WestVirginiaUniversity
THE RESEARCH REPOSITORY @ WVU

Graduate Theses, Dissertations, and Problem Reports

2015

A Safety Support System for Children's Antiloss

Sai Ram Nellutla

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Nellutla, Sai Ram, "A Safety Support System for Children's Antiloss" (2015). *Graduate Theses, Dissertations, and Problem Reports*. 6304.
https://researchrepository.wvu.edu/etd/6304

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

A Safety Support System for Children's Antiloss

Sai Ram Nellutla

Thesis submitted
to the Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science

Yanfang Ye, Ph.D, Chair
Roy Nutter, Ph.D.
Vinod Kulathumani, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2015

Keywords: Safety Support System, Children's Antiloss, Data Mining, Secure Area, Secure Path, Indoor Positioning

© 2015 Sai Ram Nellutla

Abstract

A Safety Support System for Children's Antiloss

Sai Ram Nellutla

In the recent past, crimes against children and the number of the missing children have been stayed at high. It is a tragic disaster for a family if their child is missing. Feeling safe about their children is very important for the parents. Therefore, there is an urgent requirement for safety support systems to prevent crimes against children and for antiloss, particularly when the children are on their own, such as on the ways to and from schools. Thanks to the highly development of telecommunication and mobile technologies, preventive devices such as child ID kits, family trackers have come to light. However, they haven't been impressive solutions yet as they only track current positions of the children and lack of intimations for the parents when their children are under potential dangers. In this thesis, a data mining framework is introduced, in which secure areas and secure paths of the children are learned based on their location histories. When the system predicts the children to be potentially unsafe (e.g., in a strange area or on a strange route), automatic reports will be sent to their parents. Furthermore, an indoor positioning method utilizing Bluetooth is also proposed. Based on the android platform, a prototype of the application for both children and parents is developed incorporating with the proposed techniques in this thesis.

Acknowledgements

Success doesn't come alone. It definitely needs a proper guidance and support. In my case I got 110%. First of all, I would like to express my gratitude to Dr. Yanfang Ye for her continuous guidance and all the support she gave throughout the process of this thesis, especially during the crunch times. Her thoughtful feedback always aimed me towards the correct direction. She is a wonderful person and professor. I would like to thank Dr. Vinod Kulathumani and Dr. Roy Nutter for being a part of my committee, the courses they taught, and for their time despite their overwhelming schedule. They were always inspiring and the knowledge they shared will definitely help me in future.

I thank my father Ram Chander Rao Nellutla, my mother Sarala Nellutla, and my brother Sandeep Nellutla for their support and love. Their efforts are responsible for what I am now.

I thank my boss, Tim Mitchem, for funding me throughout this journey. He is a father figure, excellent boss, and a good friend.

I thank Suma Chaganti, Ajay Krishna Teja Kavuri, Shashank Sabniveesu, and Nithin Uppalapati for being with me all the time. You guys made my journey in WVU memorable. Last but not the least I thank all my friends in WVU for their quality time with me.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Motivation

In the recent past, crimes against children and the number of the missing children have been stayed at high. [7]. According to the national center for missing and exploited children, it's reported that about 6,34,367 children are missing each year (i.e., around 1,737 children a day) [39]. Figure 1.1 shows the numbers of missing children in U.S. in the past decade. The cases of missing children involve abduction, getting lost, etc [54]. It is a tragic disaster for a family if their child is missing. Feeling safe about their children is very important for the parents. Therefore, there is an urgent requirement for safety support systems to prevent crimes against children and for anti-loss, particularly when the children are on their own, such as on the way to and from schools.

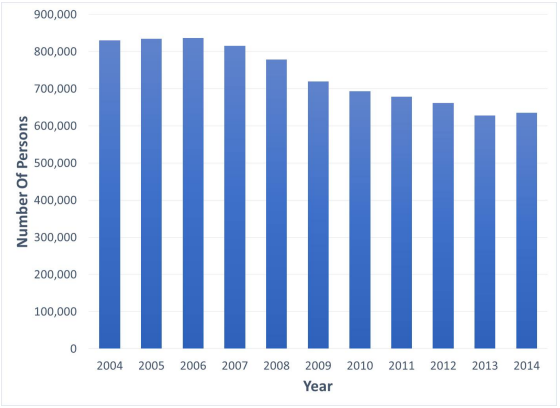


FIGURE 1.1: Numbers of missing children in U.S. in the past decade. *Source*: FBI statistics on missing children [7].

To prevent the children from missing or to help finding the children who are missing, if the locations of the children can be traced, soon after receiving the missing reports, we can quickly locate the children and take necessary measures. Thanks to the highly

development of telecommunication and mobile technologies, mobile devices that can automatically identify the geographic coordinates are becoming common [54]. As an outcome, users' position and location information can be widely used to unlock many favorable circumstances for emerging applications. There are many location tracking systems and researches for elderly [29, 41, 51], but few of them specially for children. Actually, those elderly tracking systems and researches [29, 41, 51] mainly aim at providing location or motion aware emergency detection to assist care providers or relatives to support the elderly people living alone. However, the requirement of children tracking is quite different from the elderly tracking.

Lately, few research efforts have been particularly conducted on children's tracking and their anti-loss [16, 17, 30]. In these safety support systems [16, 17, 30], each child with a cellular phone terminal can find out the phone numbers of safe volunteers who keep tracking them on the way to and from school. This kind of safety support systems aims to support volunteers who are involved in maintaining safety of the community. Ye et al. proposed a novel solution to learn the children's life patterns based on their location histories [54]. In their developed system *Soter*, safe regions and safe routes of the children are learned at the cloud side. When the children are under potential dangers, their parents will receive automatic notifications from the cloud. Their work is the first attempt applying data mining techniques for children's safety using smart devices. Several commercial location tracking devices such as child ID kits, family trackers have come to light. These buddy trackers could only display subtle locations on the map and provide a facility to the family members to watch one another. They have confined successes in providing valuable information on children's safety to parents. For example, at present day's busy world, there is an absolute chance for parents involving deeply in their works while tracking their children every minute is impossible. In other words, those systems are not smart enough: they only track the current positions of the children, but lack of intimations for the parents when their children are under potential dangers.

1.2 Research Objective

To design and develop a safety support system to prevent crimes against children and for their anti-loss, in this thesis, a data mining framework is introduced, in which secure areas and secure paths of the children are learned based on their location histories. When the system predicts the children to be potentially unsafe (e.g., in a strange area or on a strange route), automatic reports will be sent to their parents. Though current smart devices can be used in tracking the children's location outdoor using GPS or cell

towers, they may fail in indoor positioning. There are several chances for a child to be lost indoors at crowded places, such as children sliding away from parents in large indoor places (e.g., mall) when they are involved in shopping or any other work. Thus an accurate indoor positioning method which can help to track children's indoor positions is also very important. In this thesis, an indoor positioning method utilizing Bluetooth is proposed. Based on the android platform, a prototype of the application for both children and parents is developed incorporating with the proposed techniques in this thesis.

The major contributions of this thesis are summarized as follows:

- *An effective data mining framework for children's secure area and secure path detection:* To predict the potential dangers of the children, their life patterns are learned based on their location histories. First, effective clustering algorithms are applied for secure area detection; Then, efficient frequent sequential pattern mining methods are adopted to detect the secure paths for the children. A performance comparison of clustering algorithms and sequential pattern mining algorithms while finding secure areas and secure paths is done.
- *Accurate indoor positioning method:* To extend the work in [54], an indoor positioning method is further investigated in this thesis. Bluetooth with the help of the data provided by the property management is used for indoor positioning.
- *User friendly application for both children and parents:* Based on the android platform, the prototype of the application is designed and developed for both children and parents. When the system predicts the children to be potentially unsafe (e.g., in a strange area or on a strange route), automatic reports will be sent to their parents. Moreover, handy buttons in the parent console are designed to easily access communication with their children or cops, whenever immediate action is required.

1.3 Organization of the Thesis

The rest of this thesis is organized as follows:

- *Chapter* 2 discusses the related work.
- *Chapter* 3 describes the background concepts in this thesis.

- *Chapter* 4 introduces secure area detection. This chapter first defines the problem, and then introduces the basic concept of clustering analysis followed by the clustering techniques used in this thesis for children's secure area detection. Finally, a real daily case is studied and experimental results are also analyzed.
- *Chapter* 5 introduces secure path detection. This chapter first defines the problem, and then explains the basic concept of frequent sequential pattern mining followed by the efficient techniques used in this thesis for children's secure path detection. Real daily cases are studied and experimental results are also analyzed in this chapter.
- *Chapter* 6 introduces an indoor positioning method proposed in this thesis.
- *Chapter* 7 presents the system design and development.
- *Chapter* 8 concludes the work in this thesis and discusses the future work.

Chapter 2

RELATED WORK

In this section, we will discuss the related research efforts on location history mining and the methods used for indoor positioning.

2.1 Location History Mining

There have been several research works on location history mining based on geographic data, which can be categorized into two groups: mining individual location history and mining multiple users' location histories.

2.1.1 Mining Individual Location History

Mining individual location history has been an active research topic in recent years [18, 23, 27, 32, 33, 55]. The goals of individual location history mining include predicting user's current position, discovering places of interest, learning traveler patterns, user's daily movement, and discovering life patterns, etc. Ye et al. [55] proposed a life pattern mining method using *LP-normal* forms from Global Positioning System (GPS) data. Kang et al. [18] extracted significant places from the traces of coordinates and used clustering approach in detecting significant locations. Paek et al. [32] estimated the current position of the user using cell-ID sequence matching. Krumm et al. [23] used Wi-Fi signal strength to detect the motion of the user. Montoliu et al. [27] used GPS, Global System for Mobile Communications (GSM), Wi-Fi, accelerometer sensors to discover the positions of some time span, and then adopted grid-based clustering in defining the places of interest. Patterson et al. [33] used *Expectation-Maximization* to learn Bayesian models of a traveler moving through an urban environment.

2.1.2 Mining Location History of Multiple People

Researches based on applications to be developed by tracking location histories of multiple people are also performed [15, 22, 31, 47, 59]. Krumm and Horvitz [22] proposed a method which used drivers' destination and current location history to predict the destination of the person. Eagle and Pentland [31] used mobile phones to retrieve data of hundred users with which they presented ethnographic studies of device usage, relationship inference, individual behavior modeling, and group behavior analysis. GEOWHIZ [15], HITS-based inference model [59], and City Voyager [47] were few of the proposals based on recommender systems that used memory and model based collaborative filtering, tree based hierarchical graph to recommend most visited places and most accepted places of interest by users.

Apart from the above discussed works, this thesis aims to learn the life pattern of the children based on their location histories, such as children traveling to schools and back homes. When the children are under potential dangers, the system will send reports to their parents. To achieve this, children's secure areas and secure paths will be learned first.

2.2 Indoor Positioning

There are several chances for a child to be lost indoors at crowded places, such as children sliding away from parents in large indoor places (e.g., mall) when they are involved in shopping or any other work. Thus an accurate indoor positioning method which can help to track children's indoor positions is also very important. GPS is not a good resource for indoor positioning, because of its weak signals in a closed area. As alternatives, wireless positioning sensors, such as Radio Frequency Identification (RFID) [5], Ultra-Wide Band (UWB) [40], Wi-Fi [52], and Bluetooth, play significant roles in calculating the indoor position accurately.

Gao et al. [9] used particle filters to fuse signals from Wi-Fi, indoor maps and inertial navigation system to enable the localization of the user with mobile devices. But in the current urbanized generations, there are multiple crowded places and chance of visiting already visited place is less prioritized. Therefore, fetching indoor maps for all places manually is not feasible. Radaelli and Jensen [37] defined a fully organic indoor positioning system by integrating different sensors, such as Wi-Fi, Bluetooth, and video cameras. InTraTime [36] calculated the travel times indoor using location history with minimal set up costs. Shen et al. [43] explored the potential of RFID in indoor mapping and navigation.

To extend the work in [54] which proposed a novel solution to learn the children's life patterns based on their location histories, in this thesis, we will investigate an accurate indoor positioning method using Bluetooth.

Chapter 3

BACKGROUND CONCEPTS

3.1 Preliminary

3.1.1 Location Point

A location point is the measure of location, mostly represented by the *latitude* and *longitude* values of the position where a child is residing. The location point can be tracked by GPS or through cell-ID attained from GSM communication module. In this thesis, a location point p is denoted as

$p : \langle ChildID, T(TimeStamp), Lat(Latitude), Long(Longitude), cellID(s) \rangle$.

3.1.2 Location History

Location history is basically a set of location points that are collected by the application within a time window, which is

$P = \{p_1, p_2, p_3, ..., p_n\}$,

where p_i is the location point.

3.1.3 Trajectory

Trajectory is a sequence of location points generated from the children's location histories, denoted as

$Tray =< p_1, p_2, ..., p_n >$,

where $p_i \in P$, $p_{i+1}.T > p_i.T$ and $p_n.T - p_1.T \leq \Delta T$ $(1 \leq i < n)$.

3.1.4 Stay Point

Stay point is a geographic region where a child stays for certain time span. Intuitively, it basically occurs on two occasions:

- A child is stationary, that is the child stays at certain location for certain amount of time. This usually occurs at school, home, cinemas, etc.
- A child is in movement within specific perimeter, that is the child though moving lies around the same place. This usually occurs when a child is playing in the playground, shopping in a mall, visit some tourist place, etc.

As shown in Figure 3.1, there are two stay points generated from the trajectory $Tray = \langle p_1, p_2, p_3, ..., p_n \rangle$. The points marked in green represent normal points where the child wanders around and the red spots represent the stay points. *Stay point 1* represents the intuitive reading of the location where the child has been stationary for a while (occasion 1). *Stay point 2* is the center of location points $\{p_5, p_6, p_7, p_8\}$ representing the child wanders around within a certain geospatial range of a period (occasion 2).

