Mobile GPS based Traffic Anomaly Detection System for Vehicular Network

Farrukh Arslan[#1], Bilal Wajid[*2], Haroon Shafique[*3]

[#]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

*EE Department, UET, KSK, Pakistan

***Abstract* — *The quick growth in the number of mobile devices such as smart phones, wearable devices, tablets, sensor enabled vehicles etc. with a large array of sensors like GPS, Accelerometer, Gyroscope, Compass, Magnetometer, Camera etc. enables a new sensing paradigm known as Crowd Sensing. Traffic anomalies can occur due to various events like accidents, functions, celebrations, protests, disasters etc. In this paper we propose an architecture that employs crowd sensing to detect traffic anomalies and uses social media data to determine the authenticity of identified anomalies. Our prototype architecture includes an Android based navigation application for the client and a combination of J2EE application server and Hadoop as the back-end. The client application consists of an interface to report traffic anomalies apart from the basic navigation features. Anyone using this app can report an anomaly that he encounters in his route. Whenever a user reports an incident, a tweet with the exact location and incident details are posted automatically to the twitter account managed by the application. Using Recursive EM Algorithm, the authenticity of the reported anomaly is verified and if it is genuine, all the users in that particular route will get notified in advance. The system will also suggest the best possible alternative route to the same destination. The system also provides a web interface for the traffic authorities to monitor the anomalies in their locality on a real-time basis and can respond to it very immediately. Hadoop based infrastructure which is deployed in the back-end is able to process massive GPS data collected from the users using MapReduce framework. The system has been tested successfully in a simulated environment using Android emulator and GPS Location Spoof application.*

***Keywords* — *Traffic Anomaly Detection, Truth Estimation,MapReduce, Crowd Sensing, Map-Matching.*

I. INTRODUCTION

In modern days, travel is an inevitable part of our dayto day life and the advancement in technology has broughtvarious navigation apps which assist the user to find thebest possible route to his destination. With the invention ofsuch travel apps, long distance travel is now very easy and funoriented. There are many such systems already available thatcan suggest the shortest and various alternative routes to thedestination based on various criteria such as traffic conditions,distance, time etc., but we do not have a system which candetect the traffic anomalies in advance and instruct the user totake the best alternative route preventing unnecessary wastageof time. The necessity to develop such an efficient systemto re-route the traffic after detecting the authenticity of thereported anomaly is the motivation for our paper work.

There are different situations which can cause trafficanomalies resulting in traffic jams on road networks thatotherwise operate efficiently. Here, we propose a method todetect traffic anomalies, which could happen by disasters, protests, accidents etc., verify the authenticity of suchtraffic anomalies and help auser navigate to his destination in a most efficient manner by providing a fool proof mechanism.This leads to a solution for such scenarios. The numberof vehicles are increasing day-by-day but the infrastructureremains the same, so that we cannot handle the ever increasingtraffic efficiently. In such a scenario, the chance for trafficanomalies to occur is very high and it is essential to developa system which supports the user to intimate heavy traffic in roadways. Our system consists of an Android based navigation app in the client side,where the user will get all sort of valid traffic anomalies.In case of anomalies, the system willautomatically suggest the best alternative route to the destination.The application also allows the user to report new incidentswhich will come across his way. GPS trajectory data collectedfrom different users are processed by Hadoop [1], which is issued as the back-end for storing and processing. The systemalso provides a web interface for traffic authorities to monitor the incidents in their locality.

Even though there are many such applications, an efficientand fool proof system with very good processing rate islacking. Most of such systems are using databases for storingthe trajectory data and are very slow in processing and decisionmaking. Processing speed and accuracy are very muchimportantfor an anomaly detection system. This is the motivation for ourproject and our application mainly focuses on high processingrate, efficiency in route suggestion and a fool proof mechanismto determine the genuineness of the incident reports.

II. RELATED WORKS

There are so many works related to traffic monitoringand anomaly detection. Some of the research methods arepurely based on mobile traffic data, some are based on thesocial media and another line of related work is based on acombination of the studies on traffic anomalies and the analysis of social media texts (e.g. [2]).

