Effective Classification for Multi-modal Behavioral Authentication on Large-Scale Data

Shuji Yamaguchi[1], Hidehito Gomi[1], Ryosuke Kobayashi[2], Rie Shigetomi Yamaguchi[2]

[1] Yahoo Japan Corporation, Japan
[2] The University of Tokyo, Japan
{shyamagu, hgomi}@yahoo-corp.jp, kobayashi.ryousuke@sict.i.u-tokyo.ac.jp, yamaguchi.rie@i.u-tokyo.ac.jp

Abstract

We propose an effective classification algorithm for machine learning to achieve higher performance for multi-modal behavioral authentication systems. Our algorithm uses a multiclass classification scheme that has a smaller number of classes than the number of users stored in the dataset. We also propose metrics, the self-mix-classified rate, other-single-classified rate, and equal-classified rate, for use with the proposed algorithm to determine an optimal number of classes for behavioral authentication. We conducted experiments using a large-scale dataset of activity histories that are stored when 100,000 users use commercial smartphone-applications to analyze performance measures such as false rejection rate, false acceptance rate, and equal error rate obtained with our proposed algorithm. The results indicate our algorithm achieved higher performance than that for previous ones.

Keywords: Behavioral authentication, Personal data analysis, Smartphone application, Big data

1 Introduction

User authentication has been a challenging problem in computer systems. Various approaches for improving the security of authentication are needed to protect users from invalid access by malicious attackers. Many researchers have attempted to solve problems regarding passwords that are widespread but have vulnerabilities [1-3]. Many projects are striving to provide users with other authentication options such as biometrics

Numerous studies have been conducted on *behavioral authentication* to demonstrate the potential of big data for enhancing user authentication. Behavioral biometric data were analyzed to verify user identity [4]. This is significant because behavioral authentication does not require users to explicitly take a specific action [5].

Various biometric traits are being considered for behavioral authentication. For example, Fridman et al. [6] combined four behavioral modalities: text entered via a soft keyboard, application usage, websites visited, and a device's physical location. It is generally difficult for an application to ignore similar but different behavior patterns for the same person, which means the authentication accuracy is not particularly high compared with *physical* biometric authentication. With these points in mind, we considered multiple behavioral modalities should be combined to increase authentication accuracy.

Many studies on behavioral authentication have used machine learning in which user behavioral data are analyzed to extract features representing user identity. Multiclass classification is the appropriate algorithm used to classify user behavioral data to identify a specific user. However, the challenge is that, behavioral data generated by a large number of users can contain also the large corresponding number of classes. The ideal case is when the classification can classify the data into exactly the number of users. However, a real large-scale data can contain even more than one million users, and thus there is a big trade-off between accuracy and feasibility here. Thus, we ask the question: *How to determine a threshold for the number of classes that can balance the trade-off.*

There are only a few analytic algorithms to find the number of classes for the above challenge such as Silhouettes [7]. However, the number of classes obtained by these algorithms much smaller than the number of users. In this case, the data that are contained in each class are associated with many users, rarely classified to a single user. That is, a smaller number of clusters may increase the false acceptance of valid users in authentication, whereas a larger number may increase false rejection.

We proposed a classification algorithm and metrics for use with the algorithm for determining the number of classes and solving the above trade-off problem for multi-modal behavioral authentication systems [8]. We then evaluated the proposed algorithm from our experiments on behavioral authentication using the real large-scale data of a commercial smartphone application.

Since we confirmed the effectiveness of the

*Corresponding Author: Shuji Yamaguchi; E-mail: shyamagu@yahoo-corp.jp
DOI: 10.53106/160792642021092205019

proposed method in our previous paper [8], we in this paper conducted

experiments with larger scale of data to further confirm our proposed methods effectiveness. Specifically, we conducted experiments using datasets with 10 times the number of users more than that done in [8]. Note that we focus on proposing the above algorithm as a core mechanism for a behavioral authentication system that assumes a threat model. The overall design for such a system is out of the scope of this paper.

2 System and Threat Models

2.1 System Model

This section describes the multi-modal behavioral authentication system using classification that we developed for this study, which is shown in Figure 1.

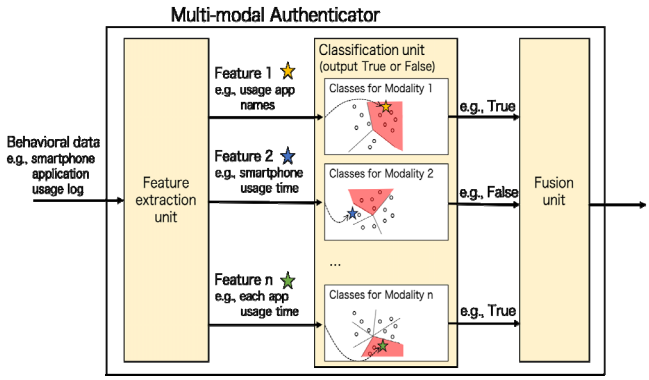


Figure 1. Overview of multi-modal behavioral authentication system using classification

The system consists of a feature-extraction unit, classification unit, and fusion unit. The feature-extraction unit analyzes user behavioral data to the system and extracts multiple features according to biometric modalities.

The classification unit maps the feature of each modal ity from the output of the feature-extraction unit into binary decision space (i.e., True or False). The classification is done by multiple sets of classes for each modality that are built and trained using the behavioral data of users. As a result, a binary decision of True is generated if a user is valid, and False otherwise. The fusion unit produces a fused decision by collecting the output of each modality from the classification unit.

At user authentication, the feature-extraction unit receives a user's behavioral data and extracts features. Next, the classification unit classifies these features into a specific class for each modality and produces the result of a binary decision about a user. Finally, the fusion unit produces a decision of the overall system.

For example, user behavioral logs of smartphone-application usage are used to extract features such as application names, smartphone-usage time, and usage time of each application (represented by stars in Figure 1), which correspond to personal habits representing user identity. In each modality of the classification unit, the output is generated depending on whether the extracted feature is classified into a user's own class (indicated by the red area in Figure 1).

Compared to physiological biometrics, such as fingerprints, behavioral biometrics is generally not as accurate or stable due to the inconsistency and mutability of human behavior. Our multi-modal behavioral authentication system aims to obtain high accuracy levels in authentication by fusing the verification results of multiple modalities, even if the verification of each modality is not highly accurate due to lack of its data.

The false acceptance rate (FAR) and false rejection rate (FRR) are used for evaluating the authentication accuracy of our multi-modal behavioral authentication system. By calculating the FAR and FRR for each threshold of the fusion unit and obtaining an equal error rate (EER), at which they are equal, the optimum threshold value can be determined.

2.2 Threat Model

The system consists of N users; $U_N = \{u_1, u_2, \ldots, u_N\}$. Each user u_i where $i \in [1, N]$ has their behavioral feature set $D_i = \{d_{i1}, d_{i2}, \ldots, d_{iM}\}$ where d_{im} indicates the m-th behavioral feature of u_i for $m \in [1, M]$.

We consider that the following types of adversaries (untrusted entities) attempt impersonation:

- *Internal Adversary*: An adversary $A = u_{i'} \in U_N$ where $i' \in [1, N]$ uses $D_{i'}$ to attempt to be authenticated as a legitimate user u_i where $i' = i$.
- *External Adversary*: An adversary $A \notin U_N$ uses a behavioral feature set $D_A \notin \{D_1, D_2, \ldots, D_N\}$ to attempt to be authenticated as a legitimate user u_i where $i \in [1, N]$.