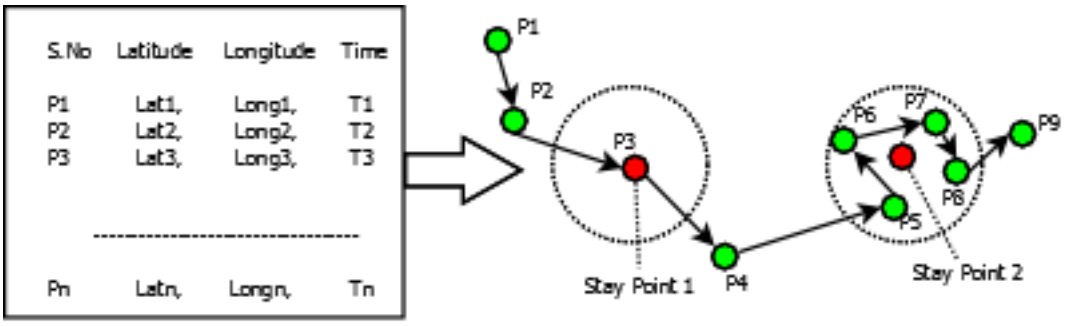


FIGURE 3.1: Location data and stay points. *Source*: Based on Zheng, Zhang, Xie and Ma [59].

A stay point can be a real or virtual location point. Given a trajectory $Tray = \langle p_j, p_{j+1}, ..., p_k \rangle$, where $\forall 1 \leq j < k \leq n$, $|Distance(p_j, p_k)| \leq \delta_D$ and $p_k.T - p_j.T \geq \delta_T$, a stay point is denoted as $sp = (Lat, Long, cell - IDs, arvT, levT)$, where

$$sp.Lat = \sum_{i=j}^{k} p_i.Lat/|Tray|, \tag{3.1}$$

$$sp.Long = \sum_{i=j}^{k} p_i.Lngt/|Tray|, \tag{3.2}$$

$$sp.CellIDs = \{p_i.CellID | p_i \in Tray\}, \tag{3.3}$$

respectively represent the average latitude, longitude and the cellIDs of the given trajectory $Tray$, while $sp.arvT = p_j.T$ and $sp.levT = p_k.T$ stand for the child's arrival and leaving times on sp.

The pseudo code for stay point detection is shown in Algorithm 1 [27, 59].

Algorithm 1 StayPoint_Detection(P, δ_D, δ_T)

Input: Location history $P = \{p_1, p_2, p_3, ..., p_n\}$, distance threshold δ_D and time threshold δ_T.
Output: A set of stay points, $SP = \{sp_i | 0 \leq i \leq n\}$

$i = 0, n =\mid P \mid$; //the number of points in location history
while $i < n$ **do**
  $j : j + 1$
  while $j < n$ **do**
    $dist = Distance(p_i, p_j)$; //calculate distance between two points
    if $dist > \delta_D$ **then**
      $\triangle T = p_j.T - p_i.T$; // calculate time span between two points
      if $\triangle T > \delta_T$ **then**
        $S.coord = ComputeMeanCoord(\{p_k \mid i \leqq k \leqq j\})$
        $SP.insert(S)$
      end if
      $i := j; break;$
    end if
    $j := j + 1;$
  end while
  return SP
end while

Irrespective of what kind of source the location is tracked from, it's highly improbable that the two readings are the same at the same place. In other words, even if a child stays at constant place, the probability of receiving same coordinates is very low. We may get a number of readings at several places where a child is stationary or where he/she hangs around for a while like home, school, etc. In order to eradicate this issue, we introduced secure area which is described in the following.

3.1.5 Secure Area

A secure area is a cluster of stay points, represented by the center of the group of stay points, which is the place that a child spends significant amount of time and/or visits frequently, such as home, school, playground, mall, etc.

3.1.6 Secure Path

Secure paths are the routes where a child travels in between the secure areas. They are technically the sequential patterns that occur frequently or most likely (i.e., the paths that a child moves from one secure area to other secure area, such as from home to school, and home to play ground). As shown in Figure 3.2, a child is assumed to be in the safe path when he/she is traveling from one secure area to other. Deviation in the path will be notified to his/her parent.

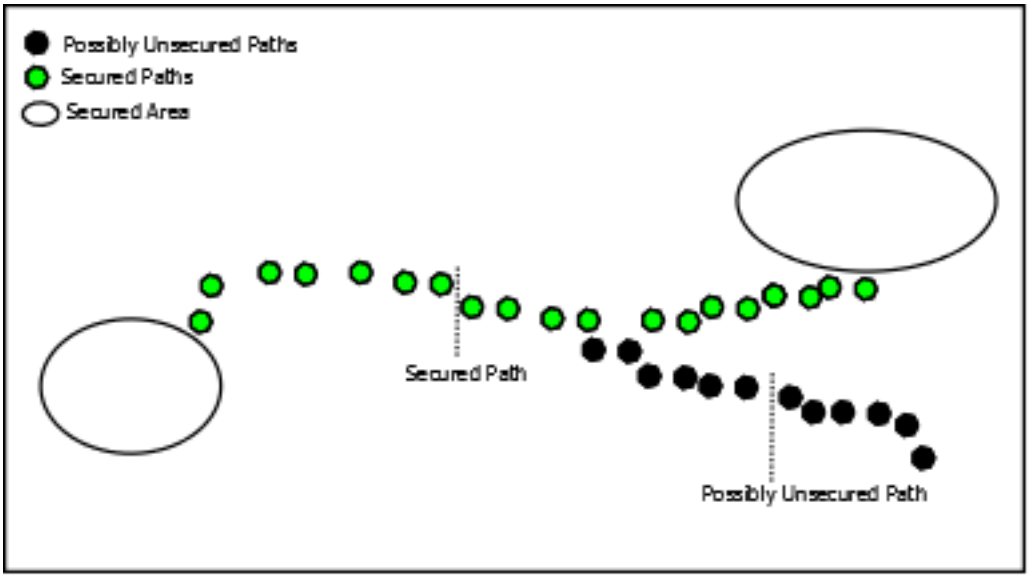


FIGURE 3.2: Example of secure path.

3.2 Life Patterns of the Children

There is a close relation between children's everyday lives and geographic locations. Taking this advantage, we can find out their general life patterns and regularities. The status of a child, whether he/she is safe or not can be anticipated based on their corresponding chronicles of location. Based on the archives formed from a child's regular traveling secure areas and secure paths which are accountable to access whether he/she is secure or in plausible threat. One such case would be when the current position of a child is not even in the least vicinity of attained secure paths and secure areas, then the child is considered to be in danger. In [54], Ye et al. gave the novel notions of children's life patterns.

- (*Arriving at the secure area.*) Parents may always worry whether their children have reached school, home or any other expected place safely or not. An example of an automatic notification would be: "*John arrived school at 8:00 AM in the*

morning". If he seems to be in the presence of school at midnight, then it will be unusual and this should be notified as a threat.

- (*Traveling on the secure path.*) Apart from worrying about reaching secure areas safely, parents also care about their safety on the ways by their own, such as "John arrives school in the following path from home: *home* → *8th Street* → *University Ave.* → *Riverview Drive* → *Evansdale Drive* → *school*".

- (*Staying at an unfamiliar area.*) A reliable application should notify when the children stay at peculiar or odd areas. Such places can be potentially dangerous for children as people pertaining to attempt crimes at that areas. And hence, if a child is noticed to be in such areas more than some threshold time, he/she will be considered to be in threat.

- (*Traveling on a strange Path.*) If a child travels through a route different from his/her regular route between secure areas, then he should be considered to be potentially in threat. One such example would be a child who is supposed to go to school from the above specified route instead takes *home* → *8th Street* → *Beechurst Ave.* → *Monangohela Blvd.* → *Chaplin Road.* Hence, the safety support system should be able to notice the change in secure paths if a consecutive unfamiliar location points are obtained while tracking the child's movement.

Chapter 4

SECURE AREA DETECTION

4.1 Problem Definition

Children often perform their activities at same places repeatedly, e.g. home, school, playground, etc. If it's reported that the children sticking around unfamiliar areas, intimations should be given to their parents immediately. For the children to be safe, detecting and marking their corresponding secure areas is important. This can be achieved by parents manually marking the secure areas of their children, but it has few hindrances. One is that it's not easy to judge the scope of the safe region manually. The other is that there is a high probability of change in secure areas, since children can be involved in multiple activities. To resolve the issues above, clustering methods are used to form self-regulated secure regions. For secure area detection, we want to discover the clusters of stay points which are the places that the children spend significant amount of time and/or visits frequently, such as home, school, playground, mall, etc.

In the following sections, we will introduce the clustering techniques used in this thesis, followed by the experimental results and analysis.

4.2 Clustering Analysis

Clustering is the process of grouping a set of identical physical or abstract objects into classes of similar objects [11]. A cluster is a collection of data objects which are identical to one another in the same cluster but nonidentical from set of objects in the other cluster. Clustering can be treated as a form of data compression, as the cluster of data objects can be considered as a single group [11]. Clustering follow the process of grouping the data based on similarity and then assign labels to relatively small number

of groups. Clustering algorithms are advantageous in case of changes and segregating the required features which represent different groups [11]. It is also called as data segmentation because it groups the data into segments based on similarity. Typical clustering methods include:

1. *Partitioning algorithms.* Partitioning algorithms divide the database D of n objects in to a set of k clusters [8]. For partitioning algorithms, k is the input parameter, and they start with a partition of D and use an iterative strategy to pull an optimal objective function. Then the partitioning algorithms designate each cluster with the gravity center or an object close to the center that makes more sense in some cases. Representative methods include k-means [25] and k-medoids [19].

2. *Hierarchical methods.* Hierarchical clustering methods accumulate the data objects into a tree of clusters [11]. Depending on whether the hierarchical decomposition is formed in a bottom up or top down fashion, hierarchal methods are divided into two types: agglomerative and divisive. In agglomerative hierarchical clustering, each object is placed in its own cluster and it further combines these fragmentary clusters into large clusters until all the objects are formed as a single cluster, or until a termination condition imposed is satisfied. Example of agglomerative clustering are AGNES [21], and ROCK [10]. Differing in determining their inter cluster likeliness, most of the hierarchical clustering methods belong to this category [11]. Unlike agglomerative clustering methods, divisive clustering methods segregate the clusters into atomic clusters until certain condition imposed on it is satisfied. Examples of divisive clustering are DIANA [20], and BIRCH [58].

3. *Density based approaches.* Density based clustering depends on its fundamental principle of density. Density based clustering methods are used especially to find arbitrary shape clusters [11]. These algorithms keep growing the cluster until a threshold is surpassed by its neighborhood. They term clusters as dense regions of objects containing at least certain number of data objects. All the data objects which does not make to a cluster are termed as outliers or noise [11]. The arbitrariness in the clusters formed by density based clustering algorithms is due to the removal of outliers. This property of density based methods make them prominent relative to partitioning methods. DBSCAN [6] and OPTICS [3] are examples of density based clustering methods.

4. *Model based methods.* Model based clustering methods use certain models for clusters and tries to optimize the fit between the data and models. In this kind of methods, the data points are viewed as coming from a mixture of probability distributions, each of which represents a different cluster [8]. That is, in model-based

clustering, with each component representing a different cluster, it is assumed that the data points are generated by a mixture of probability distributions. Thus a particular clustering method can be expected to work well when the data points conform to the model. They also guide to an approach of automatically determining the number of clusters based on standard statistics [11]. Generally, there are two approaches to model the composite of clusters, classification likelihood approach and mixture likelihood approach [8]. They differ in maximizing the integers labeling the classification and probability of the observation respectively. Expectation Maximization (EM) [28] and conceptual clustering [26] are the examples of model based clustering approach.

5. *Grid based algorithms.* Grid based clustering uses multi resolution grid data structure for its functionality [11]. Grid based algorithms perform clustering in large multidimensional space where clusters are considered as denser regions than their surroundings. Grid based clustering methods partition the data space into finite number of cells, calculate the cell density for each cell, sort the cell according to their densities, identify cluster centers and then traverse its neighbors cells [8]. Grid based clustering methods are fast with regards to processing time, as they depend on cells in each dimension in the measured space and independent of number of data objects [11]. OptiGrid [14], STING [50], and Wavecluster [42] are examples of grid based clustering algorithms.

In the following section, we will introduce the clustering methods used in this thesis for secure area detection.

4.3 Clustering Methods Used for Secure Area Detection

Since k-means is an effective clustering method and DBSCAN is capable to identify arbitrary shapes, in this thesis, we will use these two methods for secure area detection and compare their performances in our application.

4.3.1 K-means Clustering Algorithm

k-means is a partitioning method which divides the database D into k partitions based on the Euclidean distance measure. The *sum of square error* is used as the objective function to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters [13].

In a nut shell, the process of k-means algorithm is as follows [57]:

- Step 1. Selecting k objects as the center of the clusters randomly;
- Step 2. For each remaining object, find the closest of available k centers and assign it to the cluster holding that center;
- Step 3. After all the objects are assigned to their respective clusters, compute the means of the cluster and make them the new centers;
- Step 4. Repeat the process until there is no change noticed in the newly formed clusters from the previous clusters.

Figure 4.1 shows an example of using k-means algorithm to cluster the data points into different groups. The green points in the initial clustering represent the objects that are considered as centers randomly, before performing the first round. In the iteration stage, red points represent the change in center, which are essentially the new centers formed from the points in the last cluster. The final clustering represent the finely clustered data points, acquired when there is no change noticed from its previous cluster.

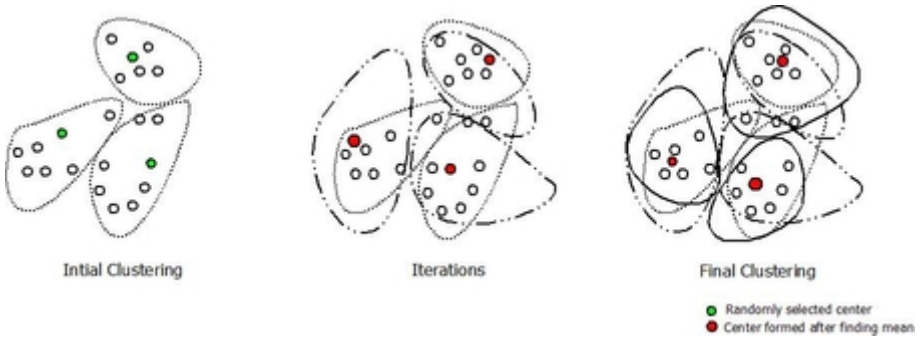


FIGURE 4.1: Clustering data points using k-means algorithm. *Source*: Based on Han and Kamber [11].

