# Self-Enhancing GPS-based Authentication Using Corresponding Address

Tran Phuong Thao, Mhd Irvan, Ryosuke Kobayashi,
Rie Shigetomi Yamaguchi, and Toshiyuki Nakata

The University of Tokyo, Japan
Graduate School of Information Science and Technology
{tpthao, irvan}@yamagula.ic.i.u-tokyo.ac.jp
{kobayashi.ryousuke, nakata.toshiyuki}@sict.i.u-tokyo.ac.jp
yamaguchi.rie@i.u-tokyo.ac.jp

**Abstract.** Behavioral-based authentication is a new research approach for user authentication. A promising idea for this approach is to use location history as the behavioral features for the user classification because location history is relatively unique even when there are many people living in the same area and even when the people have occasional travel, it does not vary from day to day. For Global Positioning System (GPS) location data, most of the previous work used longitude and latitude values. In this paper, we investigate the advantage of metadata extracted from the longitude and latitude themselves without the need to require any other information other than the longitude and latitude. That is the location identification name (i.e., the address). Our idea is based on the fact that given a pair of longitude and latitude, there is a corresponding address. This is why we use the term *self-enhancing* in the title. We then applied text mining on the address and combined the extracted text features with the longitude and latitude for the features of the classification. The result showed that the combination approach outperforms the GPS approach using Adaptive Boosting and Gradient Boosting algorithms.

**Keywords:** Location-based Authentication, Lifestyle Authentication, Global Positioning System (GPS), Text Mining, Reverse Geocoding

## 1 Introduction

Japan aims to become the first country in the world to achieve Society 5.0 (after Society 1.0 (the hunting society), Society 2.0 (agricultural society), Society 3.0 (industrial society), and Society 4.0 (information society)) which is defined as: "A human-centered society that balances economic advancement with the resolution of social problems by a system that highly integrates cyberspace (virtual space) and physical space (real space)" according to the Government of Japan [1]. It sets a blueprint for a super-smart society with the support of Artificial Intelligent (AI) and cutting-edge technology. Let's consider an example of the payment system. In 1946, John Biggins invented a credit card which can be used to

replace paper money. In 2011, Google was the first company to launch a project of mobile wallet which can be used to replace physical cash and even credit cards. Nowadays, the cashless payment system becomes a recent trend and many digital wallet services appeared such as Apple Pay (from 2014), Google Pay (from 2015 as Android Pay and from 2018 as Google Pay), Rakuten Pay (from 2016), etc. The biggest challenge for such payment systems is how to authenticate (verify) the users. The current approach is to rely on the authentication of the mobile phones using PIN code, biometrics information (i.e., fingerprinting, iris, palm vein, etc), or multi-factor method.

**Motivation** Toward the construction for a smarter mobile-based authentication system, we have several research questions. First, several studies found that a large number of users do not lock their smartphones such as 33% by B. Dirk et al. [20], 29% by S. Egelman et al. [21], 48% by L. Fridman et al. [12], or even 57% by M. Harbach et al. [22]. So we ask the question: *Is there an additional mobile-based authentication method that can support the conventional method like using PIN code or biometrics information?*. Second, in the current cashless payment system, even though the users do not need to bring their credit cards, they have to bring their phones. So, we ask another question: *Is it "human-centered" enough? Whether a new authentication system can be done via smaller wearable devices such as smartwatches, RFID chips, or satellite sensors rather than the smartphone?*. Third, imaging the scenario that a person is on the way going to a coffee shop. Before he/she arrives, the coffee shop can predict that he/she will arrive 10 minutes later with a high probability, and prepares in advance his/her usual order, and will automatically subtract the charge from his account. The person then does not need to wait time for the order and payment process. So, the final question is: *Is it possible to authenticate and predict the location (for example, the coffee shop) that the users are likely going to?*

An idea that can answer these questions is using behavioral (or habit)-based information. There are very few studies focusing on it due to the challenge of how to decide behavioral information for authentication. Inspirited from L. Fridman et al. [12], GPS location history is the most 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". If we can construct a payment system in which the users do not need to bring anything even small wearable devices such as smartwatches or RFID chips (e.g., the data can be collected via satellite sensors) and which can replace the conventional biometrics authentication, it is a successful achievement for the Society 5.0 goal.

