www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 7296-7302

A Mobile Approach Applied To Public Safety In Cities

***S.P.Maniraj*[1], *R.Thamizhamuthu*[2]**

[1]SRM University, School of Computer Science and Engineering,
Ramapuram, Chennai, India
spmaniraj@gmail.com

[2] Muthayammal Engineering College, School of Computer Science and Engineering,
Salem, India
tamilr52@gmail.com

Abstract: ***As the Population is increasing day by day natural and man-made disasters have become an important factor for public safety. The ultimate goal of defining crime and safety indexes is to provide users with safety advisory information. People are however not equally exposed and vulnerable to all crime types. Age, gender and an array of personal features, preferences and choices play a central role on the perception of an individual's safety. In this paper we design, implement and deploy an application that retrieves and conveys to the user relevant information on the user's surrounding. We propose to achieve this vision by introducing a framework for defining public safety. These information may not be readily accessible, we use the localization capabilities of a user's mobile device to periodically record and locally store the trajectory traces with which future crime index may be predicted. Time series analysis is one of the forecasting techniques has been used in order to predict future safety values. The combination of space and time indexed crime datasets, with mobile technologies has been investigated to provide personalized and context aware safety recommendations for mobile network users. The trajectory trace of the user is used to define the chance of crime to occur around the user and generalize this approach to compute the chance of a crime to occur around groups of users.***

Keywords: Context aware safety, Autoregressive Integrated Moving Average, Code Division Multiple Access, Android Virtual Device

1. Introduction

Safety is an issue of particular concern: While billions of dollars are invested annually, a significant reduction in crime levels has not been successfully achieved. We use the combination of mobile technologies and online social networks to address this problem. The overarching goal of our work is to make mobile device users aware of the safety of their surroundings. Deployment of a system is done where users are made aware of their safety in a personalized manner, through everyday experiences such as navigation, mobile authentication, choosing a restaurant or finding a place to live. Hence, to achieve this vision by introducing a framework for defining public safety is introduced. Intuitively, public safety aims to answer the question "Will location L present any danger for user A when she visits L at a future time T"? [1], [2].

An important challenge to achieving this vision is the need to properly understand and define safety. While safety is naturally location dependent, it is also inherently volatile. It not only exhibits temporal patterns (e.g., function of the season, day of week or time of day]) but also depends on the current *context* (e.g., people present, their profile and behavior). Furthermore, as suggested by the above question, public safety has a personal dimension: users of different backgrounds are likely to be impacted differently by the same location/time context [3].

In this paper we investigate the combination of space and time indexed crime datasets, with mobile technologies to provide personalized and safety recommendations for mobile users. Specifically, we first define location centric, static crime and safety labels, based on recorded crime events. We take advantage of observed crime behavior periodicities, to conjecture that location safety values are predictable. To verify this hypothesis, we investigate the ability of time series forecasting tools to predict future location crime and safety index values based on recorded crime events [4].

A user 'U' is safe at a location 'L', if the average crime index of the locations in U's trajectory trace equals or exceeds the crime index predicted for the near future at L[5]. When insufficient crime information exists at a given location, we propose to augment the "context" of the location with data collected from co-located mobile devices. The framework consists of a service provider and mobile device users. The service provider, denoted by S in the following, centralizes crime and census information and provides it upon request to clients. We assume a semi-honest, or honest-but-curious service provider. That is, the service provider is assumed to follow the protocol correctly, but attempts to learn as much user information as possible. Users take advantage of Internet connectivity to retrieve crime information.

The historical database of large number of crime incidents reported is used. Each record is labeled with a crime type (e.g., homicide, larceny, robbery, etc), the time and the geographic location where it has occurred. Since records come from different Police departments, we classify crimes into 7 categories: Murder, Forcible Rape, Aggravated Assault, Robbery, Larceny/Theft, Burglary/Arson and Motor Vehicle Theft.

We introduce iSafe, a distributed algorithm that addresses privacy concerns raised by the use of trajectory traces and associated crime and safety index values. iSafe takes advantage of the wireless capabilities of mobile devices to compute real-time snapshots of the safety profiles of close-by users in a privacy preserving manner. iSafe uses secret splitting and secure multi-party computation tools to aggregate the trajectories of co-located users without learning the private information of participants.