We consider that the above adversaries conduct two types of impersonation attacks, namely targeted impersonation attack and non-targeted impersonation attack.

- Targeted impersonation attack: An adversary A attempts impersonating a legitimate user $u_i \in U_N$ where $i \in [1, N]$ using u_i's behavioral feature $d_{im} \in D_{iM}$ where $m \in [1, M]$. A may be able to infer the feature from u_i's personal information indicated by the relationship between u_i and A.
- *Non-targeted impersonation attack*: An adversary A makes a brute force attack using publicly available data without targeting particular users.

Later in Section 6.1 we will discuss these attacks.

There are assumptions in our threat model. (1) The n users do not collude with each other. This means that there is no user in the system who can share their behavioral data with any of the other users in the system and even people outside the system. (2) The

system is secured from the attacks that attempt to learn or make use of information from the system but does not affect system resources such as data breach, network eavesdropping, contextual data theft, and hardware side-channel attack.

3 Problem Statement

In our multi-modal behavioral authentication system, the number of classes needs to be set for appropriately classifying each feature per its corresponding modality in the classification unit.

A challenge here is: "How can we determine the number of classes in each modality for achieving higher authentication performance of the overall system?"

There are previous works that proposed a method for determining the optimal number of classes when classifying a given sample dataset. For example, Silhouette analysis provides such solution by visualizing the cohesion of data per class [7]. X-means classification determines the optimal number of classes by recursively calling K-means using BIC (Bayesian Information Criterion) as an performance index [9].

These methods aim to assign sample data to appropriate classes in a well-balanced manner, but heuristically use the number of classes in many cases. Thus, they do not well assess whether the resultant design of classes are best suited for behavioral authentication systems.

To discover an appropriate number of classes in each modality, some existing works use performance measures, FAR and FRR, which have traditionally been used for biometric authentication.

However, there is a problem in calculating a precise FAR in multiclass classification. It is generally estimated that the optimized number of classes should be smaller than that of subject users in datasets of the system, but obviously leads to incorrectly accepting unauthorized persons except the legitimate person. This has not been demonstrated by existing works yet as far as we know.

In essence, this should not be counted as a false accept.

We argue that performance measures other than FAR and FRR are necessary for validating classification.

We propose a classification algorithm for use with the algorithm for optimizing the number of classes ("m-classification").

4 Our Proposal

In this section, we first describe conditions on classification and then propose metrics and a classification algorithm using the metrics for m-classification. This method was proposed in our previous paper [8], but since it is the basis of our experiments, we will introduce it in this paper as well.

4.1 Definition of Well-balanced Classification Conditions

We begin by considering our multi-modal behavioral authentication system that, for a set of N users; U_N, has K classes when classifying features of U_N. With the m-classification algorithm, a smaller K results at a more accurate rate of true acceptance, but also makes it easier to accept others. On the other hand, a larger K value makes for more accurate rejection of others but also increases the FRR. Therefore, the K value needs to be well-balanced. We consider two types of the well-balanced classification conditions:

Condition-1: All of the user's data are in the same class.

Condition-2: Other users' data are not in the same class.

4.2 Metrics for Classifying Conditions

4.2.1 Metrics for Condition-1: Self-single-classified Rate and Self-mix-classified Rate

We derived metrics corresponding to Condition-1.

First, we count each user's own features in each class and defined a maximum count class as "my class".

Let the number of features for each user and each class be defined as Dc_{uk}:

$$D_{c_{uk}} = \begin{pmatrix} dc_{11} & \cdots & dc_{1j} & \cdots & dc_{1K} \\ \vdots & \ddots & & & \vdots \\ dc_{i1} & & dc_{ij} & & dc_{iK} \\ \vdots & & & \ddots & \vdots \\ dc_{N1} & \cdots & dc_{Nj} & \cdots & dc_{NK} \end{pmatrix} \quad \textbf{(1)}$$

Then, the number of features in "my class" can be expressed as

$$D_{m_u} = \begin{pmatrix} \max(Dc_{1k}) \\ \vdots \\ \max(Dc_{ik}) \\ \vdots \\ \max(Dc_{Nk}) \end{pmatrix} \quad \textbf{(2)}$$

Let the number of features each user has be defined as $D_u = (d_{all1}, d_{all2}, \ldots, d_{allN})$. Using Dm_u and D_u, we calculate the rate of data in "my class", and use the average value of all users as the self-single-classified rate (SSR):

$$SSR = \frac{1}{N} \sum_{u=1}^{N} \frac{Dm_u}{D_u}. \quad \textbf{(3)}$$

Similarly, the rate of not entering "my class" can be represented as the self-mix-classified rate (SMR) as:

$$SMR = 1 - SSR. \quad \textbf{(4)}$$

4.2.2 Metrics for Condition-2: Other-single-classified Rate and Other-mix-classified Rate

Assuming that the number of classified users for each class Nc_k = (nc_1, nc_2, ..., nc_K), the number of others' data in each class Nco_k can be represented as Nco_k = (nc_1 - 1, nc_2 - 1, ..., nc_K - 1).

If we define the average Nco_k of all classes as the metric that can represent others in the same class, i.e., the other-single-classified rate (OSR), then

$$OSR = \frac{1}{K}\sum_{k=1}^{K} Nco_k. \quad \textbf{(5)}$$

The other-mix-classified rate (OMR), which indicates that data of other users are not in the same class, is

$$OMR = 1 - OSR. \quad \textbf{(6)}$$

4.2.3 Classification Algorithm Using Proposed Metrics

From Section 4.2.1 and Section 4.2.2, we can see that the metrics corresponding to Condition-1 and Condition-2 have been obtained. To visualize the trade-off relationship between the two conditions, we decided to use SMR and OSR, both of which indicates a smaller value as a better classification, as shown in Figure 2. In addition to the FAR and FRR concepts, the point at which the SMR and OSR become equal is defined as the equal-classified rate (ECR). The X-axis in Figure 2 represents the number of classes, so the x value of ECR shows the optimal number of classes that balances the SMR and OSR.

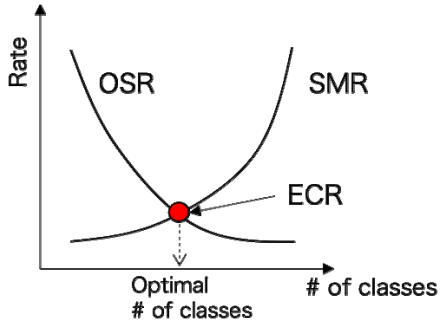


Figure 2. Overview of SMR, OSR, and ECR

To obtain an optimal number of classes, the classification process for each of the candidate classes and the SMR, OSR, and ECR calculations are repeated, and the number of classes with the smallest ECR is selected.

Figure 3 shows an example of SMR, OSR, and ECR calculations. Assuming that users A, B, and C are classified into Classes 2 to 5, the SMR and OSR for the number of classes are calculated as shown in the lower part of the figure. First, we count the Dm_u and D_u for all users then conduct a fitting calculation to Eq. 4 to obtain the SMR. The OSR is calculated by counting the Nco_k of each class and fitting the result into Eq. 5. In this example, the ECR appears when the number of classes is 4 (rounding up from 3.6). In this case, it can be said that the four classes are the most optimized.

