GPS-Based Behavioral Authentication Utilizing Distance Coherence

Tran Phuong Thao(✉) and Rie Shigetomi Yamaguchi

Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan
tpthao@yamagula.ic.i.u-tokyo.ac.jp, yamaguchi.rie@i.u-tokyo.ac.jp

Abstract. Current user authentication systems are based on PIN code, password, or biometrics traits, which can have some limitations in usage and security. Lifestyle authentication has become a new research approach in which the promising idea is to use the location history since it is relatively unique. Even when people live in the same area or have occasional travel, it does not vary from day to day. For Global Positioning System (GPS) data, previous work used the longitude, latitude, and timestamp as the classification features. In this paper, we investigate a new approach utilizing distance coherence, which can be extracted from the GPS itself without the need to require other information. We applied three ensemble classifications, including RandomForest, ExtraTrees, and Bagging algorithms. The experimental result showed that our approach could achieve 99.42%, 99.12%, and 99.25% of accuracy, respectively.

Keywords: Smartphone location-based authentication · Lifestyle authentication · Global Positioning System (GPS) · Biometrics authentication

1 Introduction

"Society 5.0" [4] has become a well-known buzzword which was introduced by the Japanese government in 2011[1]. Society 5.0 focuses on two critical keywords, **human-centered** and **smart** society with the support of Artificial Intelligent (AI), Internet of Things (IoT), big data, and cutting-edge technologies.

Let's consider an example of the electronic payment system. In 1871, Western Union debuted the electronic fund transfer (EFT), allowing people to send money to pay for goods and services without necessarily having to be physically present at the point-of-sale. In 1946, John Biggins invented the first bank-issued credit card to replace paper money (the concept of using a card for purchases and the term credit card was described in 1887 by Edward Bellamy). In 2011, Google launched a mobile wallet project to replace physical cash and credit cards.

[1] Society 5.0 follows Society 1.0 (the hunting society), Society 2.0 (agricultural society), Society 3.0 (industrial society), and Society 4.0 (information society).

J. Garcia-Alfaro et al. (Eds.): DPM 2020/CBT 2020, LNCS 12484, pp. 195–215, 2020.
https://doi.org/10.1007/978-3-030-66172-4_12

Nowadays, the cashless payment system has become a new trend. Many digital wallet services appeared, such as Apple Pay (from 2014), Google Pay (from 2015 as Android Pay and 2018 as Google Pay), Rakuten Pay (from 2016), etc. The biggest challenge is how to authenticate the users. The current approach relies on the mobile phones' authentication using PIN code, password, biometrics (i.e., fingerprinting, iris, face, etc.), or multi-factor method, which combines more than one form of authentication from independent categories of credentials.

Attacks and Vulnerabilities in Current Smartphone Authentication. Many sophisticated attacks in smartphone authentication have appeared. First, *PIN code/password-guessing attack* [15,16] tries to recover the password plaintext from its hashed form using a brute force search, which systematically checks every combination of letters, symbols, numbers and dictionary attack which uses a dictionary of common words. Second, *biometric spoofing* tries to generate synthetic or fake biometric traits of legal users to fool the capture sensors including *facial spoofing* which utilizes printed facial photographs and digital video [21] or a 3D mask [22], *fingerprinting spoofing* [23] which utilizes artificial replicas with different materials such as gelatin, latex, play-doh or silicone, and *iris spoofing* [17] which utilizes an image forging natural iridal texture characteristics [18] or even cosmetic contact lenses [19,20], and the combination of all these three spoofing types [24]. Third, *smudge attack* tries to guess the graphical password pattern in touch screen phones by analyzing the epidermal oils and smears left on the device's screen by the user's fingers [25]. Fourth, *shoulder-surfing attack* [26] uses social engineering techniques to steal the victim's personal information such as PIN code and password by looking over the victim's shoulder or by eavesdropping on sensitive information being spoken and heard or keystrokes on a device. Finally, a large number of users themselves do not lock their smartphones. [11] analyzed over 150 smartphone users and showed that 33% of the users do not use any screen lock. [12] conducted face-to-face qualitative interviews with 28 participants. 29% of the users responded that they did not lock their devices with three common reasons, including emergency personnel not identifying them, not having the devices returned if lost, and not believing they worth data. [13] run an online survey with 260 participants and a field study with 52 participants to analyze smartphone users' risk perception and behaviors. They showed that 40.9% of users use slide-to-unlock, and 16.2% of users do not use any screen lock.

Location-Based Behavioral Authentication. There are some research questions in constructing a smarter and securer mobile-based authentication. First, for mitigating the attacks above, is there an additional mobile-based authentication for supporting the conventional authentication using PIN code, password, and biometric traits (i.e., fingerprints, face, iris)? Second, imaging the scenario that a user is on the way to going to a coffee shop. Before he arrives, the coffee shop can predict that he will arrive 15 min later with a high probability, prepare in advance his usual order, and automatically subtract the charge from his account. The user then does not need to wait for the order and payment process.

So, the question is: is it possible to authenticate and predict the location (for example, the coffee shop) that the users are likely going to? Last but not least, in the situation of the COVID-19, the current smartphone-based cashless payment can reduce the chance of using cash or card, but still, the user needs to touch the smartphone screen to show the bar code to the cashier. The final question is whether the user can pay for goods when only bringing the smartphone without touching the screen?

An idea to answer the questions is using behavioral-based information. This new research's main challenge is how to decide useful behavioral information for authentication. Inspirited from L. Fridman (MIT) et al. [5], just in 2016, GPS location history is a promising approach because "It is relatively unique to each individual even for people living in the same area of a city. Also, outside of occasional travel, it does not vary significantly from day to day. Human beings are creatures of habit, and in as much as location is a measure of habit". At this time, single behavioral authentication is used as an additional method to support the conventional authentication or to combine with other behavioral authentications. In the future, if we can construct a payment system such that (i) the users do not need to bring devices, (ii) the security and privacy are ensured, and (iii) the conventional biometrics authentication can be replaced entirely, it is a step closer to Society 5.0.

Motivation. A system can achieve a high authentication accuracy when it can collect multiple factors as much as possible. However, in the users' viewpoint, a convenient system should not bring strong privacy concerns to the users by requiring too much information. From the GPS, most of the previous work utilized the longitude, latitude, and timestamp as the features for the user authentication. Given the limited information, if we can obtain metadata that carries extra independent information from the GPS itself, we can improve the accuracy. An example of GPS-based self-enhancement is [7] in which they extracted the address from the pair of longitude and latitude using a reverse geocoding.