We have extensively evaluated Android and browser plug-in implementations of iSafe, using crime and census data from the Miami-Dade County (FL) as well as data we have collected from the accounts of users and businesses in Yelp [6]. Our conclusion is that iSafe is efficient: even on a Smartphone, the computation and communication overheads are a few hundred milliseconds. The iSafe project can be found online [7], providing downloadable Chrome plug-in and Android app executables.

2. Model and Background

We consider a framework consisting of three participants, (i) a service provider, (ii) mobile device users and (iii) geosocial networks. The service provider, denoted by S, centralizes crime and census information and provides it upon request. We assume that the mobile devices are equipped with wireless interfaces, enabling the formation of transient, ad hoc connections with neighboring devices. Devices are also equipped with GPS interfaces, allowing them to retrieve their geographic location. Devices have Internet connectivity, which, for the purpose of this work may be intermittent. Users take advantage of Internet connectivity not only to communicate with the geosocial networks but also to retrieve safety information (both described in the following). Each user needs to install an application on her mobile device, which we henceforth denote as the client.

a. Geosocial networks (GSNs): Easy Tracker is an automated system that assists small public or private transit agencies in deploying bus tracking and arrival time prediction. This demo will showcase how data from GPS sensors embedded in Smart phones can be automatically processed in order to accurately estimate routes, bus stop locations, schedules, and make annotated maps with real-time bus tracking and arrival time predictions. We will also demonstrate a website portal which transit agencies can use to further interact with their bus transit systems. In addition to real-time tracking, after the set of routes and bus stop locations are estimated, the server also performs real-time route matching and classification.

Each bus is classified as following a certain route from the set of possible routes, and this information is displayed on the web inter-face by color-coding each bus icon [8]. Further-more, the server generates bus arrival time predictions for each bus stop and shows the estimated arrival times on the web interface. This real-time information enables bus riders to use the transit system more efficiently, reduce wait time, and increase overall safety and convenience.

The utility of a real-time tracking and bus arrival time prediction system is quantified. In order to estimate routes, bus stops and schedules, the devices are left in the buses (without any further driver input or interaction with the device) while the driver follows her usual route, stops at bus stops in order to pick-up or drop-off passengers, and maintains a schedule or a headway interval. Once a sufficient amount of data is collected on the back-end server, the system automatically processes the unlabeled GPS traces in order to create accurate routes, bus stops, and schedules for each route in the transit system [9].

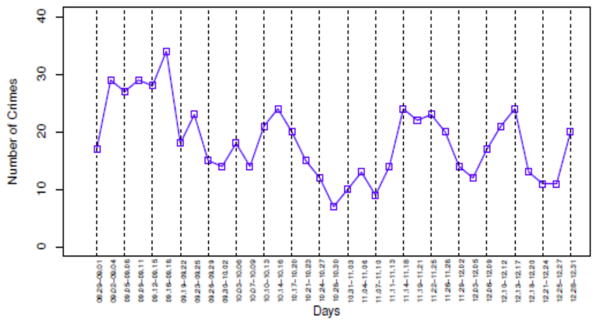

Figure1: Number of Crime reported

b. Forecasting and Error Measurement Tools: We rely on time series forecasting tools, including Auto Regressive Integrated Moving Average (ARIMA), Florida Criminal Punishment Code (FCPC), Linear (Double) Exponential Smoothing (LES) and Artificial Neural Networks (ANN). The supplemental material briefly describes each tool. Furthermore, we use the root mean squared error (RMSE) and mean absolute percent error (MAPE) [10] as error measurement metrics to evaluate the accuracy of the models considered.

The main concept is to make people safe using mobile network. The objective of this project is to make citizens aware of their surroundings using mobile devices through which he can view the places where crime has taken and also the current situation of the location. Forecasting tools has been used in order to predict future crime based on the past crime report. The novel approaches is proposed to define location and user based safety metrics and then evaluating the ability of existing forecasting techniques to predict future safety values. The disadvantages encountered in this method are the safety of the citizens is not maintained in a proper manner and the information regarding the user has not been preserved properly because of the use of social media.

3. Location Based Safety

The system such as an attempt to make people safety-aware by the use of mobile devices. User can search for any location to get the details of crime that has taken place and also he can view current condition of that place. Time series forecasting tool has been used in order to predict future crime. It generally figure 1 focuses on collecting location- and time-aware data.

