Address Authentication Method for Sustainable Social Qualification

Hosung Park ⊗, Seungsoo Nam and Daeseon Choi *

Department of Medical Information, Kongju National University, Chungnam 32588, Korea; hspark@kongju.ac.kr (H.P.); tfnk815@kongju.ac.kr (S.N.)
* Correspondence: sunchoi@kongju.ac.kr

Received: 10 February 2020; Accepted: 16 February 2020; Published: 25 February 2020

Abstract: This paper proposes an address authentication method based on a user’s location history. Address authentication refers to actual residence verification, which can be used in various fields such as personnel qualification, online identification, and public inquiry. In other words, accurate address authentication methods can reduce social cost for actual residence verification. For address authentication, existing studies discover the user’s regular locations, called location of interest (LOI), from the location history by using clustering algorithms. They authenticate an address if the address is contained in one of the LOIs. However, unnecessary LOIs, which are unrelated to the address, may lead to false authentications of illegitimate addresses, that is, other users’ addresses or feigned addresses. The proposed method tries to reduce the authentication error rate by eliminating unnecessary LOIs with the distinguishing properties of the addresses. In other words, only few LOIs that satisfy the properties (long duration, high density, and consistency) are kept and utilized for address authentication. Experimental results show that the proposed method decreases the authentication error rate compared with previous approaches using time-based clustering and density-based clustering.

Keywords: authentication; machine learning; clustering algorithm; location of interest

1. Introduction

With the rapid growth of smart devices, such as smartphones, wearable devices, and Internet of Things devices, numerous location-based services have been developed. This paper focuses on address authentication based on a user’s location history. Address authentication refers to the process of verifying that the user actually resides at the recorded home address or actually works at the recorded workplace address. Qualifications and benefits with regard to the address usually depend on the pre-registered address by the government. Therefore, the registered address must be consistent with the actual residence. However, there are many cases of inconsistencies and even unknown residences because the registered address is set by the individual moving-in report. False and omitted reports lead to the inconsistencies and there is no suitable way of discovering them. Frequent surveys by human actors may solve the inconsistency problem, but it would be very costly. An address authentication method only in case of need can be one of the solutions avoiding this waste of social cost and verifying qualification sustainably.

Address authentication can be applied to various fields such as surveys of real residence and personnel qualification (e.g., local recruitment, school admission, and real estate transactions). For examples, many cities implement a policy that real estate transactions are only open to local residents to avoid excessive real estate speculation. The condition of local resident may be a resident for more than one year. If a malicious user changed the registered address a year ago by false report for the real estate speculation, it is difficult to recognize it. For other examples, let us suppose that
a city is supporting nurturing and education costs for children more than other cities. These can be received simply by changing the address without moving to the city. Some people can change their address by false reports to get the qualifications such as local recruitment and school admission. Another application is supporting online user identification. Recently, online financial services have begun applying non-face-to-face user identification and the user can access services online without visiting the bank [1–4]. ISO/IEC 29115 [5] and 29003 [6] recommend verification of multiple personal information for non-face-to-face identification. Address authentication is one of the valid attributes for online user identification.

To the best of our knowledge, address authentication based on location history has not been studied except in our previous work [7]. This study is extension of the previous work, and we include more examples, explanations, related work, data, and experiments. In spite of the lack of related work, some studies are worth mentioning for address authentication. Studies on user authentication [8–14] identify a user by modeling movement patterns from location history. They utilize machine learning algorithms and neural networks for training the models. However, these studies cannot be applied to address authentication because the movement pattern models do not provide clues for addresses. The training and testing processes in machine learning are generally black boxes and address authentication features cannot be extracted directly from the models [15,16].

Studies on location of interest (LOI) or point of interest (POI) [17–25] are not authentication issues themselves but can be applied to address authentication. These studies utilize clustering algorithms to discover LOIs, that is, the locations where a user frequently goes and stays for long periods, according to the user’s location history. If an address is contained in one of the LOIs, the address can be regarded as authenticated. However, this approach carries the risk of false authentication because the clustering algorithms usually discover not only LOI related to the address but also many others. In other words, it misclassifies an illegitimate address (another user’s address or feigned address) as the user’s address if the illegitimate address is unfortunately contained in one of the LOIs. Furthermore, in density-based clustering, a cluster can be unexpectedly enlarged and the wider LOI worsens the misclassification problem.

This paper proposes an address authentication method based on location history. The goal of the proposed method is to reduce the authentication error rate compared with existing studies, that is, address authentication by LOI mentioned in the previous paragraph. To achieve this goal, we identify some properties of addresses to distinguish them from other LOIs, and present three conditions as follows.

- **Condition 1**: (long duration) location data on the address are kept for a long period of time.
- **Condition 2**: (high density) the address accounts for a large proportion of location data.
- **Condition 3**: (consistency) conditions 1 and 2 are repeated almost every day.

Only a few locations satisfying all three conditions are elected as LOIs. In other words, the proposed method tries to minimize the number of LOIs by eliminating LOIs that are not considered as an address. If an address is contained in one of the LOIs, then the address is authenticated. The reduction in the number of LOIs contributes to preventing wrong authentications. The proposed method exploits both a time-based clustering algorithm and a density-based algorithm to discover LOIs that satisfy the three conditions.

The performance of the proposed method is influenced by threshold parameters as well as the number of clusters (i.e., LOIs). We discover appropriate values of the threshold parameters via experiments based on real location histories in two environments: a metro area and a small area. In a small area, the parameters need to be more precisely controlled to avoid authentication errors because addresses are more likely to be located close to each other and the location histories are more likely to overlap each other.

The rest of this paper is organized as follows. Section 2 introduces background and related work. Section 3 explains the proposed method. Section 4 describes the experiments for evaluating the
proposed method. In Section 5, we discuss our experimental findings. Finally, Section 6 presents our conclusions and outline future work.