The pseudo code of k-means is shown in Algorithm 2 [11].

Algorithm 2 k-means Clustering [11]

Input: Set of data points to be clustered $D = \{d_1, d_2, d_3, ..., d_n\}$, number of clusters k.
Output: Set of k clusters formed $C = \{c_1, c_2, c_3, ...c_n\}$,
Method:

1. arbitrarily choose k data points from D as initial cluster centers;
2. repeat
3. (re)assign each data point to the cluster to which the data point is the most similar;
4. update the center of the cluster (i.e., calculate the mean value of the data points for each cluster);
5. until no change;

4.3.2 DBSCAN Clustering Algorithm

To find clusters of arbitrary shape, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [6], a density based clustering method, is introduced in this section. A density-based method creates clusters by continuously growing a cluster so long until the density of the data objects in the neighborhood exceeds some threshold. DBSCAN uses two global constants $MinPts$ (the minimum number of points in a cluster) and ξ (density of a neighborhood) in its functionality. It initiates its process with an arbitrary point p and fetch all the points in the source data D which are density reachable with respect to $MinPts$ and ξ.

Following are the key terms of DBSCAN algorithm [6]:

- *Core Object:* An object p whose ξ-neighborhood contains no less than $MinPts$ number of objects is a core object with respect to ξ and $MinPts$.
- *Directly Density Reachable:* An object M is directly density reachable from object P with respect to ξ and $MinPts$, if M is within the ξ-neighborhood of P which contains at least a minimum number of points $MinPts$.
- *Density Reachable:* An object Q is density-reachable from object P with respect to ξ and $MinPts$, if there is a chain of objects $\{p_1, p_2, p_3, ...p_n\}$ ($p_1 = P$ and $p_n = Q$), p_{i+1} is directly density reachable from p_i with respect to ξ and $MinPts$.
- *Density Connected:* An object S is density-connected to object R with respect to ξ and $MinPts$, if there is an object O such that both S and R are density reachable from O with respect to ξ and $MinPts$.

Given the data points shown in Figure 4.2, let $MinPts = 3$, P and M are core objects. M is directly density reachable from object P since M is within the ξ-neighborhood of P and P contains three points in its neighborhood. An object Q is density-reachable from object P since there are set of points between P and Q which are directly density reachable. S is density-connected to object R because O is density reachable to both S and R.

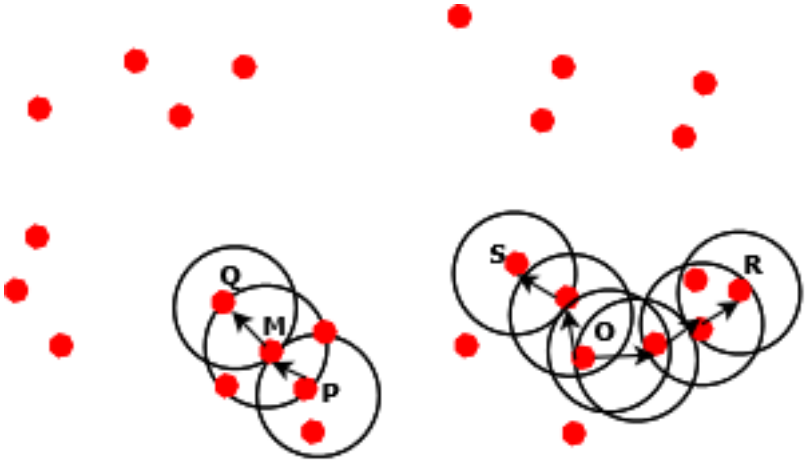


FIGURE 4.2: Density-connected and density-reachable in DBSCAN. *Source*: Based on Ester, Kriegel, Sander and Xu [6].

Algorithm 3 shows the pseudo code of DBSCAN clustering algorithm [6].

Algorithm 3 DBSCAN Clustering [6]

Input: Set of objects to be clustered $D = \{d_1, d_2, d_3, ...d_n\}$, neighborhood distance ξ, minimum number of points $MinPts$
Output: A set of density-based clusters $C = \{c_1, c_2, c_3, ...c_n\}$.

```
mark all objects as unvisited;
repeat
  randomly select an unvisited object d;
  mark d as visited;
  if ξ − neighborhood of d ≥ MinPts then
    create new cluster c, add d to c;      //core points
    let N be the set of objects in ξ − neigborhood of d;
    for each point d' in N do
      if d' is unvisited then
        mark d' as visited;
        if ξ − neighborhood of d' ≥ MinPts then
          add those points to N
        end if
      end if
      if d' is not yet a member of any cluster then
        add d' to c      // directly density reachable or density reachable points
      end if
    end for
    output c;
  else
    mark d as noise;
  end if
until no object is unvisited;
```

4.4 Experimental Results and Analysis

4.4.1 Experimental Setup

In this thesis, we develop an application based on the android platform to collect the location data. To simulate the children's life patterns, we collect the location histories from five volunteers of the students in the university. In this section, we use two weeks' data collection from one volunteer student during his daily life for experiments. The collected data contains the location points tracked by GPS, GSM and Wi-Fi with the best available resource. Each location point is tracked for every 300 seconds, that is, each day 284 location points are collected. Table 4.1 describes the data we used in the section.

Before secure area detection, we first identify the stay points from the location points reported by the application. Based on the empirical studies, in our experiments, we set distance threshold $\delta_D = 200meters$ and time threshold $\delta_T = 20minutes$ by default for

TABLE 4.1: Description of the collected data.

Sensors used for data collection	GPS, GSM and Wi-Fi
Number of days collected	14
Total number of location points collected	3,976
Time between each location point	300 seconds

stay point identification (Note that these two thresholds can also be set by the users). Using the stay point detection algorithm described in Algorithm 1, we extracted 713 stay points from the collected data. Figure 4.3 shows the number of stay points obtained per day. The drastic decrease in stay points of a particular day denotes either traveling most of the time or staying at limited places most of the time, such as the user travels or stays at home in the weekends.

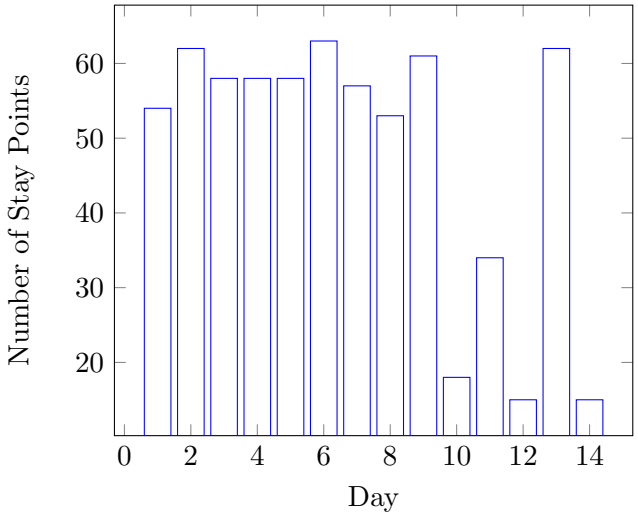


FIGURE 4.3: Number of stay points detected each day.

To evaluate the performance of different clustering algorithms, we use *F1* for measurement, which is defined as [45]

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{4.1}$$

where recall is the ratio between the number of correct positive predictions and the total number of positive examples; precision is the ratio between the number of correct positive predictions and the number of positive predictions. Based on the $F1$ score, we use the $Micro - F1$ and $Macro - F1$ for the comparisons:

$$micro - F1 = \frac{\sum_{i=1}^{|C|} 2 * Precison * Recall}{\sum_{i=1}^{|C|} Precison + Recall}, \tag{4.2}$$

$$macro - F1 = \sum_{i=1}^{|C|} \frac{F1_i}{\mid C \mid}, \tag{4.3}$$

where $\mid C \mid$ is the number of clusters formed.

Micro-F1 and Macro-F1 emphasize the performance of the algorithm on common and rare categories respectively [38].

4.4.2 Comparison of Different Clustering Methods for Secure Area Detection

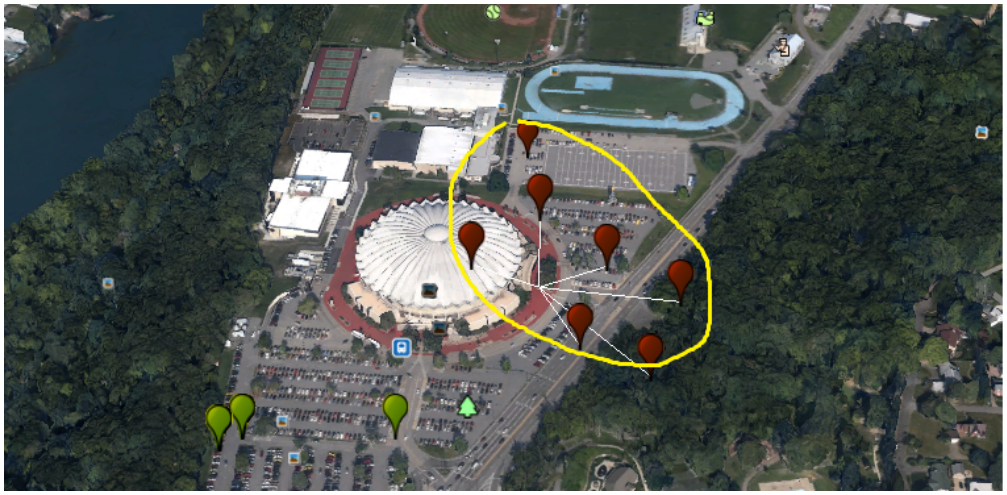
In this section, based on the identified stay points, we use k-means and DBSCAN for secure area detection and compare their performance in our application. The experimental results in Table 4.2 demonstrated that DBSCAN outperform k-means for secure area detection in our application. This is due to the capability of DBSCAN to identify clusters of arbitrary shape.

TABLE 4.2: Comparisons of different clustering algorithms for secure area detection.

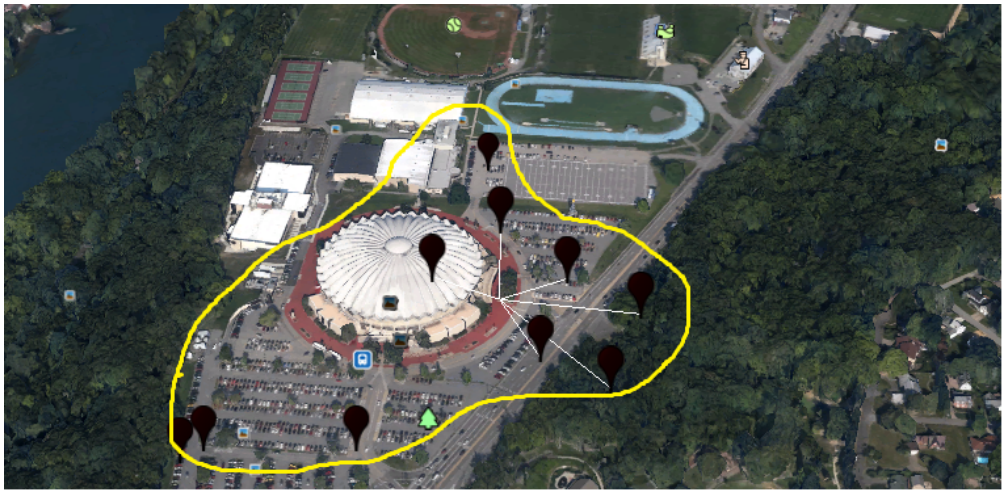
Algorithm	**Parameters**	**No. of SAs**	**micro-F1**	**macro-F1**
k-Means	$k = 2$	2	0.23	0.71
	$k = 3$	3	0.34	0.67
	$k = 4$	4	0.50	0.52
	$k = 5$	5	0.71	0.61
	$k = 6$	6	0.77	0.87
DBSCAN	$\varepsilon = 0.1, \quad MinPts = 5$	6	0.98	0.96
	$\varepsilon = 0.1, \quad MinPts = 10$	6	0.99	0.99
	$\varepsilon = 0.2, \quad MinPts = 5$	7	0.96	0.94
	$\varepsilon = 0.2, \quad MinPts = 10$	7	0.96	0.95
	$\varepsilon = 0.3, \quad MinPts = 5$	7	0.91	0.90
	$\varepsilon = 0.3, \quad MinPts = 10$	7	0.92	0.91

Figure 4.4 displays the Google Earth's views of detected secure areas (West Virginia University basketball stadium) by k-means (Figure 4.4a) and DBSCAN (Figure 4.4b), which demonstrates that DBSCAN perform better than k-means in our application. DBSCAN is capable of detecting secure areas with arbitrary shapes compared to k-means: DBSCAN discovered more meaningful regions by adding appropriate stay points into its cluster, while k-means was unable to recognize other stay points of the same secure area (marked in green) into same cluster.

Figure 4.5 shows that (1) k-means clustered two close-by secure areas (around West Virginia University Evansdale Campus) into one cluster due to its oval prone nature (Figure 4.5a); (2) DBSCAN was able to distinguish these two secured areas and formed two different clusters (Figure 4.5b).

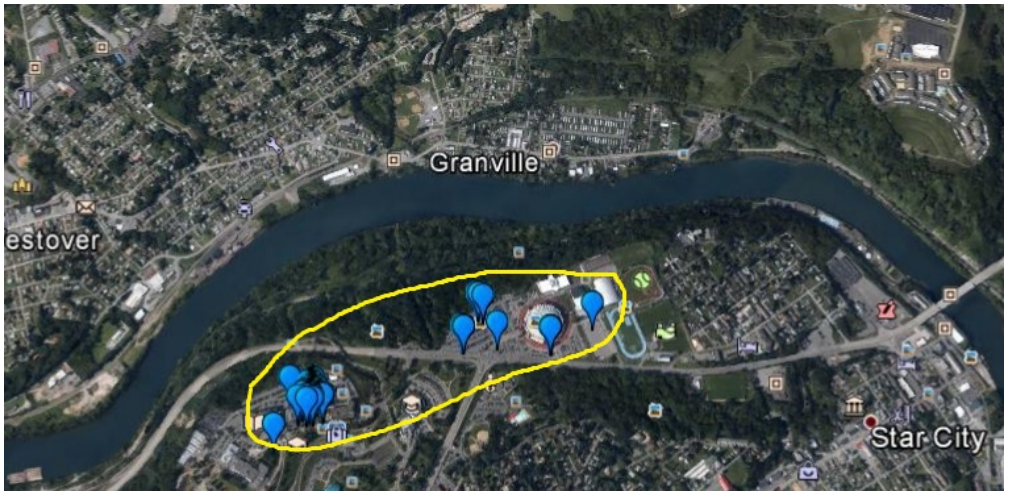


(A) Secure area formed by k-means.



