

# Incident Detection by Spatiotemporal Analysis of GPS Data

Eleonora D'Andrea and Francesco Marcelloni

Department of Information Engineering  
University of Pisa

Largo Lucio Lazzarino 1, 56122 Pisa, Italy

eleonora.dandrea@for.unipi.it, francesco.marcelloni@unipi.it

**Abstract**—We present a system to detect incidents causing traffic congestion in the road network by analyzing real-time GPS data. These data are collected from tracking devices installed in the vehicles or from drivers' smartphones. After positioning the GPS coordinates on the road map, the system assigns a traffic state to each road segment based on the velocities of vehicles, and generates alerts for incident based on a spatiotemporal analysis of these states. The system is validated by using GPS data simulated in typical traffic conditions in the city of Pisa, Italy. The results show an incident detection rate of 91.6% and an average detection time shorter than 7 minutes.

**Index Terms**—expert system, GPS, incident detection, urban mobility.

## I. INTRODUCTION

Nowadays traffic monitoring in cities has gained growing interest due to the increasing number of vehicles causing traffic congestions, bottlenecks and incidents, especially during rush hours. In particular, *incidents*, defined as “unexpected events that temporarily disrupt the traffic flow on a segment of a roadway” [1][2], can cause disruptions and congestion, and possibly generate time losses and costs. Thus, an accurate and early detection of incidents becomes of the utmost importance in traffic management [2], e.g., to effectively dispatch emergency services. For all these reasons, automatic Incident Detection Systems (IDSs) have spread in recent years, as essential components of the Intelligent Transportation Systems (ITSs) devoted to detect events, causing traffic congestions. ITSs are infrastructures that allow improving safety and management of transport networks, and addressing environmental, economic, and social needs, by combining information and communication technologies, with transport networks, vehicles and users. ITSs provide, e.g., real-time information about traffic congestion and regulation, travel time estimations, incidents, pollution levels [3]-[5]. While traditional ITSs exploit data collected from fixed location sensors placed on the road, e.g., closed-circuit cameras, infrared sensors, inductive loop sensors (ILSs) and magnetometers, new generation ITSs are based on sensor

technology within the vehicle (e.g., Global Positioning System (GPS), airbag activation detection) [1]. While traditional sensors collect data related to the total volume of traffic on a certain road, mobile sensors gather data related to the single vehicle [6]. Today, thanks to the increasing diffusion of mobile devices (e.g., smartphones), and the widespread use of GPS technology (both in vehicles and in mobile devices), we are able to collect directly and analyze traces from most of the objects (people, vehicles, etc.) moving in the road network.

In this paper, we present an IDS able to identify in real-time portions of the road network where an incident is causing traffic congestion. Our IDS exploits the pervasive use of GPS-equipped devices and smartphones, thus providing the position of users who are driving a vehicle in the road network. A spatiotemporal data analysis performed on the GPS traces of multiple vehicles, which travel along the same road segment, determines whether a traffic congestion, possibly caused by an incident, occurs in the segment. The use of mobile phones as sensor probes for monitoring real-time traffic has many advantages. From the architectural point of view, i) it allows the exploitation of the existing communication infrastructures and of the wide spread of GPS-integrated mobile phones, ii) it results to be independent of the device and the communication service provider [6], and iii) it overcomes the drawbacks of traditional monitoring systems, i.e., limited coverage of the road network and high installation and maintenance costs [7]. The aim of our IDS is to provide the drivers with the information related to traffic congestions possibly caused by incidents, in order to reduce further traffic difficulties. The IDS has been implemented as a service of a wider platform of the SMARTY project [8], and it is designed to cooperate with another service of this project devoted to detect traffic and incidents based on traffic reports from Twitter's users [9]. The platform is developed from scratch, and is built on a Service Oriented Architecture (SOA). This allows an easy integration with other services (e.g., smart parking, weather, bike sharing). Further, the scalability of our IDS is guaranteed by using a cloud computing architecture and a different instance of the service for elaborating data related to each specific geographic area. The user needs only to register to the service for receiving notifications. On registration, the user accepts the privacy policy associated with the service, according to which the user's GPS position is used only anonymously for the aim of the service. The paper is structured as follows. Section II

---

This work is partially supported by the “SMARTY” project funded by “Programma Operativo Regionale (POR) 2007-2013” - objective “Competitività regionale e occupazione” of the Tuscany Region”, and the project “Metodologie e Tecnologie per lo Sviluppo di Servizi Informatici Innovativi per le Smart Cities” funded by “Progetti di Ricerca di Ateneo - PRA 2015” of the University of Pisa.

discusses the state of the art, Section III introduces the proposed IDS, Section IV presents the experimental results, and Section V draws the conclusions.