As per the approach mentioned in [2], in the first step, behaviour pattern of alarge number of drivers is detected and stored. In the second step, the current behaviour of thedrivers and their historical driving behaviour are compared. Thechanges in the routing behaviour are considered in addition tothe change in traffic volume and alternative route to avoid theanomaly is provided. Also, the social media website Twitter is mined to get a description about the traffic anomaly.

Our work is different in the sense that it does not need historical data and can be applied to any road network. Tweetsare used to estimate the truth value of the incident reports rather than to provide a description about the anomaly. Inour work we are using the Google Directions API [3] toobtain the available routes to destination. The shortest route todestination is always the preferred one and it is recommendedto the user. Rectangle method for map-matching implemented using Java Servlet and Hadoop MapReduce [1] framework areused to know the routes travelled by each user on a realtime basis and any deviation of the user from the optimalpath is observed. Anomaly detection and the truth estimation of reported incidents are performed in parallel and users aresuggested with best possible alternative routes.

Smart Traffic Cloud [4] presents a software infrastructure toenable traffic data acquisition, manage, analyse and to presentthe results in flexible, scalable and secure manner using a cloudplatform. We adopted the software infrastructure presented in[4] to meet our needs. Map-matching algorithm and traffic anomaly detection methods are executed in the Servlet beans.The results of map-matching are stored in HDFS using theMapReduce framework for analysis.

III. SYSTEM DESIGN

Our approach to solve the problem consists of anAndroid application which assists the users in finding routes to their destination and to report incidents. The clientAndroid application communicates with a J2EE server andHadoop for map-matching, traffic anomaly detection andtruth estimation. The system architecture is shown in Figure 1.