2. Related Work and Background

To the best of our knowledge, address authentication based on location history has not been studied yet except our previous study. However, some studies are worth mentioning and the proposed method utilizes these ideas. In this section, we introduce related work and the background of our study.

2.1. User Authentication by Movement Pattern Modeling

Some user authentication studies [8–14] identify users by modeling each user’s movement pattern from their respective location histories. They utilize machine learning algorithms and neural networks for training the models.

Mahbub and Chellappa [8] authenticate a user on mobile devices in a continuous manner using movement pattern modeling. Considering the movement of a human as a Markovian motion, a modified Hidden Markov Model is trained with location history. The trained model can generate a verification score from recent location data, and the user is authenticated using the verification score.

Other studies [9–13] perform user authentication using not only movement patterns, but also various data such as gait pattern, biometric information, motion pattern, network activities, and usage of applications. They generate a behavior model with multiple types of data, or generate a model for each data and use confidence scores simultaneously for user authentication. In the training phase, the location history is used in raw data or in the processed data (LOI) mentioned in next subsection. The trained models predict the user’s behavior and authenticate the user.

Movement pattern modeling is an effective way for user authentication. However, these studies cannot be applied to address authentication because the movement pattern models do not provide clues for addresses. They try to ascertain any distinguishable pattern from those of other users in the user’s location history. They are not interested in discovering the exact location of an address. Extracting features for address authentication using movement pattern modeling is also difficult because training and testing processes in machine learning are generally black boxes [15,16].

2.2. Discovering Location of Interest

Studies on LOI or POI [17–25] are not authentication issues themselves but can be applied to address authentication. These studies assume that LOIs, that is, personally meaningful places, are the locations where a user frequently goes and stays for long periods. They, therefore, discover LOIs from location history by using clustering algorithms. For address authentication, an address can be regarded as authenticated if the address is included in one of the LOIs.

Figure 1 shows the four cases of authentication results. We mark four addresses: the owner’s two addresses (stars) and two other users’ addresses (crosses). The owner denotes an legitimate user of the current location history. In the actual experiment, the address set is composed of the owner’s address and many other users’ addresses. Address authentication methods have to accept only the owner’s address and reject other users’ addresses. True accept is the case in which the owner’s address is legitimately authenticated, that is, the owner’s address is contained in one of the clusters. False reject is the case in which the owner’s address is improperly denied, that is, the owner’s address is located outside all clusters. True reject is the case in which the other user’s address is legitimately denied, that is, the other user’s address is located outside of all clusters. False accept is the case in which the other user’s address is improperly authenticated, that is, the other user’s address is contained in one of the clusters. For address authentication, we must minimize authentication errors, that is, false rejects and false accepts.
In this subsection, we introduce clustering algorithms for discovering LOIs and discuss the problems that arise when they are applied to address authentication. Clustering algorithms for address authentication are classified into two categories: time-based clustering and density-based clustering. There are more clustering algorithms but they are inappropriate for discovering LOIs. For example, partitioning-based clustering [26,27] can separate areas but cannot discover LOIs. This is because the technique divides all the data into the predefined number of parts; in other words, it cannot remove noise (unnecessary locations).

Figure 2 shows 60 days of real location history of a user. A point represents the current location of the user recorded every minute. Time-based clustering and density-based clustering are explained with examples based on this location history.

2.2.1. Time-Based Clustering

Time-based clustering [23–25] assumes that LOIs are the locations where a user stays for long periods at a time. It therefore clusters the stream of incoming location coordinates along with the time axis, and then drops the smaller clusters where little time is spent. In other words, it determines a new LOI whenever the user stays for a significant time at a location. Figure 3 shows the concept of time-based clustering. Suppose that the user moves from location A toward location B. In location A, a first cluster is generated with all the coordinates while the user stays within a certain distance (space threshold $s$) from the reference point. When the user is out of the current cluster, a new cluster is generated and a new reference point is elected. If the user stays at a cluster for more than a certain
time (time threshold $t$), the cluster becomes the LOI. On the contrary, if the user leaves a cluster before $t$, the cluster is eliminated. In Figure 3, more coordinates in a cluster denote that the user stays for a longer time at the cluster. According to the constraint, three clusters are eliminated and two clusters at locations A and B become LOIs. Location data in eliminated clusters are also removed as noises. The two thresholds, $s$ and $t$, directly affect the clustering results and could be identified via experiments.

![Figure 3. Example of time-based clustering.](image)

Figure 3. Example of time-based clustering.

Figure 4 shows the results of time-based clustering from the location history in Figure 2. The result is the best case in our experiments. Each circle represents a major location. A circle can contain multiple clusters (different colors) due to the characteristic of time-based clustering. Even though the user revisits the same location, it forms a new cluster. Therefore, there are many clusters in the result. The authors of [23] suggest a rule of merging clusters in the same location. However, the address authentication results are same after merging clusters.

![Figure 4. Clustering results from time-based clustering.](image)

Figure 4. Clustering results from time-based clustering.

Time-based clustering is a simple and reasonable algorithm for discovering LOIs. However, in address authentication, it carries the risk of false authentications because it may discover not only an address location but also many others. In other words, it misclassifies an illegitimate address (any other address) as the user’s address if the illegitimate address is unfortunately contained in one of the LOIs. Furthermore, time-based clustering finds locations even where a user stays for a long period just for once. Suppose that a forger stays near the address location for one day only for authentication fraud. Time-based clustering may accept the forger’s address authentication.
2.2.2. Density-Based Clustering

Location history data are often represented by points collected at certain intervals. Density-based clustering [17–22] considers that LOIs are locations with high density, that is, they have denser location data than other places. It does not consider the order of location data along with the time axis. In DBSCAN [17], if a point has more than \( n \) neighbors within the radius of a circle \( r \), the point becomes a core point. High density, that is, many neighbors, denotes that the user averagely stays at that location for a long duration regardless of the number of visits. A cluster is formed with all connected core points and their neighbor points. Later studies expand this idea for their own purposes. EDBSCAN [18] can detect clusters with varied densities with multiple density parameters. DVBSCAN [19] can handle the density variations that exist within the cluster. In other words, clusters have hierarchy. Clustering algorithms with density variations are useful ideas for many applications. However, DBSCAN is more suitable for our purpose because address authentication requires to discover few clusters, which is most likely to indicate the address. It is unnecessary to detect diverse clusters.