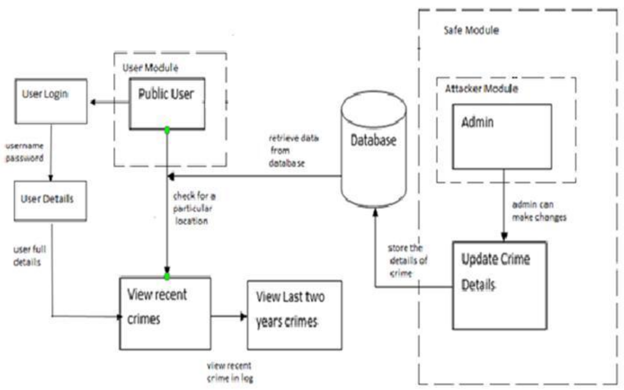

Figure 2: Architecture Diagram for Safe City application

We use a historical database of large number of crime incidents reported in a certain place (e.g.: Miami Dade county area) since 2007. The records are labelled with a crime type, the time and the geographic location where it has occurred (e.g., homicide, larceny, robbery, etc.). We briefly document two problems we encountered when pre-processing this data. Firstly, the crime type labels are non-uniform, since records come from different Police departments, (e.g., *murder* in Miami Beach vs. *homicide* in North Miami). Secondly, crime reports include many minor incidents (e.g., fire alarms issues), resulting in over 140 different crime types.

We mapped crimes into 7 categories in order to standardize and eliminate ambiguities: Murder, Forcible Rape, Aggravated Assault, Robbery, Larceny/Theft, Burglary/Arson, and Motor Vehicle Theft. Minor crime reports were removed that did not fall into these categories. Manual mapping was infeasible, due to the large number of records in the database. We have experimented with two machine learning techniques for classifying each record: the Naive-Bayes (NB)[11] classifier and the Decision Trees (DT) classifier. In order to build our training and test sets, we manually annotated a random sample of 2000 records which were collected from different police departments. Then, we split this subset of records into training and test datasets, each containing 1000 records. The accuracy was measured using a simple metric that measures the percentage of inputs in the test set that the classifier correctly labelled. For instance, a crime type classifier that predicts the correct crime type 60 times in a test dataset containing 100 crime types, would have an accuracy of 60%. On our crime dataset, the NB classifier achieved an accuracy of 91% and the DT classifier an accuracy of 98%. Since, DT classifier achieved a higher level of accuracy, so we have used the outcome of the DT classifier.

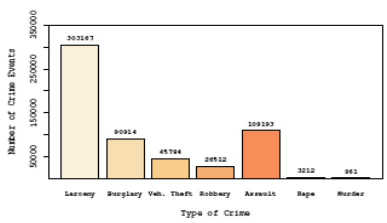

Figure 3: Outcome of DT classifier

This figure shows the crime set's distribution of the crime categories following the DT classification. Let c denotes the number of crime types. In our case, c = 7. Let CT = {CT1… CTc} denote the set of crime types. We use Census data sets [14], reporting population counts and demographic information. The data is divided into geographical extents e.g. polygons, called *census block groups*. Each block contains information about the population within (e.g., population count, various statistics).

a. Linear (Double) Exponential Smoothing (LES) Model (Exponential smoothing): Exponential smoothing is commonly applied to financial market and economic data, but it can be used with any discrete set of repeated measurements. The raw data sequence is often represented by {*xt*}, and the output of the exponential smoothing algorithm is commonly written as {*st*}, which may be regarded as a best estimate of what the next value of *x* will be. When the sequence of observations begins at time *t* = 0, the simplest form of exponential smoothing is given by the formulae.

$$\mathbf{S0 = X0}$$

$$\mathbf{St = \alpha Xt\text{-}1 + (1\text{-}\alpha)\ St\text{-}1,\ t>0}$$

Where α is the smoothing factor, and $0 < \alpha < 1$

It is also referred to as Brown's Linear Exponential Smoothing

b. Block crime and safety indexes: For a census block B and an epoch e denoted by the time interval ΔT, let C(B, ΔT) represent a c-dimensional vector, where the i-th entry denotes the number of crimes of type CT[i] recorded in block B during interval ΔT . Let $\overline{W}$ denote a c-dimensional vector of weights; each crime type of $\overline{\overline{CT}}$ (defined in Section II-B) has a weight proportional to its seriousness (defined shortly). Let BC (ΔT) denote the population count recorded for block B. We then define the *crime index* of block B during interval ΔT as,

