J Inf Process Syst, Vol.14, No.2, pp.523~538, April 2018
https://doi.org/10.3745/JIPS.03.0093

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Feature Subset for Improving Accuracy of Keystroke Dynamics on Mobile Environment

Sung-Hoon Lee*, Jong-hyuk Roh**, SooHyung Kim**, and Seung-Hun Jin**

Abstract
Keystroke dynamics user authentication is a behavior-based authentication method which analyzes patterns in how a user enters passwords and PINs to authenticate the user. Even if a password or PIN is revealed to another user, it analyzes the input pattern to authenticate the user; hence, it can compensate for the drawbacks of knowledge-based (what you know) authentication. However, users' input patterns are not always fixed, and each user's touch method is different. Therefore, there are limitations to extracting the same features for all users to create a user's pattern and perform authentication. In this study, we perform experiments to examine the changes in user authentication performance when using feature vectors customized for each user versus using all features. User customized features show a mean improvement of over 6% in error equal rate, as compared to when all features are used.

Keywords
Feature Subset, Keystroke Dynamics, Smartphone Sensor

1. Introduction

Keystroke Dynamics Authentication (KDA) is a behavior-based user authentication technology. Behavior-based user authentication has several advantages. It can overcome the limitations of existing widely-used knowledge-based authentication methods such as passwords and PINs, in which authentication can occur when a password has been exposed. Even if an attacker enters the same password or PIN, the attacker does not know a genuine user's input pattern; thus, comparing input patterns can strengthen the security of knowledge-based user authentication.

KDA began in the 1980s computer environment with research into analyzing the input patterns of users entering passwords on a keyboard to perform user authentication, and extensive research has been performed since then [1]. For conventional mobile phones, KDA was firstly applied in 2006 [2]. Since 2010, as smartphones have become widespread, there have been efforts toward applying KDA to the smartphone environment [3]. Smartphones are generally equipped with several sensors such as accelerometers, gyroscopes, and touch screens. Thus, more information can be used for analyzing user input patterns.

Manuscript received June 28, 2017; first revision August 17, 2017; accepted September 24, 2017.
Corresponding Author: Sung-Hoon Lee (sunghoon1130@etri.re.kr)
* Information Security Engineering, University of Science and Technology, Daejeon & ETRI, Korea (sunghoon1130@etri.re.kr)
** Information Security Research Division, Electronics and Telecommunications Research Institute, Daejeon, Korea ({jhroh, lifewsky, jinsh}@etri.re.kr)

One of the conventional of KDA is a time-based feature, which uses the time of events where a user presses and releases a key (key-down, key-up) [4]. In addition, time-based features are frequently used in the smartphone environment. In addition to these features, KDA in smartphones employs sensor-based features [5,6] that use the various sensors installed in smartphones (accelerometer, gyroscope, touch screen [5]. It can use sensors to analyze several aspects of a user's input pattern, including how the phone shakes when a key is pressed, touch location, and touch size. As various services use smartphones, several short PINs are being used for user authentication to protect the data saved in smartphones. Hence, active research is being performed on PIN-based KDA for the smartphone environment.

Most studies [7] use the following steps in research and experimentation for improving KDA performance as shown Fig. 1.

1. Collect data from several users using the same predetermined PIN
2. Extract time-based and sensor-based features from the collected data
3. Perform training assuming one user is the genuine user, and perform classification experiments assuming the other users are attackers
4. To obtain best EER, perform research (e.g., find customized parameter set of the classifier or transform the classification algorithm) to improve classification performance based on the experiment results

All extracted features are used, and part of the genuine user's collected data is used as training data, while the rest is used as the genuine user's test data. At this time, other users' data are used as test data. A training model is created from the genuine user's training data, and this model is compared to all features extracted from newly entered data to determine whether it is the genuine users or not. This procedure has the following limitations:

1) If all users input the predefined same PIN and a user is not familiar with the PIN, a natural input pattern is not created within short period, e.g., 5 times or 10 times in a session. Several works performed an experiment with short period dataset [8]. In a situation the user is not familiar with the same PIN, it is too short to adjust the PIN.

2) Classification is improved because users enter the PIN using their input patterns, regardless of the genuine user's input pattern; hence, the input pattern of each user is different.

3) Existing studies do not consider that classification performance reduces when all feature types are used instead of feature types with good discrimination of input patterns.

In this study, we perform experiments that confirm that higher classification performance can be achieved when collecting data from PINs that users are familiar—when collecting data from PINs that users are not familiar with, users need a time to naturally input the PINs. Users input the PINs not naturally at early session (input 5 times per session)—with and using feature types with high discrimination of users, as compared to when this is not the case. The contributions of our study are as following:

- We collect data set that imposters mimic a genuine users' input pattern.
- We confirm that each user has own input pattern even imposters try to mimic the genuine users' input pattern.
- We confirm that each user has own discriminate feature subset and classification performance is increased when using each users' feature subset.