**Contribution** Most of the previous papers utilized longitude and latitude of GPS as the features in the classification machine learning models for the user authentication. In this paper, we propose an idea of extracting metadata of the GPS itself without the need to request any other information besides the GPS.

That is the location identification name (i.e., the address) which can be inferred from the longitude and latitude. We then applied text mining on the address and combined the extracted text features with the longitude and latitude for the learning features. We made an experiment to see how the combination approach is compared with the approach using only the GPS.

Considering it reasonability, it may raise the discussion that since the address can be inferred from a pair of longitude and latitude (i.e., using reverse geocoding), so whether the entropy of the address is the same as that of the GPS (in other words, whether the address gives no additional information to the GPS). However, we should remark that, for machine learning, a pair of two float numbers (i.e., longitude and latitude) and a string of text (i.e., address) are totally different and independent. Therefore, we hypothesized that the combination approach may add a certain amount of information to the approach using only GPS. Furthermore, the model using GPS and address, of course, can be improved if they can combine with other factors such as date times, indoor location history like wifi information, web browser log, etc. However, the goal in this paper is to make clear whether information (i.e., the GPS) along with the metadata extracted from that information itself (i.e., the address) can be helpful for the better classification model. We thus excluded other factors to make the comparison clean.

To evaluate how the feasibility of our hypothesis is, concretely, we collected 14,655 GPS records from 50 users. We extracted the corresponding address using reverse geocoding. We applied text mining on the address and obtained 3,803 text features using the term frequency-inverse document frequency. We performed multi-class classification using different ensemble algorithms on total of 3,805 scaled features including longitude, latitude, and text features. The result showed that the combination approach outperforms the approach using only the GPS with the Adaptive Boosting and Gradient Boosting ensemble algorithms.

**Roadmap** The rest of this paper is organized as follows. The related work is described in Section 2. The proposed idea is presented in Section 3. The experiment is given in Section 4. The discussion is mentioned in Section 5. Finally, the conclusion is drawn in Section 6.

## 2 Related Work

### 2.1 Multimodal Location-based Authentication

The term *multi-modal* (not multi-model) is used in biometrics authentication to indicate multiple biometric data; it is opposite with *unimodal* that uses only a single biometric data. L. Fridman et al. [12] collected behavioral data of four modalities collected 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 and Wifi. The authors proposed a

location-based classification method and showed that its performance is applicable to an actual authentication system. W. Shi et al. [14] proposed an authentication framework that enables continuous and implicit user identification service for a smartphone. The data is collected from four sensor modalities including voice, GPS location, multitouch, and locomotion. A. Alejandro et al. [15] analyzed their behavior-based signals obtained from the smartphone sensors including touch dynamics (gestures and keystroking), accelerometer, gyroscope, WiFi, GPS location and app usage. They proposed two authentication models including the one-time approach that uses all the channel information available during one session, and the active approach that uses behavioral data from multiple sessions by updating a confidence score. B. Aaron et al. [16] proposed a wallet repository that can store biometric data using multiple layers: biometric layer, a genomic layer, a health layer, a privacy layer, and a processing layer. The processing layer can be used to determine and track the user location, the speed when the user is moving using GPS data. R. Valentin et al. [17] presented the context of multimodal sensing modalities with mobile devices when the GPS, accelerometer, and audio signals are utilized for human recognition. They then discussed several challenges for modality fusion such as imbalance distribution when the GPS samples have to be correlated to accelerometer readings.

### 2.2 Learning from Metadata

We introduce related work that utilize metadata extracted from the main information to gain additional knowledge or the accuracy like our paper. T. Thao et al. [11] proposed a classification method of landing and distribution webpages from drive-by-download attack. A landing webpage is the original webpage that the victim visits while a distribution webpage is the final webpage that exploits malware in the redirection chain of the attack. While most of the previous papers analyze the characteristics around the landing and distribution webpages themselves (e.g., the HTML content, the pattern of the URL, etc.), this paper found the benefits when using the metadata from the webpages. Concretely, they extracted the registration information from the domain of the webpages called Whois. Whois contains registration name, organization, registration date, update date, expiration date, address, email, etc. They then applied text mining to the Whois documents and proved that the method can increase the classification accuracy. Similarly, Whois was also used to distinguish whether a homograph domain is registered by brand companies or by attackers [23]. A Castiglione et al. [18] analyzed the format extracted from a plain-text document using digital forensics and found that such kind of evidence in the forensic environment can provide a lot of valuable hidden information such as author name, organizational information of users involved, previously deleted text, and machine-related information. B. Duy et el. [19] proposed a classification method to extract the data from a PDF document. Besides using the information from the documents itself such as title, abstract, body-text, and semi-structure, the authors used metadata surround the document including publication information (i.e., authorship,

affiliation, bibliography), journal name, header and footer, and references. Their method can improve 9.7% of accuracy from the best performing algorithm.