TABLE 1: CRIME WEIGHT ASSIGNMENT

Crime Type	Weight
Assault	0.176
Robbery	0.180
Rape	0.307
Homicide	0.336

We need to assign meaningful weights to the crime types$\overline{\overline{CT}}$. An inappropriate assignment may make a large number of "lighter" offenses over shadow more serious but less frequent crime events, (e.g., consider larcenies vs. homicides). Assigning weights to crime types is also a subjective matter: certain people are more likely to be vulnerable to certain crime categories. In the following, we restrict our definition of safety to crimes against persons e.g., assault, robbery, homicide and rape and ignore crimes against property. Although our model can be applied to both categories, the focus of this work is on physical safety. We propose to assign each crime type a weight proportional to its seriousness, defined according to the criminal punishment code, i.e., the Florida Criminal Punishment Code (FCPC). The FCPC is divided into *levels* ranging 1-10, and each level *Lk* contains different types of felonies. The higher the level, the more serious is the felony. Each felony has a *degree* [12], (i.e., capital, life, first, second and third degree, sorted in decreasing order of seriousness), with an associated punishment (years of imprisonment). Let *Lk* denote the set of felonies within level k and let Pk denote the set of corresponding punishments. Let lk = |Lk|| denote the number of felonies within level k. Then, we define the weight of crime type CT[i], $\overline{wi}$ as

$$\overline{w}_i = \sum_{k=1}^{10} \rho_k \frac{P_k[i]}{\sum_{j=1}^{l_k} P_k[j]},$$

Where $\rho_k = k / \sum_{k=1}^{10} k$ k=1, k is the weight assigned to level k (normalized to the sum of the number of levels). The weight of crime type CT[i] is the weighted sum of the per-level punishment value Pk[i] associated with the occurrence of CT[i] within the felonies of level k, normalized to the total punishment of level k. Table I shows the resulting weights. Example: We study the impact of level L8 on the weight of the “Robbery” crime. Out of the felonies represented on level 8, two are related to “Robbery”: “Robbery with a weapon” and “Home-invasion robbery”. Both are first degree felonies, therefore punishable with up to 30 years of imprisonment. The other represented felonies are “Homicide”, with 6 different counts, for a total of 135 years penalty and “Rape”, with 1 count of up to 15 years penalty. Thus, the contribution of level 8 to the weight of “Robbery” is $\frac{8}{55} \times \frac{60}{60+135+15} = 0.0415.$

4. Predicting Safety

The crime index computation can only be performed for past epochs, when all crime events have been reported. Safety information however is most useful when provided for the present or near future. One way to compute the predicted crime index of a block B for the next epoch denoted by the interval ΔT , PCI(B, ΔT) is the average crime index of the block during the same epoch in the day for the past d days, where d is a system parameter (e.g., d=7 for 1 week of recorded per-block history). This solution however is unable to detect and factor in all crime periodicities, including seasonal, weekly and daily fluctuations. As such, it may include unnecessary errors – e.g., higher number of crimes in a past August may introduce inaccuracies in the crime index considered in the current month of April. We propose to address this issue through the use of the time series forecasting techniques discussed in Section II-C.

Specifically, we use time series forecasting tools to compute long and short term predictions of the number of crimes to be committed within an area (e.g., census block, zip code, city, etc.), based on the area’s recorded history.

a. Predicting crime and safety indexes: At the beginning of each epoch (denoted by the time interval ΔT), we compute predictions for the number of crimes of each crime type to be committed at each census block B during the epoch. Let PC (B, ΔT) [[i]denote the predicted number of crimes of type CT[i]. Using a formula we compute the predicted crime index for B during interval ΔT as

PCI (B, ΔT _=min {PC (B, ΔT) ⁄BC (ΔT), 1}

The predicted safety index is then **PSI (B, ΔT) =1-PCI (B, ΔT)**

b. Modified Safety: The ultimate goal of defining crime and safety indexes is to provide users with safety advisory information. People are however not equally exposed and vulnerable to all crime types. Age, gender and an array of personal features, preferences and choices play a central role in the perception of an individual’s safety. Since such information may not be readily accessible, we use instead the localization capabilities of a user’s mobile device to periodically record and locally store her trajectory trace. This enables us to define the crime index level with which a user is comfortable: the average crime index of the locations in her trajectory.