(B) Secure area formed by DBSCAN.

FIGURE 4.4: Example of secure area detection by k-means and DBSCAN. The map in the figure is displayed using Google Earth Pro (http://www.google.com/earth/).



(A) Secure areas detected by k-means.

(B) Secure areas detected by DBSCAN.

FIGURE 4.5: Comparisons of secure area detection by k-means and DBSCAN. The map in the figure is displayed using Google Earth Pro (http://www.google.com/earth/).

Chapter 5

SECURE PATH DETECTION

5.1 Problem Definition

In children's daily lives, common routes are often reciprocated, e.g. their ways to and from schools. When a child is detected to be on a strange path based on the reported locations, an intimation should be sent to his/her parent. For the children to be safe, detection of their corresponding secure paths is important. It's not easy for parents to denote secure paths of their children manually, as they involve in many activities taking multiple paths. And it becomes much difficult, when new paths are taken by children with change in activities. To overcome such problems, frequent sequential pattern mining is applied to fetch children's secure paths automatically. Given the location histories, we aim to identify the paths which children have taken significant number of times, such as paths from homes to schools, paths to the places they always visit etc.

In the following sections, we will introduce the basic concepts of sequential pattern mining followed by the techniques used in this thesis. Experimental results for secure path detection will be also analyzed.

5.2 Sequential Pattern Mining

Sequential pattern mining is the process of mining frequently appearing ordered events or subsequences in the form of patterns [11]. An example of sequential pattern is "*customers who buy a play station are likely to buy a joy stick within a week*". In our application, the example could be "*John went through home → 8th Street → University Ave, then he is likely to pass Riverview Drive in next 10 minutes*".

Given a set of sequences, with each sequence consisting of a list of elements and each element consisting of a set of items, and given a user specified minimum support threshold of $MinSup$, sequential pattern mining discovers all the frequent subsequences whose appearances in the set of sequences are greater than or equal to $MinSup$ [2].

Let $I = \{I_1, I_2, I_3, ..., I_p\}$ be a set of all items. An *itemset* is a non-empty set of items. A *sequence* s is an ordered list of events that are occurred. A sequence s is denoted by $\langle e_1 e_2 e_3 .... e_l \rangle$, where an *event* e_i occurs before event e_{i+1}. Any $e_j \in s$ is called an *element* of s. Event refers to an itemset, which is an unordered list of items. It is denoted by $(x_1 x_2 ..... x_q)$, where x_k is an item. An item can occur only once in an event but multiple times in different events of a sequence. The *length* l of a sequence is the number of items present in a sequence. A sequence with length l is called as *l-sequence*. The *size* sz of a sequence, is the number of events or itemsets present in the sequence. An example of itemset and sequence is shown in Table 5.1. In the case of this thesis, the sequence is any trip that a child has taken between secure areas.

TABLE 5.1: Example of an itemset and sequence.

Itemset(e_i)	**Sequence**(s)	**Size**(sz)	**Length**(l)
(abc) (ab) (a) $(abcd)$	$\langle (abc)(ab)(a)(abcd) \rangle$	4	10

A sequence $\alpha = \langle a_1, a_2, a_3 .. a_n \rangle$ is called a *subsequence* of another sequence $\beta = \langle b_1, b_2, b_3, ... bm \rangle$ ($\alpha \subseteq \beta$), if there exists integers $1 \leq j_1 < j_2 ... < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2} ....., a_n \subseteq b_{j_n}$. If α is a subsequence of β, then β is called the *super sequence* of α. For example, $\alpha = \langle (ab), d \rangle$ is a subsequence of $\beta = \langle (abc), (de) \rangle$ [11].

S is a *sequence database* containing set of tuples, $\langle SID, s \rangle$, where SID is a *sequence_ID* and s is a sequence [11]. In our scenario, sequence database contains the sequences (trips taken each day) obtained each day and *sequence_ID* refers to its corresponding day. The *support* (Sup) of a sequence α is the number of tuples that contain α in the sequence database i.e., $Sup = \mid \langle SID, s \rangle | (\langle SID, s \rangle \in S) \wedge (\alpha \sqsubseteq S) \mid$ [11]. A sequence α is frequent in sequence database S if $Sup(\alpha) \geq MinSup$. An example of a sequence database is shown in Table 5.2. Given a sequence database and $MinSup$, the task of sequential pattern mining is to find the frequent sequential patterns in the database.

TABLE 5.2: Example of sequence database.

Sequence Database(S)	
$sequence_ID$	**sequence**(s)
1	$\langle(ab)(a)(abcd)\rangle$
2	$\langle(a)(abcd)(ab)\rangle$
3	$\langle(abc)(ab)(a)(abcd)\rangle$

Typical sequential pattern mining algorithms are categorized as:

1. *Apriori-like algorithms.* These algorithms are the conventional sequential pattern mining algorithms first introduced by Agrawal et al. [2]. The mining process in this kind of algorithms involves sorting the transactional database, obtaining large 1-itemsets from the sorted database based on *support*, replacing the sequences by large itemsets forming sequential database, generating all frequent sequential patterns and finally pruning the maximal sequential patterns [44]. Examples of these algorithms include Apriori [1, 2], FP-Growth algorithm [35].

2. *BFS-based algorithms.* These algorithms follow the breadth first search approach of traversal and they are similar to apriori like algorithms. All k sequences are generated at each k_{th} iteration of the algorithm while traversing the search space [44]. Examples of these algorithms include GSP algorithm [46], SPADE [56].

3. *DFS-based algorithms.* Unlike BFS-based algorithms, DFS-based algorithms follow the depth first search approach of traversal while traversing the search space [44]. They may consist of an ineffective pruning method and engender plenty of candidate sequences. Examples of these algorithms include FreeSpan [12], PrefixSpan [34], and SPAM [4].

4. *Closed sequence based algorithms.* Distinctive to other algorithms, these algorithms mine closed sequential patterns instead of full set of subsequences which satisfy the threshold *support* [44]. Due to this, there will be a significant reduction in number of subsequences reducing time and space [44]. Examples of these kind of algorithms include CloSpan [53] and BIDE [48].

5. *Incremental based algorithms.* These kind of algorithms are used for frequent and incremental database updates [44]. Examples of these kind of algorithms include SuffixTree [49] and FASTUP [24].

In the following section, we will introduce the sequential pattern mining methods used in this thesis for secure path detection.

5.3 Sequential Pattern Mining Methods for Secure Path Detection

Since Prefix Span, SPAM, and CloSpan are effective algorithms for frequent sequential pattern mining, we will use them to find secure paths for the children in this thesis and compare their performances.

5.3.1 PrefixSpan

The main idea of Prefix Span [34] is the projected databases are formed based on frequent prefixes unlike conventional algorithms where all the occurrences of frequent subsequences are projected [12]. The terms to be understood to apply PrefixSpan are as follows:

A sequence $\alpha = \langle a_1, a_2, a_3..a_n \rangle$ is called a *prefix* of another sequence $\beta = \langle b_1, b_2, b_3, ...bm \rangle$ $(m \leq n)$ if and only if: (1) $b_i = a_i$ for $(i \leq m-1)$; (2) $b'_m \subseteq a_m$; and (3) all the items in $(a_m - b_m)$ are after those in b'_m in order [34]. Given subsequences α and β such that β is a subsequence of α, i.e., $\beta \sqsupseteq \alpha$, a subsequence $\alpha^{'}$ of α (i.e., $\alpha^{'} \sqsupseteq \alpha$) is called a projection of α w.r.t. prefix β if and only if: (1) $\alpha^{'}$ has prefix β; (2) there exists no proper super sequence $\alpha^{''}$ of $\alpha^{'}$ such that $\alpha^{''}$ is a subsequence of α and also has prefix β [34].

The Prefix Span mines the sequential patterns in a sequence database S with a minimum support $MinSup$ in the following manner [34]:

- Step 1: Scan the database and find all the frequent items in the sequences.
- Step 2: The complete set of sequential patterns are partitioned into subsets such that each subset contains the prefixes of gained 1-length sequential patterns from Step 1.
- Step 3: Each of the formed subsets are mined by corresponding projected databases and keep mining them recursively.

All the patterns which has a $Sup \geq MinSup$ with respect to sequence database S are considered as frequent sequential patterns.

The algorithm of PrefixSpan is shown in Algorithm 4 [34].

Algorithm 4 PrefixSpan

Input: A sequence database S, minimum support threshold $MinSup$.
Output: The complete set of sequential patterns F

Method: Call PrefixSpan($\langle\rangle, 0, S$)
Parameters: a sequential pattern α, length of α, l, the α-projected database $S|_\alpha$ (if $\alpha \neq \langle\rangle$), otherwise the sequence database S;
Method:

1. Scan $S|_\alpha$ once, find the set of frequent items b such that
   (a) b can be assembled to last element of α to form sequential pattern; or
   (b) $\langle b\rangle$ can be appended to α to form a sequential pattern.
2. For each frequent item b, append it to α to form a sequential pattern $\alpha^{'}$, an output $\alpha^{'}$;
3. For each $\alpha^{'}$ construct $\alpha^{'}$-projected database $S|_{\alpha'}$, and call PrefixSpan ($\alpha^{'}, l + 1, S\,|_{\alpha'}$)

5.3.2 Sequential Pattern Mining using A Bitmap Representation (SPAM)

SPAM [4] is sequential pattern mining algorithm which uses a *depth-first* search strategy along with efficient pruning techniques to find sequential patterns from the sequence database S. It's searching technique integrates a vertical bitmap representation of the database with efficient *support* counting.

SPAM uses a lexicographic tree to store its sequence database for easy traversal. The lexicographic sequence tree has the following structure [53]: (1) Each node in the lexicographic tree is a sequence and the root is always a *null* sequence; (2) Each child sequence in the lexicographic sequence tree is either a *sequence-extended* sequence or an *item-set extended* sequence; (3) The left sibling is always less than the right sibling in lexicographic graphic tree.

In order to find frequent sequential patterns, SPAM traverses the lexicographic tree using a breadth first strategy. At each node n, if the support Sup of the sequence-extended [4] or itemset-extended [4] sequence is greater than $MinSup$ the sequence is stored and the depth first search is repeated recursively on the sequence s.

Since the depth first strategy used has a huge search space, SPAM applies *S-step pruning* and *I-step pruning* mechanisms to prune s-extensions and i-extensions of a node n in the lexicographic tree using Apriori based algorithms. The aim of these pruning techniques is to decrease the candidate items which are responsible to extend a node n at S-step and I-step [4]. The S-step and I-step pruning of SPAM algorithm are described as follows [4]:

- *S-step pruning:* Consider a sequence s at node n of a lexicographic tree. If its sequence extended sequences are $\alpha = (\alpha, \{a_j\})$ and $\beta = (\beta, \{a_k\})$ and suppose α is frequent but β is not frequent, then $(s, \{a_j\}, \{a_k\})$ and $s, \{a_j, a_k\}$ cannot be frequent, as both contain subsequence β. Hence, a_k is pruned from both S_m and I_m where m is any node corresponding to frequent sequence-extended child of s.
- *I-step pruning:* Consider an item extended sequence $s = (s^{'}, \{a_1, ...a_n\})$. If it's item-extended sequences are $\alpha = (s^{'}, \{a_1, ...a_n, a_j\})$ and $\beta = (s^{'}, \{a_1, ...a_n, a_k\})$ and suppose $a_j < a_k$ and α is frequent but β is not frequent, then $(s^{'}, \{a_1, ...a_n, a_j, a_k\})$ cannot be frequent. Hence, a_k is pruned I_m where m is any node corresponding to frequent itemset-extended child of s.

SPAM uses vertical bitmap representation to perform efficient counting. That is for each item in the data set there is a bit corresponding to each transaction in the data set. If the item is present in the transaction, the bit is set to 1; otherwise the bit is set to 0. Also all the transactions of each sequence in the database will appear together in the bitmap. The bitmap idea is also extended to itemsets and sequences using *bitwise* of two bitmaps of different items. Sequences are partitioned into different sets based on their lengths when read in the data set to make counting efficient [4]. The candidate generation in SPAM is done by *S-Step* process and *I-Step* process. They are described as follows [4]:

- *S-step process:* If $B(\alpha_i)$ and $B(\beta)$ are bitmaps of sequence α and item β respectively, the S-step appends the itemset of item β to α_i. The new sequence produced will have a bitmap again, say $B(\alpha_g)$, with a property that if a bit has value 1, the corresponding transaction γ must contain β and before the transactions of γ, all the other itemsets in α_g should be present.
- *I-step process:* The I-step works as follows: If $B(\alpha_i)$ and $B(\beta)$ are bitmaps of sequence α_i and item β respectively, the S-step appends the item β to the last itemset of sequence α_i. The new sequence formed is α_g with β as an additional element compared to α_i in its last itemset. The bitmap of α_g should have a property that if a bit has value 1, the corresponding transaction γ must contain the last itemset in α_g, and before the transactions of γ, all the other itemsets in α_g should be present.

5.3.3 Closed Sequential Pattern Mining (CloSPan)

Closed sequential pattern mining [53] is the process of disregarding sequences which has one or more super-sequence(s) with the same *support* as the current sequence i.e., all

the sequences whose super sequences appear the same number of times as them will be neglected [53]. To avoid large number of frequent sub-sequences for long patterns, Closed Sequential Pattern Mining (CloSPan) has come to light.

The approaches proposed to find closed or maximum frequent patterns by CloSpan [53] are: (1) greedily finding the *final* closed pattern set; and (2) finding a closed pattern candidate set and conducting *post-pruning* on it.