In density-based clustering, the boundary of each cluster is relatively precise because the cluster shape is flexible, unlike time-based clustering. In time-based clustering, the cluster shape is fixed at a circle. Density-based clustering therefore can better reflect the overall data. However, a cluster can be unexpectedly enlarged because the size of the cluster is not predefined.

Figure 5 shows the results of density-based clustering from the location history in Figure 2. In comparison with Figure 4, each cluster is larger and irregular. Even though the average number of clusters in density-based clustering is less than that in time-based clustering, it is still too many due to the same reason with time-based clustering. Furthermore, the average cluster size is larger. If clusters become larger, the probability of false reject increases because addresses of other users are more likely to be contained in the clusters. The authentication results therefore are adversely affected by wide clusters as well as many clusters. Specifically, the results are affected by the sum of all cluster areas.

3. Proposed Method

The purpose of the proposed method is to minimize authentication errors for more accurate address authentication. We also utilize clustering algorithms as seen in studies on LOI. However, Figures 4 and 5 show that the approach used in existing studies carries the risk of authentication errors.

For address authentication, the ideal case is to discover only an address cluster. The address cluster is defined as a cluster containing the user’s address among multiple LOIs. However, overly strict conditions raise the risk that the address cluster is also eliminated (false reject). Therefore, instead of selecting only one cluster, the proposed method uses the following design principle.
• Minimize the number of estimated address clusters (EACs) by eliminating other locations of interest (LOIs) that are not considered as an address cluster.

With the design principle, the proposed method discovers EACs and eliminates other LOIs based on three conditions pertaining to the address: long duration, high density, and consistency. The process of discovering EACs reduces the number of clusters compared with existing studies on LOIs while carefully avoiding the risk of eliminating the address cluster. This approach finds the properties of a particular place and eliminates other places based on the said properties. It can be applied to discovering other meaningful locations. Looking ahead, if meanings and properties of many LOIs are discovered, they can be used in various fields.

3.1. Overview

The overall procedure for discovering the EACs is shown in Figure 6. Each procedure relates to each property of the address. After three procedures, the remaining clusters that satisfy the conditions of the address are elected as EACs.

The first procedure, long-duration check, discovers the places where the user stays for a long period at a time using time-based clustering. The procedure eliminates unnecessary locations such as bus stations, grocery shops, and streets. However, other unnecessary locations, such as business trips, hospitals, shopping centers, and vacation places, can remain because the user typically spends a long time at such locations.

The second procedure—high-density check—discovers the places that are characterized by high density among the places remaining after the first procedure is complete. The procedure eliminates locations pertaining to unusual events such as medical treatment, shopping, and short business trips. However, other locations caused by long-duration events such as long-duration business trips and travel can still be retained because their density is also high.

The third procedure—consistency check—discovers the places that consistently satisfy the above two conditions. In other words, the procedure discovers the locations where the user stays for many days among the places remaining after the second procedure is complete. People usually stay at an
address almost every day except in special cases. The last condition is different from the long-duration and high-density checks. For example, a gym where the user exercises for one hour a day can pass a consistency check but is filtered in the long-duration check or high-density check. By contrast, a place where the user stayed for a week on a business trip can pass long-duration and high-density checks but cannot satisfy the consistency condition. The remaining places finally become EACs.

If the user’s address is contained in one of the EACs, the address is authenticated. The procedures cannot guarantee perfect discovery of address location and elimination of unnecessary locations. Suppose that a user was admitted to the hospital for 50 days of 60 days, i.e., the whole period of the location history. The address authentication may fail. However, cases such as this example are very unusual.

3.2. Collecting Location History

For collecting real location history, we developed a smartphone application and recruited participants. The application records current location with a timestamp every minute for 24 h a day. The average duration of collecting location histories is approximately 63 days. Smartphones can measure the current location through GPS, cellular networks, and Wi-Fi networks. The application selects GPS when the user is outdoors because the location from GPS is generally more accurate than other locations. When the user is indoors, the application selects a network-based location because GPS do not work indoors. In Appendix A, Figure A1 shows the samples of smartphone application screen (location collector) and location history data.

Location information from location systems including GPS and networks contains uncertainty due to errors and variations in the measured phenomena. Even though the user is located in the exact same place, the recorded location can be little different. The coordinate switches from the address also do not exactly match with the coordinates from the location history. A small margin is needed to identify the relation between them. Therefore, a cluster is better than a single point to express a location and it is also appropriate for address authentication.

3.3. Discovering Estimated Address Clusters

3.3.1. First Procedure: Long-Duration Check

To satisfy the first condition (long duration), the proposed method exploits time-based clustering. Other location data, which are not included in the clusters, become noises and are eliminated because they do not satisfy the first condition with regard to long duration.