When enough crime information exists to enable the prediction of the near-future crime index of a location, we introduce the concept of *personalized safety*: the user is safe if her comfortable crime index level equals or exceeds the predicted crime index of her current location. However, crime information is not always available or detailed enough to allow a confident prediction of location crime index values.

c. Mobile Safety: We have implemented the location centric static safety labeling component of iSafe for a mobile application using Android. We used the Android Maps API to facilitate the location based service employed by our approach. We represent safety using three color labels ranging from green (safe) to red (unsafe).This ensures both (i) privacy – the user trajectory and her requests for block safety indexes never leave her phone and (ii) performance – frequent block safety index requests are performed locally, while infrequent census block safety index updates are performed periodically to ensure an accurate copy of the device’s cache [13].

The ultimate goal of defining crime and safety indexes is to provide users with safety advisory information. Once the mobile network is woven and connections are established, the information can be gathered from the network. For a mobile network, the crime details are needed to be updated timely by the service provider and not by the users. Hence, the malicious users cannot post incorrect information about the location. Reviews about the application can be posted and read by the user which other users can see and download the application.

5. Personalized, Context-Aware Safety

Every android application has its starting activity. In the default project it would be main activity class but since we are using SDK from Parse.com our starting activity differs from the default one. The structure of android project is mostly the same, but also may differ depending on the project needs and IDE tool. We will describe basic structure while using the ADT. When programmer uses ADT the project structure is generated automatically. Even further, ADT is also generating the ready- made application “Hello word”. The GUI version of ADT is the easiest way to create an Android project but the advance programmer can be also using the set of tools which can be run in terminal session. The terminal tool called “ant” can debug the Android project and create sample structure even if developer uses any other programming tool than Eclipse.

a. ARIMA Model: ARIMA models have been successful in forecasting time series in a variety of domains, including economics, marketing and sales, power systems, determine the appropriate model form, social problems etc. ARIMA incorporates autoregressive (p), integration (d) and moving average terms (q) to provide higher fitting and forecasting accuracy. ARIMA uses the input data to determine the appropriate model form. The ARIMA forecasting procedure consists of four steps, (1) identifying the ARIMA (p, d, q) structure, (2) estimating the unknown parameters, (3) fitting tests on the estimated residuals and (4) forecasting future outcomes based on the historical data.

The formulation of the ARIMA model depends on the characteristics of the series. Generally, it is originated from the autoregressive model AR (p), the moving average model MA (q) and the combination of AR (p) and MA (q), the ARMA (p, q) model. Like most time series, ours is non-stationary. Hence we cannot apply stationary ARIMA processes directly. One way of handling non-stationary series is to apply *differencing* (d) so as to make them stationary. Then, to find the best ARIMA model, we used the autocorrelation (ACF) and partial autocorrelation (PACF) functions for preliminary estimations of the AR (p) and MA (q) components. The ACF function is a set of correlation coefficients between the series and lags of itself over time while the PACF function is the partial correlation coefficients between the series and lags of itself over time. We use the Corrected Akaike Information Criterion (AICc) as the primary criterion in selecting the orders of a fitted ARIMA model which acts as an estimator of the expected discrepancy between the true model and a fitted candidate model. We choose the ARIMA model that has the minimum AICs value. We use T-statistics with 95% confidence interval to test the significance of the parameters in the fitted ARIMA model.

Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. Developers mostly choose the popular Eclipse Integrated Development Environments for development. The innovative Android is positioned well to confront the current challenge of mobile market place.

b. Android Software Development Kit: The Android SDK includes an emulator, some tools for performance poling and debugging. Eclipse IDE is natural choice for Android developers. Android Development tool (ADT) is a plug-in use to enhance and boost the performance of Eclipse IDE. It provides faster and easier way of creation and debugging of Android application. Note that further plugins are also available to support other IDEs such as IntelliJ and Net Beans for the Android developers.

c. Dalvik virtual machine: It is specially designed for Android platform and optimized for mobile devices, where resource constrains is an issue (like low memory, small size, and lower processing power). Dalvik is register based virtual machine and its interpreter is optimized for faster execution. Dalvik is capable of executing programs written in Java. It does not understand the java code directly, rather a dx tool is use to convert java code into byte code (which is then executed by Dalvik). The purpose of conversion java code into byte code is to optimize the code to be easily compiled over the limited resourced mobile device. Android support the execution of multiple instances of Dalvik VM simultaneously.