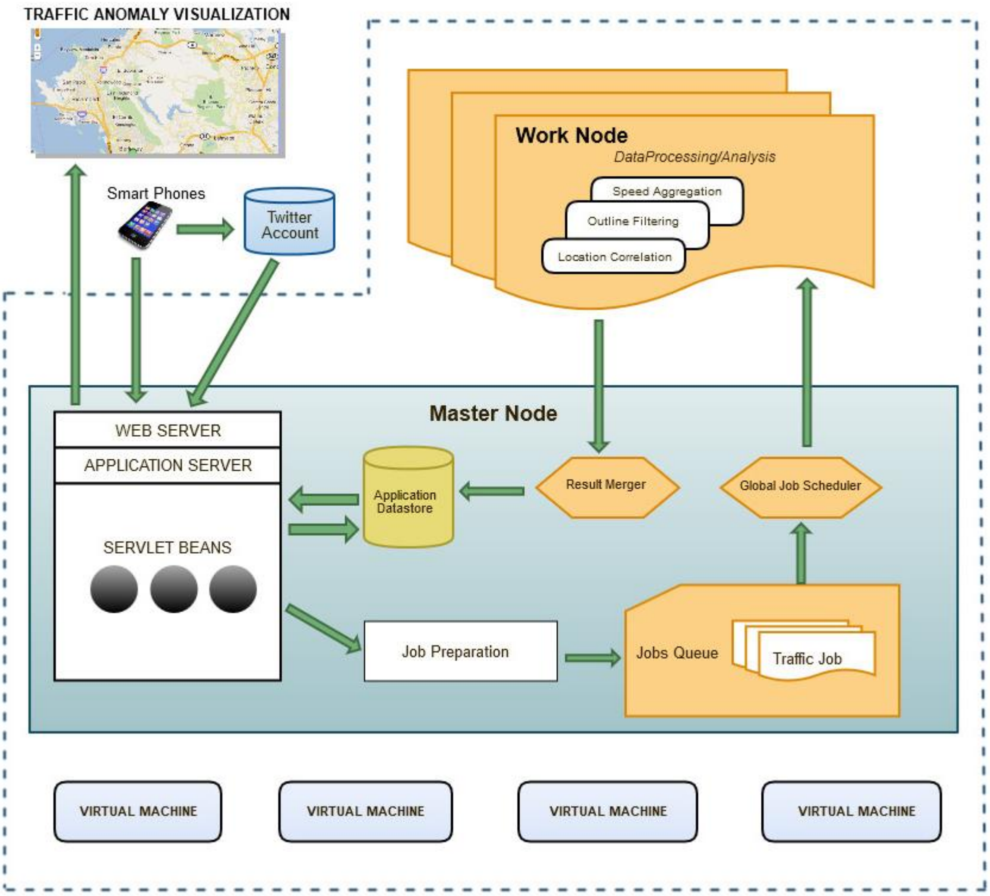

Fig. 1. System Architecture

A. Data Collection and Usage

Two different data sets are required by the system for traffic anomaly detection. The first set of data includes the GPS coordinates obtained from the mobile phones of the users that can be used to monitor the current route selected by the users on a real-time basis. The Android application installed in smart phones will send the GPS coordinates to the server at frequent intervals for map-matching. In the map-matching phase the GPS coordinates thus obtained are verified to check whether they lie on the best shortest path or on a different route. If the number of users on the best route is lesser than the alternative route then the chances of occurrence of anomaly is large. The GPS data collected from the users are in the following format:

([IMEI Number]: [Timestamp], [Latitude], [Longitude])

The IMEI Number of the mobile phone is used to uniquely identify each user so that the route travelled by each user can be tracked separately. The second set of data includes the incident reports by users which are basically tweets posted to the application's Twitter account. Each tweet includes a time stamp, location coordinates and incident type. The tweets for analysis can be retrieved using the Twitter API. These are analysed frequently using the Recursive Expectation Maximization Algorithm [5] to determine the truth value of the incidents reported. After truth estimation, if the incident reports are found to be genuine then all the users travelling in that particular route will be notified about the incident and they will be suggested an alternative route if available. The tweets posted to the application's twitter account are in the following format:

([Incident-type] reported at [Latitude], [Longitude] on [Timestamp]).

B. Android Application

In the proposed system users are provided with an Androidapplication [6] as shown in Figure 2 for navigation betweentwo points and also to report the incidents. Google DirectionsAPI [3] is used to find the routes from a user's currentlocation to destination and the route/s will be drawn in GoogleMaps activity of the application. Anyone with the applicationcan report incidents. Application provides an interface with various incident report types to report incidents. While reporting incidents a tweet containing the incident details will beautomatically posted to the applications twitter account. Theapplication is capable of alerting the users of any anomalies onthe way beforehand and suggests an alternate route if availableto avoid the anomaly.

C. Rectangle Method for Map Matching

In our system, we have used the rectangle method [7]for matching the collected GPS coordinates of the users withGoogle Maps to find any abnormalities in their routing pattern.Massive volume of GPS data needs to be handled for map-matching. The rectangle method is found to be practical with a very high accuracy rate evenfor low sampling rates.

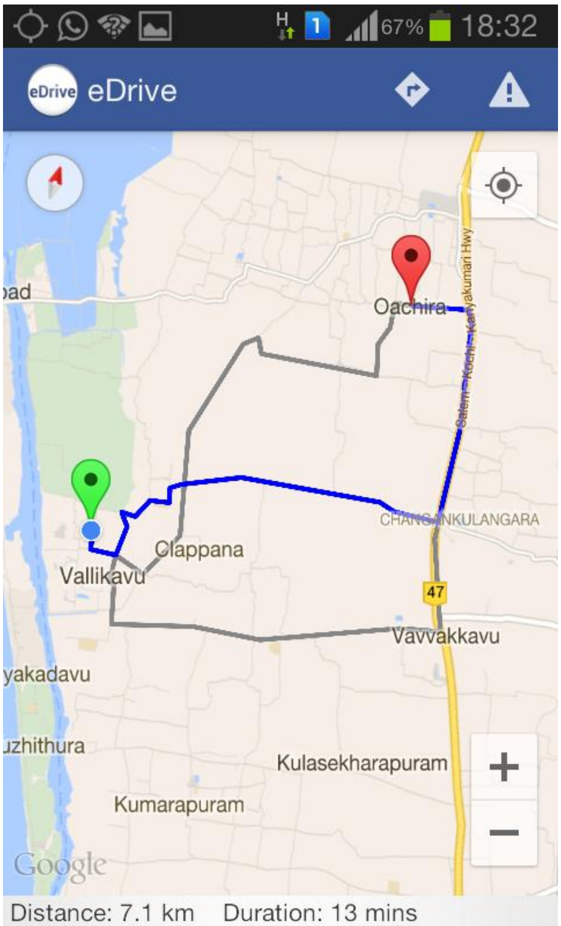

Fig. 2. eDrive Android application showing preferred route in blue andalternate routes to destination in grey

Therectangle method works by generating rectangles or boxes fora route in a finite number of steps mentioned below.