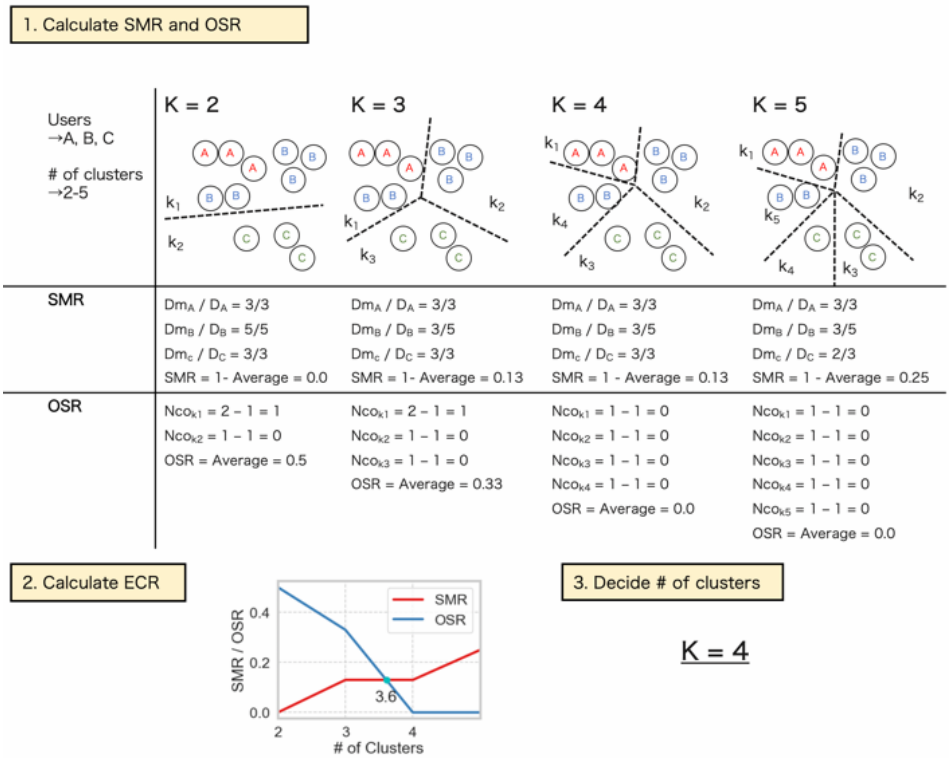


Figure 3. Example procedure for determining optimal class number using the SMR and OSR metrics

5 Experiments

5.1 Experimental Framework

We conducted two kinds of experiments.

In Experiment 1, we adapted the metrics proposed in Section 4 to actual data to determine the optimal number of classes. We used the three feature patterns shown in Section 5.3: (a) application only, (b) timestamp only, and (c) application and timestamp.

We conducted Experiment 2 to evaluate the performance of our multi-modal authentication system. We used the three Experiment 1 feature patterns as the multi-modal authentication modalities and prepared three patterns for m-classification optimization, (a) using our proposed optimization algorithm calculated in Experiment 1, (b) using the same number of classes as the number of users, and (c) optimizing the number of classes with the FAR, FRR, and EER.

We used the smartphone-application activity histories of 100,000 users who used our commercial application "Yahoo! Smartphone Security" during the period between February 1 and 29, 2020. Target users were randomly chosen from among users who used the application every day during the above period. Note that only Android users were surveyed because the application is only available for Android OS. Details for our experimental dataset are given in Section 5.2.

We conducted the experiments with the m-classification and validation window size set for 1 week and the test window size set for 1 day. During the 1 month target period, we conducted shifting windows both to m-classification and test data for each day to update the classes daily.

In Experiments 1 and 2, we adopted K-means as an m-classification algorithm using the Euclidean distance as the distance metric and the K-means++ initialization scheme [10] from the scikit-learn [11] v0.20.2 for the centroid initialization. All of the source codes for these experiments were developed using Python 3.5.2.

5.2 Datasets

5.2.1 Activity Histories of Smartphone-application

We focus on the users' activity histories that are stored in our commercial application Yahoo! Smartphone Security on their smartphones. It is a free security application running on Android devices provided by Yahoo Japan Corporation that protects a user's smartphone from any threats by detecting virus applications, malicious billing sites, and phishing sites by monitoring behavior on their smartphones. It records installed applications and their related activities on the smartphones. With the data, it detects some suspicious behavior that has a high security risk such as unauthorized communications and access to malicious sites. Our application is no longer available because it stopped providing service on March 23, 2020, but the collected activities data are used for our research purposes with the consent of the users. More details for data collection process are described in Section 5.2.2. Since the data are collected from the production-based application, to our knowledge, it is unprecedentedly more large-scale than that obtained in previous studies. Note again that our research group conducted this experiment using the above sets of data after gaining social understanding.

5.2.2 Data Collection Process

We collected the activity history of smartphone-application for our research purposes through appropriate procedures. The method for acquiring the data from our application for this study follows the guidelines of the privacy act in our country.

We have fully explained to users how to get and use data. Our application is not pre-installed when a user buys a smartphone, but is installed by the user on their own intention. Also, data will not be obtained unless the user consents to the permission to access the data on Android OS. Thus, the process of notifying the user of collecting activity history of smartphone-application and obtaining the user's consent is always performed. Our application is only targeted to our country and these procedures are provided so that users can understand how their data is handled. We also provide users with a privacy policy page in Japanese. (Privacy Policy: https://about.yahoo.co.jp/docs/info/terms/chapter1.html#cf2nd), as well as a commentary page with diagrams (Privacy Center: https://privacy.yahoo.co.jp).

5.2.3 Details and Statistics of Datasets

Table 1 lists three items in an activity history log: the timestamp at access (timestamp), hash value of user id (identifier), and package name of the application (application name).

Table 1. Activity history items

Item name	Description	Example
Timestamp	Timestamp when the application was started	1496740589
Identifier	Hash value of user id	-
Application name	Package name of smartphone-application	com.android.chrome

We collected 11.2 GB of smartphone logs of 100,000 users between February 1 and 29, 2020. The data included 170,239,146 activities and 21,056 package names of the applications (in Table 2).

Table 2. Our experimental data

Data	Value
Target application	Yahoo! JAPAN Smartphone Security
Data Size	11.2 GB
Target period	February 2020
Number of users	100,000
Number of activities	170,239,146
Number of application names	21,056

Figure 4 shows a histogram of the user-activity numbers. Most users performed fewer than 100,000 activities. As the number of activities increased, the number of people decreased, but some users performed more than 100,000 activities. Figure 5 shows a histogram of the numbers of applications operated by each user. The figure indicates that many users used fewer than 30 applications. Note that the Y-axis in Figure 4 and Figure 5 are expressed logarithmically. Figure 6 shows a histogram of timestamps and their frequencies. Users were most active at lunchtime and in the evening, whereas they were less active from midnight to the early morning hours.

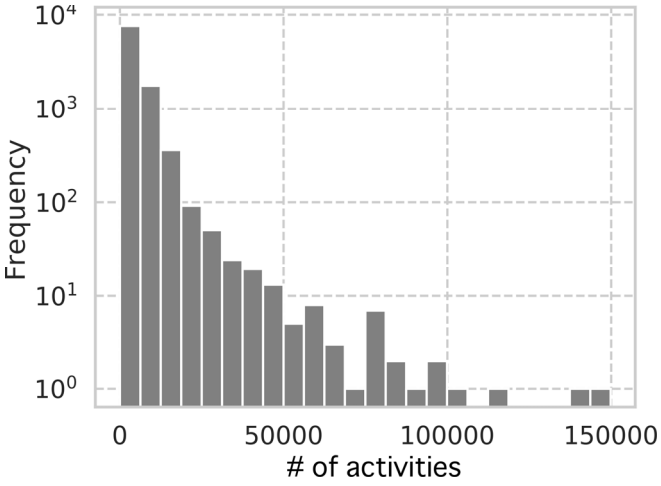


Figure 4. Histogram of activities

We consider the data in Figure 6 as clearly representing user lifestyles because there are numerous activities in the morning and during lunch breaks but only a few late at night and in the early morning. The data shown in Table 2, Figsure 4, Figure 5, and Figure 6 indicate that the numbers of activities and applications and that of the usage timestamps are sufficiently dispersed with respect to the number of users. Based on these observations, we determined that these data could be used as a modality of biometrics for verifying user identities.