The Android system architecture comprise of four layers. The lowest of all is Linux kernel layer, used as an abstraction between hardware and the remaining software stack of Android. The basic reason to choose Linux 2.6 as kernel, as it is an open source and has proven driver model. It makes Android a robust operating system structure. Android rely on kernel for memory management, security model, network stack and process management. The Android current architecture relies onMSM7200A Qualcomm chipset for following features. *Android SDK* It is a software development kit that enables users to develop applications for Android platform. So to start developing applications you need download this kit first. Every Android version has a separate SDK released to enable developers to develop applications with latest features. Android ADK includes the below components API libraries Debugger Emulator Sample.

Android Emulator Android SDK includes a virtual mobile device to test Android applications without any real device called the emulator. This emulator has all the features of a real mobile device, but it cannot place any calls. Android emulator is basically an application that provides a virtual mobile device where you can run, test Android applications. These are some of the basic terms to be

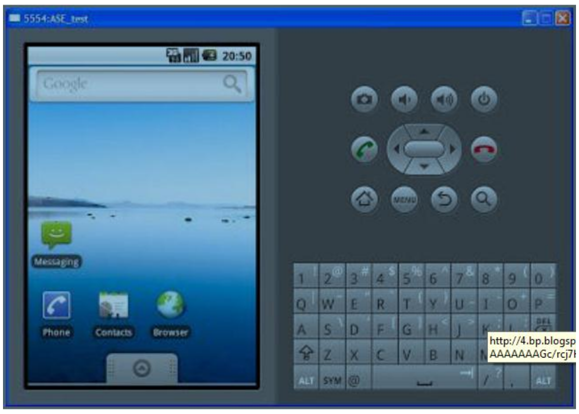

Figure 4 Android Emulator

The structure of android project is mostly the same, but also may differ depending on the project needs and IDE tool. We describe basic structure while using the ADT. Even further, ADT is also generating the ready- made application “Hello word”. The GUI version of ADT is the easiest way to create an Android project but the advance programmer can be also using the set of tools which can be run in terminal session. The terminal tool called “ant” can debug the Android project and create sample structure even if developer uses any other programming tool than Eclipse.

6. Evaluation Results

a. Android SDK: The figure (a) shows an Android device emulator. This emulator can be used to run an Android Virtual Device (AVD), which emulates a real Android phone. Such an emulator is displayed in the following screenshot AVDs allow you to test your Android applications on different Android versions and configurations without access to the real hardware. During the creation of your AVD you define the configuration for the virtual device. This includes, for example, the resolution, the Android API version and the density of your display. You can define multiple AVDs with different configurations and start them in parallel. This allows you to test different device configurations at once [14].

b. Recent Incidents Updation Page: The figure (b) is a representation of the window for recent updation of crime. In this, the admin can add new crime details by entering city name, place, date of crime and crime description.

c Report Page: The figure (c) is a representation of the report page window where a user can type a place name and thus, it will show the location where the crime has occurred and the relevant date and time.

d. Risk Indicator Page: The figure (d) is a representation of the risk indicator which indicates the safety level of a particular area with a green, yellow or red colour. Green indicates that the area is safe; yellow indicates less safe area and red indicates danger.