1) The whole route is covered by a grid with squarecells separated at a distance of 0.02 miles. The gridcovers all the cells from source to destination whichincludes all the line segments in the route. The gridincludes one extra cell than the total cells needed tocover the whole route in both directions.
2) In this step, all the cells that are touched by the routeline segments are identified. For this, the vertices ofthe route line segments are traversed and the cellswhich holds these vertices are selected and marked. If two

vertices are identified in disjoint cells, all theother cells that lie in between will also be marked.

3) Once a cell is marked, the surrounding eight cells arealso selected and marked.
4) Traversing all the vertices in the route guarantees allthe points in a specified distance to be included inany one of the marked cells.
5) This step merges the marked cells in to rectangleswhich does not overlap with each other. For this, two different methods are used. In the first method, thehorizontally adjoined cells are merged into rectanglesof larger width but with the height of a single cell.
6) All the rectangles are then compared with the rectangles created in the next row and are merged if theyare of same width and have horizontal position.
7) The second method uses a vertical approach justopposite to the horizontal one.
8) Once these two approaches are finished, the numberof rectangles created using these two methods are thencompared.
9) After comparison, the set of least number of rectangles are given to the application for further processing.
10) GPS coordinates obtained from the user ischecked to determine whether it lies within any ofthe rectangles generated in the previous step.
11) If the GPS coordinates lies within the rectangle thenthe user is travelling along the route else the user hasdeviated from the route.

D. Truth Estimation Problem

Determining the correctness of the anomalies reported by the users is the key challenge in all such systems and is commonly called as truth estimation problem. Anyone using the application can report an anomaly as shown in Figure 3 and the reliability of the person is usually unknown. While reporting, the application sends the current location of the user to the server to get the approximate location of the incident. Most of the fact finding algorithms [8] available to solve the problem can be applied on a batch of data and the entire algorithm needs to be re-executed on the whole dataset when a new data arrives. This is indeed not suitable for a GPS based traffic anomaly detection system because here the data arrives frequently. To overcome this difficulty, in our system, we adopted a streaming fact finder that updates the previous estimates recursively based on the new data arrived. This recursive Expectation Maximization algorithm [5] solves the truth estimation problem on a real-time basis and makes our system fool proof. The recursive formula for the fact finder algorithm on streaming data is derived in [5]. In theory of estimation, the following formula [9] estimates the parameterin a recursive fashion in successive time intervals.

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \{(k+1)I_c(\hat{\theta}_k)\}^{-1}\,\psi(X_{k+1},\hat{\theta}_k) \qquad (1)$$

Where $\hat{\theta}_k$is the estimation parameter up to the time interval k. The CRLB (Cramer Rao lower bound) of the estimation parameter at time k is represented by $I_c^{-1}(\hat{\theta}_k)$and the score vector of the observed data at time interval $k+1$ with estimation parameter $\hat{\theta}_k$is represented by $\psi(X_{k+1},\hat{\theta}_k)$. The estimation parameter in the new time interval $(\hat{\theta}_{k+1})$can be calculated recursively using the formula 1. Based on, the estimation vector $\hat{\theta}_k$is defined as $\hat{\theta}_k = (\hat{a}_1^K, \hat{a}_2^K, \dots \hat{a}_M^K; \hat{b}_1^K, \hat{b}_2^K, \dots \hat{b}_M^K)$ and $I_c^{-1}(\hat{\theta}_k)$is given as [10]:

$$I_c^{-1}(\hat{\theta}_k)\; i,j = \begin{cases} 0 & i \neq j \\ \dfrac{\hat{a}_i^k \times (1-\hat{a}_i^k)}{N \times d} & i = j \in [1, M] \\ \dfrac{\hat{b}_i^k \times (1-\hat{b}_i^k)}{N \times (1-d)} & i = j \in [M, 2M] \end{cases} \qquad (2)$$