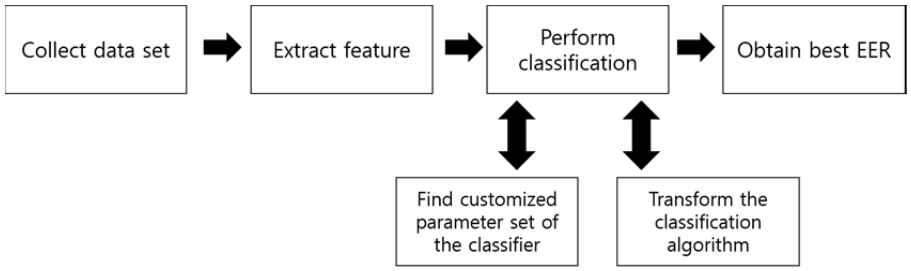

Fig. 1. Extracting time-based features from a time event.

When data were collected in previous research, all users were asked to input the same PIN, whereas we allowed each user to use a familiar PIN (6 digits of the user's phone number remaining after removing the telecom ID number). In addition, after other users observed the genuine user entering his/her PIN (a peeking over the shoulder attack scenario), we asked them to imitate the genuine user's PIN input pattern. In this manner, we could confirm whether KDA can distinguish between the genuine user and the attacker even when the attacker imitates the genuine user's input pattern.

Additionally, we confirmed this through experiments on feature type vectors with high discrimination which can better express each user's input patterns by distinguishing feature vectors by the data types (time, accelerometer, linear accelerometer, gyroscope, uncalibrated gyroscope) obtained from all feature extracted from the data.

Additionally, we also confirmed that classification performance is increased when using a feature type (time, accelerometer, linear accelerometer, gyroscope, uncalibrated gyroscope) that represents the user's input pattern versus using all feature types.

The structure of the rest of the paper is as follows: in Section 2, we describe and compare previous research, and in Section 3, we describe the experimental procedure. In Section 4, we analyze experiments and their results, and finally, we provide the conclusions in Section 5.

2. Related Works

Research in KDA for smartphones has been actively pursued since 2013 [7]. Time-based extraction is the conventional method of extracting features from keystroke dynamics. The moment at which a key is pressed is referred to as key down (KD, D), and the moment at which it is released is referred to as Key Up (KU, U). When a user enters their PIN, KD and KU times are saved, and DD, DU, UD, and DD features are extracted, as shown in Fig. 2 and Table 1. Sensor-based features use the maximum, minimum, and average values of sensor changes within a certain time period (for example, the DU time period) as features [5].

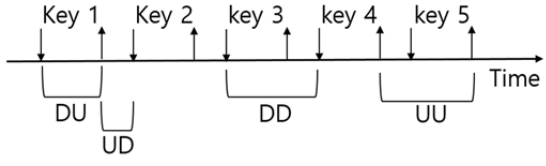

Fig. 2. Extracting time-based features from a time event.

Table 1. Time-based features

Feature	Description
DU	Refers to the time between the pressing and release of a key; shows how long the user pressed a single key. Also referred to as dwell time [5].
UD	Refers to the time between the released of one key is and the pressing of the next key. Also referred to as flight time [5].
DD	Refers to the time between the pressing of one key and the next key.
UU	Refers to the time between the release of one key and the next key.

Table 2. Research study on keystroke dynamics in the smartphone environment

Study	Input string	Feature	Normalization	Feature selection	Classifier	EER (%)
Saevanee et al. [9]	10-D	F, D, P	-	X	Neural network	1
Zheng et al. [10]	4, 8-D	A, F, D, P, S	O	X	Distance-based	3.65
Giuffrida et al. [5]	8-9 C	A, G, F, D	-	X	OCSVM, kNN	0.08
Teh et al. [11]	4, 16-D	F, D, P, S	-	X	Statistical-based	5.49
Pisani and Larena [12]	10 C	F	O	X	Immune algorithm	13
Ho [6]	4-D	A, F, D, S	-	X	Distance-based	15

C=character, D=digit, A=accelerometer-based feature, G=gyroscope-based feature, F=flight feature, D=dwell feature, P=pressure feature; S=size feature.

Table 2 summarizes the research on keystroke dynamics for smartphones. In a previous study [7], 4-digit PINs and 8-digit PINs were collected from 80 users, which was equivalent to 11,062 data points. Dwell (DU) and flight (UD) were used as time features, the x, y, and z axes were obtained from an accelerometer, and a Euclidean norm was used to extract features.

The time features and sensor features were combined and an error equal rate (EER) of 3.5% was obtained. In another study [5], 8-character (internet) and 9-character alphabet strings were used as passwords. Eleven categories of features (RMS, RMSE, Min, Max, AvgDeltas, NumMax, NumMin, TTP, TTC, RCR, SMA) were extracted from x, y, z axes of an accelerometer and gyroscope sensors during the dwell period. Data were collected from 40 users and experiments were performed, resulting in a performance with an EER of 0.5%.

Four-digit PIN and 16-digit PIN data were collected from 150 people and touch size and touch pressure were used as features [11]. Dwell and flight were also used as time feature. As PIN length increased from 4 to 16, the EER decreased from 8.55% to 5.49%, which was the performance improvement of approximately 35.39%. Performance differed slightly depending on PIN length; however, user convenience decreased because the users had to enter more PIN digits.

