Receiver Operating Characteristic (ROC) curve to perform the comparison of each algorithm between with EBS method and without EBS method. We have provided some examples of ROC curve in Fig. 2. In the graph, the x-axis denotes as the false positive rate (FPR) which is the ratio of genuine users has misclassified as imposters to the total number of genuine users. Y-axis, on the other hand, labels as the true positive rate (TPR). TPR is the ratio of a number of the genuine user has classified correctly to the total number of genuine users. If a line is closer to the (0.0, 1.0) coordinate, we can determine the line has high performance. In Fig. 2, we have observed the dashed lines on three graphs are closer to the (0.0, 1.0) coordinate than the solid lines. The dashed line is the performance of median vector proximity [1] algorithm with EBS method in CMU dataset. The solid line, however, is the performance of median vector proximity algorithm without EBS method in CMU dataset. From the ROC curve, we measure equal error rate (EER). The EER is an FPR value of the point on the intersection of a diagonal line (from the top left corner of the graph to the bottom right corner of the graph) and the ROC curve. Besides that, we can evaluate a method outperforms another method if its area under the curve (AUC) is larger than those of another method.

## 4. Experimental Result

As aforementioned, we have used CMU benchmark dataset only which is an appropriate dataset for our experiment. We have tested the dataset with five distance-based algorithms which include Euclidean distance, Manhattan distance, Mahalanobis distance, Manhattan (scaled) distance[5], and median vector proximity [1].

In Table 1, we have shown the average of equal error rate and its standard deviation for five algorithms. We have performed two different experiments. In the first experiment, we have created a model by using the whole dataset. However, in the second experiment, we have used EBS method, which distributes the dataset into multiple sub-datasets. The results of the first and second experiment have shown in Table 1 labelled with "Without EBS" and "With EBS" respectively.

**Table 1. The average of equal error rate with their standard deviation for five algorithms on the CMU dataset. The significant improvement on the performance of the algorithm is in bold. bold.**

| Algorithm | EER | |
|---|---|---|
| | Without EBS | With EBS |
| Euclidean | 0.171 (0.095) | 0.171 (0.096) |
| Manhattan | 0.153 (0.092) | **0.144 (0.081)** |
| Mahalanobis | 0.110 (0.065) | 0.112 (0.058) |
| Median Vector Proximity | 0.080 (0.062) | **0.070 (0.069)** |
| Manhattan (scaled) | 0.096 (0.069) | **0.090 (0.059)** |

From Table 1, we have observed three algorithms have significant improvement in their performance. The algorithms are manhattan distance, manhattan (scaled) distance, and median vector proximity. Manhattan distance has improved 0.9% with EBS method, manhattan (scaled) distance has decreased 0.4% of the EER and median vector proximity has reduced 1% of the EER. Euclidean distance has shown the same result in both experiments. Although Mahalanobis has slightly increased in the average of EER, but the standard deviation of EER has reduced 0.7% at the same time.

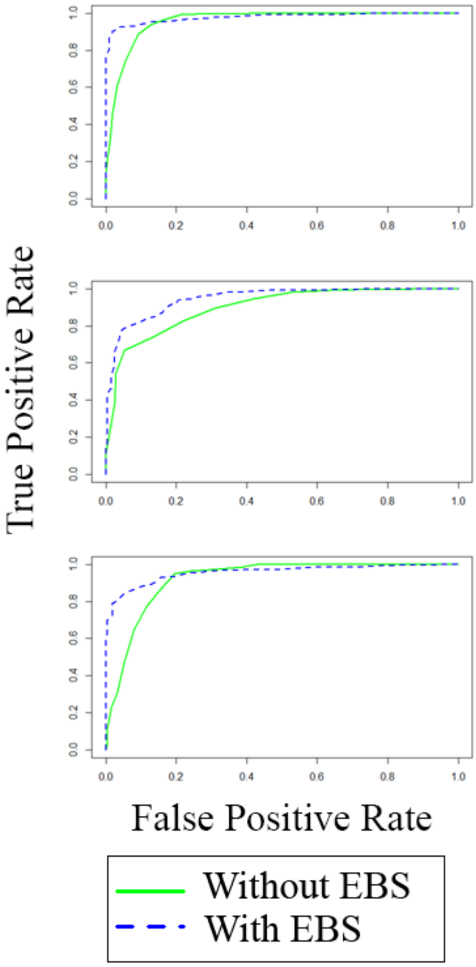

**Figure 2. ROC curves for median vector proximity algorithm with and without EBS in CMU dataset**

## 5. Related work

Al-Jarrah [1] has proposed median vector proximity in his paper. In his paper, instead of using mean, he has used median which produces very effective performance in his result. We have reproduced his work and performed the work with our proposed method, Ensemble-By-Session method. Surprisingly, the median vector proximity algorithm has improved in its performance by 1% with our EBS method.