and

$$\psi(X_{k+1},\hat{\theta}_k)_{i,j} = \begin{cases} 0 & i \neq j \\ \sum_{j=1}^{N} \hat{Z}_j^{k+1}\left(\dfrac{S_iC_j}{\hat{a}_i^k} - \dfrac{1-S_iC_j}{1-\hat{a}_i^k}\right) & i = j \in [1, M] \\ \sum_{j=1}^{N} (1-\hat{Z}_j^{k+1})\left(\dfrac{S_iC_j}{\hat{b}_i^k} - \dfrac{1-S_iC_j}{1-\hat{b}_i^k}\right) & i = j \in [M, 2M] \end{cases} \qquad (3)$$

The formula to recursively update the estimation parameter scan be derived by combining equations 1, 2 and 3 as follows:

$$\hat{a}_i^{k+1} = \hat{a}_i^k + \frac{1}{Nd(k+1)} \times \left[\sum_{j \in SJ_i^{k+1}} \hat{Z}_j^{k+1}(1-\hat{a}_i^k) - \sum_{j \in \bar{S}J_i^{k+1}} \hat{Z}_j^{k+1}\hat{a}_i^k\right]$$
$$\hat{b}_i^{k+1} = \hat{b}_i^k + \frac{1}{Nd(k+1)} \times \left[\sum_{j \in SJ_i^{k+1}} (1-\hat{Z}_j^{k+1})(1-\hat{b}_i^k) - \sum_{j \in \bar{S}J_i^{k+1}} (1-\hat{Z}_j^{k+1})\hat{b}_i^k\right] \qquad (4)$$

In the equations above, $\hat{Z}_j^{k+1}$ is unknown and can be estimatedby its approximation $\tilde{Z}_j^{k+1}$ as [9]:

$$\tilde{Z}_j^{k+1} = f\left(\tilde{a}_i^{k+1}, \tilde{b}_i^{k+1}, X_{k+1}\right) = \frac{A_j^{k+1} \times d}{A_j^{k+1} \times d + B_j^{k+1} \times (1-d)} \quad (5)$$

where

$$A_j^{k+1} = \prod_{i=1}^{M} \left(\tilde{a}_i^{(k+1)}\right)^{s_i c_j^{k+1}} \left(1 - \tilde{a}_i^{(k+1)}\right)^{\left(1 - s_i c_j^{k+1}\right)}$$

$$B_j^{k+1} = \prod_{i=1}^{M} \left(\tilde{b}_i^{(k+1)}\right)^{s_i c_j^{k+1}} \left(1 - \tilde{b}_i^{(k+1)}\right)^{\left(1 - s_i c_j^{k+1}\right)}$$

$$\tilde{a}_i^{k+1} = \hat{a}_i^k \times \frac{s_i^{k+1}}{s_i^k}$$

$$\tilde{b}_i^{k+1} = \hat{b}_i^k \times \frac{s_i^{k+1}}{s_i^k}$$

Based on known values like $\hat{a}_i^k, \hat{b}_i^k, X_k, X_{k+1}$at time interval$k+1$, $\tilde{Z}_j^{k+1}$can be represented as a function [9]:

$$\tilde{Z}_j^{k+1} = g\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_k, X_{k+1}\right) = \frac{C_j^{k+1} \times d}{C_j^{k+1} \times d + D_j^{k+1} \times (1-d)} \quad (6)$$

where

$$C_j^{k+1} = \prod_{i=1}^{M} \left(\hat{a}_i^k \times \frac{s_i^{k+1}}{s_i^k}\right)^{s_i c_j^{k+1}} \left(1 - \hat{a}_i^k \times \frac{s_i^{k+1}}{s_i^k}\right)^{\left(1 - s_i c_j^{k+1}\right)}$$

$$D_j^{k+1} = \prod_{i=1}^{M} \left(\hat{b}_i^k \times \frac{s_i^{k+1}}{s_i^k}\right)^{s_i c_j^{k+1}} \left(1 - \hat{b}_i^k \times \frac{s_i^{k+1}}{s_i^k}\right)^{\left(1 - s_i c_j^{k+1}\right)}$$