3. Experimental Process

3.1 Data Acquisition Scenario

There are several scenarios where a PIN can be exposed. There are cases where the server where the PIN number is stored is captured through hacking by a malicious attacker (hacker), and where the PIN

number is revealed to a malicious attacker as it is entered by a genuine user through a peeking over the shoulder attack. In addition, there are the following three cases where a malicious attacker can attempt authentication with an exposed PIN:

1. When authentication is attempted on the attacker's phone with a stolen PIN.
2. When authentication is attempted on the attacker's phone with a PIN stolen through an over the shoulder attack.
3. When authentication is attempted on the genuine user's stolen phone with a PIN through an over the shoulder attack.

This study aims to confirm whether the input pattern of a genuine user can be distinguished from that of an imposter on the genuine user's phone when a PIN is exposed to a malicious attacker through a peeking over the shoulder attack and the malicious attacker attempts authentication by entering the PIN on the genuine user's phone. When the genuine user enters the PIN, imposters watch the genuine user, and they can view the user's PIN and input pattern.

The PINs were 6 digits from each user's personal mobile phone numbers. In Korea, mobile phone (cellular phone) numbers use 3 digits for each telecom's unique number and 8 digits for an individual's unique number. We used the first 6 digits of the 8 unique digits. We asked the users to input a number familiar to them so that we could represent the user's input pattern in detail. We conducted data collection with each user's 6 digits of the 8 individual's phone number by consent of users. We described the reason why used the 6 digits and the 6 users approved.

3.2 Data Collection

Data collection was performed with 6 users and Samsung Galaxy S6 phone. The training data used to create the users' keystroke dynamics patterns were collected by entering the PIN 10 times each on two isolated sessions, for a total of 20 samples as the training data of each user. After collected the training data, one of the 6 users input the PIN 5 times and other 5 users watched the one user's input pattern as a genuine user entered the test data 5 times. And then, test data of the 5 users, assuming impostor, were collected. The collected data is shown as Fig. 3.

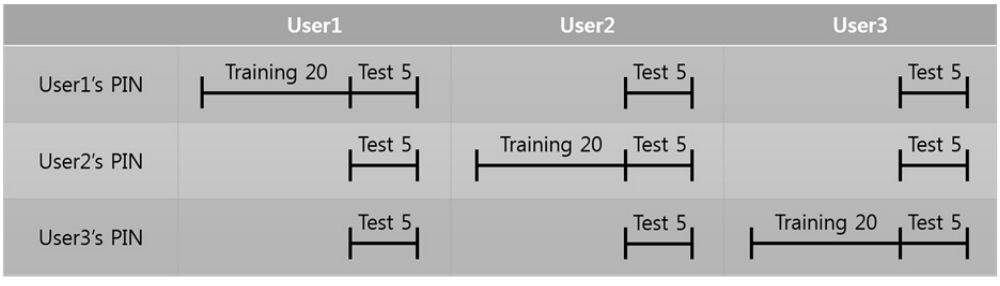

Fig. 3. Data collection size of training data and test data for each user.

The data collected from the smartphones consisted of time data, sensor data, and touch data. Time data were recorded when the PIN digits were pressed and released using the JAVA API System and nanotime method. Sensor data were collected from the following 4 sensors: accelerometer (Acc), linear

acceleration (Lacc), gyroscope (Gyr), and uncalibrated gyroscope (Ugyr). The details of each sensor is summarized in Table 3. As shown in Fig. 2, all sensor change values were stored for the X, Y, and Z axes during the time period starting from when the first PIN digit was pressed until the last PIN digit was released. When the sensor change data was saved, the time of the changes was saved. These saved times were used to extract sensor features within specific time periods.

Table 3. Description of sensors [13]

Sensor	Description
Accelerometer	Measures the acceleration applied to the device and is always influenced by the force of gravity.
Linear acceleration	Measures the acceleration applied to the device without the force of gravity.
Gyroscope	Measures the rate of rotation around the devices' local X, Y and Z axis with drift compensation.
Uncalibrated gyroscope	Measures the rate of rotation around the devices' local X, Y and Z axis without drift compensation. Also provides estimated drift around X, Y and Z axis.

The rate of data gathering can be adjusted by setting parameter values for the method used to collect sensor data. Android API provides preset parameter settings (FASTEST, GAME, UI, NORMAL). The FASTEST setting returns sensor changes in the fastest time possible, providing the highest rate. The UI setting returns sensor changes as suitable for a user interface, providing the lowest rate [13]. The FASTEST setting can collect sensor change values in the most detail; however, it does not always return sensor values at fixed intervals. After collecting and checking actual data, it was confirmed that error was present. This is because the method for providing rate is based on average rate rather than an always fixed rate. The data collected when sensor values changed were saved at an average of every 4900000 ns (every 0.004 seconds) at FASTEST setting. The FASTEST setting is used to collect sensor values for our study.

3.3 Feature Extraction

Features were extracted from the data collected from a smartphone to create user patterns. For time-based features, the times when a user pressed (KD) and released (KU) each digit were used to extract DU, UD, DD, UU features as shown in Fig. 4. For sensor-based features, the average, minimum, and maximum values of the sensor changes during the DU period were extracted as features [5]. When recording sensor data, sensor change values were recorded with the time of the values. Time information recorded along with sensor change values is used when extracting features for the sensor change values that occur during a certain period (DU period) as shown in Fig. 4. The time events saved when the accelerometer changes values do not match the DU time precisely; however, there are methods to extract sensor features for each key within the DU time period nanosecond on average and create input patterns. In addition, there are methods for creating input patterns using the maximum, minimum, and average values for all PIN periods as features.