Contribution. In this paper, we propose an idea to extract the distance coherence features from the GPS itself without any other information besides the GPS. The locations at close time clocks may have some closer correlation in physical distance than the locations at far time clocks for each user. The idea is inspired by the fact that a human needs time to move from one location to another. Since this concept can reflect a movement "lifestyle" of the users, we hypothesized that it might improve the accuracy. Although it may be not 100% correct when the user goes forward and then backward within the considered period of time, we combine the proposed distance coherence features with the previous ones. To evaluate how feasible the approach is, we collected 107,637 GPS records from 348 users. We applied three ensemble machine learning classification (RandomForest, ExtraTrees, and Bagging) on a total of 13 features, including the distance coherences features. The experimental result showed that our approach outperforms the approach without

the distance coherence features with the accuracy of 99.42% (for RandomForest), 99.12% (for ExtraTrees), 99.25% (for Bagging).

Considering its reasonability, it may raise a question. Since we infer the distance coherence from the GPS and timestamp, whether the distance coherence's entropy is the same as that of the GPS and timestamp? In other words, whether the distance coherence gives no additional information to the GPS and timestamp. However, for each sample, the corresponding distance coherence is computed from a sample and other samples with a close timestamp with the considered sample. Therefore, the GPS, timestamp, and distance coherence are independent variables. Of course, we can improve the model if we combine the GPS and timestamp with other factors such as Wifi information, web browser log, etc. However, this paper aims to clarify whether the distance coherence extracted from the GPS and the timestamp can improve the classification model. We thus excluded other factors to make the comparison clean.

Roadmap. The rest of this paper is organized as follows. The related work is introduced in Sect. 2. The proposed method is described in Sect. 3. The experiment is presented in Sect. 4. The threat model is presented in Sect. 5. The discussion about future work is shown in Sect. 6. Finally, the conclusion is drawn in Sect. 7.

2 Related Work

This section presents related work focusing on multimodal authentication using human-smartphone interactions and other factors. The term *multimodal* (not *multimodel*) is used to indicate the biometrics authentication using multiple biometric data. It is the opposite with *unimodal*, which uses only a single biometric data.

2.1 Multimodal Authentication for Smartphone

L. Fridman et al. [5] analyzed four modal behavioral data from active mobile devices, including text stylometry typed on a soft keyboard, application usage patterns, web browsing behavior, and physical location of the device from GPS (outdoor) and Wifi (indoor). They collected the data from 200 users in more than 30 d. The authors proposed a parallel binary decision-level fusion architecture for classifiers based on four biometric modalities. A. Alejandro et al. [8] analyzed multimodal data from four biometric data channels (including touch gestures, keystroking, accelerometer, and gyroscope) and three behavior profiling (including WiFi, GPS location, and app usage). They obtained the data during the natural human-smartphone interaction of 48 users, on average, ten days per user. They proposed two authentication models named the one-time approach that uses all the channel information available during one session, and an active approach that uses behavioral data from multiple sessions by updating a confidence score. W. Shi et al. [6] proposed an authentication framework

that enables continuous and implicit user identification service for a smartphone. They collected the data from four sensor modalities, including voice, GPS location, multitouch, and locomotion. They conducted a preliminary empirical study with a small set of users (seven). The result showed that the four modalities are enough for mobile user identification. R. Valentin et al. [10] analyzed multimodal sensing modalities with mobile devices when the GPS, accelerometer, and audio signals are utilized for human recognition. They collected the data from four existing datasets which consist of 491 users. They applied four variants of deep learning for interpreting user activity and context as captured by multi-sensor systems. M. Upal et al. [14] investigated user authentication methods using the first non-commercial multimodal data, which focuses on three smartphone sensors (front camera, touch sensor, and location service). They collected the data from 48 users for two months. Their benchmark results for face detection, face verification, touch-based user identification, and location-based next place prediction showed that more robust methods fine-tuned to the mobile platform are needed to achieve satisfactory verification accuracy. T. Thao et al. [7] extracted the addresses given the longitudes and latitudes from the GPS records. They then applied the text mining on the addresses. They collected the data from 50 users for about four months. Their experimental result showed that the combination between the text features and the GPS data could improve the classification accuracy. B. Aaron et al. [9] proposed a wallet repository that can store biometric data using multiple layers: a biometric layer, a genomic layer, a health layer, a privacy layer, and a processing layer. They used the processing layer to determine and track the user location, the speed when the user is moving using GPS data.

2.2 Other Multimodal Authentication

Besides using human-smartphone interactions, multimodal authentication also uses other factors. T. Kaczmarek et al. [27] investigated a new hybrid biometric based on a human user's seated posture pattern in an average office chair throughout a typical workday. Their experimental results on a population of 30 users showed that the posture pattern biometric could capture a unique combination of physiological and behavioral traits and can authenticate the users with 91% of accuracy. M. Ivan et al. [28] proposed an approach which combines the PIN code and the pulse-response. For the experiment process, they collected biometric information from 10 users. The result showed that each human body exhibits a unique response to a signal pulse applied at the palm of one hand and measured at the other's palm. The experimental result for user authentication achieved 88% of accuracy when taking the records weeks apart. W. Louis et al. [30] and R. Alejandro et al. [32] constructed a continuous authentication system based on electrocardiogram (ECG) and electroencephalogram (EEG). Their approaches achieved 1.57% and 0.82% of the false-negative rate, respectively. E. Simon et al. [29] extracted distinct patterns from eye movement (it is different from iris) with 21 features for user authentication. The data was collected

from 30 users in 2 weeks with three scenarios (no prior knowledge, the knowledge gained through description, and knowledge gain through observation). The experimental result achieved 3.98% of equal error rate.

3 Proposed Approach

This section describes our proposed method, including data collection, feature extraction and selection, and our learning method.

3.1 Data Collection

We created a navigation application named MITHRA (Multi-factor Identification/auTHentication ReseArch) in the project of the University of Tokyo to collect the users' GPS information. The application is available on both iOS and Android smartphones. We developed the application run in the background. We collected the data from 348 users with 107,637 GPS records, including pairs of longitude and latitude for four months from January 11 to April 26 in 2017[2]. Compared to the existing works (see Sect. 2), the number of users in our dataset is higher than most of the papers and is only lower than [10], which could collect the information from 491 users. We recruited the participants randomly. The users live and work in random areas. The GPS data was measured every minute. The value of the longitudes and latitudes were collected with the precision up to 6 decimal places (e.g., 36.xxxxxx) corresponding to 0.1 m.

Privacy Consent. The privacy consent is shown to the users during the installation process. The installation can only be done if the users accept the terms and conditions agreement. Even after successfully installing the application, the users can choose to start or stop using the application anytime. Any personal information of the users such as name, age, gender, race, ethnicity, income, education, etc. is not collected. We collected only the email addresses the user identity used to distinguish the users from each other. Although the application collects the GPS information, the users do not need to disclose their home location, office location, etc. Our project was reviewed by the Ethics Review Committee of the Graduate School of Information Science and Technology, the University of Tokyo. Finally, all the users who installed the application agreed to participate in our project.