5.3 Feature Extraction

We classified users according to the names of the applications that they had used so far and the time frames during which they had been using those applications. The purpose of this operation was to

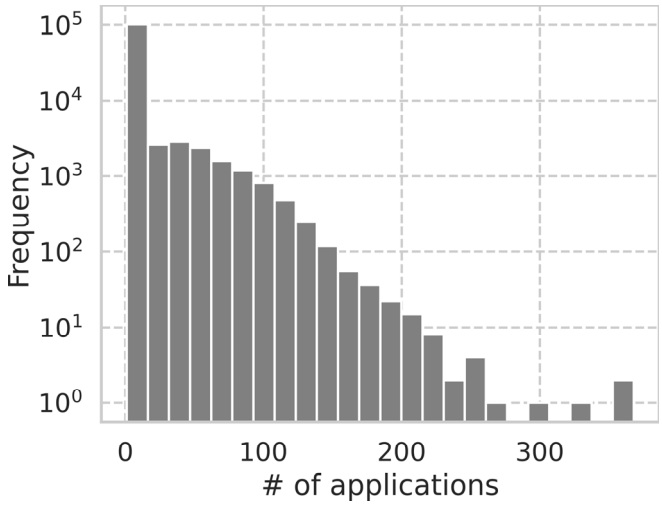


Figure 5. Histogram of applications

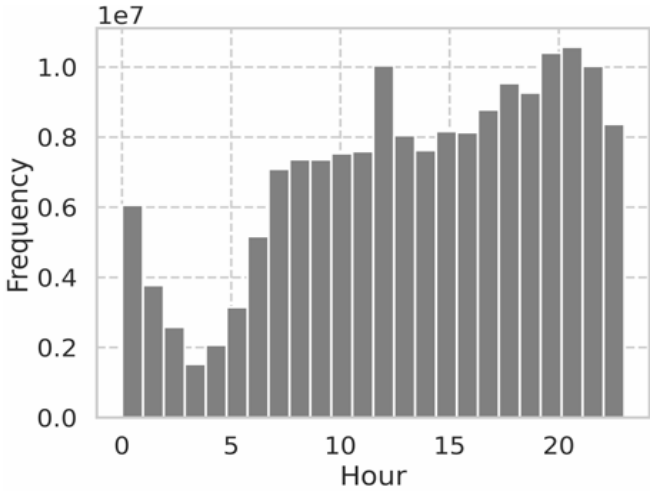


Figure 6. Histogram of timestamps

extract lifestyle features for authentication use by identifying whether they were continuously categorized into the same class. The application-usage histories include the package names of applications and the timestamps at which users ran the applications. These types of information were extracted from the history datasets and separately converted into each feature. An example of executing the feature- extraction procedure is shown in Figure 7.

In Step 1, we extracted a list of the user's past smartphone-application activities by specifying a certain period (e.g., between 11/1 and 11/2). To equalize the number of elements when carrying out m-classification, we narrowed down the applications before creating templates and test data from the time series data. Thus, we omitted "popular" applications which were ranked as the top 20 frequently-used ones in all logs because they were too common to employ for identifying users.

In Step 2, a feature was created for each user in the form of an array based on the application name and timestamps from the template and test data. We examined the following three patterns shown in Figure 7:

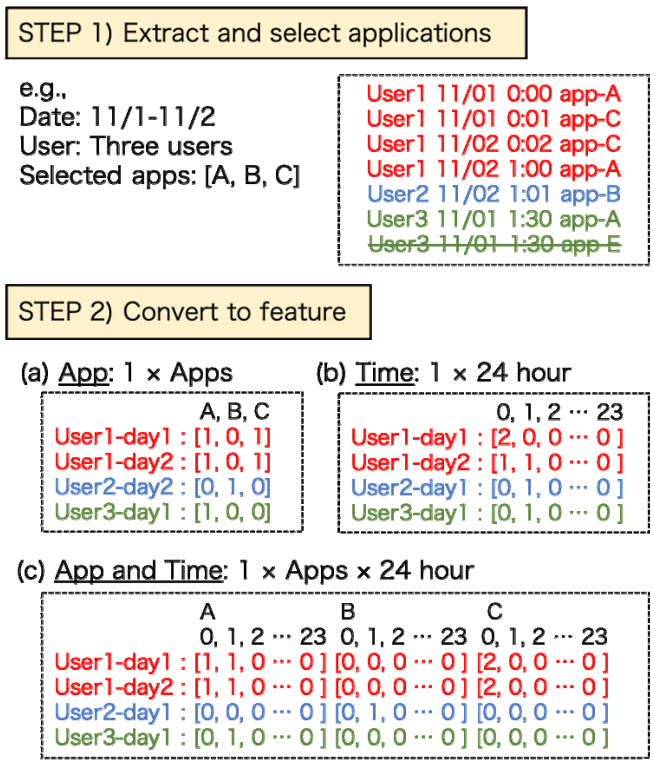


Figure 7. Example of executing feature-extraction procedure

(1) Application only pattern. In this pattern, an application is used as a feature. This feature indicates the application usage of target applications and is represented by a one-dimensional array of length equal to the number of target applications. For example, in the case of four target applications (A: using, B: not using, C: using, D: using), the feature is represented as [1, 0, 1, 1].

(2) Timestamp only pattern. Application timestamps are used as a feature in this pattern. For example, since users often use their smartphones at night, the launches of all applications are counted for each user every hour then summarized in the form of a 24 * 1 array, in which the values are arranged from 00:00 to 23:00.

(3) Application and timestamp pattern. Using a combination of applications and timestamps, habits such as checking a weather application every morning can be used as features. In this case, the number of launches for each application, each hour, and each user are counted and summarized in the form of a 24 * (number of applications) array, in which the values are arranged from 00:00 to 23:00.

5.4 Experiment 1: Calculating SMR, OSR, and ECR

We analyzed the SMR, OSR, and ECR for the aforementioned three patterns depending on the feature used, (a) application only, (b) timestamp only, or (c) application and timestamp, as mentioned in Section 5.3. At the time of feature-extraction, we selected the top 20 frequently-used applications from among all our experimental data. For (c), to reduce the dimensions, the number of applications was limited to the top 10 and the timestamp feature was set to 12 dimensions by dividing 24 hours into 2-hour increments. Each result is shown in Figure 8.

The X-axis in each figure represents the number of classes when K-means is applied and the Y-axis represents the SMR and OSR. The coordinates denoted by the intersection of the SMR and OSR lines represents the ECR. The results are shown from day-1 to day-5. The optimal number of classes on day-1 is 13 in (a), 7 in (b), and 8 in (c) after the X coordinate of the ECR was rounded off.