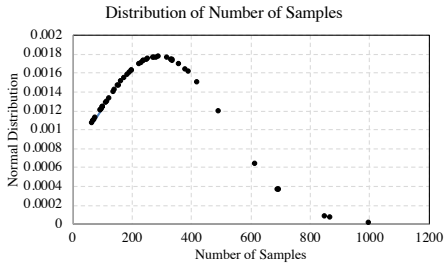
## 3 Our Method

### 3.1 Data Collection

The data is collected from 50 Internet users who installed a smartphone navigation application (named MITHRA (Multi-factor Identification/auTHentication ReseArch)) created by a project of the University of Tokyo. The application runs in the background and collects the GPS data including the longitude and latitude. The application works for both Android and iOS versions. The data is collected for four months from January 11th to April 26th in 2017. Although the GPS was collected from smartphones in this project, nowadays, GPS can be collected from smaller devices such as smartwatch or smartband.

The privacy agreement is showed during the installation process. The application installation is only done if the users accept the agreement. Even after the installation is finished, the users can choose to start participating or stop using the application anytime during the experimental period. The project was reviewed by the Ethics Review Committee of the Graduate School of Information Science and Technology, the University of Tokyo. No personal information such as age, name, race, ethnicity, income, education, etc. is collected. Only the email address is used as the user identity in the collected data. Finally, all 50 users agreed to participate in our project.

The total number of samples (i.e., GPS records including pairs of longitude and latitude) from the 50 users is 14,655. Each user has a different number of samples that range from 67 to 999. The distribution curve and distribution summary of the number of samples are described in Figure 1. The Kurtosis and Skewness scores are computed as 1.864 and 1.537, respectively. Both of the scores lie in the range $[-2, +2]$ which are acceptable for normal distribution [2, 4, 5].




| Metric | Value |
|---|---|
| Mean | 293.100 |
| Standard Error | 31.899 |
| Median | 237.000 |
| Standard Deviation | 225.561 |
| Kurtosis | 1.864 |
| Skewness | 1.537 |
| Range | 932.000 |
| Min | 67.000 |
| Max | 999.000 |
| Confidence Level (95.0 %) | 64.104 |

**Fig. 1:** Distribution Curve and Distribution Summary of The Number of Samples

### 3.2 Feature Selection

All the samples are unique for each user. Each GPS sample is a pair of a longitude and a latitude represented by two float numbers with 6 decimal digits, and can be positive or negative values.

**Address Extraction** Reverse geocoding is the process of reverse coding of a pair of latitude and longitude to a corresponding address. There are several reverse geocoding methods but we use Google's reverse geocoding [3] since it is the most reliable, fast, and stable. Especially, different from other services, Google's reverse geocoding can provide pinpoint addresses. For each request containing a pair of longitude and latitude, the service returns the addresses and the corresponding tags with the following values: *rooftop* (the most precise geocode), *range interpolated* (the address is interpolated between two precise points like intersections), *geometric center* (such as a polyline (street) or polygon (region)), and *approximate* (the approximated address). Since it is possible that multiple addresses with different tags are returned, we sorted each tag in the descending order of precision scale (rooftop $\rightarrow$ range interpolated $\rightarrow$ geometric center $\rightarrow$ approximate) and selected the first one (the most precise one).