A pseudocode for a long-duration check is presented in Procedure 1. It is similar to the algorithm in TBC [23], the well-known time-based clustering algorithm, and is slightly simplified. The location history of a user consists of series of location data (timestamps, latitudes, and longitudes). If a new coordinate $\text{loc}$ is located within the space threshold $s$ from a reference point of the current cluster $\text{cl}$, the $\text{loc}$ is joined to $\text{cl}$ (lines 2–3). The distance is calculated as a Euclidean distance. If not, a new cluster is generated and $\text{loc}$ becomes a new reference point (line 9). Before clearing the current cluster $\text{cl}$, if $\text{cl}$ persists beyond the time threshold $t$, then $\text{cl}$ is added to candidate clusters $\text{cls}$ (lines 5–6). In other words, the current cluster satisfies the first condition. All other clusters, which do not persist for more than $t$, are eliminated (line 8). The remaining clusters $\text{cls}$ become candidate address clusters and location data in the $\text{cls}$ are used as the input in the next procedure.
Procedure 1: Long-duration check (time-based clustering)

Input: series of location data \( locs \),
distance threshold \( s \),
time threshold \( t \)

Output: candidate clusters \( cls \)

1. for all \( locs \) do
2.   if \( \text{distance (cluster } cl, \text{ location } loc ) < s \) then
3.     add \( loc \) to \( cl \)
4.   else
5.     if \( \text{duration (} cl \text{ )} > t \) then
6.       add \( cl \) to \( cls \)
7.   end if
8.   clear \( cl \)
9.   add \( lo \) to new \( cl \)
10. end if
11. end for all
12. return \( cls \)

In terms of performance, the first procedure has two implications: a decrease in algorithm complexity and an increase in authentication accuracy. The decrease in algorithm complexity is caused by the difference between the first and second procedures. The computational complexity of the second procedure is high and exponentially increases in proportion to the number of nodes, that is, location data. In contrast, the complexity of the first procedure is low because it only performs as many comparison operations as the number of nodes. The first procedure eliminates a number of unnecessary locations and leads to a considerable decrease in the complexity of the second procedure. The increase in authentication accuracy can be inferred from the comparison with the case using only the second procedure, i.e., density-based clustering. We assumed that eliminating unnecessary locations via the first procedure can contribute to the next procedure to discover more precise clusters. In a location history, many location data are located near the address, and these location data can make the address cluster wider. For example, in a real case in our experiments, if a user frequently goes to a market very close to the address, density-based clustering can discover a cluster containing both the address and the market. In the proposed method, the first procedure eliminates unnecessary location data near the address, such as moving path and even market, and then the second procedure discovers a more precise address cluster that does not contain the market.

3.3.2. Second Procedure: High-Density Check

After time-based clustering, the remaining clusters can overlap with each other because different locations from the previous cluster always trigger a new cluster. Even though the user revisits the same location, time-based clustering forms a new cluster. For the high-density check of each location, we need to merge clusters at the same location. Density-based clustering could be a simple solution for this task. The location data remaining after the first procedure are clustered by density-based clustering, and each location is naturally merged as a cluster. Note that a cluster can be unexpectedly enlarged in density-based clustering. However, the proposed method can prevent the problem because unnecessary noises, except major places, are eliminated in the first procedure. The flexible clusters identified by density-based clustering contribute to the accuracy of address authentication because the area of each user’s address cluster is different.
In density-based clustering, existing studies \cite{17–22} use the radius parameter \( r \) and number of neighbors \( n \). However, we choose the local density parameter \( l \) instead of \( n \) because the total number of all location data varies from user to user. If the total number of all location data is different, the proper \( n \) also needs to be changed. The rate parameter \( l \) however is constant. A local density is calculated as the number of neighbors divided by the number of original location data (and not the input of the second procedure).

After density-based clustering, the procedure checks the density of each merged cluster. The density of each cluster is calculated by dividing the number of location data included in the cluster by the total number of original location data. Clusters that do not satisfy the density threshold \( d \) are eliminated from candidate address clusters. For example, in case \( d \) is 5\% and the total number of original location data is 10,000, clusters with less than 500 location data are eliminated. The remaining clusters are used as the input in the next procedure.

The pseudocode for high-density check is presented in Procedure 2. The procedure consists of two parts: density-based clustering (lines 1–23) and high-density check (lines 24–28). For density-based clustering, we exploit DBSCAN \cite{17}, the well-known density-based clustering algorithm. As mentioned in 2.2.2, DBSCAN algorithm is more suitable for the purpose of address authentication. If location data \( \text{loc} \) do not have enough neighbors (if local density is less than the local density threshold \( l \)), the location data are labeled as noise (lines 4, 19, and 20). By contrast, \( \text{loc} \) has enough neighbors, and it becomes a core point and a new cluster \( \text{cl} \) is generated (lines 4–5). According to the rule of DBSCAN, the cluster \( \text{cl} \) includes \( \text{loc} \)'s neighbors \( \text{ns} \) if the neighbors are not already contained in other clusters (lines 8–11). The procedure also allocates neighbors \( \text{ns} \) to a seed set \( \text{seeds} \) and checks the core point condition of each neighbor (lines 6, 7, and 13). If a neighbor is a core point, the neighbor's neighbors are also added to the current cluster \( \text{cl} \) and allocated to the seed set \( \text{seeds} \) (lines 13 and 14). This process is repeated until all connected core points and their neighbors are contained in the cluster \( \text{cl} \) (lines 7–17). If the density of the generated cluster \( \text{cl} \) is lower than the cluster density threshold \( d \), \( \text{cl} \) is eliminated. Consequently, only clusters that satisfy the condition of high density remain in the candidate clusters \( \text{cls} \). The candidate address clusters \( \text{cls} \) are used as the input in the next procedure.
Procedure 2: High-density check (density-based clustering)

Input: location data after long-duration check \( \text{locs} \),
- local density threshold for clustering \( l \),
- radius threshold for clustering \( r \),
- the number of original location data \( \text{numOrigLocs} \),
- cluster density threshold \( d \)

Output: candidate clusters \( \text{cls} \)

```plaintext
forall \( \text{locs} \) do
  if label (loc) == unprocessed then
    \( \text{ns} \) = loc’s neighbor locations within \( r \)
    if \( (|\text{ns}| / \text{numOrigLocs}) > l \) then
      add loc to new cluster \( \text{cl} \)
      seed set \( \text{seeds} \) = \( \text{ns} \)
      forall \( \text{seeds} \) do
        if label (seed) == Noise then
          add seed to \( \text{cl} \)
        else if label (seed) == unprocessed then
          add seed to \( \text{cl} \)
          \( \text{ns} \) = seed’s neighbor locations
          if \( (|\text{ns}| / \text{numOrigLocs}) > l \) then
            add \( \text{ns} \) to \( \text{seeds} \)
        end if
      end for
      add \( \text{cl} \) to \( \text{cls} \)
    end if
  else
    label (loc) = Noise
  end if
end forall
forall \( \text{cls} \) do
  if \( (|\text{locs in cl}| / \text{numOriginLocs}) < d \) then
    delete \( \text{cl} \) from \( \text{cls} \)
  end if
end forall
return \( \text{cls} \)
```