Given two sequences s = $\langle e_1 e_2 e_3 .. e_m \rangle$ and an item a, $s \diamond a$ means s concatenates with a. The concatenation can be I-step extension, $s \diamond_i a = \langle e_1 e_2 .. e_m \cup (a) \rangle$; or S-step extension $s \diamond_s a = \langle e_1 e_2 .. e_m, (a) \rangle$. For example if $\langle (a) \rangle$ is a sequence, then $\langle (ab) \rangle$ is an I- step extension and $\langle (a)(b) \rangle$ is a S-step extension. If a sequence $s = \alpha \diamond \beta$, then α is the prefix of s, and β is the suffix of s. A sequence database $S_s = \{s \mid s^{'} \in S, s^{'} \alpha \diamond s$ s.t. α is minimum prefix of $s^{'}$ containing s. Two kind of projections exists, namely physical projection and pseudo projection. In physical projection, S_s is stored in actual table. In pseudo projection, S_s is not physically produced, instead pointers to the projected point are saved for each sequence [53].

Algorithm 5 provides a generic methodology for depth first search in the prefix search tree. For each sequence s and its projected database S_s, the algorithm performs I-step extension and S-stem extension respectively and recursively until all frequent sequences which has prefix s are found. The return statement in line 8 checks for the $MinSup$ in the projected database. Once fails, it is not required to expand s anymore. In order to exit the recursion as soon as possible, an extra termination condition is added at line 16 by CloSpan so as to form closed sequences. CloSpan finds the closed sequences in two steps. In the first step candidate set is found which is larger than the usual closed sequence set. In the second step all the non-closed sequences are removed.

Algorithm 5 Prefix Span with additional condition [34]

Input: A sequence s, a projected DB S_s and $MinSup$
Output: The frequent sequence set F

insert s to F;
scan S_s once, find every frequent item a such that
    (a) s can can be extended to $(s \diamond_i a)$, or
    (b) s scan can be extended to $(s \diamond_s a)$;
if no valid a **then**
    return ;
end if
for valid a **do**
    Call PrefixSpan$(s \diamond_i a, S_{s \diamond_i s}, MinSup, F)$;
end for
for valid a **do**
    Call PrefixSpan$(s \diamond_i a, D_{s \diamond_i a}, MinSup, F)$;
end for
return ;

Algorithm 6 is the preprocessing step that CloSpan does. It includes sorting every itemset, removing infrequent items and empty sequences. Then it calls CloSpan recursively by performing depth first search on the prefix search tree and building respective prefix sequence lattice. Once the above step is done, it eliminates non closed sequences.

Algorithm 6 ClosedMining$(S, MinSup, F^{'})$ [53]

Input: A database S_s and $MinSup$
Output: The complete closed sequence set $F^{'}$

remove infrequent items and empty sequences, and sort each itemset of a sequence in S_s;
$s^1 \leftarrow$ all frequent 1-item sequence;
$s \leftarrow s^1$
for sequence $s \in s^1$ **do**
    CloSPan$(s, S_s, MinSup, F^{'})$;
end for
eliminate non-closed sequences from L;

Though CloSpan is similar to Prefix Span, the search space pruning play a vital role making a significant difference between them as shown in Algorithm 7. That is before mining successive super sequences from a discovered sequence and it's corresponding projected database, CloSpan checks whether a discovered sequence exists such that $s \sqsubseteq s^{'}$ or $s^{'} \sqsubseteq s$, and $I(D_s) = I(D_{s^{'}})$. If the condition is satisfied, it means that all its descendants are already discovered because given two sequences, $s \sqsubseteq s^{'}$ and $I(D_s) = +(D_{s^{'}})$, $\forall \Upsilon$, support$(s \diamond \Upsilon) = support(s^{'} \diamond \Upsilon)$.

Algorithm 7 CloSPan($s, S_s, minSup, F''$) [53]

1: Input: A sequence s, a projected database S_s and $MinSup$.
2: Output: The prefix search lattice F''.

3: Check whether a discovered sequence s^i exists s.t. either $s \sqsubseteq s'$ or $s' \sqsubseteq s$, and $I(S_s) = I(S_{s'})$;
4: **if** such super-pattern or sub-pattern exists **then**
5: modify the link in F'',
6: **return** ;
7: **else**
8: INSERT s into F'';
9: **end if**
10: Scan S_s once, find very frequent item a such that
11: (a) s can be extended to ($s \diamond_i a$), or
12: (b) s can be extended to ($s \diamond_s a$)
13: **if** no valid a available **then**
14: **return** ;
15: **end if**
16: **for** valid a **do**
17: Call CloSPan($s \diamond_i a, S_{s \diamond_i a}, MinSup, F''$)
18: **end for**
19: **for** valid a **do**
20: Call CloSPan($s \diamond_s a, S_{s \diamond_s a}, MinSup, L$)
21: **end for**
22:
23: **return** ;

There are two approaches to check the equivalence of projected databases: the containment, and the size of projected database. Checking for the containment initially, i.e. checking all the sequences if they are subsequences or super sequences of the current sequence is expensive. It's costly even to extend the sub-sequence and super-sequence from the current set when a new sequence is extended from the current sequence. CloSpan devised an approach which uses hash index on the size of projected database. Only sequences whose projected database size is equal to current sequence are tested. The cost becomes negligible with this approach relative to total run time [53].

The mechanism for fast condition works this way: A hash table is maintained such a way that the hash key is $I(S_s)$ with s as it's corresponding value thus making the pair $\langle I(S_s), s \rangle$. When a sequence s comes in, the value $I(S_s)$ is calculated and the hash table is checked if the value of $I(S_s)$ already exists. If exists, it means that some other sequence s' contains the same projected database size $I(S_{s'}) = I(S_s)$ as s. If $s \sqsubseteq s'$, then the value $I(S_{s'})$ is unaltered, else if $s' \sqsubseteq s$ then $I(S_{s'})$ is replaced with $I(S_s)$. CloSpan uses the sequences stored in the prefix sequence lattice as a value in its hash table. Instead of storing the whole sequence, it just stores the pointer to the node containing corresponding sequence in the prefix sequence lattice. Thus lines 3-8 of Algorithm 7

perform quicker with hashing. Checking as shown in Algorithm 8. Line 16 in Algorithm 8 discovers the backward super-pattern $s^{'} \sqsubseteq s$. It makes sure that s does not need to grow any descendant by deleting the duplicate sub trees produced by $s^{'}$ and recovering only one such tree for super pattern s.

Algorithm 8 checkProjectedDBSize(s, k, H) [53]

```
1: Input: A sequence s, it's key k and hash table H
2: Output: An updated hash table H

3: l_sup ← ∅, l_sub ← ∅
4: index hash table with key k
5: find a list of pairs ⟨k, s'⟩;
6: for each pair ⟨k, s'⟩ do
7:    if support(s) = support(s') then
8:       if s' ⊑ s then
9:          l_sup ← l_sup ∪ {⟨k, s'⟩};
10:      end if
11:      if s ⊑ s' then
12:         l_sub ← l_sub ∪ {⟨k, s'⟩};
13:      end if
14:   end if
15: end for
16: if l_sup not empty then
17:    remove all pairs in l_sup from H
18:    merge descendant sub trees (of s' in l_sup) in L';
19: end if
20: if l_sub not empty then
21:    merge descendant sub trees (of s' in l_sub) in L;
22:    return ;
23: end if
24: insert ⟨k, s⟩ into H;
```

A fast subsumption checking algorithm is used by CloSpan. The *support* of the sequence is used as it's hash function. For a sequence s, CloSpan checks whether there are any other sequences containing same support as s and then checks them if any of them is a super sequence for s. As *support* is so dense to be hash key the sum of sequence identifiers $\tau(S_s)$ is proposed to be a hash key for its sparsity. But, the equivalence of $\tau(S_s)$ doesn't imply the equivalence of *support*, so for the sequences with same $\tau(S_s)$ their *support* has to be checked for eliminating invalid candidates. Ultimately the containment is checked to see whether the sequence can be absorbed. The elimination is done by early termination of equivalence by CloSpan.

5.4 Experimental Results and Analysis

5.4.1 Experimental Setup

Based on the collected location histories and secure areas that are obtained, sequences are formed determining the paths that the children has taken during the training period. Before applying a sequential pattern mining technique to obtain frequent sequential patterns, the original sequences formed are segregated into sub-sequences based on "day". As a result, sequence is formed for each day forming a sequence database ultimately. As children's activities differ on regular days with weekends, in order for the pattern mining algorithm not neglecting the weekend patterns due to less support, a separate bucket is maintained for weekends and normal days.

The partitioned sequence buckets act as input for the sequential pattern mining algorithm in order to fetch two variant (week days and weekends) sequential patterns based on children's location histories.

To simulate the children's life patterns, we collected the location histories from five volunteers who are students in the university. Secure paths are obtained by applying sequential pattern mining algorithms on two weeks of data. The obtained sequential patterns are tested against four weeks of data using the measures shown in Table 5.3.

TABLE 5.3: Performance Measures.

Metric	**Description**
TP	The number of notifications that predicted the child to be unsafe are actually when the child is unsafe
TN	The number of notifications that predicted the child to be safe are actually when the child is safe
FP	The number of notifications that predicted the child to be unsafe are actually when the child is safe
FN	The number of notifications that predicted the child to be safe are actually when the child is unsafe
Precision	TP / $(TP + FP)$
Recall	TP / $(TP + FN)$
Accuracy	$(TP + TN)$ / $(TP + TN + FP + FN)$

Children being the commuters of similar path, most of the times they take the same route. Taking this fact into consideration, empirically a *support* value of 0.6 (i.e., the paths which are used at least 60% of the time) are considered to be secure in the coarse of this experiment.

5.4.2 Comparison of Different Sequential Pattern Mining Methods to Detect Secured Paths

In this section, based on the collected location histories and the identified secure areas, we use CloSPan, SPAM, and PrefixSpan for secure path detection and compare their performances.

Figure 5.1 shows the run time comparisons of three algorithms CloSPan, SPAM, and PrefixSpan for varying *Support*. From Figure 5.1, we can see that (1) CloSpan outperform SPAM Prefix Span, and (2) both CloSpan and SPAM are much quicker than Prefix Span. This is mainly due to the early termination condition applied by CloSPan.

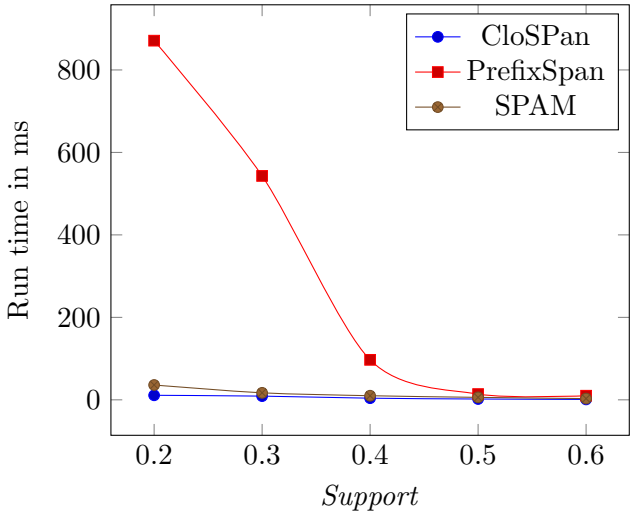


FIGURE 5.1: Run time comparisons of different algorithms.

Table 5.4 shows the precision, recall and accuracy obtained with CloSPan for varying supports. CloSpan well performs with all *support* values. The possible reason for the reduced *precision* is that while collecting the data, the volunteer could have taken a secure path less frequently which ultimately connects to his/her secure area. For example, a child may sometimes drop by his/her friend's home, on the way home from school. As these paths may be taken rarely, they does not fall in closed frequent patterns with high support value. As a result, false positives are obtained on testing.

TABLE 5.4: Precision and recall of experiments conducted.

Support	**# of alerts made**	**Precision**	**Recall**	**Accuracy**
0.4	104	0.97	0.94	0.98
0.5	117	0.95	0.94	0.99
0.6	122	0.94	0.96	0.99

The life patterns of the children can be learned using the safe routes obtained by CloSPan. Figure 5.2 shows a situation where the daily route from home to school is

violated. The red, green and brown routes simulate the paths of the child taking similar routes to and from school. The violation of secure path can be seen on the route with light blue color.

FIGURE 5.2: An example of violation from detected secure paths. The map in the figure is displayed using Google Earth Pro (http://www.google.com/earth/).

Chapter 6

INDOOR POSITIONING

6.1 Motivation

Though current smart devices can be used in tracking the children's location outdoor using GPS or cell towers, they may fail in indoor positioning. There are several chances for a child to get lost indoors at crowded places, such as children sliding away from parents in large indoor places (e.g., mall) when they are involved in shopping or any other work. Thus an accurate indoor positioning method which can help to track children's indoor positions is also very important. In this section, an indoor positioning method utilizing Bluetooth is introduced.

6.2 Proposed Method

In our application, we propose an indoor positioning method to track a child's location using Bluetooth ID's and 4G LTE or Wi-Fi. The technique can be both semi-automated and full automated depending on whether the child is traveling alone or with the parents aside. The difference would fall in providing a unique *buildingID* manually in case of semi-automated mode. The indoor position tracking system we developed works on the coordination with the building owners subscribing to the application by providing the building map and coordinates on the map where the Bluetooth sensors are placed in the building. In the following, we describe how our developed indoor position tracking system works.

The requirements for the functioning of the tracking system falls into two sections: user requirements and subscriber requirements. The user requirements for the tracking system are Bluetooth and Wi-Fi or GSM connection. The subscribers require an image

of the building map, address of the building, Bluetooth devices, and the coordinates of the places on the map where they are going to place the Bluetooth devices.

The subscriber has to provide the image of the building map for each floor and the corresponding coordinates on the floor map where the Bluetooth chips are placed, along with their device ID's D_{id}. The coordinates on the image (x_i, y_i) can be easily found out with many free applications available online by uploading the image and touching on the screen at the place where the Bluetooth chips are placed. Also the address of the building should be provided. Once the subscriber provides the following details:

- Address of the building,
- Image of the building plan,
- Bluetooth ID's $BID = \{b_{id_1}, b_{id_2}, b_{id_3}, ...b_{id_n}\}$,
- Coordinates of Bluetooth on map $C_{bth} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ...(x_k, y_k)\}$,

then they are stored in the cloud with a unique *buildingID* created for each new subscriber as shown in Table 6.1 and the building ID is reported to the subscriber to advertise.