3.2 Feature Extraction and Selection

We categorized the features into two groups: (i) the features extracted from the GPS and the timestamp, and (ii) the features using the distance coherence score.

[2] Although we collected the GPS from smartphones in this project, we can also collect the GPS from many smaller devices such as smartwatches or smartbands nowadays.

GPS and Datetime. There are seven features in this group. Two features were extracted from the GPS, including the latitudes and the longitudes represented by float numbers. The valid ranges for the latitudes and the longitudes are the continuous range $[-90, +90]$ and $[-180, +180]$, respectively. Five features were extracted from the timestamp, including month, day, hour, minute, and day of a week (i.e., seven days from Monday to Sunday) represented by integer numbers. The valid ranges for these features are the intervals $[1, 12]$, $[1, 31]$, $[0, 23]$, $[0, 59]$, and $[1, 7]$, respectively. We did not extract the year as a feature because all the data samples were collected in the same year (2017).

Distance Coherence. There are α features in this group (we will soon explain how to choose α). Each z-th feature ($z \in [1, \alpha]$) represents the distance coherence (similarity score) between each data sample with the average of all the other samples in the dataset that belong to the same user and that occur before or after p hours for every $p \in [0, z]$ with the considered sample. $p = 0$ is when the other samples occur in the same hour with the considered sample.

More concretely, the features are computed as follows (see Fig. 1). Let $\{\mathsf{dc}_z\}$ denote the set of α features where $z \in [1, \alpha]$. Let s_i denote each sample in the dataset where $i \in [1, n]$ and n denotes the number of samples (in our dataset, $n = 107,637$). For each feature dc_z, let $K_z = \{s'_j\}$ (where $j \in [1, n]$ and $j \neq i$) denotes the set of all the other samples such that s_i and s'_j belong to the same user U_t (where $t \in [1, 348]$). State differently, s_i and s'_j have the same label U_t. Let $lat(s_i)$ and $lat(s'_j)$, $lon(s_i)$ and $lon(s'_j)$, and $hour(s_i)$ and $hour(s'_j)$ denote the latitude, the longitude, and the hour features for s_i and s'_j, respectively. For each dc_z, K_z is chosen such that:

$$hour(s_i) = hour(s'_j) \pm p \qquad \text{for } \forall p \in [1, z] \tag{1}$$

The average coordinate s''_j is determined from all the samples s'_j in K_z such as:

$$lat(s''_j) = \mathsf{average}(lat(s'_j)) \quad \forall s'_j \in K_z \tag{2}$$

$$lon(s''_j) = \mathsf{average}(lon(s'_j)) \quad \forall s'_j \in K_z \tag{3}$$

The features are finally calculated as the distance between s_i and s''_j:

$$\mathsf{dc}_z(s_i) = \sqrt[2]{(lat(s_i) - lat(s''_j))^2 + (lon(s_i) - lon(s''_j))^2} \tag{4}$$

From Eq. 1, we can observe that K_z chosen for dc_z is a subset of $K_{z'}$ chosen for $\mathsf{dc}_{z'}$ for all $z, z' \in [1, \alpha]$ such that $z' > z$. It may raise the question that whether all the α features have a correlation. However, the averages from even correlated sets are completely different (for example, $\mathsf{average}(1, 2, 3) = 2$ which is different from $\mathsf{average}(1, 2, 3, 4) = 5$). All the features dc_z are thus independent variables. A numeric example for how to calculate the distance coherence features will be given in Appendix A.

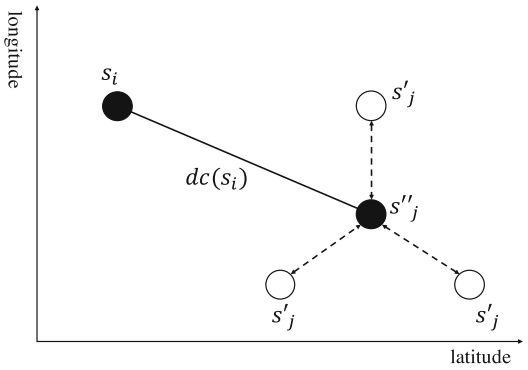

Fig. 1. Distance coherence (similarity score)

We now explain how the concrete value for α is. In our approach, we use three advanced classification machine learning algorithms, which are RandomForest, ExtraTrees, and Bagging (explained in more detail in Sect. 3.3). We experimented with every α from 1 and increased it gradually. We found that the best α for RandomForest, ExtraTrees, and Bagging is 3, 4, and 5, respectively, at which the algorithms reach the peak performance (Sect. 4.3). Since α reflects the movement lifestyle of the users, it is reasonable for α to be not large. For instance, the GPS (latitude, longitude) of a user U_t at 15:00 may have some physical distance coherence with the GPS records at 14:00 and 16:00 than the GPS records at 13:00 and 17:00. In the rest of this paper, we use α-DC to denote the approach in which α distance coherence features are used, and $\{\mathsf{lat}, \mathsf{lon}, \mathsf{mon}, \mathsf{day}, \mathsf{hour}, \mathsf{min}, \mathsf{weekday}, \mathsf{dc}_1, \mathsf{dc}_2, \cdots, \mathsf{dc}_6\}$ to denote the set of the thirteen features related to both the GPS and timestamp and the distance coherence.

Feature Distribution. We describe the distribution statistics for the features in Table 1, including the mean, standard error, median, standard deviation, Kurtosis score, skewness score, min value, and max value. A normal distribution check for the features is not necessary [31]. The negative and positive values in the latitude and the longitude in the "Min" and "Max" columns indicate that the users who used to commute in Japan might travel abroad during the data collection. This kind of data can create noises during the training and testing processes. However, we did not remove it because the data reflects the users' natural behavior. Although the noises may lower the accuracy, we want to measure how practical the approach is when using real data without being manipulated.

Table 1. Feature distribution

Feature	Mean	SE	Median	SD	Kurtosis	Skewness	Min	Max
lat	35.262	0.014	35.376	4.554	151.722	−10.935	−36.858	43.907
lon	136.783	0.034	137.846	11.165	248.09	−15.101	−121.979	174.799
month	3.321	0.002	3.000	0.753	−0.260	−0.777	1.000	4.000
day	17.328	0.026	19.000	8.600	−1.075	−0.285	1.000	31.000
hour	13.421	0.019	14.000	6.388	−0.820	−0.417	0.000	23.000
min	28.919	0.053	29.000	17.357	−1.186	0.038	0.000	59.000
weekday	3.986	0.006	4.000	1.966	−1.215	−0.016	1.000	7.000
dc_1	4,104.198	137.84	191.318	45,214.683	976.703	27.756	0.000	2,545,473.711
dc_2	4,359.489	137.598	239.163	45,140.323	988.797	27.801	0.000	2,548,301.562
dc_3	4,586.805	140.910	259.640	46,228.488	995.124	27.895	0.000	2,549,190.471
dc_4	4,678.671	140.658	272.654	46,147.07	978.139	27.653	0.004	2,554,832.383
dc_5	4,781.704	141.784	276.978	46,516.486	1,002.699	28.001	0.048	2,567,773.385
dc_6	4,822.694	143.361	284.604	47,033.864	1,013.685	28.284	0.017	2,568,234.888

SE (Standard Error), SD (Standard Deviation), DC: Distance Coherence

3.3 Learning

This section explains the machine learning algorithms chosen for our model and the evaluation method. In the dataset, each user has a different label. Each label has a different set of records.