By combining equation 6 with equation 4, the final recursivecomputation of the estimation parameters can be obtained asfollows:

$$\hat{a}_i^{k+1} = \hat{a}_i^k + \frac{1}{Nd(k+1)} \times \left[\sum_{j \in SJ_i^{k+1}} g\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_k, X_{k+1}\right)\left(1 - \hat{a}_i^k\right) - \sum_{j \in \bar{S}J_i^{k+1}} g\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_k, X_{k+1}\right)\hat{a}_i^k \right]$$

$$\hat{b}_i^{k+1} = \hat{b}_i^k + \frac{1}{Nd(k+1)} \times \left[\sum_{j \in SJ_i^{k+1}} \left(1 - g\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_k, X_{k+1}\right)\right)\left(1 - \hat{b}_i^k\right) - \sum_{j \in \bar{S}J_i^{k+1}} \left(1 - g\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_k, X_{k+1}\right)\right)\hat{b}_i^k \right] \quad (7)$$

Also, the updated correctness of the variable reported can becalculated dynamically as follows [9]:

$$\tilde{Z}_j^{k+1} = f\left(\hat{a}_i^{k+1}, \hat{b}_i^{k+1}, X_{k+1}\right) \quad (8)$$

The estimation parameter of the new anomaly reports bythe sources consecutively over a time period can be trackedby using equation 7.

IV. IMPLEMENTATION AND TESTING

An Android application called eDrive has been developedwhich communicates with a Java EE server deployed usingGlassFish. The GPS trajectory data collection process, map-matching algorithm and the traffic anomaly detection processhas been implemented in separate servlet beans. Servlet beans are used for dynamic scaling as a large number of users willbe interacting with the users in real-time. Virtual machines areused to simulateHadoop cluster where one Virtual Machinewith superior processing capability is made the Master Node. After map matching the GPS coordinates of each user is storedin separate files in HDFS using MapReduce for a short period.All incident reports are automatically posted to the applicationstwitter account. Incident reports are grouped based on theincident type, latitude, longitude and the timestamp. These tweets are constantly monitored by the system for new reportsbased on the timestamp. When a new report arrives, it will be extracted using the Twitter API and is provided as Boolean input to the recursive EM algorithm and updates the probabilityvalue accordingly. This functionality is implemented in thetraffic anomaly detection bean. The final result is then send tothe users and to the traffic authorities. The application has beentested using Android emulator and a GPS spoof application.The test result is shown in Figure 4.