**Address Text Mining** Given the address texts, the word tokenization is perform to build a dictionary of features. The texts are then transformed to feature vectors. $N$-grams of words which are the sequences of $N$ consecutive characters from the given words are counted. Let $T = \{t_1, t_2, \cdots, t_n\}$ denote the set of $n$ texts $t_i$ for $i = [1, n]$. Given a word $w$, the *Term Frequency* ($tf$) of $w$ in $t_i$ and the *Inverse Document Frequency* ($idf$) used for measuring how much information that $w$ provides or whether $w$ is common or rare in all the texts $T$ are computed as: $tf(w, t_i) = \frac{occ(w,t_i)}{|corpus(t_i)|}$ and $idf(w, T) = log(\frac{1+n}{1+df(w,T)}) + 1$ where $corpus(t_i)$ denotes the set of all tokenized unique words in $t_i$. $|corpus(t_i)|$ denotes its length. $occ(w, t_i)$ denotes the occurrence count of $w$ in $t_i$. $df(w, T)$ denotes the number of text $t_i \in T$ that contains $w$. The *Term Frequency Inverse Document Frequency* ($tf$-$idf$) is then computed to downscale weights of $w$ that may occur in many texts: $tf\text{-}idf(w, t_i) = tf(w, t_i) \cdot idf(w, T)$. Finally, each word tokenized from the texts is used as a feature with its $tf$-$idf$.

**Feature Scaling** While the longitude and the latitude range from $[-180, +180]$ and from $[-90, +90]$ respectively, the text features mostly range from (0, 1]. We thus use *Standard* (*Z-score*) scaling to normalize the features. For a given original feature $f_x$, the new standard for $f_x$ is computed as: $f'_x = \frac{f_x - \mu}{\sigma}$ where $\mu$ and $\sigma$ denote the mean and the standard deviation of the training samples, respectively. Standard scaling rescales the features by subtracting the mean and scaling the features to unit variance (the data distribution is centered around 0 and a standard deviation of 1).

### 3.3 Training

There are multiple users and each user has a different set of data. The conventional solution is the one-class classification in which each label corresponds to each user. However, when the number of users is large, the authentication performance can be low. Furthermore, if there is a new user participating in the system, it is not scalable since the classification should be trained again. Therefore, the one-class classification is transformed into a more lightweight approach known as the multi-class classification. Each user has a different classifier with binary classes representing whether or not a new sample belongs to that user.

**One-vs-rest Multi-class Classification** We use *One-vs-rest* strategy that consists in fitting one classifier per class (the samples belonging to the considered class are labelled as positive and the other samples of the other classes are marked as negatives). Suppose there are $q$ classifiers $c_i$ where $i \in \{1, \cdots, q\}$. Given a new sample $x$, the one-vs-rest approach classifies $x$ into the label $k$ such that: $\hat{y} = \text{argmax } c_k(x)$ where $k \in \{1, \cdots, q\}$ and all the classifiers are applied to $x$ and predict $k$ which has the highest confidence score. $\hat{y}$ represents the approximate score. Besides the computational efficiency and scalability when there is a new class, an advantage of this approach is the interpretability. The learner can gain knowledge about the class by inspecting its corresponding classifier.

**Ensemble Algorithms for One-vs-rest** Since our data contains a large number of samples (14,655) with a large number of features (3,805), instead of using traditional algorithms we use advanced *ensemble techniques*. The ensemble technique combines base estimators to produce one optimal predictive estimator with better performance using two approaches: boosting and averaging. In boosting approach, the base estimators are built sequentially. Each base estimator is used to correct and reduce the bias of its predecessor. We use two algorithms AdaBoost [6] (Adaptive Boost) and Gradient Boost [7]. In averaging approach, the estimators are built independently and their predictions are then averaged based on the aggregated results. The combined estimator reduces the variance. We use three algorithms: ExtraTrees [8], Bagging [9] (Bootstrap Aggregating), and Random Rorest [10].

### 3.4 Validation

**Stratified KFold** First, the data is shuffled. The numbers of samples of the classes are imbalanced, ranging from 66 to 999 (0.45% to 6.82 % of 14,655 samples). Using normal $k$-fold cross validation can lead to the problem that there may exist a class such that all the samples from the class belong to the test set, and the training set does not contain any of them. The classifier then cannot learn about the class. We thus used *Stratified k-fold* to deal with such imbalanced data. It splits the data in the train and the test sets and returns stratified folds made by preserving the percentage of samples for each class.

**Metrics** To evaluate the model, we measure the accuracy ($acc$), precision ($pre$), recal ($rec$), and F1 score ($F1$) with the following formulas: $acc = \frac{tp+tn}{tp+fp+fn+tn}$, $pre = \frac{tp}{tp+fp}$, $rec = \frac{tp}{tp+fn}$, $F1 = 2 \times \frac{rec \times pre}{rec+pre}$ where $tp, tn, fp, fn$ denote the true positive, true negative, false positive, and false negative values obtained from the confusion matrix, respectively. Accuracy is a good metric when the class distribution is similar; but for imbalanced classes F1-score is a better metric.