Average Ensemble Classifications. The dataset contains 107,637 samples with a large number of labels (348 users). Instead of using the traditional algorithms, we use *average ensemble classifications* to get better performance. The average ensemble algorithms build several base estimators independently and produce one optimal predictive estimator by averaging all the base estimators' predictions. The combined estimator is better than any single base estimator by reducing the variance to control over-fitting. The common algorithms include:

- RandomForest [1]: implements a meta estimator that fits some decision tree classifiers on various randomized sub-samples and uses averaging to create the best predictive estimator. When each estimator is built, a bootstrap is created by randomly sampling the dataset with replacement. The sub-samples' size is set to be the same as the size of the original input sample. A decision tree is usually trained by recursively splitting the data (converting the non-homogeneous parent into the two most homogeneous child nodes). The algorithm selects an optimal split on the features selected at every node.
- ExtraTrees [2]: produces the best predictive estimator in a way like RandomForest. However, there are some differences. While RandomForest uses the optimal split, ExtraTrees uses the random split. While RandomForest sets the $bootstrap = True$ by default, ExtraTrees sets the $bootstrap = False$ by default. It means that while RandomForest supports drawing sampling with replacement, ExtraTrees supports drawing sampling without replacement.

– Bagging (Bootstrap Aggregating) [3]: uses all the features for splitting a node while RandomForest and ExtraTrees select only a subset of randomized features for splitting a node.

Stratified K-Fold. We shuffled the data at first and then used a k-fold cross validation. Since the numbers of samples of the users are imbalanced, using the normal k-fold cross validation can lead to the following problem. There may exist a class c_k ($k \in \{1, 2, \cdots, 348\}$) in which all the samples belong to the test set; and the training set does not contain any samples. The classifier, therefore, cannot learn about the class c_k. To solve this problem, we used *Stratified k-fold* cross-validation object, which is a variation of k-fold and can deal with imbalanced data in each class. As presented in Fig. 2, it splits the data in the train and the test sets. It returns stratified folds made by preserving the percentage of samples for each class.

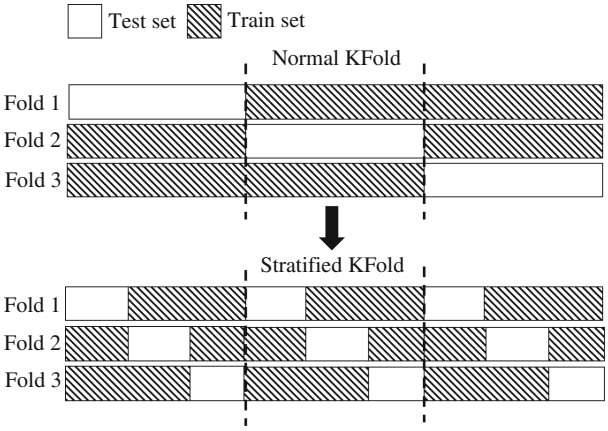

Fig. 2. A stratified KFold

Evaluation Metrics. To evaluate our approach, we measure the following metrics:

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}, \quad precision = \frac{tp}{tp + fp}, \quad recall = \frac{tp}{tp + fn} \tag{5}$$

$$F1 = 2 \times \frac{recall \times precision}{recall + precision}, \quad FPR = \frac{fp}{fp + tn}, \quad FNR = \frac{fn}{fn + tp} \tag{6}$$

where tp, tn, fp, fn denote the true positive, true negative, false positive, and false negative values, respectively. FPR and FNR denote the false positive rate and false-negative rate, respectively. The accuracy is a good metric when the distribution for each label is almost similar. However, for an imbalanced dataset, F1-score is the better metric.

4 Experiment

This section presents the experimental setup, the results obtained after applying the classification, and how to find the best α for each algorithm.

4.1 Experimental Setup

We implemented the program using Python 3.7.4 on a computer MacBook Pro 2.8 GHz Intel Core i7, RAM 16 GB. The machine learning algorithms are executed using *scikit-learn*[3] library version 0.22.

For each ensemble algorithm, the number of base estimators $n\_estimators$ is set to 100. The k value in the stratified k-fold cross validation is set to $k = 10$. Since the categorical labels are represented in text strings (such as 'user001', 'user002', etc.), the labels are transformed to numerical values using the *label encoding*. While the *ordinal encoding* encodes a label to an integer array and the *one-hot encoding* encodes it to a one-hot numeric array, the *label encoding* encodes it to the values between 0 and $q - 1$ where q is the number of distinct labels of all the classes. The label encoding is the most lightweight method and uses less disk space. Since the data is imbalanced, to avoid the situation that F1 is not between precision and recall, we calculate the three metrics (precision, recall, and F1 score) for each label and find their average weight by the number of true instances of each class. This process can be done by setting the parameter $average = weighted$ in the *sklearn.metrics*. For the accuracy, this parameter is not necessary. Since the values of the distance coherence features are small, we scaled them up to $\times 10^4$. For each of the three algorithms (RandomForest, ExtraTrees, and Bagging), we experimented with different α's. We applied the classification 107,637 samples with 348 labels, which correspond to 348 users.

4.2 Main Result

The main result is presented in Table 2. In the table, NoDC represents the approach not using distance coherence features, while α-DC represents the approach using α distance coherence features. As proved later in Sect. 4.3, RandomForest, ExtraTrees, and Bagging reach the best performance at $\alpha = 3$, $\alpha = 4$ and $\alpha = 5$, respectively. Thus, we chose 3-DC, 4-DC, and 5-DC to compare with NoDC in this table (although only 1-DC can already beat NoDC (see Sect. 4.3)).

The result shows that our approach α-DC outperforms NoDC in all the cases. Comparing all the algorithms using NoDC only with each other, Bagging

[3] https://scikit-learn.org/stable/.

gives the best result with 98.69% of F1 score with 0.02% of false-negative rate. Comparing all the algorithms using our approach with each other, RandomForest gives the best result with 99.42% of F1 score and merely 0.01% of false-negative rate even though RandomForest just reaches $\alpha = 3$ (which is less than $\alpha = 4$ for ExtraTrees and $\alpha = 5$ for Bagging). Comparing the improvement between α-DC and NoDC, ExtraTrees gives the best result when 2.34% of F1 score is increased ($\triangle = +2.34$) and 0.04% of false-negative rate is reduced ($\triangle = -0.04$).