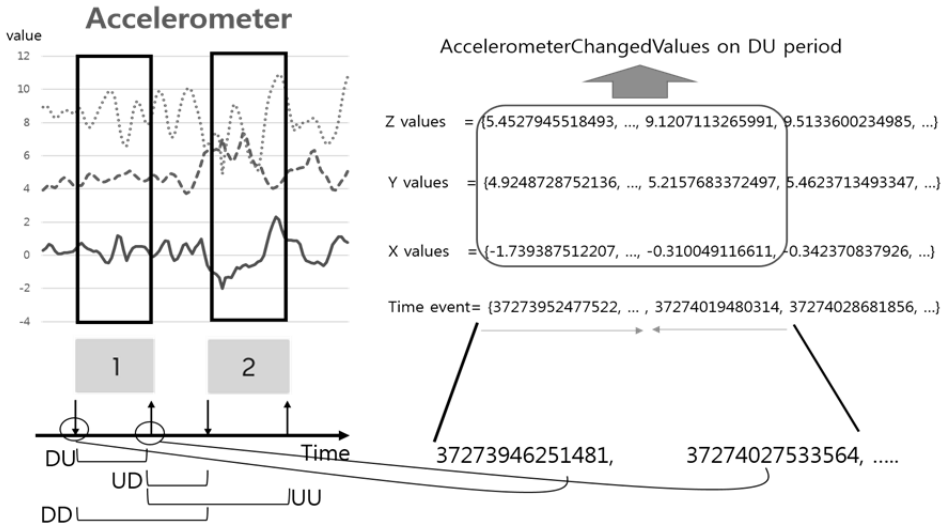

Fig. 4. Data collection and feature extraction of accelerometer.

3.4 Feature Normalization

Extracted features have different ranges. The range of time feature values is considerably larger than that of sensor feature values. Thus, sensor features, which have a small range, have almost no effect on classification. Because of this, feature values must be normalized so that they are within the same range. This is referred to as normalization or scaling. Frequently used normalization methods include min-max scaling, which uses minimum and maximum values, and Z-score which uses means and distributions [14].

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

$$X_{norm} = \frac{x - \mu}{\sigma} \tag{2}$$

	Time-based features			Accelerometer based features		
Rawdata 1	150993291,	64827375,	86165916, ...,	-2.470816135406, ...	-1.3060370683670, ...	-1.928529477119, ...
Rawdata 2	135486708,	65056583,	70430125, ...,	-3.253720045089, ...	-1.4389152526855, ...	-2.416892197999, ...
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Normalization	↓	↓	↓	↓	↓	↓
Normalized feature 1	0.67129408,	0.61422717,	0.59163097, ...,	0.70449309, ...	0.54831287, ...	0.6918538, ...
Normalized feature 2	0.35626903,	0.61879821,	0.17383426, ...,	0.327765, ...	0.46319018, ...	0.32575366, ...
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Fig. 5. Feature normalization from raw data features.

Using the min-max formula (1) for normalization creates a range from 0 to 1. Using the Z-score formula (2) for normalization creates a range that varies depending on features. Normalization is calculated according to the same feature type. For example, as shown in Fig. 5, the actual time features and sensor features go through the normalization process and are converted into feature values with the same range. In the data set which we collected, min-max scaling showed a better performance than Z-score with an average EER of 2 to 3%. Thus, min-max scaling was used in the experiments in this study.

3.5 Classification

A process is required for using the features extracted from raw data to train a user's pattern and create a classification model, which is compared to newly entered data to determine whether the user is genuine or not. Only the genuine user's data are required for training, and the classification results produce true or false. And impostor's data is not collected in KDA of real world. Hence, a One-Class Classifier (OCC) is required. In this study, a one-class SVM (OCSVM) was used to distinguish users. An OCSVM is a modification of the SVM classifier, in which only one person's data are used for training and the classification boundaries of training data are formed as maximum margins so that users are distinguished by whether or not newly entered data are included within that boundary line [15].

The open source scikit-learn program was used for OCSVM training and classification [16]. In classification using the OCSVM, the extent to which a boundary line (the curve that determines classification) is created along a margin determines whether or not it is a flexible classifier. This affects classification performance. Default parameter setting values were used (kernel: rbf, nu: 0.5, degree: 3, gamma: 'auto').

In addition, a threshold value is an important factor that affects classification performance. Methods for setting the threshold value are divided into methods that set the value as calculated by a predetermined algorithm and heuristic methods [17] that set the optimal value determined through experimentation. In this study, we used a heuristic method for finding the optimal threshold values for our experiments.

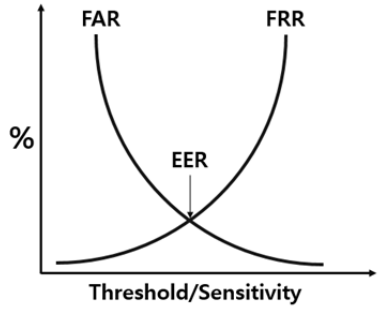

Fig. 6. The relationship between the FAR, FRR, and EER.