5.5 Experiment 2: Multi-modal Authentication

We developed a multi-modal authentication system using our proposed approach and evaluated its performance. We used three optimized classes for classification unit as shown in Section 5.4. We used majority vote rule for fused decision on the fusion unit as proposed in [12]. We compared our proposed algorithm with the following algorithms regarding the aforementioned patterns:

(a) Proposed: uses our proposed optimization algorithm for calculation in Experiment 1.

(b) Baseline: optimizes the number of classes with the FAR, FRR, and EER.

The traditional FAR, FRR, and EER metrics are used to evaluate the performance of our multi-modal authentication system. The procedure for calculating these metrics are shown as follows.

5.5.1 Caluculation of FRR and FAR

We calculated the performance metrics of the FRR and FAR to evaluate the performance and feasibility values from a security perspective because these measures have traditionally been used for evaluating biometric authentication systems.

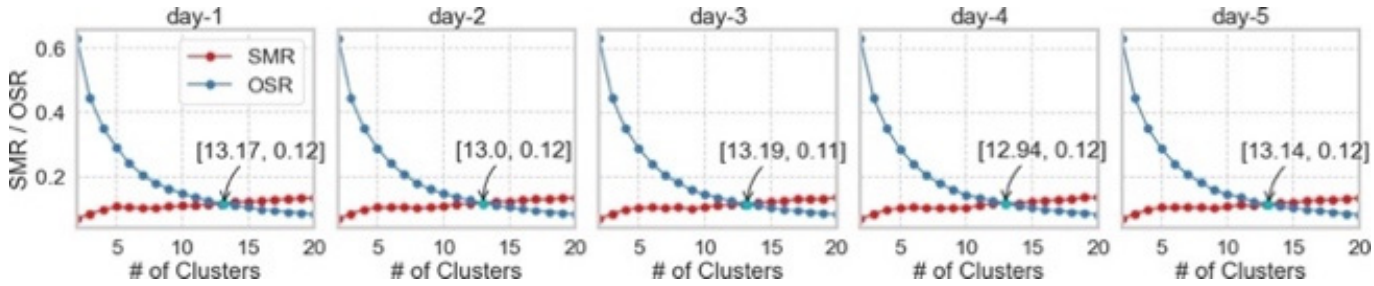
We first calculated the user's score function $c(k)$, after which we determined whether the user is regarded as valid. The kth feature was included in the K features obtained from all users during the specified time period. To accomplish this, we defined the threshold α as the rate of users that were accurately estimated to be legitimate.

Next, we defined $\delta^{(k)}$ as the indicator and specified that a user is accurately identified:

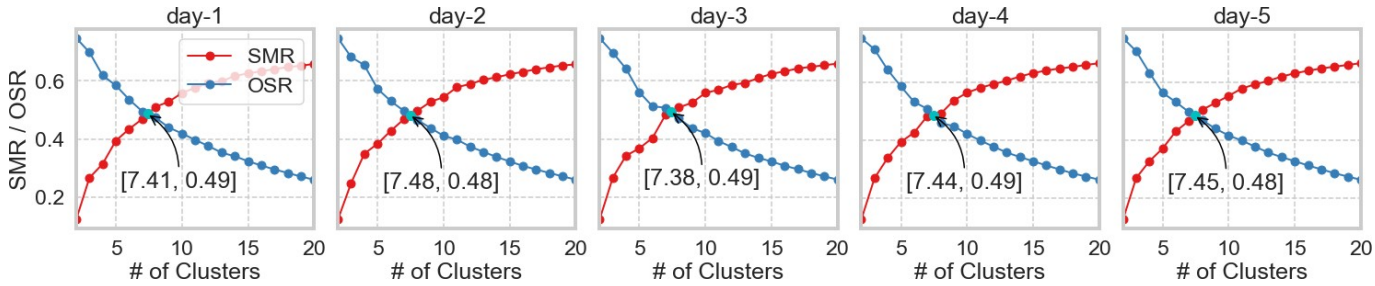
$$\delta^{(k)} = \begin{cases} 1 & \text{if } c(k) \geq \alpha \\ 0 & \text{otherwise.} \end{cases} \quad \textbf{(7)}$$

Using the above relationship, the FRR of the developed system FRR is represented as

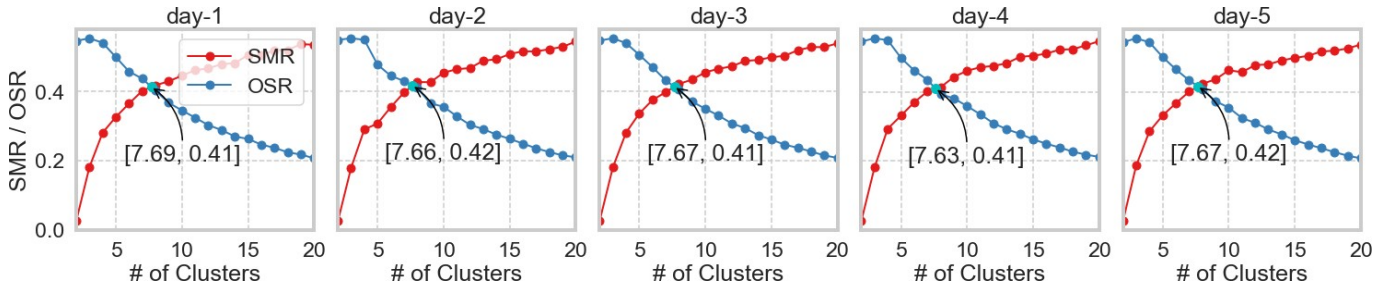
$$FRR = 1 - \sum_{k=1}^{K_r} \delta^{(k)} K_r, \quad \textbf{(8)}$$



(a) application only pattern



(b) Timestamp only pattern



(c) Application and timestamp pattern

Figure 8. Results of SMR, OSR, and ECR for three patterns

where K_r is the number of authentication attempts triggered by user activities.

Similarly, we obtained the FAR of the proposed algorithm, *FAR*, as follows:

$$FAR = \sum_{k=1}^{K_a} \delta^{(k)} K_a, \tag{9}$$

where K_a is the number of authentication attempts triggered by user activities.

5.5.2 Results

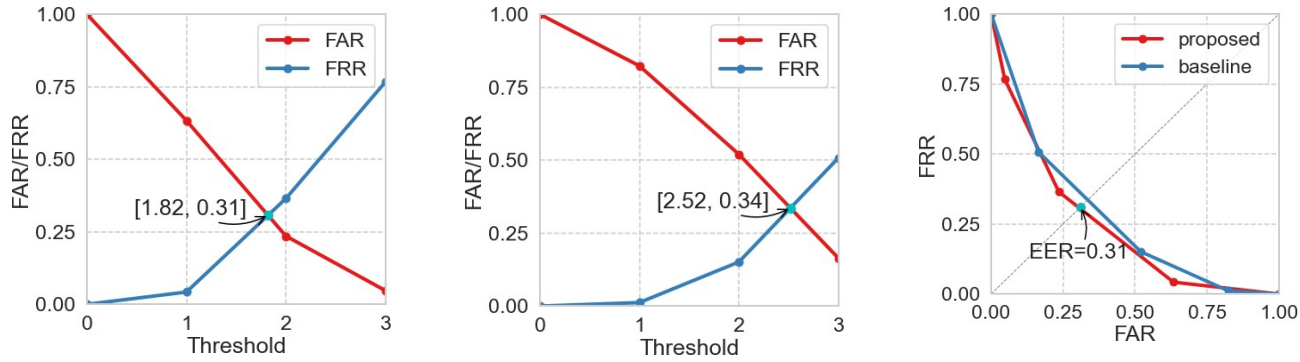
The results are shown in Figure 9. Figure 9(a), and Figure 9(b) respectively show the results of the proposed algorithm and Baseline. The X-axis in each figure in Figure 9(a) represents the threshold value of multi-modal authentication that indicates the number of authentication modalities. The Y-axis represents the FAR and FRR, and the coordinate of the indicated intersection of the two lines represents the EER. Note that it is widely known that a smaller Y-coordinate for an EER point indicates a better authentication system. Therefore, from a comparison of the results shown in Figure 9(a), and Figure 9(b), our proposed algorithm (shown in Figure 9(a)) obtained the best results with an EER of 0.31. Hence, the value of the X-coordinates at the EER point should be adopted as a threshold for judging whether the user is regarded as valid. In Figure 9(a), this is rounded up to 2. We believe that this value is correct and matches our intuition.