Table 2. Result for distance coherent with different ensemble algorithms

Measure	RandomForest			ExtraTrees			Bagging		
	NoDC	3-DC	△	NoDC	4-DC	△	NoDC	5-DC	△
F1	97.95	99.42	**+1.47**	96.77	99.11	**+2.34**	98.69	99.24	**+0.55**
Accuracy	97.97	99.42	**+1.45**	96.80	99.12	**+2.32**	98.69	99.25	**+0.56**
Precision	98.05	99.45	**+1.40**	96.90	99.15	**+2.25**	98.75	99.28	**+0.53**
Recall	97.97	99.42	**+1.45**	96.80	99.12	**+2.32**	98.69	99.25	**+0.56**
FPR	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**
FNR	0.03	0.01	**−0.02**	0.05	0.01	**−0.04**	0.02	0.01	**−0.01**

NoDC: the approach without distance coherence features,
α-DC ($\alpha = 3, 4, 5$): the approach using distance coherence features,
△: the improved score between α-DC and NoDC.

4.3 Best Alpha (α) for Each Algorithm

This section explains the experiment to find the best α for each algorithm. First, α is set to 1 and is then gradually increased until the performance becomes convergent or reduced after reaching the peak. The result and its graphs are presented in Table 3 and Fig. 3. The proposed approach using RandomForest, ExtraTrees, and Bagging got the best performance at $\alpha = 3$, $\alpha = 4$, and $\alpha = 5$, respectively. Figure 3 shows that in all the algorithms, the graph almost has the cone shape (the result is gradually increased, gets the peak, and then is reduced or becomes convergent), not a zigzag shape (in which we cannot predict where is the peak). The result also shows that by even just using 1-DC ($\alpha = 1$), our approach can already beat NoDC.

4.4 Computation Time

For the best algorithms (5-DC using Bagging, 4-DC using ExtraTrees, and 3-DC using RandomForest), the average computational time for the training and cross-validation processes from 5 execution times is 2,272 s (38 min), merely 270 s (4.5 min), and 596 s (10 min) respectively. It is not a big deal for the server. When the number of users is much more increased (e.g., to thousands), it is not complicated to transform the current model from the one-class classification to a multi-class classification where each user has a different classifier with binary labels representing whether or not a sample belongs to that user.

Table 3. Result for each alpha

		1-DC	2-DC	3-DC	4-DC	5-DC	6-DC
RandomForest	F1	99.11	99.41	**99.42**	99.38	99.36	99.31
	Accuracy	99.11	99.42	**99.42**	99.38	99.37	99.31
	Precision	99.15	99.44	**99.45**	99.41	99.39	99.35
	Recall	99.11	99.42	**99.42**	99.38	99.37	99.31
	FPR	0.00	0.00	**0.00**	0.00	0.00	0.00
	FNR	0.01	0.01	**0.01**	0.01	0.01	0.01
ExtraTrees	F1	97.27	98.90	98.98	**99.11**	99.11	99.11
	Accuracy	97.30	98.91	98.99	**99.12**	99.12	99.11
	Precision	97.40	98.95	99.03	**99.15**	99.15	99.15
	Recall	97.30	98.91	98.99	**99.12**	99.12	99.11
	FPR	0.00	0.00	0.00	**0.00**	0.00	0.00
	FNR	0.04	0.02	0.02	**0.01**	0.01	0.01
Bagging	F1	99.03	99.07	99.10	99.14	**99.24**	99.23
	Accuracy	99.04	99.07	99.10	99.14	**99.25**	99.23
	Precision	99.07	99.11	99.14	99.18	**99.28**	99.26
	Recall	99.04	99.07	99.10	99.14	**99.25**	99.23
	FPR	0.00	0.00	0.00	0.00	**0.00**	0.00
	FNR	0.01	0.01	0.01	0.01	**0.01**	0.01

5 Threat Model

This section presents the threat model, including the focused attack, the adversary's probability, and the assumptions.

5.1 Targeted Attack

Most of such authentication systems, not just our approach but other previous biometrics-based authentication, focus on protecting against insider threats in which the adversary tries to impersonate the authentication of an authorized user in the system. As mentioned in Sect. 1, at this time, the behavioral-based authentication should be used as an additional approach to support the conventional PIN code, password, or biometric authentications. So let's run an example in which our approach is combined with PIN code-based authentication. Let $Pr_{\mathcal{A}}$ denote the probability that the adversary $\mathcal{A}$ can break the system. $Pr_{\mathcal{A}}$ is defined as:

$$Pr_{\mathcal{A}} = Pr_{guess} \cdot Pr_{forge} \tag{7}$$

where Pr_{guess} and Pr_{forge} denote the probability that $\mathcal{A}$ can correctly guess the PIN code and the average probability that $\mathcal{A}$ can fool the classifier, respectively. Pr_{forge} is the false-negative rate which is the percentage of identification

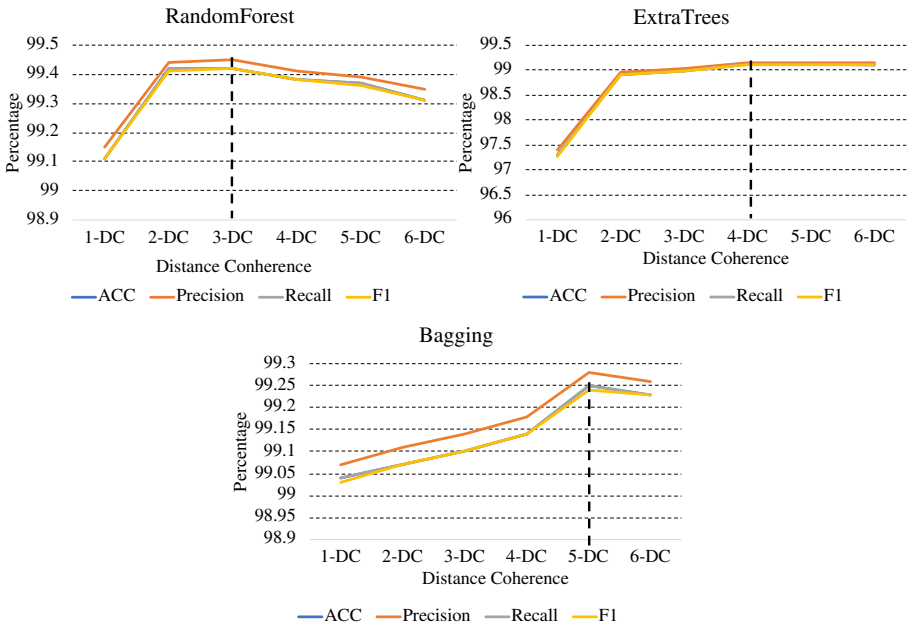

Fig. 3. Different alpha's for distance coherence

instances in which the unauthorized users are incorrectly accepted. Table 2 shows that all the 3-DC, 4-DC, and 5-DC approaches corresponding to the three different algorithms have the same 0.01% of false-negative rate. Thus, $Pr_{forge} = 10^{-4}$. Let τ and σ denote the number of digits in the PIN code and the number of guessing candidates for each PIN code digit. If $\mathcal{A}$ has n_t tries before the device is locked with many wrong PIN codes, we have $Pr_{guess} = \frac{n_t}{\sigma^\tau}$. Therefore:

$$Pr_{\mathcal{A}} = 10^{-4} \cdot \frac{n_t}{\sigma^\tau} \tag{8}$$

Most of the new smartphone operation systems nowadays require six digits for PIN code. Typically, there are ten digits of candidates from 0 to 9 for each digit. The users often have 4 to 6 PIN code tries for Android and iOS before the device is locked. Therefore, $Pr_{\mathcal{A}} \simeq 4 \cdot 10^{-10}$ to $6 \cdot 10^{-10}$.

Suppose the attacker can guess the PIN code after shoulder surfing and then robs the user's smartphone. Since the application is designed such that every GPS record is sent to the server in realtime and the GPS history is not stored in the user smartphone, the attacker cannot see the log from the robbed phone to imitate the user's behavior. Also, there is no function of downloading the GPS log from the server to the smartphone because it is a doubtable action from a (suspicious) user. The only action that the attacker can manipulate on the GPS tracking application is to turn it on/off or uninstall it. If the attacker continues to use the smartphone without being able to search for the history log from the smartphone application), the probability for the attacker $Pr_{\mathcal{A}}$ is now 0.01%.

Even though it is not 0% for the best case, it is still much better than 100% for $\mathcal{A}$ to break the system without our approach. Similarly, if the *collusion attack* in which an authorized user shares his/her PIN code to others occurs, $Pr_{\mathcal{A}}$ is also 0.01%. If the colluded user tells others his/her personal location history, every single continuous GPS record cannot be imitated. It is why we investigated the idea of using behaviors (especially long-term and continuous).

The model assumes that the server storing the GPS cannot be accessed or corrupted by the adversary. The data is encrypted, and only the trusted server can decrypt it. The data is transmitted via a secure network. Each smartphone is used by only a unique user. The smartphone and the server are protected against the side-channel attack collecting the user data via timing information, power consumption, electromagnetic leaks, or sound. Finally, we assumed that the users are honest in sending their data to the server, which performs the classification.

5.2 Security Scenario Discussion

In this section, we discuss other security scenarios from using smartphones.

What if Two Users Live and Work in the Same Areas? As mentioned in Sect. 3.1, since our project recruited the users randomly, the users live and work in random areas. Even if in the rare case, when two users live and work in the same area, they cannot have the same GPS tracking for every single hour. Each user has many activities at different timestamps, not just at home and office (such as shopping, outdoor exercising, picking children at schools, etc.). Furthermore, we can collect indoor positioning inside the home and the office building besides the GPS such as WiFi or Bluetooth beacons. Since this paper aims to investigate the benefit of the extra information (i.e., the distance coherence) from the GPS itself, we do not consider to collect indoor location information. However, it is entirely possible since we can collect the GPS and the indoor location information independently. Let's consider the case when legal users have the same trajectory within a period of time (e.g., older people in a senior home have daily activities confined to the surroundings). Since the longitude and latitude values have 6 decimal places (see Sect. 3.1), the precision is 0.1 m. With this precision, two users cannot have the same movement log in a long period.

How Does the System Work When Individuals Are Outside Their Routine or When the Attacker Follows (imitates) the User's Behavior? Since these questions are not just for the GPS-based location authentication but the general behavioral-based authentication, we discuss from the general to specific perspectives. We emphasize that a single-factor behavioral-based authentication is used to support (not to replace) the conventional approaches such as password or biometrics; or it is combined with other behavioral factors to build up a multi-factor behavioral-based authentication. Suppose a user is outside

his/her routine or the attacker tries to imitate the user's behavior. In that case, the password/biometric or other routines are used to lower the false rejection and false acceptance rates. Although behavioral-based authentication has not yet been commonly used, researchers proved that this new but promising research is possible for real applications. For instance, Google has launched the Project Abacus [33] in 2016 to collect smartphone sensor signals (i.e., front-facing camera, touchscreen and keyboard, gyroscope, accelerometer, magnetometer, ambient light sensor, etc.). They demonstrated that human kinematics could convey important information about user identity and serve as a valuable component of multi-modal authentication systems. Among many behaviors, location is a typical factor in identifying users. Human beings are creatures of habit, and in as much as location is a measure of habit [5]. Also, the location is easy to collect since it is available in most modern smartphones.

Is It a Problem When a User Gets a New Phone? It has no problem since the smartphone is just the device/tool, not the method. The user can register a location-based authentication system with an account and its application installed in his smartphones. As long as the user does not share his account with others and an account can only log in one smartphone at a specific timestamp, his unique GPS data can be collected regardless of how many smartphones are used and whether the user shares his smartphones with others.

6 Future Work

This section describes an idea for future work based on the separation of daily and weekly distance coherences. In our current approach, for each sample s_i, the distance coherence features are calculated by grouping the other samples, which have the corresponding clock hours close to the clock hour of s_i regardless of the dates. We thus call it daily distance coherence. An example is given in the first chart of Fig. 4. We can calculate the features chosen for the sample s_i at the timestamp 7:00 April 10, 2020 (Friday) using the samples at $7{:}00 \pm \alpha$ on any date of the same user.

However, another promising method may improve the accuracy or F1 score. For each sample s_i, we can calculate the distance coherence features by grouping the other samples, which have the clock hours close to the clock hour of s_i on only the days with the same day of the week. We thus call it weekly distance coherence. We consider the example in the second chart of Fig. 4. Suppose s_i occurred at 7:00 April 10, 2020 (Friday); we can calculate the featured chosen for s_i from the samples at $7{:}00 \pm \alpha$ on every Friday such as April 03, 2020, or April 17, 2020, etc. These features may reflect the lifestyle of the users that we are aiming for in this paper. For example, a worker goes to work every weekday but goes to the usual supermarket every Saturday around 10:00; a student has a training course at a usual stadium every Thursday around 15:00. The weekly distance coherence can measure these habits. Remark that the weekly distance coherence features are not covered in the daily ones. Each feature is computed