TABLE 6.1: A sample of building subscriber's record in the database.

Building ID	
Bluetooth ID	**Coordinates**
b_{id_1}	(x_1, y_1)
b_{id_2}	(x_2, y_2)
b_{id_3}	(x_3, y_3)
b_{id_4}	(x_4, y_4)

The procedure for tracking takes place in two ways: automated and semi-automated. They differ in fetching the *buildingID* of the building they enter. In the semi-automated mode the $buildingID$ is provided by the parent when they enter indoors. This is useful especially when the parent and the child are entering a building together. While in automated mode, the $buildingID$ is fetched automatically from the database with the GPS coordinates tracked using GSM or Wi-Fi.

Once the $buildingID$ is fetched, its corresponding map is pulled by the application from the database. The child application acts as the listener while the parent's application grabs data from the cloud to fetch child's location. That is, as soon as the child's application listen for a beacon containing the Bluetooth ID (b_{id}), it immediately notifies the cloud server with the $buildingID$ and Bluetooth ID(b_{id}) received. With the help of $buildingID$, the cloud pulls the subscribers record in the database and checks for

the Bluetooth ID(b_{id}) received. If the Bluetooth ID(b_{id}) is available in the database, its corresponding coordinates (x_i, y_i) are sent to the application along with the Bluetooth ID(b_{id}) as shown in Figure 6.1. The application will compare the sent and received Bluetooth ID(b_{id}) to see whether they are the same in the coordinates on the building map.

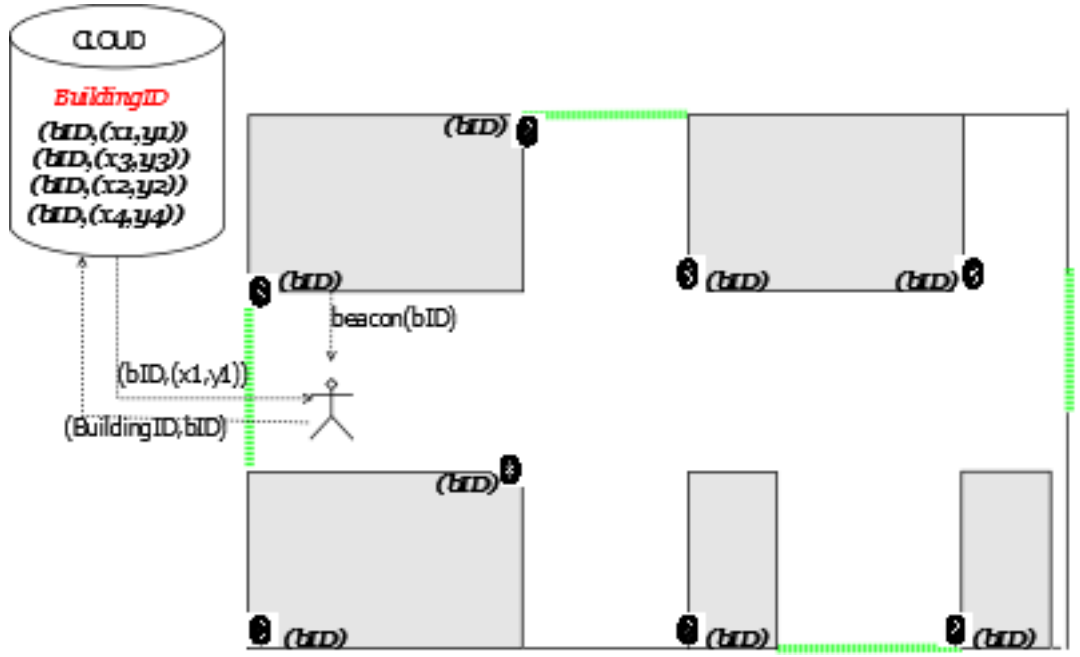


FIGURE 6.1: Indoor positioning system.

6.3 Experimental Results and Analysis

The experiment is conducted using mobile devices containing dual mode Bluetooth Low Energy (BLE) sensors. Each mobile device used contains unique name and device ID (bID). This experiment is conducted on Samsung, OnePlus, and Google Nexus devices. One device acts as the child application and other devices act as BLE devices placed in the building.

The experiment is conducted five times each in five different buildings with seven Bluetooth devices placed at a distance of fifteen meters and containing different floors. The distance of fifteen meters is chosen empirically so as to make sure the proximities of the devices doesn't overlap. For all the five buildings, their maps including all the floors, Bluetooth IDs and their corresponding coordinates are registered with the application before conducting the experiment. Each device in the building sends one beacon per second.

The application was able to detect the beacons of all the BLE devices as soon as it entered its proximity and place the location pointer at the corresponding coordinates obtained from the cloud with in two seconds approximately. Table 6.2 shows the Average distance error obtained from the experiment performed.

TABLE 6.2: Indoor positioning results.

Building No.	**Average Distance Error(meters)**
Engineering Sciences Building	12
Engineering Research Building	14
Evansdale Library	17
Allen Hall	13
Percival Hall	14

This experiment is performed as an initiative to acquire at least minimum positioning indoors. The moderate amount of accuracy is the problem experienced with this system. Since BLE devices are placed at large distances, no update of location is done until the next BLE device is detected. This can be significantly reduced by placing the BLE devices at proper distances making sure their proximities doesn't overlap. However, it's not easy to identify which direction the child has left from a particular point as the next update is obtained only after next BLE is detected. This drawback is relaxed to an extent by displaying both the last visited point and current point on the building map. Display of last visited location along with current location and placement of BLE devices at appropriate distances can provide better security to the child. Thus, the proposed system provides a cheap indoor positioning environment with reasonable accuracy.

Chapter 7

SYSTEM DESIGN AND DEVELOPMENT

Based on the android platform, a prototype of the application for both children and parents is developed incorporating with the proposed techniques in this thesis. In this chapter, we will present the system design and development of the application.

7.1 System Architecture

The primary goal of the developed system is to get the parents notified when their children's movement patterns are observed to be unusual or potentially dangerous. The architecture of the application is shown in Figure 7.1. It contains 3 components: child application, parent application and cloud server.

First, child needs to be registered in the application by providing corresponding parent's user name. On successful registration, child can login to the application with valid credentials. The child application will run in the background and continuously collect outdoor and indoor location information. The outdoor location information is obtained from GPS, GSM or Wi-Fi based on availability. On the other hand, indoor location information is obtained using Bluetooth beacons. Location information (indoor/outdoor) obtained as such is sent to the cloud database for further functionality.

The cloud server is the heart of the application. Cloud server manages all the users registered with the application, stores the location information from child's application, performs computation over the datasets and triggers notifications to parent's application. The system is mainly responsible for generating stay points on a daily basis. Once the users pass the threshold training time, the secure areas pertaining to each users are

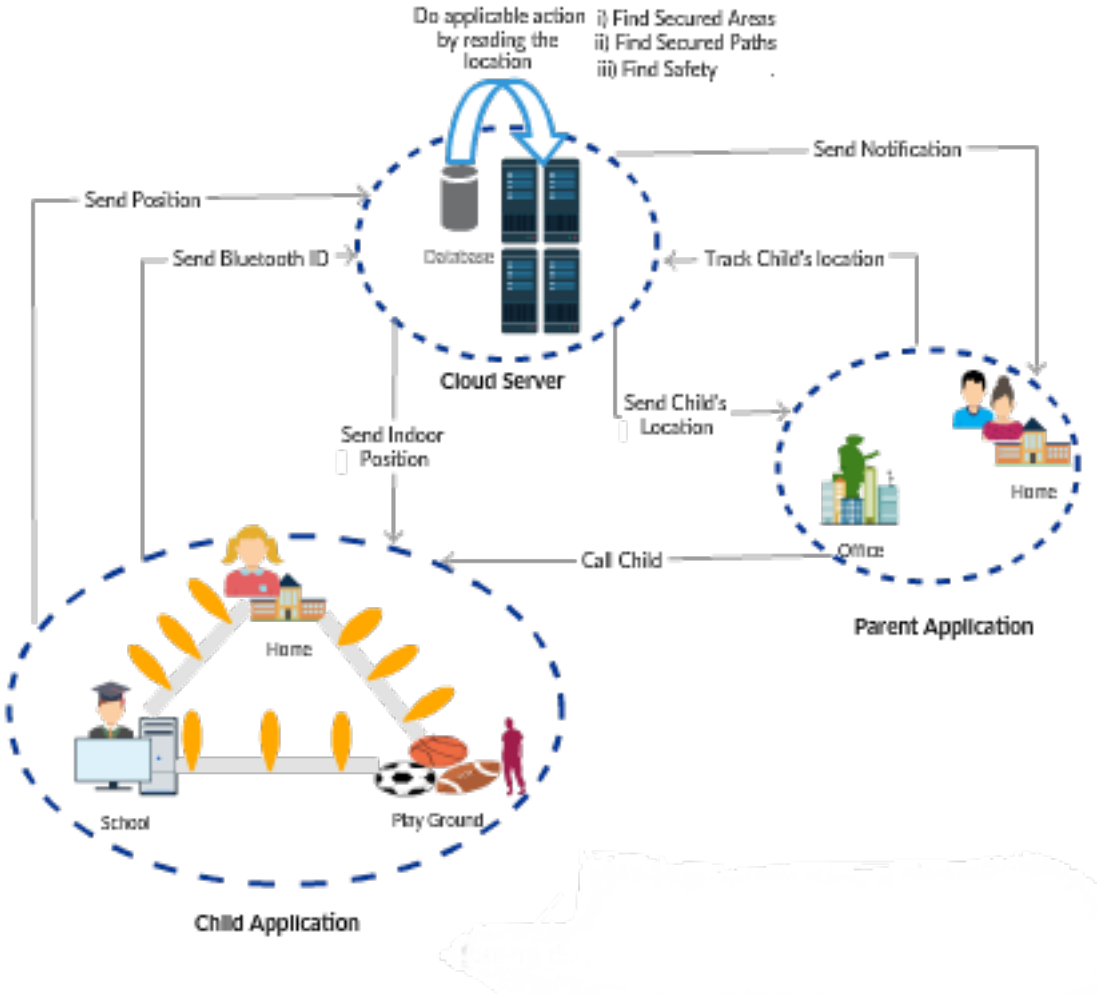


FIGURE 7.1: Architecture design of the application.

derived. Based on secure areas of the children, their corresponding secure paths are found. On detection of any unfamiliarity in the child's paths, the situation is reported to the parent's application.

The parent's application has the ability to monitor child's activity in real time. In the current busy world, the real time monitoring is not feasible, so we facilitate notifications in emergency situations. Upon receiving this alert, parent can communicate with child or report the situation to police through handy interface.

If the child has entered any building subscribed, indoor positioning is turned on displaying the map of the building and current position of the child. Whenever a next position is tracked, the immediate preceding position is also shown to let the parent know which path the child has used. A manual registration for indoors is also available based on the building ID for improved reliability.

We help to ensure child's safety by integrating all these components into one system which learns, detects, and notifies the location patterns. Figure 7.2 depicts the above described procedure through a data flow model.

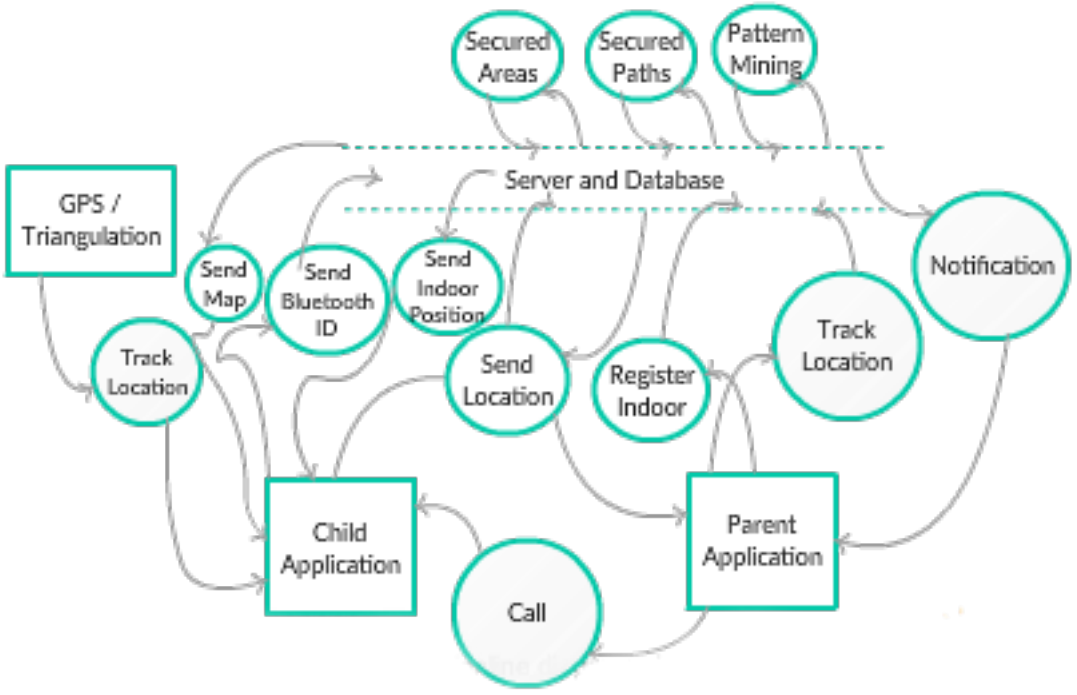


FIGURE 7.2: Data flow design of the application.

7.2 System Development and Applications

The application is built on Android platform using Android Studio and tested on One Plus 2, Samsung S5 Active, and Asus Nexus 7 devices. The application incorporates the Android's latest material design containing Cardviews and Recyclers. The application backend is built on Parse.net, a Mobile Backend as Service (MBaaS). Parse directly connects with android without any middleware and also provides a user interface for admin control of the application. It uses cloud code hooks for sending notifications to the application through cloud. Table 7.1 shows the software and hardware requirements for the application.