## 4 Experiment

We use Python 3.7.4 on a MacBook Pro 2.8 GHz Intel Core i7, RAM 16 GB. The addresses are extracted using Google reverse geocoding [3]. The machine learning algorithms are executed using *scikit-learn* 0.22 [13].

### 4.1 Parameter Setting and Data Pre-processing

Our experiment is designed with three different plans. Let $\#\mathsf{class}, n, \lambda, \eta$ denote the number of classes, the number of samples, the number of text features, and the number of combined features. The first plan is $\#\mathsf{class} = 10$, $n = 2,671$, $\lambda = 836$, $\eta = 838$. The second plan is $\#\mathsf{class} = 30$, $n = 9,068$, $\lambda = 2,514$, $\eta = 2,516$. The third plan is $\#\mathsf{class} = 50$, $n = 14,655$, $\lambda = 3,803$, $\eta = 3,805$. All the five ensemble algorithms are performed with three approaches (using the GPS only, using the address only, and using the combined GPS and address). For each algorithm, the number of base estimators is set to $n\_estimators = 100$. The feature scaling is necessary for the third approach. Since all the addresses are fortunately returned with the rooftop tags, we can obtain the most precise address. $k$ in the stratified $k$-fold is set to $k = 2$. Since the labels of the classes are independent categories represented in string type (i.e., 'user1', 'user2', etc.), we transformed the categorical labels to numerical values using *label encoding* which is the most lightweight and uses less disk space (compared with ordinal encoding or one-hot encoding). Since the data is imbalanced, to avoid the situation that F1 cannot be between precision and recall, we calculate the precision, recall, and F1 score for each label and find their average weight by the number of true instances of each class using the parameter $average =$ '$weighted$' in the *sklearn.metrics*. For the accuracy, this parameter is not necessary.

### 4.2 Result

The result is shown in Table 1. All the scores are reduced when the number of classes is increased. It is common for most of the multi-class classifications. The result shows that using the address only cannot beat using the GPS only. Let $\triangle$ denote the difference of F1 score between the combination approach and using only the GPS. The magnitude $\mid \triangle \mid$ is increased when the number of classes is increased. $\triangle$ is negative for ExtraTrees, Random Forest, and Bagging but is positive for AdaBoost and GradientBoost. It indicates that the combination approach outperforms the GPS approach using the ensemble boosting algorithms. Therefore, we suggest using the boosting algorithms for our approach.