Montalvão's et al. [6] have published two attractive analyses in their paper. They have described all subjects have a similar rhythmic profile (but with different time length in term of complete a full password) when the same password is given. Furthermore, they have discovered that different keyboard device does not affect much to the corresponding profile. The second analysis in their paper inspects the length of the password can affect the error rate. The longer the password length, the lesser the error rate. These analyses have provided very helpful information for every researcher who has researched about the keystroke dynamics.

Killourhy and Maxion [5] have presented interesting results with several machine learning algorithms in their dataset, CMU dataset. We have found that the CMU dataset is an appropriate dataset for our method because it consists of the session information which allows us to distribute the dataset into multiple sub-dataset based on the session value. Besides that, each user for each session has enough instances to train a model.

## 6. Conclusion

In this paper, we have proposed ensemble-by-session (EBS) method. EBS has split the dataset into multiple sub-datasets. From the result, some algorithms have shown significant improvement of the performance of the EBS method. We believe that the distribution of the dataset based on the session can improve the performance of the algorithm.

For the future work, we would like to work on the small dataset which is similar in our real life case. We will insert our password 1-3 times for a particular account per day. Hence, we would like to test with the small dataset that could bring high accuracy and performance in the result.

## Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. NRF-2015R1D1A1A01061328).

## References

[1] M. M. Al-Jarrah, "An anomaly detector for keystroke dynamics based on medians vector proximity," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 6, pp. 988–993, 2012.

[2] S. Cho, C. Han, D. H. Han, and H.-I. Kim, "Web-based keystroke dynamics identity verification using neural network," *Journal of organizational computing and electronic commerce*, vol. 10, no. 4, pp. 295–307, 2000.

[3] R. Giot, M. El-Abed, and C. Rosenberger, "Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis," in *Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP), 2012 Eighth International Conference on.* IEEE, 2012, pp. 11–15.

[4] S. Z. S. Idrus, E. Cherrier, C. Rosenberger, and P. Bours, “Soft biometrics for keystroke dynamics,” in *Image analysis and recognition*. Springer, 2013, pp. 11–18.

[5] K. S. Killourhy, R. Maxion et al., “Comparing anomaly-detection algorithms for keystroke dynamics,” in *Dependable Systems & Networks, 2009. DSN’09. IEEE/IFIP International Conference on*. IEEE, 2009, pp. 125–134.

[6] J. Montalv˜ao, E. O. Freire, M. A. Bezerra Jr, and R. Garcia, “Contributions to empirical analysis of keystroke dynamics in passwords,” *Pattern Recognition Letters*, vol. 52, pp. 80–86, 2015.

[7] K. Revett, “A bioinformatics based approach to user authentication via keystroke dynamics,” *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 7–15, 2009.

[8] Z. Syed, S. Banerjee, and B. Cukic, “Leveraging variations in event sequences in keystroke-dynamics authentication systems,” in *High- Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on*. IEEE, 2014, pp. 9–16.

[9] X. Wang, F. Guo, and J.-f. Ma, “User authentication via keystroke dynamics based on difference subspace and slope correlation degree,” *Digital Signal Processing*, vol. 22, no. 5, pp. 707–712, 2012.

[10] E. Yu and S. Cho, “Keystroke dynamics identity verification - its problems and practical solutions,” *Computers & Security*, vol. 23, no. 5, pp. 428–440, 2004.

[11] R. Moskovitch, C. Feher, A. Messerman, N. Kirschnick, T. Mustafić, A. Camtepe, B. Lőhlein, U. Heister, S. Mőller, L. Rokach et al., “Identity theft, computers and behavioral biometrics,” in *Intelligence and Security Informatics, 2009. ISI’09. IEEE International Conference on*. IEEE, 2009, pp. 155–160.

[12] A. N. H. Nahin, J. M. Alam, H. Mahmud, and K. Hasan, “Identifying emotion by keystroke dynamics and text pattern analysis,” *Behaviour & Information Technology*, vol. 33, no. 9, pp. 987–996, 2014.

[13] R. Giot and C. Rosenberger, “A new soft biometric approach for keystroke dynamics based on gender recognition,” *International Journal of Information Technology and Management*, vol. 11, no. 1-2, pp. 35–49, 2012.

[14] T. G. Dietterich, “Ensemble methods in machine learning,” in *Multiple classifier systems*. Springer, 2000, pp. 1–15.