The results of the three patterns were also plotted on the receiver operating characteristic (ROC) curve shown in Figure 9(c). As the curve shows better results as it gets closer to the origin, we also confirmed that our proposed algorithm is superior to the others.

6 Discussion and Future Work

6.1 Security Consideration

We described impersonation attacks in Section 2.2. Here, we discuss how users can be protected against these attacks by our proposed approach.



(a) Proposed optimization algorithm (b) Baseline (c) ROC curve

Figure 9. Results of FAR, FRR, and EER

6.1.1 Non-targeted Impersonation Attack

An adversary can obtain the information about smartphone-application usage from some statistics that are publicly available (e.g., Google Play Store). However, we consider that the probability of success of this attack is very low because the total number of applications is extremely large, over 3.6 million applications are released at Google Play Store until 2017 [13], so the number of combinations of applications is enormous, and because the number of combinations is sufficiently larger than that for passwords and PINs which have short characters.

6.1.2 Targeted Impersonation Attack

An adversary may be able to infer a set of smartphone applications that a user uses from their nationality because the popularity of applications depends on each country. The adversary may also be able to narrow down the user's smartphone-application usage from their friend list and activities posted at social media if the adversary can browse them. Behavior authentication using user activities including our smartphone-based algorithms has a potential problem against this type of attack. For this reason, we adopt a multi-modal approach for behavior authentication. Even if some behavioral features of an attack match with those managed in one modality, the attack cannot succeed the overall system invalidly because each modality uses different behavioral features. Thus, we consider that our proposed approach is resilient to such attack.

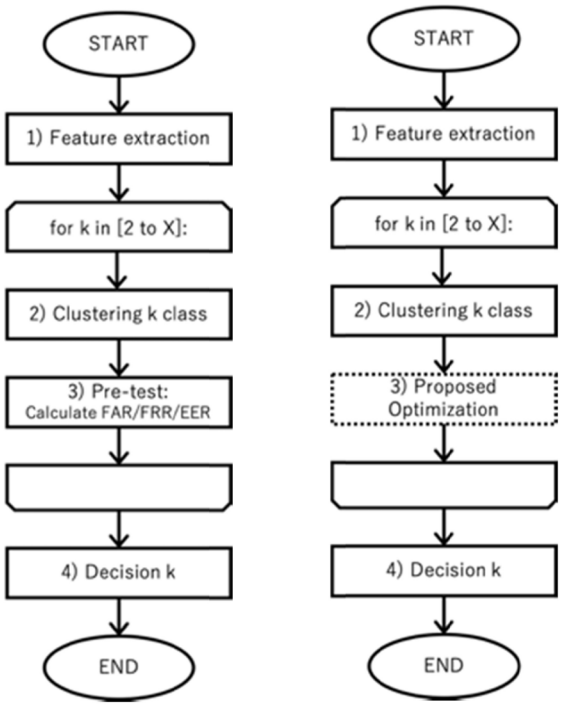
6.2 Feature Engineering

We used three features patterns in our experiments, (a) application only, (b) timestamp only, and (c) application and timestamp, as described in Section 5.3. From the results in Figure 8(a), we see that both SMR and OSR have small values and that they intersect at about 0.1. Therefore, we confirmed that with the proposed algorithm, we obtained well-balanced classes in which user activities were well classified in an identifiable fashion. In contrast, Figure 8(b) and Figure 8(c), which contain time elements, have large values. This is because the timezone feature consists of limited variations (from 00:00 to 23:00), so different users will often show similarities. In Figure 8(c), the lines are not smooth, which we believe is because the curse of dimensionality occurs due to the relatively high dimension of the feature vectors. Although this paper focused on optimizing the number of classes, we expect that our proposed method improve in performance from these data if we improve the method used for extracting features, such as by reducing their dimensions. This is a topic for future work.

6.3 Low Computational Complexity

Behavioral authentication using data based on machine learning is extremely useful because it can achieve highly accurate authentication compared to simple template matching. However, for some consumer services, the number of users can easily exceed 100,000,000, resulting in a vast amount of data and requiring massive amounts of calculation. Since the situations of users and the services provided can also change daily, it is not sufficient to create the template just once, so daily updates are essential. Therefore, we examined the computational complexity required to optimize the number of classes.

We began by comparing our proposed method with the computational complexity for the Baseline pattern. Optimization was done using the steps introduced in Figure 10. For the FAR calculated with Baseline, the number of false acceptances calculated by the number of other increases as $O(n^2)$, so it is often calculated by sampling users. In this experiment, we compared two patterns: calculating the FAR from all combinations and calculating with respect to two other sampled users for each user. To simplify the comparison, the number of calculation steps was used as the measure of calculation amount. We assumed that the number of users was 100,000 and the optimal number of classes was between 2 and 20.



(a) Typical (b) Proposed

Figure 10. Scheme for determining the number of clusters (k: number of clusters)

Figure 11 shows that our proposed method overwhelmingly reduced the number of required calculation steps by omitting the pre-testing phase compared with the other methods. Accordingly, we are convinced that extraordinary benefits can be obtained by assuming its realistic deployment.

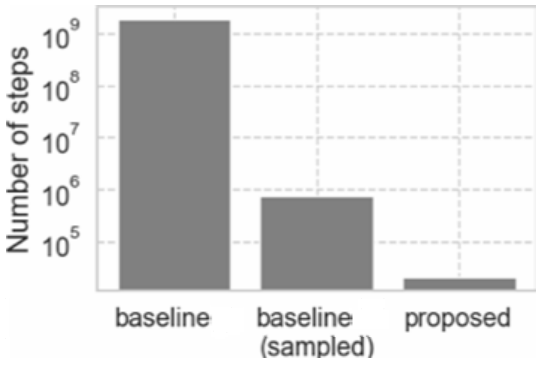


Figure 11. Results of number of calculation steps (Y-axis is log scale)

Existing machine learning schemes typically have four pre-testing phases as shown in Figure 10(a): (1) extracting features from original data; (2) developing a template for a specific number of classes; (3) pre-testing the template to obtain a FAR and a FRR, and then repeating the above phases 2 and 3 for each of the candidate numbers of classes; and 4) adopting the number of classes that results in the best EER. Since these phases generally require increasing computational complexity as the number of users grown, the scheme's efficiency is an important issue when using real behavioral data for authentication.

6.4 Optimization of Entire Multi-modal Authentication

The proposed method is based on the idea that multi-modal authentication can be improved by optimizing each modality. The experimental results indicate that this way of thinking can provide better results than with previous methods. Although the proposed method is better than those we compared in this study, it could be better because we did not consider correlations between modalities. Therefore, optimizing the overall system as well as each modality may enable us to obtain better results. We consider this to be the next step towards improving accuracy.