**Table 1:** GPS-based, Address-based, and Combination Approaches

| Alg. | Score (%) | #Classes = 10 | | | | #Classes = 30 | | | | #Classes = 50 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **LL** | **A** | **LLA** | △ | **LL** | **A** | **LLA** | △ | **LL** | **A** | **LLA** | △ |
| Ada Boost | Acc | 99.36 | 99.21 | **99.40** | | 96.79 | 96.44 | **97.29** | | 95.09 | 95.04 | **95.67** | |
| | Pre | 99.43 | 99.28 | **99.45** | | 96.86 | 96.54 | **97.32** | | 95.26 | 95.25 | **95.80** | |
| | Rec | 99.36 | 99.21 | **99.40** | | 96.79 | 96.44 | **97.29** | | 95.09 | 95.04 | **95.67** | |
| | F1 | 99.37 | 99.23 | **99.41** | +0.04 | 96.78 | 96.44 | **97.28** | +0.50 | 95.11 | 95.07 | **95.70** | +0.59 |
| Gradient Boost | Acc | 99.74 | 99.51 | **99.81** | | 97.24 | 97.07 | **97.35** | | 95.55 | 95.36 | **95.91** | |
| | Pre | 99.74 | 99.52 | **99.82** | | 97.29 | 97.11 | **97.39** | | 95.63 | 95.46 | **96.05** | |
| | Rec | 99.74 | 99.51 | **99.81** | | 97.24 | 97.07 | **97.35** | | 95.55 | 95.36 | **95.91** | |
| | F1 | 99.74 | 99.51 | **99.81** | +0.07 | 97.21 | 97.06 | **97.35** | +0.14 | 95.55 | 95.36 | **95.94** | +0.39 |
| Extra Trees | Acc | 99.59 | 99.18 | 99.14 | | 98.14 | 97.14 | 97.35 | | 97.78 | 96.02 | 96.28 | |
| | Pre | 99.60 | 99.19 | 99.15 | | 98.16 | 97.17 | 97.37 | | 97.80 | 96.05 | 96.31 | |
| | Rec | 99.59 | 99.18 | 99.14 | | 98.14 | 97.14 | 97.35 | | 97.78 | 96.02 | 96.28 | |
| | F1 | 99.59 | 99.17 | 99.13 | −0.46 | 98.13 | 97.10 | 97.33 | −0.80 | 97.78 | 96.01 | 96.28 | −1.50 |
| Random Forest | Acc | 99.70 | 99.14 | 99.44 | | 97.93 | 96.85 | 97.06 | | 97.33 | 95.58 | 95.79 | |
| | Pre | 99.71 | 99.16 | 99.45 | | 97.95 | 96.85 | 97.08 | | 97.37 | 95.60 | 95.87 | |
| | Rec | 99.70 | 99.14 | 99.44 | | 97.93 | 96.85 | 97.06 | | 97.33 | 95.58 | 95.79 | |
| | F1 | 99.70 | 99.12 | 99.44 | −0.26 | 97.92 | 96.81 | 97.03 | −0.89 | 97.33 | 95.56 | 95.79 | −1.54 |
| Bagging | Acc | 99.59 | 99.44 | 99.55 | | 97.81 | 97.03 | 97.44 | | 97.07 | 95.61 | 96.23 | |
| | Pre | 99.60 | 99.45 | 99.57 | | 97.82 | 97.11 | 97.49 | | 97.13 | 95.65 | 96.27 | |
| | Rec | 99.59 | 99.44 | 99.55 | | 97.81 | 97.03 | 97.44 | | 97.07 | 95.61 | 96.23 | |
| | F1 | 99.59 | 99.44 | 99.55 | −0.04 | 97.78 | 97.03 | 97.43 | −0.35 | 97.07 | 95.59 | 96.22 | −0.85 |

(LL): the GPS approach, (A): the address approac, (LLA): the combination approach
Acc: accuracy, Pre: precision, Rec: recall

### 4.3 Analysis

We analyze how well the data points fit the data mean in boosting and averaging algorithms by measuring the variance and the bias. Let $y$ and $\hat{y}$ denote the true samples and the predicted samples. The variance is calculated as: $var(\hat{y}) = \frac{\sum(x_i - \bar{x})}{n-1}$ where $x_i$, $\bar{x}$, and $n$ denote the sample in the dataset, the sample mean, and the number of samples. The bias is calculated as: $bias(\hat{y}) = mean((mean(\hat{y}) - y)^2) - var(\hat{y}) - 0.01$ where $mean((mean(\hat{y}) - y)^2)$ is the sum of squared errors. We set the number of estimators varying $\{1, 2, \cdots, 10\}$. We choose the second plan (#class=30, $n = 9,068$) for the analysis. The graphs of variance and bias are showed in Figure 2 and 3. Ten curves GPS(Ada), Com(Ada), GPS(Gra), Com(Gra), GPs(Ext), Com(Ext), GPS(Ran), Com(Ran), GPS(Bag), and Com(Bag) represent the GPS approach or the combination approach using the five algorithms with the first 3 letters as the abbreviations.

In Figure 2, the variances are separated into two different groups. The first group that has higher variance score includes only the combination approach regardless of which algorithm is used. The other group includes the remaining curves which are only the GPS approach regardless of which algorithm is used. This indicates that the text features make the data samples spread out from the mean and from one another rather than the GPS. In both the groups, the aver-