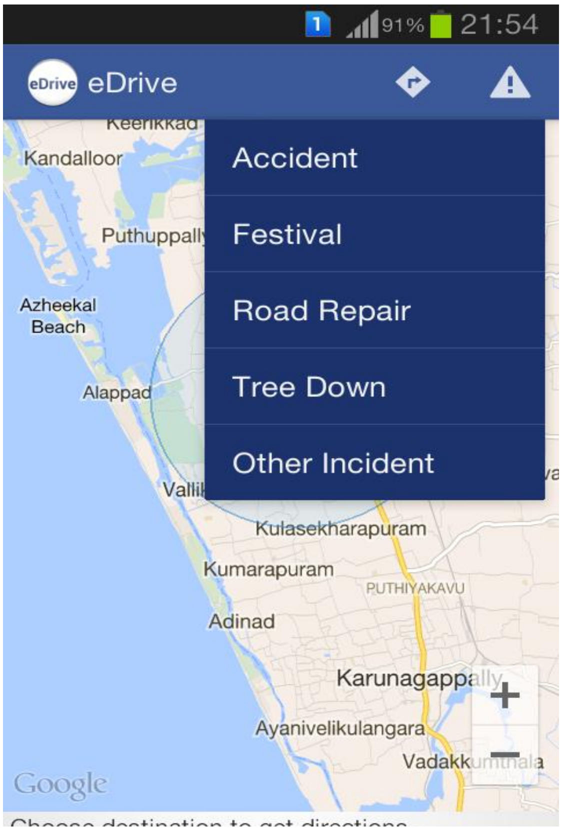

Fig. 3. eDrive incident report menu

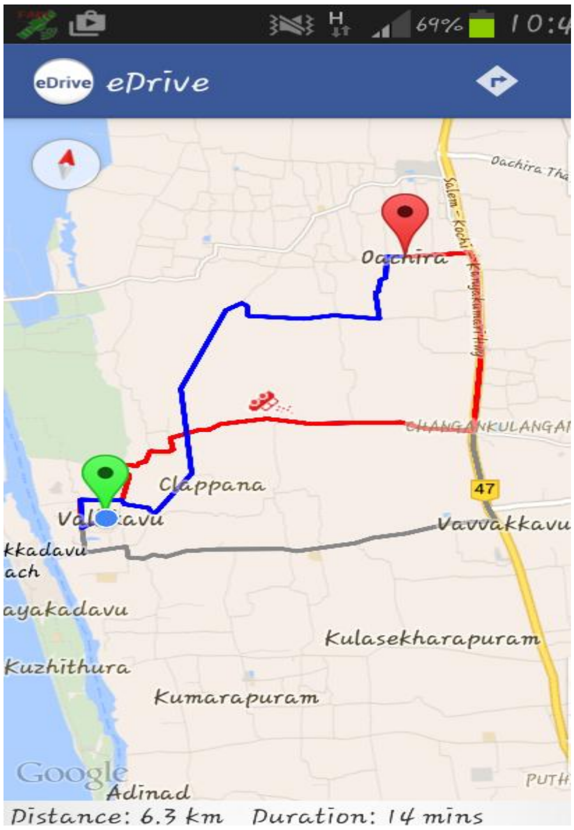

Fig. 4. Test Run with anomaly detection and route suggestion

V. CONCLUSION

In this work, we have proposed a Hadoop based trafficanomaly detection system which is scalable and foolproof.The user is provided with an Android application with basicnavigation features and incident report functionality. The Android application communicates with a JavaEE server. Thearchitecture uses the MapReduce framework for distributedand parallel processing of traffic data. To track users a map-matching algorithm, which is efficient even when the GPS datasampling rate is less, is used. We have used a truth estimationalgorithm to calculate the authenticity of the incidents reported.The proposed system is scalable and can handle massivevolumes of GPS data. The usage of Twitter for incidentreports makes the information available to the social mediaand thus makes the incident reports available even for otherapplications if required. With this project we have aimed toexplore the power of big data processing, social media analysis,expectation maximization and GPS navigation using mobiledevices.

REFERENCES

[1] Hadoop MapReduce, http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[2] Bei Pan, Yu Zheng, David Wilkie, Cyrus Shahabi. Crowd Sensing ofTraffic Anomalies based on Human Mobility and Social Media. InSIGSPATIAL GIS 13.

[3] Google Directions API, https://developers.google.com/maps/documentation/directions/start

[4] WenQiang Wang, Xiaoming Zhang, Jiangwei Zhang, Hock BengLim.Smart Traffic Cloud: An Infrastructure for Traffic Applications. In2012 IEEE 18th International Conference on Parallel and DistributedSystems.

[5] Dong Wang, Tarek Abdelzaher, Lance Kaplan, Charu Aggarwal. Recursive Fact-finding: A Streaming Approach to Truth Estimation inCrowdsourcing Applications. In ICDCS '13 Proceedings of the 2013IEEE 33rd International Conference on Distributed Computing Systems,Pages 530-539.

[6] Android Development, https://developer.android.com/guide/index.html

[7] Rectangle Method for Map Matching, https://github.com/gglnx/google-maps-utility-library-v3/blob/master/routeboxer/docs/examples.html

[8] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovering social sensing: A maximum likelihood estimation approach. In The 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 12), April 2012.

[9] D.M. Titterington. Recursive parameter estimation using incompletedata. Journal of the Royal Statistical Society. Series B (Methodological), 46(2):pp. 257267, 1984.

[10] D. Wang, L. Kaplan, T. Abdelzaher, and C. C. Aggarwal. On Scalabilityand robustness limitations of real and asymptotic confidence bounds insocial sensing. In The 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 12), June 2012.