6.5 Effectiveness with Multiple-datasets

In this paper, we focused on smartphone usage logs, so we used only one dataset. We extracted multiple features from one dataset and confirmed that our proposed method can be applied to each feature.

To further confirm the effectiveness of our proposed method, it is better for us to experiment with other datasets and obtain more reliable results. We consider this to be a next step in our proposal.

7 Related Work

7.1 Physiological Authentication

Physiological biometric authentication mechanisms, which are based on human physical characteristics that are assumed to be relatively unchanging such as fingerprints [14], face [15], iris [16]. Since such human body features are unique, these physiological biometrics can provide higher overall authentication accuracy than behavioral biometrics. However, such physiological authentication usually requires additional hardware. Previous studies on physiological biometrics used the FAR and FRR as metrics for evaluating their user-verification methods whereas we focused on proposing an effective classification method using specific metrics for behavioral authentication.

7.2 Behavioral Authentication

In previous behavioral authentication studies, behavioral traits including user activities and movements were continuously collected and used for user authentication (*continuous authentication*) [4, 17-23]. The authentication mechanisms in these studies included voice [24], walking gait [25], key strokes [26], touchscreen use [27]. Our work in this paper does not specialize such a continous approach for authentication and rather specifically focused on developing the algorithm of gaining higher performance for multi-modal behavioral authentication systems instead of proposing the overall system.

In one such study, Neal et al. [18] proposed an authentication mechanism using behavioral biometric traits such as application, Bluetooth, Wi-Fi, and mobile device usage. They obtained user identification rates averaging 80, 77, 93, and 85%, respectively.

7.3 Authentication Using Smartphone-applications

Other studies proposed behavioral authentication mechanisms for smartphone users using a data mining method [21-22, 28-29]. However, these mechanisms have the following two problems. (1) They rely on user data that must be carefully handled from a privacy viewpoint (e.g., location data), and (2) users have their own preferences regarding the types of behavioral data they are willing to allow use for authentication, even though service providers often require them to use specific mechanisms and data. Therefore, we should ideally examine possibilities that cover a broad range of potential behavioral data.

Tang et al. [21] collected application histories and GPS data and used them to carry out experiments that they claimed clearly reflected user habits. However, their experiment involved only ten participants and the authentication accuracy was as low as 0.7. Kobayashi et al. [28] used Wi-Fi information captured by smartphone sensors to conduct an authentication experiment involving 100 participants and achieved an accuracy was 0.932. This result is better than Tang et al.'s, but lower than can be achieved with biometrics such as fingerprints.

7.4 Contextual Device Information

A number of studies examined user-authentication mechanisms using information gathered from wearable devices. For example, Susuki et al. [30] proposed a cost-effective user model for a behavioral authentication system that obtains activity information from a wearable device. Using the daily and hourly features of human activity information, they conducted an experiment involving 70 participants. Their resulting model achieved an accuracy rate of 89.28%.

7.5 Authentication Using User Activities

Authentication methods using content posted on social media rather than information obtained from sensors have also been investigated. For example, Sultana et al. [31] argued that social interactions can be used to identify individuals' unique behavioral patterns. Specifically, they analyzed online social content provided by 241 Twitter users and concluded that social-behavioral biometric features have properties such as uniqueness, stability, and recognition accuracy for a set of frequent and non-frequent online social networking users.

Dandapat et al. [19] developed a dynamic authentication system that mines a user's daily activities to extract passwords. Although all these studies are related to ours in that they attempted to extract user characteristics and features from their behavioral traits, they were all small-scale studies.

Previous studies on behavioral authentication also adopted the FRR and FAR for evaluating their verification methods in the same way as physiological authentication. However, we proposed metrics for classifying in a machine learning approach for behavioral authentication.

7.6 Machine Learning for Authentication

Cluster-based machine learning has been shown to be effective for achieving high authentication accuracy. Specifically, there are two approaches to machine learning clustering: binary and multiclass classification. With binary classification approaches, such as those that use support vector machines (SVMs) [6], Binary classification approaches include support vector machines (SVMs) [6], which builds an authentication model for a specific user that assigns new example data to one category or another to identify the user. With multiclass classification methods, clusters for a number of users are developed to classify user behavioral data into one specific user identity. There are existing studies applying multiclass classification methods for physiological authentication though few studies for behavioral authentication. Other multiclass classification approaches include unsupervised learning algorithms such as K-means [32], Gaussian mixture model [33], and the auto associative neural network [34].

8 Conclusion

We proposed an effective classification algorithm for machine learning and metrics, SSR, SMR, OSR, OMR, and ECR, for use with the proposed algorithm for validating classification of behavioral datasets. We evaluated the proposed algorithm from our experiments on behavioral authentication using almost 170,000,000 activities from 100,000 users from our production Android applications along with user consent after obtaining appropriate ethical approval and analyzed performance measures such as the FAR, FRR, and EER. The results indicate that our proposed algorithm achieved higher performance than that for existing algorithms. To evaluate the feasibility of the proposed algorithm, we used large-scale activity logs that are stored when users use real smartphone-applications. We collected almost 170,000,000 activities from 100,000 users as experimental data from our production Android application.

We analyzed various performance measures such as authentication accuracy, FRR, FAR, and EER in relation to our multi-modal behavioral authentication and compared the results with previous algorithms.

The results indicate that the best pattern of our proposed algorithm had an EER of 0.31 and reduced computational complexity, confirming our algorithm's effectiveness.

These are significant results in the area of multi-class behavioral authentication. To obtain further performance improvements, our observations indicate further research directions as follows: (1) the need to update the feature-extraction algorithm by considering dimension reduction for more accurate classification, and (2) for multi-class authentication, the need to consider not only optimization of each modality but also the correlations for each modality to optimize the entire multi-modal authentication system.

Acknowledgments

The authors are grateful to Dr. Tran Thao and Dr. Mhd Irvan, The University of Tokyo, for their technical contribution and warm encouragement. We also wish to acknowledge Professor Toshiyuki Nakata, The University of Tokyo, for his kind technical advice. We also gratefully acknowledge the work of Yahoo! JAPAN Smartphone Security team members about the creation of experimental data.

References

[1] J. Bonneau, C. Herley, P. C. Van Oorschot, F. Stajano, The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes, *IEEE Symposium on Security and Privacy*, Oakland, CA, 2012, pp. 553-567.

[2] J. Yan, A. Blackwell, R. Anderson, A. Grant, Password Memorability and Security: Empirical Results, *IEEE Security & Privacy*, Vol. 2, No. 5, pp. 25-31, September-October, 2004.

[3] K. C. Wang, M. K. Reiter, How to End Password Reuse on the Web, *Network and Distributed Systems Security*, San Diego, CA, 2019, pp. 1-15.

[4] W. Meng, D. S. Wong, S. Furnell, J. Zhou, Surveying the Development of Biometric User Authentication on Mobile Phones, *IEEE Communications Surveys and Tutorials*, Vol. 17, No. 3, pp. 1268-1293, Thirdquarter, 2015.

[5] E. Shi, Y. Niu, M. Jakobsson, R. Chow, Implicit Authentication through Learning User Behavior, *International Conference on Information Security*, Deerfield Beach, FL, 2010, pp. 99-113.

[6] L. Fridman, S. Weber, R. Greenstadt, M. Kam, Active Authentication on Mobile Devices via Stylometry, Application Usage, Web Browsing, and GPS Location, *IEEE Systems Journal*, Vol. 11, No. 2, pp. 513-521, June, 2017.