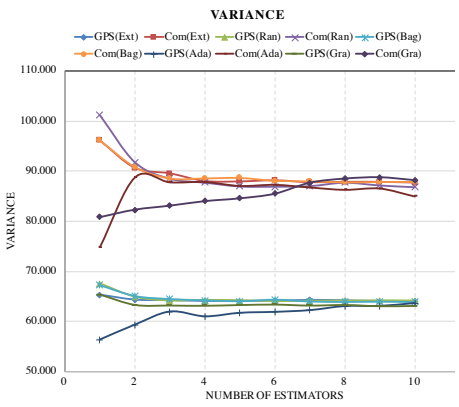



**Fig. 2:** Variance

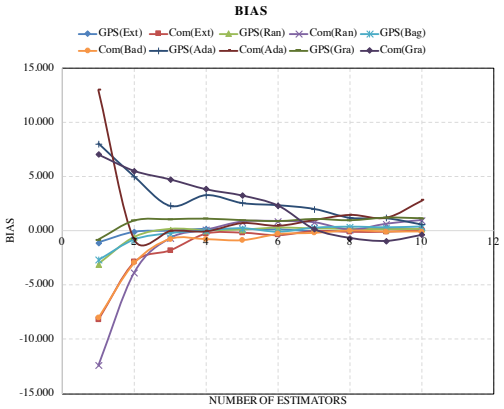



**Fig. 3:** Bias

aging algorithms give a higher variance than the boosting algorithms for the first estimator. After a number of estimators, the variances become convergent and constant in each group. Considering the bias in Figure 3, there is no separation like the variance. Also, the boosting algorithms give higher bias than the averaging algorithms for both the GPS and combination approaches. After a number of estimators, the bias becomes convergent to zero. It is reasonable when there is a trade-off between the variance and bias. In summary, while the GPS approach give low bias and lower variance, the combination approach gives low bias and higher variance since text features may contain more noise. Higher variance does not mean lower accuracy or F1. It just explains how the data samples spread out from the mean and from one another.

## 5 Discussion

In this section, we present our plans for future work and the threat model.

**Feature Improvement** First, the addresses were extracted in English. Since most of the collected GPS records are in Japan, applying semantic text mining on Japanese addresses may have some other meanings. For example, "Minato-ku" means the name of a ward in Tokyo, but in Japanese Kanji, it also means a port; and in fact, Minato ward is near to a port. It is possible since the Google reverse geocoding supports the Japanese language. Second, we used ($tf$-$idf$) as the text features in the current experiment. $tf$-$idf$ is used when the lengths of every text are too different, so it is possible that a word may occur much more times in long texts than shorter texts. However, the addresses have almost the same length, the word occurrence count or $tf$ may be enough for the text features. Third, the standard scaling was used for normalizing the features. Combining it with other feature scalings (e.g., Robust scaling) may give some extra information.

Four, the number of features extracted from the addresses is large, e.g., 3,805 features for 50 users. To improve the accuracy, the number of features can be reduced by selecting only the most important features. A threshold $\phi$ can be determined, then only the top features with $tf\text{-}idf \geq \phi$ are chosen. A large $\phi$ is not necessary since all the algorithms are compared in the same dataset and parameter setting, so it is fair.

**Threat Model and Other Behavioral Data** In such authentication system, the threat model deals with insider attack in which an authorized user tries to obtain the authentication from the other authorized users. Collusion attack (the authorized users in the system share their data together) and outsider attack (i.e., network eavesdropping, device hacking, etc.) are assumed in the model. Mitigating the attack can be done via improving the authentication accuracy. Proposing new approaches to achieve a high accuracy with a low false positive rate is the main goal in such authentication system. Besides extending the GPS collection, we launched a project to collect other behavioral data, i.e., calorie burning, distance, heart rate, ladders, sleep, speed, steps using wearable devices like smartwatch. A first result from this project can be found in [24]. Integrating different behavioral data for authentication is a future work.

## 6 Conclusion

In this paper, we proved that the GPS-based authentication can improve itself using its metadata. The address can be inferred from the GPS without the need to extract any other information. We collected 14,655 GPS records from 50 users. We extracted the address using the reverse geocoding. We applied text mining on the addresses and selected 3,803 text features using the tf-idf. Ensemble boosting and averaging algorithms are applied to the multi-class classification. The result showed that the combining approach outperforms the GPS approach using AdaBoost and GradientBoost.

## References

1. Cabinet Office, the Government of Japan, Society 5.0. Available: `https://www8.cao.go.jp/cstp/english/society5_0/index.html`
2. George D, Mallery P (2010) SPSS for Windows Step by Step: A Simple Guide and Reference (17.0 update). In: Allyn and Bacon, Boston.
3. Google Geocoding and Reverse Geocoding. Available: `https://developers.google.com/maps/documentation/geocoding/intro`
4. Thao TP, Sawaya Y, Nguyen-Son H, Yamada A, Kubota A, Sang T, Yamaguchi R (2020) Influences of Human Demographics, Brand Familiarity and Security Backgrounds on Homograph Recognition. In: arXiv:1904.10595. Available: `https://arxiv.org/abs/1904.10595`
5. Thao T, Sawaya Y, Nguyen-Son H, Yamada A, Kubota A, Sang T, Yamaguchi R (2020) Human Factors in Homograph Attack Recognition. In: 18th Int. Conf. on Applied Cryptography and Network Security (ACNS'20).