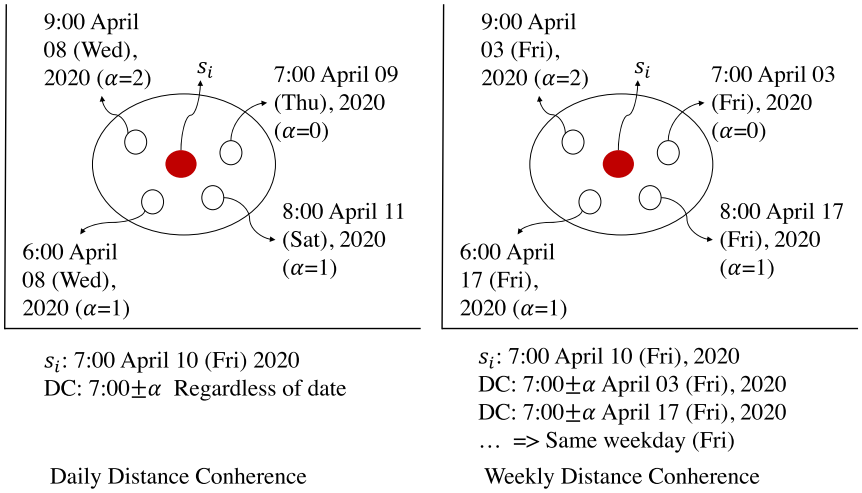

Fig. 4. Daily and weekly distance coherence

from the average of all the samples chosen for the main sample. Even though the set of the samples selected for the weekly case is a subset of the set, their averages are different in the daily case.

7 Conclusion

This paper has shown that using the distance coherence score as the additional features can improve user authentication. We collected 107,637 GPS records, including longitude, latitude, and timestamp from 348 users in Japan. The three average ensemble algorithms, including RandomForest, ExtraTrees, and Bagging, are applied to the classification and are evaluated using stratified k-fold. The experimental result showed that our approach outperforms the approach without the distance coherence in all the cases. The accuracy can reach up to 99.42%, 99.12%, and 99.25% using RandomForest, ExtraTrees, and Bagging, respectively. The F1 score can be improved even 2.34%, and the false-negative rate can be reduced by 0.04% using ExtraTrees.

Appendix

A Numeric Example (for Distance Coherence Extraction)

In this section, we give a numeric example for the distance coherence extraction in Sect. 3.2. Suppose the data consists of 7 samples $\{s_1, s_2, \cdots, s_7\}$ from 2 users $\{user1, user2\}$ as showed in Table 4. We explain how to calculate the distance coherence for each sample $\{dc_{11}, dc_{12}, dc_{13}, dc_{21}, dc_{22}, dc_{23}, dc_{24}\}$. Suppose α (the number of distance coherence feature) is set to $\alpha = 1$.

Table 4. Numeric example for calculating distance coherence

Sample ID	User/class	Timestamp	Longitude	Latitude	Distance coherence
1	User1	2020/01/16 10:55	lon_{11}	lat_{11}	dc_{11}
2	User1	2020/01/17 11:55	lon_{12}	lat_{12}	dc_{12}
3	User1	2020/01/17 12:50	lon_{13}	lat_{13}	dc_{13}
4	User2	2020/01/16 21:30	lon_{21}	lat_{21}	dc_{21}
5	User2	2020/01/17 22:10	lon_{22}	lat_{22}	dc_{22}
6	User2	2020/01/18 21:45	lon_{23}	lat_{23}	dc_{23}
7	User2	2020/01/19 20:10	lon_{24}	lat_{24}	dc_{25}

– For s_1, the hour extracted from the timestamp is $hour(s_1) = 10$. We find all the samples s_i that belong to the same class ($user1$) and have $hour(s_i)$ such that $(hour(s_1) - \alpha) \leq hour(s_i) \leq (hour(s_1) + \alpha)$ regardless of the date and the second. Only s_2 satisfies the conditions (i.e., $hour(s_2) = 11$). Thus:

$$dc_{11} = \sqrt[2]{(lon_{11} - lon_{12})^2 + (lat_{11} - lat_{12})^2} \tag{9}$$

– For s_2, $hour(s_2) = 11$. s_i from $user1$ that satisfy $(hour(s_2) - \alpha) \leq hour(s_i) \leq (hour(s_2) + \alpha)$ are s_1 and s_3 ($hour(s_1) = 10, hour(s_3) = 12$). Thus:

$$dc_{12} = \sqrt[2]{(lon_{12} - \frac{lon_{11} + lon_{13}}{2})^2 + (lat_{12} - \frac{lat_{11} + lat_{13}}{2})^2} \tag{10}$$

– For s_3, $hour(s_3) = 12$. s_i from $user1$ that satisfies $(hour(s_3) - \alpha) \leq hour(s_i) \leq hour(s_3) + \alpha)$ is only s_2 ($hour(s_2) = 11$). Thus:

$$dc_{13} = \sqrt[2]{(lon_{13} - lon_{12})^2 + (lat_{13} - lat_{12})^2} \tag{11}$$

– For s_4, $hour(s_4) = 21$. s_i from $user2$ that satisfy $(hour(s_4) - \alpha) \leq hour(s_i) \leq (hour(s_4) + \alpha)$ are s_5, s_6, and s_7 ($hour(s_5) = 22, hour(s_6) = 21, hour(s_7) = 20$). Thus:

$$dc_{21} = \sqrt[2]{(lon_{21} - \frac{lon_{22} + lon_{23} + lon_{24}}{3})^2 + (lat_{21} - \frac{lat_{22} + lat_{23} + lat_{24}}{3})^2} \tag{12}$$

– For s_5, $hour(s_5) = 22$. s_i from $user2$ that satisfy $(hour(s_5) - \alpha) \leq hour(s_i) \leq (hour(s_5) + \alpha)$ are s_4 and s_6 ($hour(s_4) = hour(s_6) = 21$). Thus:

$$dc_{22} = \sqrt[2]{(lon_{22} - \frac{lon_{21} + lon_{23}}{2})^2 + (lat_{22} - \frac{lat_{21} + lat_{23}}{2})^2} \tag{13}$$

– For s_6, $hour(s_6) = 21$. s_i from $user2$ that satisfy $(hour(s_6) - \alpha) \leq hour(s_i) \leq (hour(s_6) + \alpha)$ are s_4, s_5, and s_7 ($hour(s_4) = 21, hour(s_5) = 22, hour(s_7) = 20$). Thus:

$$dc_{23} = \sqrt[2]{(lon_{23} - \frac{lon_{21} + lon_{22} + lon_{24}}{3})^2 + (lat_{23} - \frac{lat_{21} + lat_{22} + lat_{24}}{3})^2} \tag{14}$$

– For s_7, $hour(s_7) = 20$. s_i from $user2$ that satisfy $(hour(s_7) - \alpha) \leq hour(s_i) \leq (hour(s_7) + \alpha)$ are s_4 and s_6 $(hour(s_4) = hour(s_6) = 21)$. Thus:

$$dc_{24} = \sqrt[2]{(lon_{24} - \frac{lon_{21} + lon_{23}}{2})^2 + (lat_{24} - \frac{lat_{21} + lat_{23}}{2})^2} \quad (15)$$