[7] L. Kaufman, P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 2009.

[8] S. Yamaguchi, H. Gomi, R. Kobayashi, T. P. Thao, M. Irvan, R. S. Yamaguchi, Effective Classification for Multi-modal Behavioral Authentication on Large-Scale Data, *Asia Joint Conference on Information Security*, Taipei, Taiwan, 2020, pp. 101-109.

[9] D. Pelleg, A. W. Moore, X-means: Extending K-means with Efficient Estimation of the Number of Clusters, *International Conference on Machine Learning*, Vol. 1, Stanford, CA, 2000, pp. 727-734.

[10] D. Arthur, S. Vassilvitskii, K-means++: The Advantages of Careful Seeding, *Symposium on Discrete Algorithms*, New Orleans, Louisiana, 2007, pp. 1027-1035.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, 2011.

[12] J. Kittler, M. Hatef, R. Duin, J. Matas, On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 226-239, March, 1998.

[13] Appfigures, *iOS Developers Ship 29% Fewer Apps In 2017, The First Ever Decline – And More Trends To Watch*, https://blog.appfigures.com/ios-developers-ship-less-apps-for-first-time/.

[14] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, A. K. Jain, Performance Evaluation of Fingerprint Verification Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 1, pp. 3-18, January, 2006.

[15] A. Ouamane, M. Belahcene, A. Benakcha, S. Bourennane, A. T-Ahmed, Robust Multimodal 2D and 3D Face Authentication Using Local Feature Fusion, *Signal, Image and Video Processing*, Vol. 10, No. 1, pp. 129-137, January, 2016.

[16] P. Loya, A. R. Pais, SIRIS - Secure IRIS Authentication System, *International Conference on Security of Information and Networks*, Jaipur, Rajasthan, India, 2012, pp. 148-152.

[17] K. O. Bailey, J. S. Okolica, G. L. Peterson, User Identification and Authentication Using Multi-modal Behavioral Biometrics, *Computers & Security*, Vol. 43, pp. 77-89, June, 2014.

[18] T. J. Neal, D. L. Woodard, A. D. Striegel, Mobile Device Application, Bluetooth, and Wi-Fi Usage Data as Behavioral Biometric Traits, *International Conference on Biometrics Theory, Applications and Systems*, Arlington, VA, 2015, pp. 1-6.

[19] S. K. Dandapat, S. Pradhan, B. Mitra, R. R. Choudhury, N. Ganguly, ActivPass: Your Daily Activity is Your Password, *Conference on Human Factors in Computing Systems*, Vol. 1, Seoul, Korea, 2015, pp. 2325-2334.

[20] H. Gomi, S. Yamaguchi, K. Tsubouchi, N. Sasaya, Continuous Authentication System Using Online Activities, *International Conference on Trust, Security and Privacy in Computing and Communications*, New York, NY, 2018, pp. 522-532.

[21] Y. Tang, N. Hidenori, Y. Urano, User Authentication on Smart Phones Using a Data Mining Method, *International Conference on Information Society*, London, UK, 2010, pp.

173-178.

[22] A. Alzubaidi, J. Kalita, Authentication of Smartphone Users Using Behavioral Biometrics, *IEEE Communications Surveys and Tutorials*, Vol. 18, No. 3, pp. 1998-2026, Thirdquarter, 2016.

[23] T. Neal, D. Woodard, Surveying Biometric Authentication for Mobile Device Security, *Journal of Pattern Recognition Research*, Vol. 11, No. 1, pp. 74-110, 2016.

[24] G. Peng, G. Zhou, D. T. Nguyen, X. Qi, Q. Yang, S. Wang, Continuous Authentication with Touch Behavioral Biometrics and Voice on Wearable Glasses, *IEEE Transactions on Human-Machine Systems*, Vol. 47, No. 3, pp. 404-416, June, 2017.

[25] W. Xu, Y. Shen, Y. Zhang, N. Bergmann, W. Hu, Gait-Watch: A Context-aware Authentication System for Smart Watch Based on Gait Recognition, *International Conference on Internet-of-Things Design and Implementation*, Pittsburgh, PA, 2017, pp. 59-70.

[26] D. Buschek, A. De Luca, F. Alt, Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices, *Conference on Human Factors in Computing Systems*, Seoul, Korea, 2015, pp. 1393-1402.

[27] S. Budulan, E. Burceanu, T. Rebedea, C. Chiru, Continuous User Authentication Using Machine Learning on Touch Dynamics, *Neural Information Processing*, Istanbul, Turkey, 2015, pp. 591-598.

[28] R. Kobayashi, R. S. Yamaguchi, Behavioral Authentication Method Utilizing Wi-Fi History Information Captured by IoT Device, *International Workshop on Secure Internet of Things*, Oslo, Norway, 2017, pp. 20-29.

[29] Y. Ashibani, Q. H. Mahmoud, A Behavior Profiling Model for User Authentication in IoT Networks Based on App Usage Patterns, *Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp. 2841-2846.

[30] H. Susuki, R. S. Yamaguchi, Cost-Effective Modeling for Authentication and Its Application to Activity Tracker, *International Workshop on Information Security Applications*, Jeju Island, Korea, 2015, pp. 373-385.

[31] M. Sultana, P. P. Paul, M. L. Gavrilova, User Recognition from Social Behavior in Computer-Mediated Social Context, *IEEE Transactions on Human-Machine Systems*, Vol. 47, No. 3, pp. 356-367, June, 2017.

[32] P. Kartik, S. M. Prasanna, R. V. Prasad, Multimodal Biometric Person Authentication System Using Speech and Signature Features, *IEEE Region 10 Conference*, Hyderabad, India, 2008, pp. 1-6.

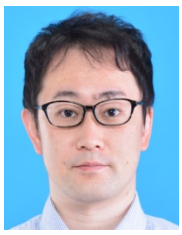
[33] J. Wang, Y. Li, X. Ao, C. Wang, J. Zhou, Multi-modal Biometric Authentication Fusing Iris and Palmprint Based on GMM, *Workshop on Statistical Signal Processing*, Cardiff, UK, 2009, pp. 349-352.

[34] A. K. Sao, B. Yegnanarayana, Face Verification Using Template Matching, *Transactions on Information Forensics and Security*, Vol. 2, No. 3, pp. 636-641, September, 2007.

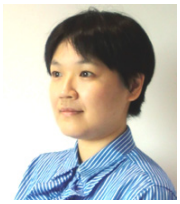
Biographies

Shuji Yamaguchi Joined Yahoo! JAPAN Corporation in 2009. He engaged in development and operation for Yahoo! JAPAN ID system including registration, authentication and federation. From 2015, he works as an Engineer of Yahoo! JAPAN Research.

Hidehito Gomi received the Ph.D degree in informatics from Kyoto University, Japan in 2012. He is currently Senior Chief Researcher of Yahoo! JAPAN Research at Yahoo Japan Corporation. His research areas primarily focus on security and privacy for Internet systems.



Ryosuke Kobayashi is currently a collaborative researcher, Social ICT Research Center, Graduate School of Information Science and Technology, The University of Tokyo. His research interests include behabioral authentication.



Rie Shigetomi Yamaguchi received PhD degree in Information Science and Technology from the University of Tokyo in 2006. She is currently a Project Associate Professor, Social ICT Research Center, Graduate School of Information Science and Technology in The University of Tokyo. Her major is privacy protection and information security.