3.6 Equal Error Rate

EER (Equal Error Rate) is the index used for performance evaluation of behavior based authentication. EER is calculated through FAR (False Acceptance Rate) and FRR (False Reject Rate). FAR means the ratio of impostor user data to genuine user's data, while FRR means the ratio of genuine

user's data to impostor user's data. The FAR and the FRR are correlated with each other, and the point at which the two indicators are equal is referred the EER as shown in Fig. 6.

4. Results

Table 4 shows the results of the OCSVM experiment, in which the time features and sensor features for each user were combined to create a user pattern and perform classification. The mean EER is 13.05%. When the user pattern is created according to each feature type, we observe performance with an EER of 0% to 11.67% and a mean EER of 5.27%. In the feature type with the highest EER, all false reject rates (FRRs) are 0%. This implies that genuine users' experiment data are classified as being the genuine user.

Table 4. OCSVM classification results (unit: %, highlighted parts: feature vectors with a low EER for user)

	Time			Acc			Lacc			Gyr			Ugyr			All		
	EER	FRR	FAR	EER	FRR	FAR	EER	FRR	FAR	EER	FRR	FAR	EER	FRR	FAR	EER	FRR	FAR
User1	11.67	20	3.33	6.67	0	13.33	6.67	0	13.33	18.33	0	36.67	21.67	0	43.33	3.33	0	6.67
User2	0	0	0	31.67	0	63.33	41.67	0	83.33	33.33	0	66.67	15	0	30	30	0	60
User3	25	0	50	11.67	20	3.33	40	0	80	10	0	20	16.67	0	33.33	10	0	20
User4	21.67	0	43.33	25	0	50	33.33	20	46.67	25	0	50	11.67	0	23.33	21.67	0	43.33
User5	26.67	20	33.33	26.67	0	53.33	13.33	0	26.67	3.33	0	6.67	3.33	0	6.67	13.33	20	6.67
User6	13.45	20	6.9	0	0	0	41.03	20	62.07	13.79	0	27.59	0	0	0	0	0	0
Avg.	16.41	10	22.81	16.94	3.33	30.55	29.33	6.66	52.01	17.29	0	34.6	11.39	0	22.77	0	3.33	22.77

Good performance is obtained for more than one feature type per user. Even when users who were assumed to be attackers watched and imitated the genuine user's input pattern, the imitation was difficult. In the case of user 2 and user 6, an EER of 0% is obtained for Time, Acc, and Ugyr feature types, while in the case of user 5, an EER of 3.33% is obtained for the Gyr and Ugyr feature types. The FAR for user 2 is relatively high. In this case, an imposter is classified as a genuine user for feature types other than Time; hence, the EER is high. Impostors entered PIN of user 2 as the input pattern of user 2 but did not mimic the input pattern in time. However, it seems to have imitated the similarity in sensor.

For all users except user 1, training with single feature types provides the same or better performance compared to training with all feature types. depending on the PIN or the user's input style, there are feature types that better express a user's input pattern, which distinguishes him/her from other users.

4.1. Features for a Certain Period vs. Features for total Period

Sensor features can be extracted for certain periods to make the user input pattern slightly more detailed, or features can be extracted for total period of the PIN input to create the user input pattern. We observed the data to examine the difference between using a total period and a certain period.

In the case of certain periods, the average, minimum, and maximum of the sensor change values for the x, y, and z axes were extracted during the DU period of PIN for 54 features (the average, minimum, and maximum features per each axis on 6 DU periods). In the case of total periods, features were extracted based on the sensor changes during the period of PIN input for 9 features (the average,

minimum, and maximum features per each axis on a total period). The results obtained for each feature type are shown in the Table 5. In the case of sensor feature types based on the accelerometer, the mean EER for the classification results of the features extracted during the DU period is 2.5% better than that for the case of total period because more features are used in the DU period than in the total period. There is no significant difference between the features extracted during the DU period and those extracted during total period. In the case of the linear accelerometer, the EER for the case of the DU period is 9.8% lower than that for the case of the total period. The difference between the mean EERs for user LSH and CMR is more than 20%. In the case of the gyroscope, the performance for the total period is better than that for the DU period for a few users. However, on average, the performance for the DU period performance improves by 6.64% compared to that for the total period. Compared to other sensors, the uncalibrated gyroscope exhibits the largest performance difference, with results showing that the EER for the case of the DU period decreases by 16.41% compared to that for the case of total period. For all users, an overall improvement in performance in the DU period is observed compared to all periods. As a result, rather than extracting sensor features from all periods to create and compare user input patterns, extracting sensor features from the DU periods for each key and using more features to create input patterns provides better unique input patterns for users and better classification performance.

Table 5. Classification performance (EER) on DU time period vs. total time period

	Acc		Lacc		Gyr		Ugyr	
	DU	**Total**	**DU**	**Total**	**DU**	**Total**	**DU**	**Total**
User1	6.67	6.67	6.67	30	18.33	18.33	21.67	36.67
User2	31.67	28.33	41.67	40	33.33	35	15	30
User3	11.67	13.33	40	41.67	10	31.67	16.67	35
User4	25	26.67	33.33	45	25	50	11.67	41.67
User5	26.67	30	13.33	33.33	3.33	0	3.33	18.33
User6	0	0	41.03	44.83	13.79	8.62	0	5.17
Avg.	16.94	17.5	29.33	39.13	17.29	23.93	11.39	27.8

4.2. Training Data and Performance According to Change

In classification that uses machine learning, if there are more training data, more genuine user patterns can be analyzed and better performance can be obtained. We reduced the number of training data and performed experiments to check whether there was a difference compared to when we used 20 training data. Five training data were used, and the data used as training data were changed, as shown in Fig. 7. We tested the difference between using 20 training data and using 5 training data, when the data used as training data were changed.