3.3.3. Third Procedure: Consistency Check

To verify the consistency condition of each candidate cluster, the proposed method utilizes the number of days that the user stayed in the cluster. Each cluster’s consistency ratio is calculated by dividing the number of days included in the cluster by the total number of days in the user’s location history. If a cluster’s consistency ratio does not exceed a threshold \( c \), then the cluster is eliminated. For example, in case \( c \) is 50% and the total number of days in the user’s location history is 60, clusters with less than 30 days are eliminated. Note that previous procedures already eliminate locations where the user does not stay long enough. The remaining clusters finally become the EACs.

The pseudocode for the consistency check is presented in Procedure 3. Each location data in a cluster \( \text{cl} \) have a timestamp as well as a coordinate. The procedure can identify the days the user stays in the cluster \( \text{cl} \) from the timestamps of the location data in \( \text{cl} \). If the consistency ratio of \( \text{cl} \), that is, the
rate of the number of days in $cl$, does not exceed the consistency threshold $c$, and $cl$ is eliminated (lines 2–4). After the consistency check, the output $cls$ becomes an EAC.

**Procedure 3: Consistency check**

| Input: | candidate clusters after high-density check $cls$, the number of days in original location data $numDays$, consistency threshold $c$ |
|--------|--------------------------------------------------------------------------------------------------|
| Output: | estimated address clusters $cls$ |
|        | forall $cls$ do                                                                                |
|        | 1. $days = \text{the number of days in } cl$                                                   |
|        | 2. if $(days / numDays) < c$ then                                                               |
|        | 3. delete $cl$ from $cls$                                                                       |
|        | 4. end if                                                                                        |
|        | 5. end forall                                                                                   |
|        | 6. return $cls$                                                                                  |

### 3.4. Address Authentication Based on EACs

If an address is contained in one of the EACs, then the address is authenticated. It is, however, difficult to compare the address and EACs because the shapes of the EACs are flexible. Therefore, the proposed method adds the address coordinate to the location data at the start of the second procedure and checks whether they are eliminated or not after all the procedures are complete. Because the address coordinate has no timestamp, it cannot be added in the first procedure. If the address does not satisfy the first condition, it is naturally eliminated in the second procedure because no other location data exist near the address coordinate.

The other manner of authenticating an address is by using the local density threshold $l$. After discovering the EACs, the proposed method stores them. When authenticating an address, the proposed method checks the number of the address’s neighbors in an EAC. In other words, if the local density (in an EAC) of the address is higher than the local density threshold $l$, we can consider that it belongs to the EAC and authentication is successful. This condition is the same for the finding core node in the second procedure. The results of the two authentication approaches are identical, but the second is more resource efficient because it can reuse the discovered EACs.

The process of discovering EACs reduces the number of clusters compared with the existing clustering approaches while carefully avoiding the risk of eliminating address clusters. The reduced clusters can help achieve our goal of decreasing the authentication error rate. Note that the misclassification problem worsens when the number of clusters increases and each cluster grows, as mentioned above.

Figure 7 shows an example of discovering EACs. The first figure shows the real location history of a user during 60 days. After the long-duration check, we see that many location data are eliminated. Multiple clusters overlap each other in the same location. Different colors denote different clusters. After the high-density check, overlapped clusters are merged as a cluster and some clusters are eliminated. The second procedure eliminates clusters with low density, that is, clusters with few location data. After the consistency check, only one cluster remains and becomes the EAC. The third procedure eliminates clusters that have few days. Because the user’s address is contained in the EAC, the address is authenticated. The number of EACs can vary with the life pattern of the user. In this example, the EAC to the left is the user’s home and that to the right is the university. The two EACs may be discovered because the user is a university student. This study focuses on home address authentication but the proposed method can authenticate the user’s office address too because the properties of the two addresses are very similar.
The order of the three procedures is also important. If the long-duration check is performed after the high-density check, a cluster can be unexpectedly enlarged because of the characteristics of density-based clustering. Moreover, time-based clustering is difficult because many location data are missing in the stream of location data with the time axis. The consistency check must be performed last because it can check the consistency after cluster generation.