6. Zhu J, Hui Z, Saharon R, Thevor H (2009) Multi-class AdaBoost. In: Statistics and Its Interface, vol. 2, pp 349-360.
7. Friedman J (2001) Greedy Function Approximation: A Gradient Boosting Machine. In: The Annals of Statistics, 29(5):1189-1232.
8. Geurts P, Damien E, Wehenkel L (2006) Extremely randomized trees. In: Machine Learning, 63(1):3-42.
9. Louppe G, Geurts P (2012) Ensembles on Random Patches. In: Machine Learning and Knowledge Discovery in Databases, pp 346-361.
10. Breiman L (2001) Random Forests. In: Machine Learning, 45(1):5-32.
11. Thao T, Yamada A, Murakami K, Urakawa J, Sawaya Y, Kubota A (2017) Classification of Landing and Distribution Domains using Whois's Text Mining. In: 16th IEEE Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-17), pp. 1-8.
12. Fridman L, Steven W, Rachel G, Moshe K (2016) Active Authentication on Mobile Devices via Stylometry, Application Usage, Web Browsing, and GPS Location. In: IEEE Systems Journal, 11(2):513-521.
13. Scikit-learn. Available: `scikit-learn.org`
14. Shi W, Yang J, Jiang Y, Yang F, Xiong Y (2011) SenGuard: passive user identification on smartphones using multiple sensors. In: IEEE 7th Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob'11), pp 141-148.
15. Alejandro A, Aythami M, Vera-Rodriguez R, Julian F, Ruben T (2019) MultiLock: Mobile Active Authentication based on Multiple Biometric and Behavioral Patterns. In: Multimodal Understanding and Learning for Embodied Appl. (MULEA'19), pp 53-59.
16. Aaron B, Christopher D, Barry G, David K (2018) System and method for real world biometric analytics through the use of a multimodal biometric analytic wallet. In: US patent, US20180276362A1.
17. Valentin R, Catherine T, Sourav B, Nicholas L, Cecilia M, Mahesh M, Fahim K (2018) Multimodal Deep Learning for Activity and Context Recognition, in Interactive, Mobile, Wearable and Ubiquitous Tech. (IMWUT'18), article no. 157.
18. Castiglione A, Santis A, Soriente C (2007) Taking advantages of a disadvantage: Digital forensics and steganography using document metadata. In: Journal of Systems and Software, 80(5):750-764.
19. Duy B, Guilherme F, Siddhartha J (2016) PDF text classification to leverage information extraction from publication reports. In: Elsevier Journal of Biomedical Informatics, vol. 61, pp 141-148.
20. Dirk B, Shu L, Mitch K, Aaron S, Charles C, John D (2013) Modifying smartphone user locking behavior. In: 9th SOUPS'13 Symposium, article no. 10, pp 1-14.
21. Egelman S, Jain S, Portnoff R, Liao K, Consolvo S, Wagner D (2014) Are you ready to lock?, in ACM Conf. on Comp. and Comm. Security (CCS'14), pp 750-761.
22. Harbach M, Zezschwitz E, Fichtner A, Luca A, Smith M (2014) It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In: 10th USENIX Conf. on Usable Privacy and Security (SOUP'14), pp. 213-230.
23. Thao T, Sawaya Y, Nguyen-Son H, Yamada A, Omote K, and Kubota A (2019) Hunting Brand Domain Forgery: A Scalable Classification For Homograph Attack. In: 34th Int. Information Security and Privacy Conf. (IFIP Sec'19), pp. 3-18.
24. Thao T, Takahashi M, Shigeta N, Irvan M, Nakata T, and Yamaguchi R (2020) Human Factors in Exhaustion and Stress of Japanese Nursery Teachers: Evidence from Regression Model on A Novel Dataset. In: 13th Int. Conf. on Advances in Computer-Human Interactions (ACHI'20).