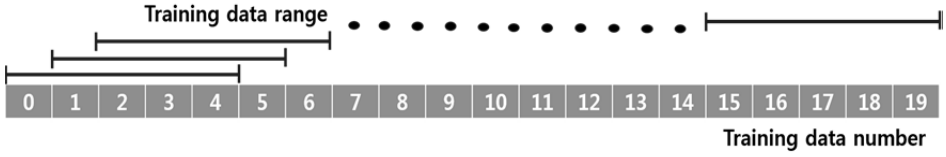

Fig. 7. Extracting time-based features from a time event.

Five data from the genuine user's 20 training data were used for training, and five data from the genuine user's test data and an impostor's test data were used to check performance. In Table 6, darker cell colors represent higher feature type discrimination and lighter cell colors represent lower feature type discrimination.

Table 6. OCSVM Classification Results (Unit: %, Highlighted Parts: Feature vectors with a low EER for user)

	Time	Acc	Lacc	Gyr	Ugyr
User1					
User2					
User3					
User4					
User5					
User6					

Table 6 shows that each user has feature types that had the exact same high discrimination as using all the training data for User 2, User 5, and User 6. For User 1, it was not exactly the same as using all the training data, but it can be seen that the Acc and Lacc feature types had the highest discrimination. In the case of User 3, the Gyr feature type had relatively high discrimination, and it can be seen that the Time, Acc, and Ugyr feature types also had a fair amount of discrimination. Just like in Table 2 where the EER for each of User 3's feature types was on a similar level with the exception of the Lacc feature type, here there is a similar distribution according to changes in the training data. User 4 showed a similar EER for each feature type as the total data, so on Table 3 there was no feature type which stood out.

Fig. 8 shows an EER change graph for each feature type when a sliding window was set to 5 according to the user. For User 1's Gyr feature type, there were almost no features with discrimination which can distinguish between other users, so it can be seen that it almost cannot distinguish between other users. It can be seen that when 20 items of training data were used, the classification performance deviation of the Acc and Lacc, which showed the best classification performance, depended on changes in the training data. For Acc, when the training data items 0 to 13 were used, a good classification performance was shown, but when training data items 14 and higher were used, the results showed that the classification went well and later did not go well. For Lacc, the classification performance gradually improved in training data items 0 to 11, but then the performance became poor again until training data item 18. However, finally after items 14 and 15, good performance was shown again. The time feature was a feature type in which performance gradually became good with almost no performance deviation. It can be seen that the EER performance gradually improved according to changes in the training data. However, when all 20 training data items were used, it showed a 5% performance drop compared to Acc and Lacc.

For User 2, the Gyr feature type was almost unable to distinguish different users just as with User 1, and Acc and Lacc showed poor classification performance. These feature types showed the worst performance in the tests using 20 items as well. The Ugyr feature type showed similar performance improvements as the Time feature type until the 13th data item, but from the 14th item on, its

performance dropped. The Time feature type showed comparatively steady classification performance with an EER under 10%, and it was the feature type that showed the best performance in the tests using 20 data items.