4. Experimental Evaluation

4.1. Experimental Settings

As no public test set of location history with corresponding addresses is available, we construct our own set of tests. We conduct two types of experimental environments: a metro area and a small area. For the metro area, we gather 27 location histories and addresses from 27 people (10 females and 17 males) living in Seoul, Korea. The participants are office workers who use Android smartphones, with ages in the twenties to forties. Twenty-four participants used smartphones manufactured by Samsung, including Galaxy S8, Galaxy S7, Galaxy S6 Edge, Galaxy Note 8, Galaxy Note 5, and Galaxy Note 4; the remaining 3 participants used smartphones manufactured by LG, namely, the G5, V30, and sXcreen. In a small area, we gather 26 location histories and addresses from 26 undergraduates (12 females and 14 males) living near Kongju National University, Kongju, Korea. The participants who use Android smartphones are in their twenties except 2 males in their thirties. Eighteen participants used smartphones manufactured by Samsung, such as Galaxy S8, Galaxy S7, Galaxy S6, Galaxy Note 5, and Galaxy Note 4; the remaining 8 participants used smartphones manufactured by LG, namely, the G3, V10, and V20. Figures 8 and 9 show the addresses in the metro area and small area, respectively. The average duration of the location histories was approximately 63 days. The location data were collected every minute during the day. For each user’s test (for each location history), the subject denotes the legitimate user’s address and others denotes other users’ addresses. The address authentication methods have to accept only the subject and reject others. The experimental results are presented as the average values of all users’ address authentications. Location histories are collected via an application on Android smartphones, and the procedures for address authentication are implemented and performed in Python on a Linux server.
Table 1 shows the parameter ranges. For example, when \( s \) is 50 m and \( t \) is 3 h, time-based clustering generates a cluster only when the user stays for more than 3 h within a 50-m area in the first procedure. When \( r \) is 10 m and \( l \) is 0.5\%, density-based clustering generates a cluster including location data whose local density is more than 0.5\% within a 10-m area in the second procedure. When \( d \) is 10\%, clusters whose density is less than 10\% are eliminated in the second procedure. When \( c \) is 50\% and the number of days in the user’s location history is 60, clusters with less than 30 days are eliminated in the third procedure.
The performance of the proposed method is influenced by threshold parameters as well as the number of clusters. For example, if clusters are too small, the proposed method has a higher risk of false rejection because the address of the user is more likely to be out of the clusters. By contrast, if clusters are too large, the probability of false acceptance increases because addresses of other users are more likely to be contained in the clusters. We discover appropriate values of the threshold parameters via experiments based on real location histories.

### 4.2. Experimental Results for the Metro Area

The performance of the proposed method is compared with the results of previous studies on TBC [23] and DBSCAN [17]. Table 2 shows the address authentication results of three approaches in the best cases. The performances are described in terms of three error factors: false positive rate (FPR), false negative rate (FNR), and average error rate (AER). FPR means the rate of falsely accepting illegitimate addresses and FNR means the rate of falsely rejecting legitimate addresses. AER is the average value of FPR and FNR. The term # clusters denotes the average number of clusters for 27 location histories. The best cases are chosen based on AER. Note that our goal is to reduce the authentication error rate.

#### Table 2. Address authentication results for the metro area.

| Method       | FPR  | FNR  | AER  | # Clusters |
|--------------|------|------|------|------------|
| TBC          | 0.0188 | 0.0370 | 0.0279 | 24.21 |
| DBSCAN       | 0.0296 | 0.0 | 0.0148 | 13.82 |
| Proposed method | 0.0108 | 0.0 | 0.0053 | 6.81 |

The parameter values in the best case are summarized as follows. For TBC, the threshold parameters $s$ and $t$ are 250 m and 4 h, respectively. For DBSCAN, the parameters $l$ and $r$ are 0.7% and 200 m, respectively. For the proposed method, the parameters $s$, $t$, $l$, $r$, $d$, and $c$ are 200 m, 3 h, 0.8%, 30 m, 24%, and 65%, respectively.

Both DBSCAN and the proposed method have no false rejects, that is, FNRs are 0. However, the FPR of the proposed method is lower than that of other approaches because the average number of clusters is less than half of them. Even though the average number of clusters in DBSCAN is less than that of TBC, the FPR of DBSCAN is higher than that of TBC because of the size of each cluster. The average cluster size in DBSCAN is the highest. As mentioned in the Introduction, the presence of wide and many clusters worsens the misclassification problem in address authentication.

The primary objective of the simulations is to identify the proper values of parameters. However, Seoul is a big city, and a dataset of 27 people is too small for the metro area. Addresses are located far apart from each other, and location histories rarely overlap. The average distance among the addresses is almost 10 km. Thus, cases of authentication errors are fundamentally very small. We therefore acknowledge that our dataset for the metro area is limited with regard to evaluating the authentication accuracy and identifying proper parameter values for the proposed method. Accordingly, we construct a new dataset for the small area.
4.3. Experimental Results for the Small Area

Figure 9 shows a map of the small area including only a university and its environs. The addresses are located close to each other. The location histories frequently overlap because the participants are university students. Therefore, the probability of authentication errors is higher than that in the metro area.

Table 3 shows the address authentication results of the three approaches in the best cases for the small area. The performances are described in terms of three factors: FPR, FNR, and AER. The number of clusters is the average number of clusters for 26 location histories. The term area denotes the average cluster areas for all users, and each user’s area is the sum of all cluster areas. The best cases are chosen based on AER.

| Method       | FPR   | FNR   | AER   | # Clusters | Area (m²) |
|--------------|-------|-------|-------|------------|-----------|
| TBC          | 0.0473| 0.0385| 0.0429| 113.42     | 320533    |
| DBSCAN       | 0.0415| 0.0385| 0.04  | 11.2308    | 112703    |
| Proposed     | 0.02  | 0.0   | 0.01  | 3.7558     | 4522      |

The parameter values in the best case are summarized as follows. For TBC, the parameters $s$ and $t$ are 60 m and 5 h, respectively. For DBSCAN, the parameters $r$ and $l$ are 20 m and 0.8%, respectively. For the proposed method, the parameters $s$, $r$, $l$, $d$, and $c$ are 60 m, 1 h, 20 m, 1%, 21%, and 20%, respectively. We cannot try all combinations because the number of parameters is large. Therefore, we found a local optimum for a parameter, fixed it, and then found another local optimum for another parameter. To approximate the global optimum as much as possible, if an unusual result is found, we return to the related parameters and repeat the experiment.

Compared with results for the metro area, all authentication errors increase due to the environments. The AER of TBC increases by 1.5% from 2.79% to 4.29%. The AER of DBSCAN increases by 2.52% from 1.48% to 4%. Finally, the AER of the proposed method increases by 0.47% from 0.53% to 1%. The AER of the proposed method is slightly higher in comparison with that of TBC and DBSCAN. TBC constructs many clusters and DBSCAN constructs large-scale clusters. Therefore, the area of the proposed method is much smaller than that of TBC and DBSCAN. In a stricter environment, that is, when more users exist in a small area, the differences in AER between the proposed method and other methods may be bigger.