TABLE 7.1: Software and hardware requirements of the application

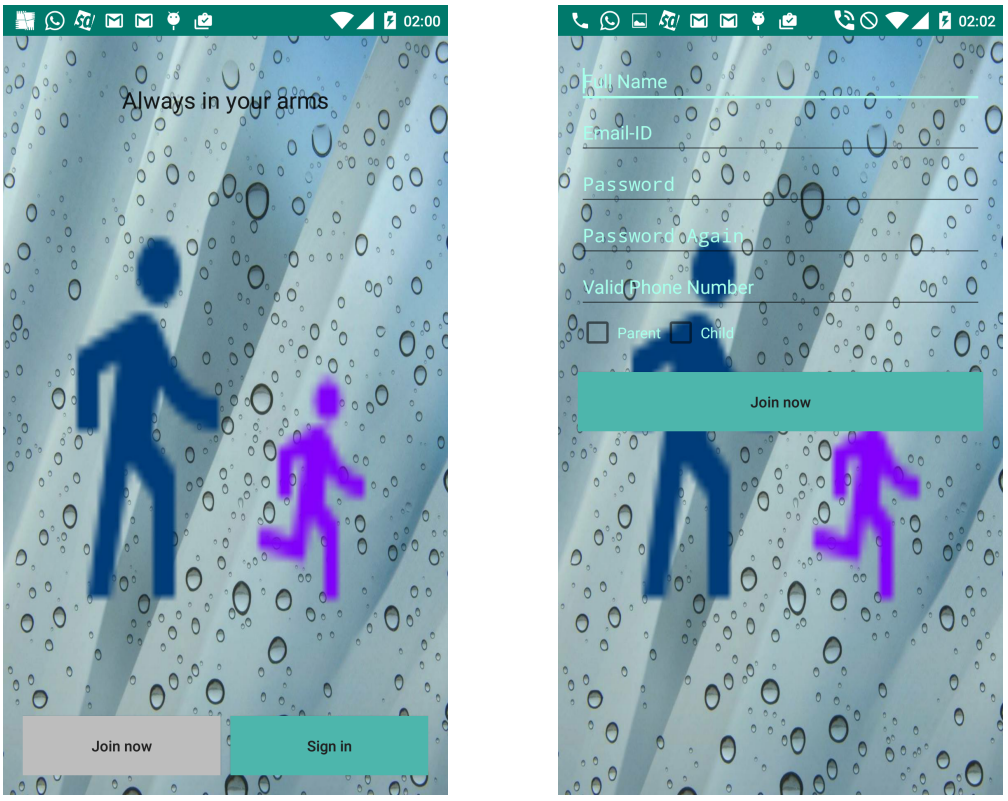
Resource	**Requirement**
Communications	GSM, 4G LTE, Wi-Fi, Bluetooth, GPS
OS	Android 4.3 - Android 6.0
RAM	2 GB minimum
Permissions	Location, Calls, Internet, Bluetooth
Space	30 MB

The interfaces of the application and the functionality of each component are described below.

7.2.1 Generic Module

The following screens are common to parent and child.

- *Welcome Screen:* Figure 7.3a shows the welcome screen of the application. It contains two buttons: Join now and Sign in. Join now is for the first time users to get registered with the application. Sign in is for the registered users to enter the application.
- *Sign up Screen:* Figure 7.3b shows the sign up screen of the application. In this screen, user provides information (Username, Email ID, Password, and Phone number) for the application. This screen also contains two checkboxes: parent and child. The user must select an appropriate checkbox. Once the user enters the information, all the information collected is placed in the Parse cloud database.

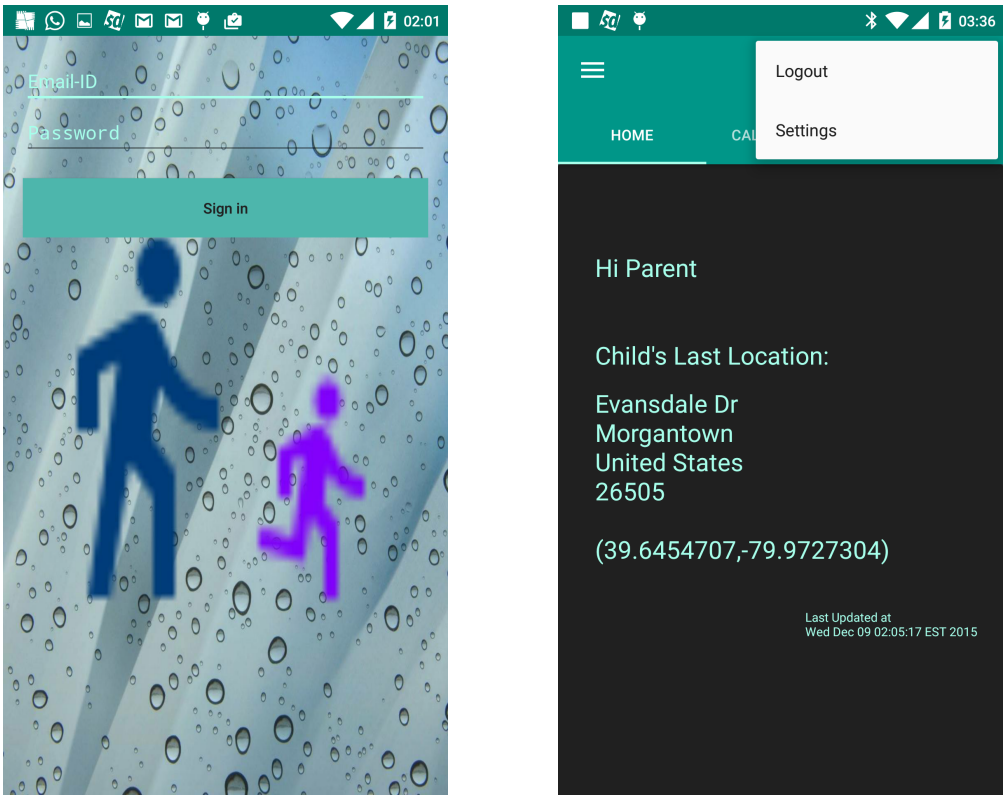


(A) Welcome screen. (B) Sign up screen.

FIGURE 7.3: Welcome and sign up screen.

- *Sign in Screen:* Figure 7.4a shows the sign in screen of the application. The user must enter username and password. Then the application detects if it is a child or parent with the username provided. It validates user based on username and password combination. After a successful login, user is redirected to home page of the application.

- *Logout:* A user can logout from the application through the logout option provided in the action bar as shown in Figure 7.4b.



(A) Sign in screen. (B) Logout screen

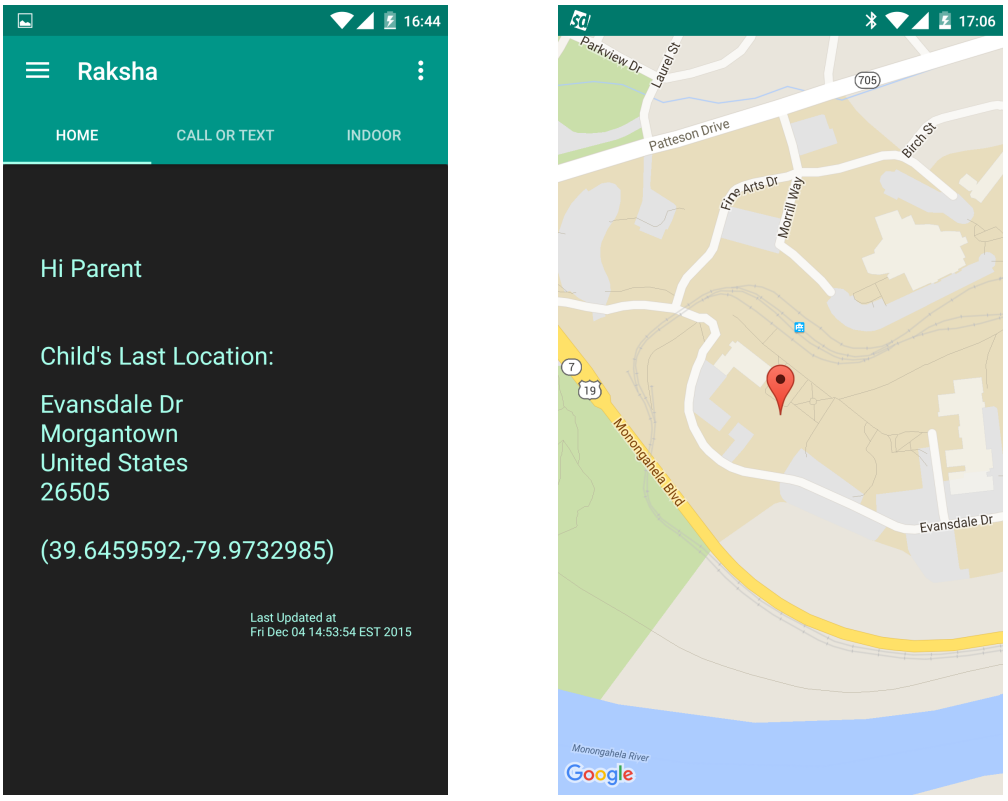
FIGURE 7.4: Sign in and logout screen.

7.2.2 Parent Module

The following are the main features provided to the parents:

- *Home Screen:* Figure 7.5a shows the home screen of the parent module. In this screen, parent can view the child's current location. Along with the location information of the child, the last updated time is also displayed. The parent can also view the location on Google map with a long press on the screen. This screen also provides navigation to other tabs where the parent can use available handy features.

- *Outdoor Map:* Figure 7.5b shows the corresponding map of the outdoor location displayed on home screen. The map is pulled up when the user long touches the location fragment of the home screen. The service used in displaying outdoor locations is Google maps API.

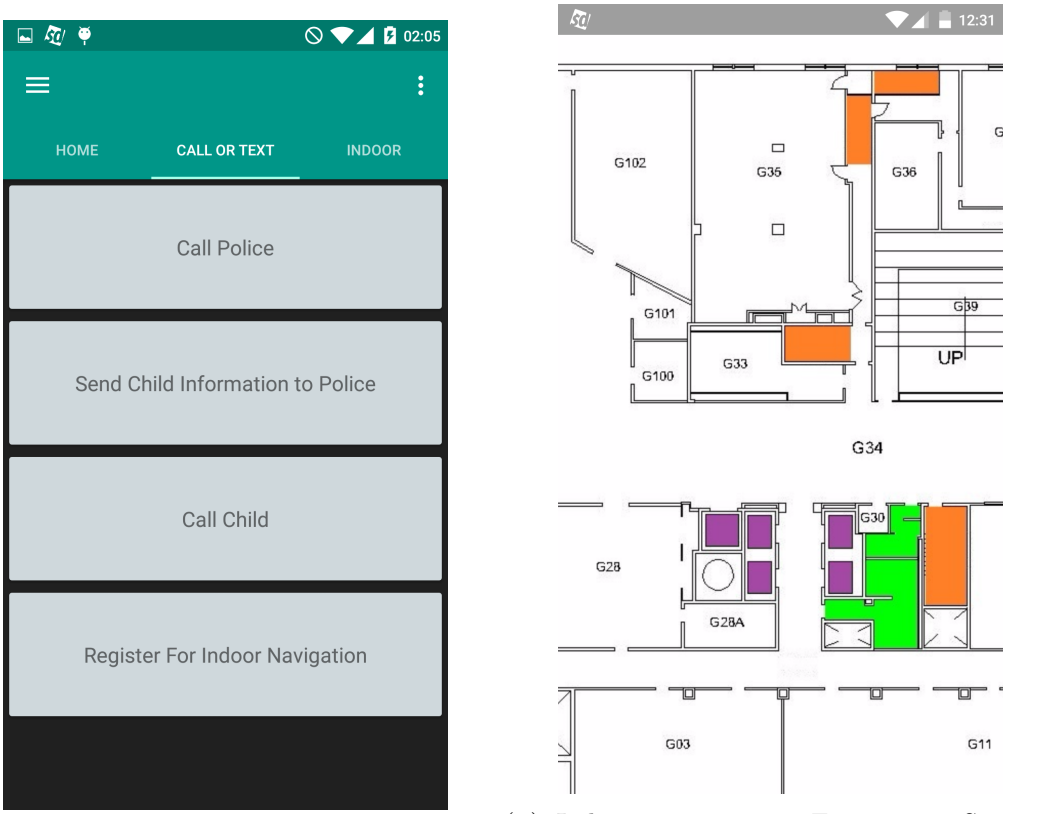


(A) Parent's home screen. (B) Outdoor location map.

FIGURE 7.5: Parent's home screen and outdoor map view. The map in the figure is displayed using Google maps (http://www.google.com/maps/).

- *Features Screen:* Figure 7.6a shows the handy features available in the application. The user can quickly call cops, send them a message with child's details and last tracked location if the child is in danger. It also includes features of call child/parent and register for indoor positioning.

- *Indoor Map:* Figure 7.6b shows the indoor map screen. The indoor map is pulled up automatically when the user enters indoor in automatic mode or through provided building ID in semi-automated mode.

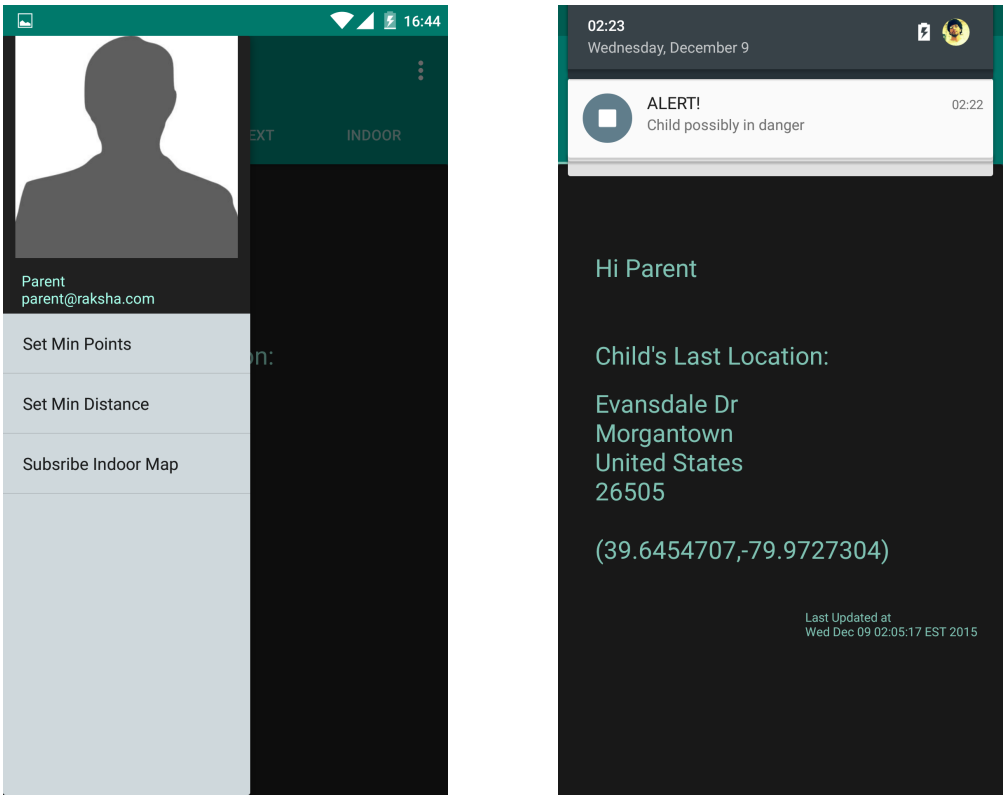


(A) Features screen.