References

1. Breiman, L.: Random Forests. Mach. Learn. **45**(1), 5–32 (2001)
2. Geurts, P., Damien, E., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
3. Louppe, G., Geurts, P.: Ensembles on random patches. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2012), pp. 346–361 (2012)
4. Cabinet Office, the Government of Japan, Society 5.0. https://www8.cao.go.jp/cstp/english/society5_0/index.html. Accessed 26 Apr 2020
5. Fridman, L., Steven, W., Rachel, G., Moshe, K.: Active authentication on mobile devices via Stylometry, application usage, web browsing, and GPS location. IEEE Syst. J. **11**(2), 513–521 (2016)
6. Shi, W., Yang, J., Jiang, Y., Yang, F., Xiong, Y.: SenGuard: passive user identification on smartphones using multiple sensors. In: IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2011), pp. 141–148 (2011)
7. Thao, T.P., Irvan, M., Kobayashi, R., Yamaguchi, R.S., Nakata, T.: Self-enhancing GPS-based authentication using corresponding address. In: Singhal, A., Vaidya, J. (eds.) DBSec 2020. Self-enhancing GPS-Based Authentication Using Corresponding Address, vol. 12122, pp. 333–344. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49669-2_19
8. Alejandro, A., Aythami, M., Vera-Rodriguez, R., Julian, F., Ruben, T.: MultiLock: mobile active authentication based on multiple biometric and behavioral patterns. In: International Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA 2019), pp. 53–59 (2019)
9. Aaron, B., Christopher, D., Barry, G., David, K.: System and method for real world biometric analytics through the use of a multimodal biometric analytic wallet. US patent, US20180276362A1 (2018). https://patents.google.com/patent/US20100050253. Accessed 26 Apr 2020
10. Valentin, R.: Multimodal deep learning for activity and context recognition. In: Publication: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT 2018), (2018) Article no. 157
11. Dirk, B., Shu, L., Mitch, K., Aaron, S., Charles, C., John, D.: Modifying smartphone user locking behavior. In: 9th Symposium on Usable Privacy and Security (SOUPS 2013), pp. 1–14 (2013). article no. 10
12. Egelman, S., Jain, S., Portnoff, R., Liao, K., Consolvo, S., Wagner, D.: Are you ready to lock?. In: 21st ACM Conference on Computer and Communications Security (CCS 2014), pp. 750–761 (2014)
13. Harbach, M., Zezschwitz, E., Fichtner, A., Luca, A., Smith, M.: It's a hard lock life: a field study of smartphone (un) locking behavior and risk perception. In: 10th USENIX Conference on Usable Privacy and Security (SOUP 2014), pp. 213–230 (2014)

14. Upal, M., Sayantan, S., Vishal, P., Rama, C.: Active user authentication for smartphones: a challenge data set and benchmark results. In: 8th International Conference on Biometrics Theory, Applications and Systems (BTAS 2016) (2016)
15. Hashcat - Advanced Password Recovery. https://hashcat.net/hashcat/. Accessed 27 Apr 2020
16. John the Ripper password cracker. https://www.openwall.com/john/. Accessed 27 Apr 2020
17. Marsicoa, M.D., Michele, N., Daniel, R., Wechsler, H.: Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols. Pattern Recogn. Lett. **57**, 17–23 (2015)
18. Venugopalan, S., Savvides, M.: How to generate spoofed irises from an iris code template. IEEE Trans. Inf. Forensics Sec. **6**(2), 385–395 (2011)
19. Bowyer, K.W., Doyle, J.S.: Cosmetic contact lenses and iris recognition spoofing. Computer **47**(5), 96–98 (2014)
20. Daksha, Y., Naman, K., James, S.D., Richa, S., Mayank, V., Kevin, W.B.: Unraveling the effect of textured contact lenses on iris recognition. IEEE Trans. Inf. Forensics Secur. **9**(5), 851–862 (2014)
21. Chingovska, I., Anjos, A., Marcel, S.: On the effectiveness of local binary patterns in face anti-spoofing. In: IEEE International Conference of Biometrics Special Interest Group (BIOSIG 2012) (2012)
22. Erdogmus, N., Marcel, S.: Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect. In: IEEE 6th International Conference on Biometrics: Theory Applications and Systems (VISAPP 2013), pp. 1–6 (2013)
23. Anthony, R.: System for and method of securing fingerprint biometric systems against fake-finger spoofing. US Patent US7505613B2 (2009). https://patents.google.com/patent/US7505613B2/en. Accessed 27 Apr 2020
24. David, M., et al.: Deep representations for iris, face, and fingerprint spoofing detection. IEEE Trans. Inf. Forensics Secur. **10**(4), 864–879 (2015)
25. Adam, J.A., Katherine, G., Evan, M., Matt, B., Jonathan, M.S.: Smudge attacks on smartphone touch screens. In: 4th USENIX Conference on Offensive Technologies (WOOT 2010), pp. 1–7 (2010)
26. Malin, E., Mohamed, K., Emanuel, V.Z., Heinrich, H., Florian, A.: Understanding shoulder surfing in the wild: stories from users and observers. In: ACM CHI Conference on Human Factors in Computing Systems, pp. 4254–4265 (2017)
27. Kaczmarek, T., Ercan, O., Gene, T.: Assentication: user deauthentication and lunchtime attack mitigation with seated posture biometric. In: 16th Conference on Applied Cryptography and Network Security (ACNS 2018), pp. 616–633 (2018)
28. Ivan, M., Kasper, R., Marc, R., Gene, T.: Authentication using pulse-response biometrics. Commun. ACM **60**(2), 108–115 (2017)
29. Simon, E., Kasper, B.R., Vincent, L., Ivan, M.: Preventing lunchtime attacks: fighting insider threats with eye movement biometrics. In: 22nd Annual Network and Distributed System Security Symposium (NDSS 2015) (2015)
30. Louis, W., Komeili, M., Hatzinakos, D.: Continuous authentication using one-dimensional multi-resolution local binary patterns (1dmrlbp) in ECG biometrics. IEEE Trans. Inf. Forensics Secur. **11**(12), 2818–2832 (2016)
31. Thao, T.P., et al.: Human factors in exhaustion and stress of Japanese nursery teachers: evidence from regression model on a novel dataset. In: 13th International Conference on Advances in Computer-Human Interactions (ACHI 2020), pp. 124–129. http://www.tpthao.com/pdf/2020_ACHI.pdf

32. Alejandro, R., Stephen, D., Ivan, C., Giulio, R.: STARFAST: a wireless wearable EEG/ECG biometric system based on the ENOBIO sensor. In: International Workshop on Wearable Mycro and Nanosystems for Personalised Health (pHealth 2008) (2008)
33. Neverova, N., et al.: Learning human identity from motion patterns. IEEE Access **4**, 1810–1820 (2016)