Figures 10–12 show the relation between the AER and the cluster area in TBC, DBSCAN, and the proposed method, respectively. The cluster area and error rate vary with the parameters in each method.

The three graphs show that FNR decreases as the cluster area increases. A higher cluster area raises the possibility that the subject is contained in the clusters (true accept). However, FNR becomes very low and stable if the cluster area exceeds a certain size. This is because the address cluster can satisfy the conditions of long duration (TBC), high density (DBSCAN), and the three procedures (the proposed method). The difficult part is eliminating unnecessary clusters that may cause a false accept. In TBC and DBSCAN, FPR consistently increases according to the cluster area. The higher cluster area also raises the possibility that others are included in one of the clusters (false accept). The graph of the proposed method shows more complex aspects because the final clusters (EACs) are generated by three procedures rather than one condition. However, FPR of the proposed method is less than other methods because the cluster area is much small. We assume that many clusters cause authentication errors. These graphs show that not only many but also wide clusters adversely affect the authentication results. According to the applications, the proposed method can prioritize FPR or FNR. These experiments use AER, the average error rate, for identifying the proper parameters.
Figure 10. Error rate according to the cluster area (TBC).

Figure 11. Error rate according to the cluster area (DBSCAN).

Figure 12. Error rate according to the cluster area (Proposed method).
4.4. Experimental Results on Precision

Authentication errors are more likely to occur when addresses are located close to each other. In other words, the proposed method is limited as it cannot distinguish between addresses located too close to each other. This is caused by the inherent uncertainty of location information during the collection of location history. This section describes the precision of the proposed method, meaning the minimum distance between addresses that the proposed method can distinguish. In small area of Figure 9, the minimum distance among addresses is ~19 m. The distances are measured between the centers of two buildings. The proposed method have 2% FPR and we find out that most errors are occurred around four addresses located very close to each other.

To investigate the precision of the proposed method, we exclude two addresses from an experiment as shown in Figure 13. Now, the minimum distance among addresses is ~43 m and only one address is located in a small block separated by roads.

Table 4 shows the address authentication results of the three approaches in the best cases for the small area after excluding two addresses. For TBC, the parameters \( s \) and \( t \) are 40 m and 5 h, respectively. For DBSCAN, the parameters \( r \) and \( l \) are 15 m and 0.6%, respectively. For the proposed method, the parameters \( s, t, r, l, d, \) and \( c \) are 40 m, 3 h, 20 m, 0.8%, 24%, and 30%, respectively.

The proposed method have no authentication errors in the new environment. In comparison with the results for the small area, the number of clusters and average cluster areas decrease, and parameters become stricter. The error rates of TBC and DBSCAN also decrease slightly but still have a certain amount of errors. In the experiment, we can say that the precision of the proposed method is small unit in block or distance of 43 m.
Specifically, the observation for precision can be little hasty due to the small amount of location history data. We mention this problem again in “Discussion” section. However, overall experiments can show the differences and tendencies of authentication accuracy and precision compared with other approaches.

5. Discussion

Address authentication requires that some conditions be followed while constructing the data set. The location history must be collected for more than a certain period of time. Two weeks of a business trip cannot affect four months of location history, but it can greatly affect one month of location history. We consider that more location histories are needed for precise experiments, such as the address for each building and address for each block. Moreover, we assume that the proposed method is superior to other methods in a stricter environment, a small area with more location histories. However, it is difficult to collect data because location history is very sensitive and is regarded as personal information.

We assume that the actual residence is the location at which the user stays the longest and most frequently. This is our definition of the actual residence and the conditions of the proposed method as well as an assumption. People normally return to and sleep at the residence almost daily, and it is difficult to distinguish the residence with other features. In our experimental results, the third condition, that is, “how many days the user stays in there”, was 20%. In contrast, the special cases for which the user stays other places over 80% are very unusual. It is a more special case if the duration of location history is longer. Consequently, we regard the proposed method as practical for the following two reasons. First, it is very special case that the user stays other places for very long periods (over 80%). Second, it is difficult to identify the residence by using other features.

This paper proposed algorithms for address authentication. To apply the algorithms to mobile devices or online services, more considerations such as data security, privacy, and computational complexity are required. These considerations exceed scope for this paper but would be a good focus for future work. For examples, for privacy and data security, if the proposed method operates on smartphones, a possible approach is to only provide the results of address authentication rather than the location history.

We collect location histories by smartphone application for experiments. Another way is to use map service. For example, Google map users can obtain their own location histories by JSON files. Address authentication method can use the location histories in times of need without the data collection period. If a map service provider is allowed to share a user’s location history, cooperation with the provider can be a good way for service deployment.

6. Conclusions and Future Work

This paper proposed an address authentication method based on a user’s location history. To reduce the authentication error rate compared with existing studies using LOI, the proposed method identified three properties of addresses that distinguish them from other LOIs. Among LOIs, we discovered estimated address clusters that satisfied the properties of long duration, high density, and consistency. By eliminating unnecessary clusters, that is, inappropriate LOIs, the proposed method can reduce the risk of misclassification, which leads to authentication errors. The experimental results showed that the proposed method decreased the authentication error rate (1%) compared with authentication approaches using time-based clustering (4.29%) and density-based clustering (4%) alone. The proposed method also provide appropriate values of the threshold parameters based on real location histories.

Our future work will apply this methodology of discovering address location to other meaningful places. Other places may contain other properties pertaining to the locations. If we can find the properties, we can also discover the locations and meanings of the places. Furthermore, we will try to collect more location histories for more precise experiments.
Author Contributions: Conceptualization, H.P. and D.C.; methodology, H.P.; software, H.P. and S.N.; validation, H.P. and D.C; formal analysis, H.P. and S.N.; investigation, H.P. and S.N.; resources, D.C.; data curation, S.N.; writing—original draft preparation, H.P.; writing—review and editing, D.C.; visualization, S.N.; supervision, D.C.; project administration, D.C.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2016-0-00173, Security Technologies for Financial Fraud Prevention on Fintech) and the research grant of the Kongju National University in 2019.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Availability of Data

![Figure A1. Screenshot of smartphone application and sample of location history.](image)

The datasets generated and/or analyzed during the current study are not publicly available due to the privacy issue. According to the participant agreement, we cannot reveal location history and address data. Therefore, we provide samples of smartphone application screen (location collector) and location history in Figure A1. For convenience of participants, application contains some Korean words.