(B) Indoor map screen: Engineering Sciences Building WVU.

FIGURE 7.6: Features Screen and Indoor Map.

- *Settings Screen:* Figure 7.7a shows the settings screen of the application. Here the user can set preferred minimum distance and minimum time parameters.
- *Notification:* A notification is displayed on the parent's device whenever the child is detected to be in danger as shown in Figure 7.7b



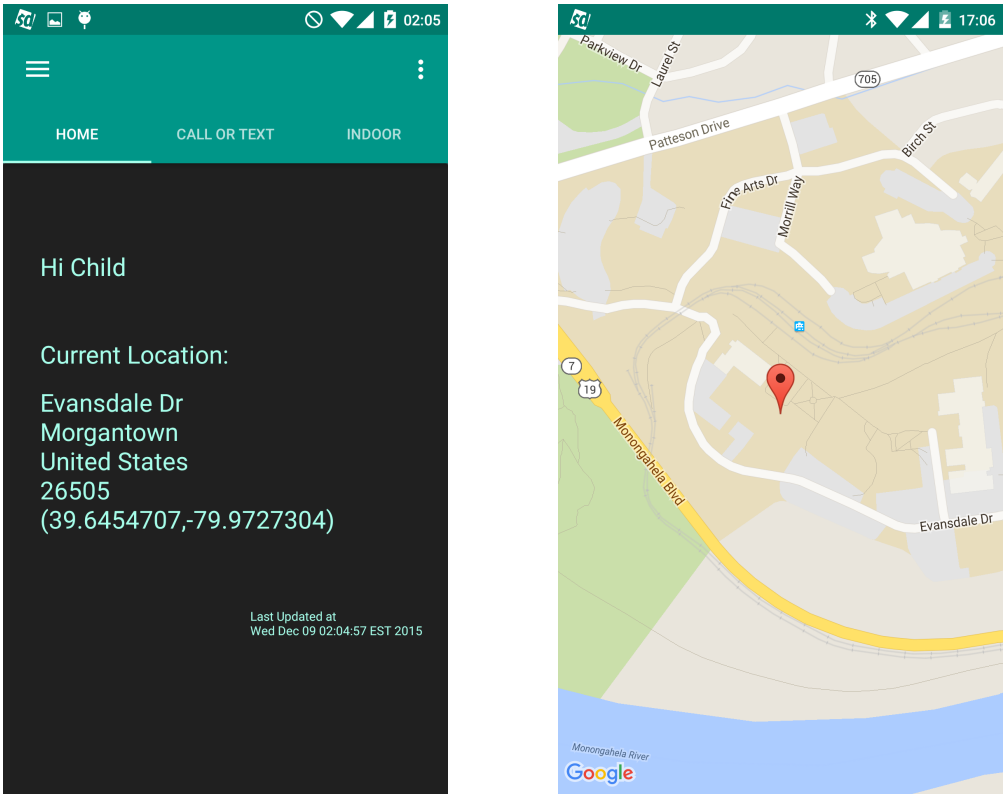
(A) Settings screen. (B) Notifications frame

FIGURE 7.7: Settings and notifications screen.

7.2.3 Child Module

The following are the main features provided to the child:

- *Home Screen:* Figure 7.8a shows the home screen of the child module. In this screen, the child's location tracked is displayed and updated to the parse cloud database. The child can also view the location on Google map with a long press on the screen as shown in Figure 7.8b.



(A) Child's home screen. (B) Map view

FIGURE 7.8: Child's screen and map view. The map in the figure is displayed using Google maps (http://www.google.com/maps/).

Chapter 8

CONCLUSION AND FUTURE WORK

In this thesis, an efficient child tracker and safety predictor model with unconventional methodologies that grasp children's life patterns is presented. Based on the children's location histories, the proposed data mining framework uses clustering and sequential pattern mining techniques to detect the children's secure areas and secure paths. When the system predicts the children to be potentially unsafe (e.g., in a strange area or on a strange route), automatic reports will be sent to their parents. Furthermore, an indoor positioning method utilizing Bluetooth is also proposed. Based on the android platform, a prototype of the application for both children and parents is developed incorporating with the proposed techniques in this thesis. This technique can be applied on the applications which need an automated anti-loss tracking system. One such example that is applicable in the future would be the anti-loss for automatic driver less cars.

The application developed could only provide accuracy to some extent with the help of Bluetooth. Providing an accurate indoor tracking system using evolving technologies like Visual Light Communication and maintaining energy accuracy trade off over all would make this application more accurate and reliable. In our future work, we will further investigate on more accurate positioning. The application developed is limited to an android platform mobile devices. Implementing it on several other platforms like Windows and iOS would make it much platform independent. One of our future works will be making the application available for both Windows and iOS platforms.

Bibliography

[1] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM*, 22(0-89791-592-5):207–216.

[2] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.

[3] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM.

[4] Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM.

[5] Bekkali, A., Sanson, H., and Matsumoto, M. (2007). Rfid indoor positioning based on probabilistic rfid map and kalman filtering. In *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, pages 21–21. IEEE.

[6] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[7] FBI. Ncic missing person and unidentified person statistics for 2014 pursuant to public law 101-647, 104 statute 4967, crime control act of 1990 requirements. `https://www.fbi.gov/about-us/cjis/ncic/ncic-missing-person-and-unidentified-person-statistics-for-2014`. Accessed: 2015-12-07.

[8] Gan, G., Ma, C., and Wu, J. (2007). *Data clustering: theory, algorithms, and applications*, volume 20. Siam.

[9] Gao, Y., Yang, Q., Li, G., Chang, E. Y., Wang, D., Wang, C., Qu, H., Dong, P., and Zhang, F. (2011). Xins: The anatomy of an indoor positioning and navigation

architecture. In *Proceedings of the 1st International Workshop on Mobile Location-based Service*, MLBS '11, pages 41–50, New York, NY, USA. ACM.

[10] Guha, S., Rastogi, R., and Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE.

[11] Han, J. and Kamber, M. (2001). Data mining: concepts and techniques.

[12] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., and Hsu, M.-C. (2000). Freespan: frequent pattern-projected sequential pattern mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359. ACM.

[13] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):pp. 100–108.

[14] Hinneburg, A. and Keim, D. A. (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering.

[15] Horozov, Tzvetan, Narasimhan, Nitya, Vasudevan, and Venu (2006). Using location for personalized poi recommendations in mobile environments. In *Proceedings of the International Symposium on Applications on Internet*, SAINT '06, pages 124–129, Washington, DC, USA. IEEE Computer Society.

[16] Kakuda, Y. and Inoue, S. (2011). New safety support system for children on school routes using mobile ad hoc networks. *IEICE transactions on communications*, 94(1):18–29.

[17] Kakuda, Y., Ohta, T., Inoue, S., Kohno, E., and Akiyama, Y. (2009). Performance improvement of hiroshima city children tracking system by correction of wrong registrations on school routes. In *Autonomous Decentralized Systems, 2009. ISADS '09. International Symposium on*, pages 1–5.

[18] Kang, J. H., Welbourne, W., Stewart, B., and Borriello, G. (2004). Extracting places from traces of locations. *ACM*, 72(103043):110–118.

[19] Kaufman, L. and Rousseeuw, P. (1987). *Clustering by means of medoids*. North-Holland.

[20] Kaufman, L. and Rousseeuw, P. J. (1990). Divisive analysis (program diana). *Finding Groups in Data: An Introduction to Cluster Analysis*, pages 253–279.

[21] Kaufman, L. and Rousseeuw, P. J. (2009). Agglomerative nesting (program agnes). *Finding Groups in Data: An Introduction to Cluster Analysis*, pages 199–252.

[22] Krumm, John, Horvitz, and Eric (2006). Predestination: Inferring destinations from partial trajectories. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, UbiComp'06, pages 243–260, Berlin, Heidelberg. Springer-Verlag.

[23] Krumm, J. and Horvitz, E. (2004). Locadio: Inferring motion and location from wi-fi signal strengths. In *in First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous.*

[24] Lin, M.-Y. and Lee, S.-Y. (1998). Incremental update on sequential patterns in large databases. In *Tools with Artificial Intelligence, 1998. Proceedings. Tenth IEEE International Conference on*, pages 24–31. IEEE.

[25] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[26] Michalski, R. S. and Stepp, R. E. (1983). *Learning from observation: Conceptual clustering.* Springer.

[27] Montoliu, Raul., Gatica-Perez, and Daniel (2010). Discovering human places of interest from multimodal mobile phone data. In *Proceedings of 9th International Conference on on Mobile and Ubiquitous Multimedia (MUM,',','), Limassol, Cyprus.*

[28] Moon, T. K. (1996). The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60.

[29] Mori, Y., Kojima, H., Kohno, E., Inoue, S., Ohta, T., Kakuda, Y., and Ito, A. (2011). A self-configurable new generation children tracking system based on mobile ad hoc networks consisting of android mobile terminals. In *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, pages 339–342. IEEE.

[30] Morii, K., Taketa, K., Mori, Y., Kojima, H., Kohno, E., Inoue, S., Ohta, T., and Kakuda, Y. (2012). A new generation children tracking system using bluetooth manet composed of android mobile terminals. In *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*, pages 405–407. IEEE.

[31] Nathan, E. and Alex, P. (2006). Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268.

[32] Paek, J., Kim, K.-H., P.Singh, J., and Govindan, R. (2011). Energy-efficient positioning for smartphones using cell-id sequence matching. In *MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 293–306. ACM New York, NY, USA ©2011.

[33] Patterson, D. J., Liao, L., Fox, D., and Kautz, H. (2003). Inferring high-level behavior from low-level sensors. In *UbiComp 2003: Ubiquitous Computing*, pages 73–89. Springer Berlin Heidelberg.

[34] Pei, J., Han, J., Mortazavi-Asl, B., and Pinto, H. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. *ACM*, pages 215–225.

[35] Pei, J., Han, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM*, 29(1-58113-217-4):1–12.

[36] Prentow, S., T., Blunck, Henrik, Kjærgaard, B., M., Stisen, Allan, Grønbæk, and Kaj (2014). Accurate estimation of indoor travel times: Learned unsupervised from position traces. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS '14, pages 90–99, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[37] Radaelli, L. and Jensen, C. S. (2013). Towards fully organic indoor positioning. In *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA '13, pages 16–20, New York, NY, USA. ACM.

[38] Rijsbergen, C. (1979). Van: Information retrieval. *London: Butterwoths.*

[39] ryan et al., J. D. (2013). National center for missing and exploited children: Annual report 2013.

[40] Sahinoglu, Zafer, Gezici, Sinan, Guvenc, and Ismail (2008). Ultra-wideband positioning systems. *NewYork: Cambridge University, Inc.*

[41] Saranya, J. and Selvakumar, J. (2013). Implementation of children tracking system on android mobile terminals. In *Communications and Signal Processing (ICCSP), 2013 International Conference on*, pages 961–965. IEEE.

[42] Sheikholeslami, G., Chatterjee, S., and Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB*, volume 98, pages 428–439.

[43] Shen, B., Zheng, Q., Li, X., and Xu, L. (2015). A framework for mining actionable navigation patterns from in-store rfid datasets via indoor mapping. *Sensors*, 15(3):5344–5375.

[44] Slimani, T. and Lazzez, A. (2013). Sequential mining: patterns and algorithms analysis. *arXiv preprint arXiv:1311.0350*.

[45] Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021. Springer.

[46] Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *ACM SIGMOD Record*, volume 25, pages 1–12. ACM.

[47] Takeuchi, Y. (2005). An outdoor recommendation system based on user location history. Master's thesis, School of Frontier Sciences, University of Tokyo.

[48] Wang, J. and Han, J. (2004). Bide: Efficient mining of frequent closed sequences. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 79–90. IEEE.

[49] Wang, K. and Tan, J. (1996). Incremental discovery of sequential patterns. In *1996 ACM SIGMOD Data Mining Workshop: Research Issues on Data Mining and Knowledge Discovery (SIGMOD'96)*, pages 95–102.

[50] Wang, W., Yang, J., Muntz, R., et al. (1997). Sting: A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, pages 186–195.

[51] Weston, D. C. and Barney, J. A. (2004). Method of game play using rfid tracking device. US Patent 6,761,637.

[52] Xiang, Zhe, Song, Song, Chen, Jin, Wang, Hao, Huang, Jian, Gao, and Xingxin (2004). A wireless lan-based indoor positioning technology. *IBM Journal of research and development*, 48(5.6):617–626.

[53] Yan, X., Han, J., and Afshar, R. (2003). Clospan: Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177.

[54] Ye, Y., Li, T., and Shen, H. (2015). Soter: Smart bracelets for children's safety. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(4):46.

[55] Ye, Y., Zheng, Y., Chen, Y., Feng, J., and Xie, X. (2009). Mining individual life pattern based on location history. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 1–10. IEEE.

[56] Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1573-0565):31–60.

[57] Zaki, M. J. and Meira Jr, W. (2011). Fundamentals of data mining algorithms.

[58] Zhang, T., Ramakrishnan, R., and Livny, M. (1996). Birch: an efficient data clustering method for very large databases. In *ACM SIGMOD Record*, volume 25, pages 103–114. ACM.

[59] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA. ACM.