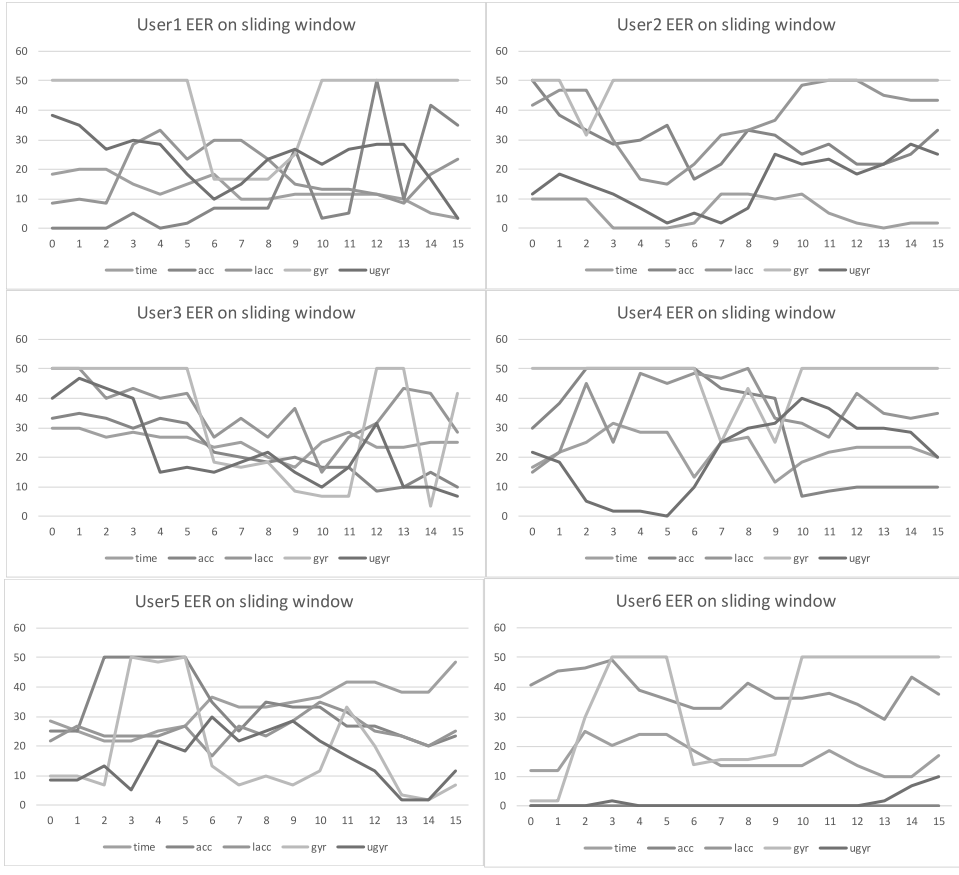

Fig. 8. Users EER on change of Training data. X axis is start index of training data range (window size: 5) and y axis is EER (%).

For User 3, steady EER performance improvements were shown according to the training data changes, with the exception of the Gyr and Lacc feature types. Acc showed particularly good performance, and it had the second best classification performance in the tests using 20 items. Gyr, which showed the best performance in the tests with 20 items, had the largest deviation in classification performance according to changes in the training data, and from training data item 6 to 16, it showed good performance, and in the interval from training data 14 to 19, it showed the best performance. However, overall it showed the worst performance. When all 20 training data items were used, at an EER of 10% it showed merely 1.67% preceding performance with the Acc feature type.

In the case of User 4, the Ugyr feature type showed the best EER performance in the interval up to training data item 10, and it showed EER performance improvements according to changes in the training data. However, in the later training data interval, it showed a performance drop. For the Time feature type, it showed a relatively even ER performance with no large deviation.

For User 5, the Ugyr feature type showed relatively good classification performance compared to other feature types, and in the training data interval after 13, it showed performance slowed to 0%. The Ugyr feature type also showed good performance in the interval after item 13, and these two feature types showed the best performance compared to when 20 training data items were used.

User 6 showed steady performance during changes to the training data, which was almost the same as the results when 20 training data items were used. The Acc and Ugyr feature types showed steady performance with an EER near 0% during training data changes, and this was almost the same as the performance when 20 training data items were used, which also showed an EER of 0%. The Time feature type also showed an even performance during changes to the training data with no EER deviation and an EER around 15%, and it showed the same performance in the results for 20 training data items.

In order to see the difference in EER performance according to changes in training data, a sliding window was used with the window value set at 5, and the results showed a performance which was broadly similar to when 20 training data items were used. For some users, different feature types showed a more even EER distribution.

4.3. Performance Changes According to Combinations of Feature Types

Using the aforementioned experiment results, we combined feature types with high discrimination for each user and performed experiments to see if it would produce improved classification performance as shown in Table 7. When a single feature type showed good performance (User 2), features were not combined, but when several features showed equally good performance, features were combined. In the case of User 6, Acc and Ugyr both obtained an EER of 0%, and it was confirmed that the two feature types show the same performance when combined. A maximum of 2 or 3 features were combined per user.

Table 7. Classification performance on feature type combination

User	Feature Type Combination	EER	FRR	FAR	vs. B.S.F.	vs. A.F
User1	Time + Acc + Lacc	0	0	0	+6.67	+3.33
User2	Time	0	0	0	-	-
User3	Acc + Gyr	8.3	0	16.67	+1.7	+1.7
User4	Time + Ugyr	10	0	20	+1.67	+11.67
User5	Gyr + Ugyr	3.33	0	6.67	0	+10
User6	Acc + Ugyr	0	0	0	0	0

B.S.F.=best single feature type, A.F.=all feature types.

Combinations of features showed better performance than a single feature combination. In the case of User 1, they improved performance over a single feature by 6.67%, and they improved performance over a combination of all features by 3.33%. For User 3, a combination of the Acc and Gyr feature types showed a 1.7% performance improvement over both a single feature combination and a combination of all features. For User 4 as well, it showed a performance improvement over a single feature combination and a combination of all features by 1.67% and 11.67% respectively. Through combinations of feature types, an FRR of 0% was achieved for all users. This means that after training with the genuine user's data and creating a model, the genuine user was always distinguished by the user's entered data. This

shows that it is possible to avoid classifying the genuine user as a non-genuine user and giving the user the inconvenience of reentering their PIN.

5. Conclusion

By comparing input patterns, keystroke dynamics can distinguish a genuine user from an impostor during PIN or password-based authentication, even when the PIN or password has been exposed. It can be used to compensate for the existing limitations in commonly used smartphone PIN authentication.

In the past, time-based and sensor-based features have been used in keystroke dynamics to distinguish users' input patterns. However, the PIN and input style of each user is different. It was confirmed through experiments that the performance of features customized for each user and that of all features are different and better performance can be obtained when feature type that better express a user's input pattern are used. To do this, we collected two additional sensors (linear acceleration and uncalibrated gyroscope), while accelerometer and gyroscope sensor data were collected in previous studies. We compared the case where the all feature types were used and the case where each feature type was used. The performance improvement was 5.27% on the average when using the feature type for each user rather than using the all feature types. Based on these experimental results, we confirmed the improvement of the maximum EER of 11.67% by combining the high discriminatory feature types. We used the familiar PIN to represent the input patter of each user, and reflected the difference in PINs for each user in real life.