References

1. Kiljan, S.; Simoens, K.; Cock, D.D.; Eekelen, M.V.; Vranken, H. A survey of authentication and communications security in online banking. *ACM Comput. Surv. (CSUR)* 2017, 49, 61.

2. Kiljan, S.; Vranken, H.; van Eekelen, M. Evaluation of transaction authentication methods for online banking. *Future Gener. Comput. Syst.* 2018, 80, 430–447.

3. Mohammed, M.M.; Elsadig, M. A multi-layer of multi factors authentication model for online banking services. In Proceedings of the International Conference on Computing, Electrical and Electronic Engineering (ICCEE), Khartoum, Sudan, 26–28 August 2013; pp. 220–224.

4. Nagaraju, S.; Parthiban, L. Trusted framework for online banking in public cloud using multi-factor authentication and privacy protection gateway. *J. Cloud Comput.* 2015, 4, 22.
5. Information security—Security techniques—Entity authentication assurance framework. Standard ISO/IEC 29115:2013; International Organization for Standardization: Geneva, Switzerland, 2013.

6. Information technology—Security techniques—Identity proofing. Standard ISO/IEC TS 29003:2018; International Organization for Standardization: Geneva, Switzerland, 2018.

7. Park, H.; Kwon, D.; Nam, S.; Choi, D. POSTER: Address Authentication Based on Location History. In Proceedings of the Asia Conference on Computer and Communications Security (ASIACCS), Incheon, Korea, 4–8 June 2018; pp. 845–847.

8. Mahbub, U.; Chellappa, R. PATH: person authentication using trace histories. In Proceedings of the IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 20–22 October 2016; pp. 1–8.

9. Fridman, L.; Weber, S.; Greenstadt, R.; Kam, M. Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. *IEEE Syst. J.* 2017, 11, 513–521.

10. Shila, D.M.; Srivastava, K.; O’Neill, P.; Reddy, K.; Sritapan, V. A multi-faceted approach to user authentication for mobile devices—Using human movement, usage, and location patterns. In Proceedings of the IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 10–12 May 2016; pp. 1–6.

11. Rachuri, K.K.; Hossmann, T.; Mascolo, C.; Holden, S. Beyond location check-ins: Exploring physical and soft sensing to augment social check-in apps. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), St. Louis, MO, USA, 23–27 March 2015; pp. 123–130.

12. Buthpitiya, S.; Dey, A.K.; Griss, M. Soft authentication with low-cost signatures. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), Budapest, Hungary, 24–28 March 2014; pp. 172–180.

13. Shi, E.; Niu, Y.; Jakobsson, M.; Chow, R. Implicit authentication through learning user behavior. In Proceedings of the International Conference on Information Security (ISC), Boca Raton, FL, USA, 25–28 October 2010; pp. 99–113.

14. Ye, Y.; Zheng, Y.; Chen, Y.; Feng, J.; Xie, X. Mining individual life pattern based on location history. In Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM), Taipei, Taiwan, 18–21 May 2009; pp. 1–10.

15. Kulesza, T.; Burnett, M.; Wong, W.K.; Stumpf, S. Principles of explanatory debugging to personalize interactive machine learning. In Proceedings of the 20th International Conference on Intelligent User Interfaces, Atlanta, GA, USA, 29 March–1 April 2015; pp. 126–137.

16. Liu, S.; Wang, X.; Liu, M.; Zhu, J. Towards better analysis of machine learning models: A visual analytics perspective. *Vis. Inform.* 2017, 1, 48–56.

17. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.

18. Ram, A.; Sharma, A.; Jalal, A.S.; Agrawal, A.; Singh, R. An enhanced density based spatial clustering of applications with noise. In Proceedings of the IEEE International Advance Computing Conference, Patiala, India, 6–7 March 2009; pp. 1475–1478.

19. Ram, A.; Jalal, S.; Jalal, A.S.; Kumar, M. A density based algorithm for discovering density varied clusters in large spatial databases. *Int. J. Comput. Appl.* 2010, 3, 1–4.

20. Zhou, C.; Frankowski, D.; Ludford, P.; Shekhar, S.; Terveen, L. Discovering personally meaningful places: An interactive clustering approach. *ACM Trans. Inf. Syst. (TOIS)* 2007, 25, 12.

21. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W.Y. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 791–800.

22. Gupta, A.; Miettinen, M.; Asokan, N.; Nagy, M. Intuitive security policy configuration in mobile devices using context profiling. In Proceedings of the International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing, Amsterdam, The Netherlands, 3–5 September 2012; pp. 471–480.

23. Kang, J.H.; Welbourne, W.; Stewart, B.; Borriello, G. Extracting places from traces of locations. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* 2005, 9, 58–68.

24. Birant, D.; Kut, A. ST-DBSCAN: An Algorithm for Clustering Spatial-temporal data. *Data Knowl. Eng.* 2007, 60, 208–221.
25. Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; Thalmann, N.M. Time-aware point-of-interest recommendation. In Proceedings of the 36th International Conference on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland, 28 July–1 August 2013; pp. 363–372.

26. Ashbrook, D.; Starner, T. Learning significant locations and predicting user movement with GPS. In Proceedings of the Sixth International Symposium on Wearable Computers, Seattle, WA, USA, 7–10 October 2002; pp. 101–108.

27. Lloyd, S. Least squares quantization in PCM. IEEE Trans. Inf. Theory 1982, 28, 129–137.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).