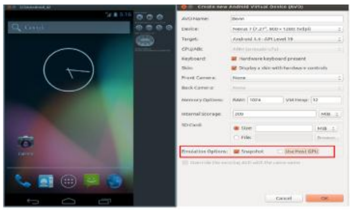

Figure a: Android SDK

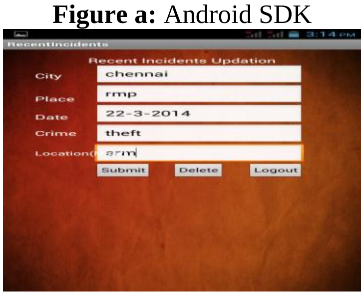

Figure b: Recent Incidents Updation Page

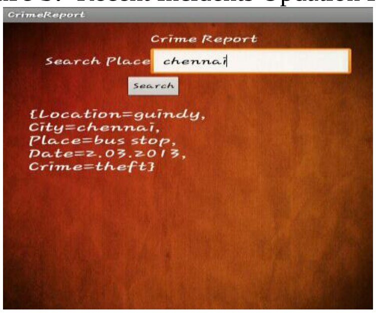

Figure c: Report Page

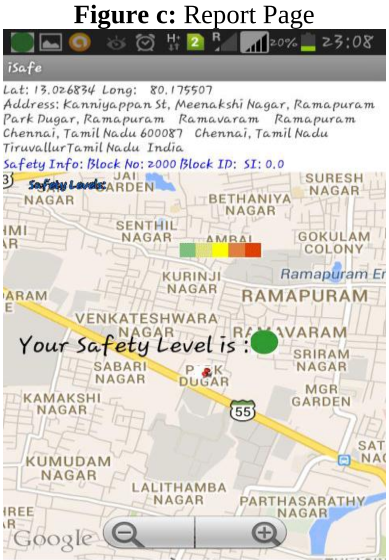

Figure d: Risk Indicator Page

7. Conclusion and Future work

Mobile Approach applied to Public Safety in cities is a new topic drawing attention because of the increasing rate in crimes in cities across the world. Previous methods implement the use of social media ,as opposed to which we proposed a system in which we have prohibited the use of social media because of its lack in providing safety to the users as personal identity of the users were revealed in the social networking sites. The proposed method can take advantage to maintain time as maps are not used which were used in the existing system as it consumes a lot of the user's time in the process of loading. Moreover, user information is preserved in the proposed system. Furthermore this noble method can be very useful in maintaining the safety of the citizens as safety is considered to be of utmost importance in the present time.

In future, safety to citizens can be provided by posting real time images and videos of crime events. Also, all types of information can be provided with extra level of safety to the users. In future, as soon as police authorities updates any crime in their database, automatically it gets updated in the database of safe city application.

8. References

[1]. Vasco Furtado, Leonardo Ayres, Marcos de Oliveira, Eurico Vasconcelos, Carlos Caminha, Johnatas DOrleans, and Mairon Belchior. Collective intelligence in law enforcement the wikicrimes system. Information Sciences, 180(1):4 – 17, 2010.

[2]. Gorr and A. Olligschlaeger. Crime hot spot forecasting: Modeling and comparative evaluation. Draft final report, U.S. Department of Justice, Office of Justice Program, National Institute of Justice, 2001.

[3]. S Chainey, S Reid, and N Stuart. When is a Hotspot a Hotspot? A Procedure for Creating Statistically Robust Hotspot Maps of Crime. Kidner, D and Higgs, G and White, S, 2002.

[4]. John E. Eck, Spencer Chainey, James G. Cameron, Michael Leitner, and Ronald E. Wilson. Mapping crime: Understanding hot spots. Special, U.S. Department of Justice, Office of Justice Program, National Institute of Justice, August 2005.

[5]. Jaime Ballesteros, Mahmudur Rahman, Bogdan Carbunar, and Naphtali Rishe. Safe Cities. A Participatory Sensing Approach. In Proceedings of the 37th IEEE International Conference on Local Computer Networks (LCN), 2012.

[6]. Anna Yu, Athanasios Bamis, Dimitrios Lymberopoulos, Thiago Teixeira, and Andreas Savvides. Personalized Awareness and safety with mobile phones as sources and sinks. In International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08), 2008.

[7]. A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In 8th ACM Conference on Embedded Networked Sensor Systems, pages 85–98, 2010.

[8] S. Chainey and J. Ratcliffe. GIS and Crime Mapping. Wiley, 2005.

[10] E. Jefferis. A multi-method exploration of crime hot spot: A summary of findings. Technical report, U.S. Department of Justice, Office of JusticeProgram, National Institute of Justice, 1999.

[9] N. Barberis, A. Shleifer, and R Vishny. A model of investor sentiment. Journal of Financial Economics, 49:307–243, 1998.

[10] Deborah L. Estrin. Participatory sensing: applications and architecture. In Proceedings of the 8th international conference on Mobile systems, applications, and services, 2010.

[11] H.Brian Hwarng and H.T Ang. A simple neural network for ARMA(p,q) time series. Omega, 29(4):319 – 333, 2001.

[12] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). Bulletin of the EATCS, 82:72–107, 2004.

[13] A. C. Tamhane and D. D Dunlop. Statistics and data analysis: From elementary to intermediate. Upper Saddle River, NJ: Prentice Hall, 2000.

[14] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

Author Profile

S.P.Maniraj received the B.E in Computer Science and Engineering from Anand Institute of Higher Technology in 2008 and M.Tech. Embedded System Technologies from Anna University Tirunelveli in 2010 respectively. He now with Assistant Professor in SRM University, Chennai.

R.Thamizhamuthu received the B.Tech in Computer Science and Engineering from Muthayammal Engineering College in 2009 and M.E. Computer Science and Engineering from GTEC in 2011 respectively.