In future research, we plan to collect experimental data from more users and study algorithms that can select feature types and individual features on each feature type that better represent user input patterns. In addition, we will compare to state-of-the art automated feature selection approach such as filter and wrapper approach.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0126-17-1007, Development of Universal Authentication Platform Technology with Context-Aware Multi-Factor Authentication and Digital Signature). This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00097, Development of Biometrics-based Key Infrastructure Technology for On-line Identification).

References

[1] R. S. Gaines, W. Lisowski, S. J. Press, and N. Shapiro, *Authentication by Keystroke Timing: Some Preliminary Results*. Santa Monica, CA: RAND Corporation, 1980.

[2] N. L. Clarke and S. M. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal Information Security*, vol. 6, no. 1, pp. 1-14, 2007.

[3] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, article no. 408280, 2013.

[4] P. H. Pisani and A. C. Lorena, "A systematic review on keystroke dynamics," *Journal of the Brazilian Computer Society*, vol. 19, no. 4, pp. 573-587, 2013.
[5] C. Giuffrida, K. Majdanik, M. Conti, and H. Bos, "I sensed it was you: authenticating mobile users with sensor-enhanced keystroke dynamics," in *Proceedings of the International Detection of Intrusions and Malware, and Vulnerability Assessment*, Egham, UK, 2014, pp. 92-111.
[6] G. Ho, "Tapdynamics: strengthening user authentication on mobile phones with keystroke dynamics," Stanford University, Stanford, CA, *Technical Report*, 2014.
[7] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile device," *Computers & Security*, vol. 59, pp. 210-235, 2016.
[8] R. Giot, B. Dorizzi, and C. Rosenberger, "A review on the public benchmark databases for static keystroke dynamics," *Computers & Security*. Vol. 55, pp. 46-61, 2015.
[9] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, "Continuous user authentication using multi-modal biometrics," *Computers & Security*, vol. 53, pp. 234-246, 2015.
[10] N. Zheng, K. Bai, H. Huang, and H. Wang, "You are how you touch: user verification on smartphones via tapping behaviors," in *Proceedings of the IEEE 22nd International Conference on Network Protocols*, Raleigh, NC, 2014, pp. 221-232.
[11] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "Recognizing your touch: towards strengthening mobile device authentication via touch dynamics integration," in *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, Brussels, Belgium, 2015, pp. 108-116.
[12] P. H. Pisani and A. C. Lorena, "Emphasizing typing signature in keystroke dynamics using immune algorithms," *Applied Soft Computing*, vol. 34, pp. 178-193, 2015.
[13] Android Developers-SensorManager [Online]. Available: https://developer.android.com/reference/android/hardware/SensorManager.html.
[14] S. Raschka, "About feature scaling and normalization-and the effect of standardization for machine learning algorithms," 2014 [Online]. Available: http://sebastianraschka.com/Articles/2014_about_feature_scaling.html.
[15] K. A. Heller, K. M. Svore, A. D. Keromytis, and S. J. Stolfo, "One class support vector machines for detecting anomalous windows registry accesses," in *Proceedings of the Workshop on Data Mining for Computer Security*, 2003, pp. 1-8.
[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
[17] Wikipedia, "Heuristic," [Online]. Available: https://en.wikipedia.org/wiki/Heuristic_(computer_science).

Sung-Hoon Lee https://orcid.org/0000-0002-7129-825X

He received B.S. Degree in Department of Computer Engineering from Chungju National University, Korea in 2011. Since August 2011, he is a Ph.D. candidate of Information Security Engineering from University of Science and Technology (UST) and currently the student researcher in the Electronics and Telecommunications Research Institute (ETRI), Korea. His research interests are behavior-based authentication, user authentication, and mobile security.

Jong-hyuk Roh https://orcid.org/0000-0002-6269-7566

He received the B.S., M.S., and Ph.D. degree in Computer Engineering from Inha University, Korea. He is currently the principal researcher in the Electronics and Telecommunications Research Institute (ETRI), Korea. His research interests are machine learning, pattern analysis, behavior-based authentication, and computer security.

SooHyung Kim https://orcid.org/0000-0002-3190-0293

He received the B.S. and M.S. degrees in computer science from Yonsei University, Seoul, Korea in 1996 and 1998, respectively, and the Ph.D. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2016. His research interests include identity management, mobile security, biometrics and digital payment.

Seung-Hun Jin https://orcid.org/0000-0002-9781-7863

He received his B.S. Degree in School of Computer Science and Engineering from Soongsil University, Seoul, Korea in 1993, the M.S. Degree in School of Computer Science and Engineering from Soongsil University, Seoul, Korea in 1995, and the Ph.D. degrees in Department of Computer Science and Engineering (Information Security) from Chungnam National University, Daejeon, Korea in 2004. He is currently an Assistant Vice President/Principal Researcher Information Security Research Division in ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea. His research interests include information security (PKI, authentication/authorization technique, privacy protection), mobile payment system, and computer/network security.