## II. THE STATE OF THE ART

Various approaches for extracting knowledge by means of GPS traces of vehicles moving in the road network have been proposed in the literature [10]. In the following, we provide a brief description of the most relevant works related to incident detection based on GPS traces, exploiting statistical methods, computational intelligence techniques, and rule-based expert systems. In [11], the authors employ a neural network model to detect incidents. They combine GPS data collected from ILS with travel time and speed collected from probe vehicles, and simulate 108 incidents (lasting for 15, 30, and 60 min.) in Brisbane, Australia under different traffic conditions. In [2], the authors detect freeway incidents, using GPS data collected from ILS in the 5.8 km city-bound area in Singapore. In [12], the authors adapt freeway incident detection algorithms to the urban scenario. They simulate 1800 incidents (lasting for 5 and 10 min.) in Dublin, Ireland, during rush hours, and use traffic volume and traffic occupancy, as inputs to a support vector machine model. In [1], the authors use real-time GPS data to detect incidents. They identify anomalous road segments, based on the velocity of vehicles, and study their behavior by means of a spatial and temporal analysis. In the simulated experiments in Coventry, UK, they employ 10 segments on a 10-miles route, and 10-20 vehicles per segment. In [13] the authors detect blocked or congested road segments by means of GPS data, and by considering a road congested if its average velocity is below 10 km/h. The IDS in [15] collects real-time GPS data from probe vehicles on a segmented road network in order to detect 30-minutes workdays incidents in Calgary, Canada, by analyzing, with statistical techniques, travel times and acceleration noise of vehicles. In [14], the authors present Nericell, a system for monitoring road and traffic conditions by means of accelerometer, GSM, and GPS data extracted from users' smartphones.

Our IDS integrates and re-elaborates some ideas discussed in the above-mentioned papers by proposing an organic solution for detecting incidents. First, it pre-processes the GPS traces. Then, it classifies the road segment with a traffic state, i.e., *blocked*, *non-blocked*, or *absent*, based on the velocities of the vehicles travelling along the segment. Finally, it determines possible incidents by performing a spatiotemporal analysis of the classified segments, and notifies the registered users by reporting the road segments involved in the traffic congestion. The strengths of our IDS with respect to similar works in the literature are summarized as follows: i) it does not use sensor probe vehicles (e.g., taxis, buses) or fixed sensors installed in the city. In fact, the GPS traces are directly collected from smartphones or tracking devices, resulting in a very low-cost framework; ii) it does not require historical data (e.g. collections of GPS traces, weather data, traffic conditions) or other vehicular data (e.g., acceleration noise, travel times), since it does not exploit statistical or supervised learning techniques; iii) it has been implemented as an event-driven infrastructure, built on an SOA architecture. In this way, it can be integrated with other services, and can be scaled to manage road networks with different sizes. Finally, regarding the

possible real-world applications, the proposed IDS could be a valuable tool for traffic and city administrations to regulate traffic and vehicular mobility in the case of incidents, and, more importantly, to effectively dispatch emergency services.

In this paper, we represent the road network exploiting the *digital map* provided by the well-known open source, free-license framework Open Street Map (OSM) [16]. The digital map provided by OSM consists of an oriented graph whose main elements are: *nodes*, *tags*, and *segments*. *Nodes* represent GPS positions on the map, corresponding to Points Of Interest (POIs), intersections, changes in bearing of the road (curves), traffic lights, pedestrian crossings, etc. Each node is associated with an id and with a list of *tags* associating information with the node (e.g., traffic light, crossing, road type, POI, speed limit, travelling direction). Two nodes are joined by a linear *segment*, thus, a road is described as the conjunction of consecutive *segments*. By leveraging on digital maps, the objects (GPS points) are positioned on the segments according to a membership function. In addition, with the aim of locating more precisely a traffic event on the road, too long segments of the map were split into sub-segments, based on the speed limit. With this elaboration, we obtained segments having an average size of about 50 m. We wish to point out that, in our work, the length of each segment is not fixed, but is adapted to the particular road (urban or highway). On the contrary, other works in the literature (e.g., [1]) employ fixed road segments by focusing on highways with lengths from 200 to 500 meters.

## III. THE PROPOSED IDS

The architecture of our IDS is shown in Fig. 1. It consists of three modules (depicted as gray rectangles in Fig. 1): *Pre-Processing*, *Segment Traffic Classification* and *Incident Notification*. Each module receives inputs or produces outputs (both represented as white parallelograms in Fig.1). The *Pre-Processing* module map-matches each GPS point to the corresponding road segment, and determines the travelling direction of the corresponding vehicle. The *Segment Traffic Classification* module assigns a *traffic state* to each segment based on considerations made on the number of vehicles in the segment, their velocities, and the speed limit of the segment. The *Incident Notification* module analyzes the traffic states previously found in order to confirm the presence of an incident, and sends an *alert for incident* related to the involved area based on a spatiotemporal analysis required to study the temporal and spatial context of the event. Indeed, a detected low velocity on urban roads may be caused by the presence of traffic lights and pedestrian crossings [7], while in the case of highways, it may easily indicate traffic congestion or incidents. Thus, detecting a low average velocity on a segment of road does not necessarily imply an incident or a traffic congestion situation, especially in the case of urban context, which is certainly the most difficult to analyze.

More precisely, after the pre-processing phase, each segment of the map is observed for a time interval  $T$  sampled every  $f$  seconds. At each sampling time in  $T$ , the traces of the vehicles that have travelled along the segment are collected, processed and saved. The procedure is then repeated on each segment of the map in the following time interval, and so on. We denote with  $T_t$  the  $t$ -th time interval analyzed. As in other

papers [7][17], we set  $f = 30$  seconds. This value balances precision and energy consumption, when the GPS traces are collected by using a smartphone: higher sampling frequencies would consume too fast the battery. The value of  $T$  is set to 2 minutes. We verified that this value allows having a reliable estimate of the travelling velocity in the segment when the number of vehicles is low, and permits to manage the corresponding complexity when the traffic is high.

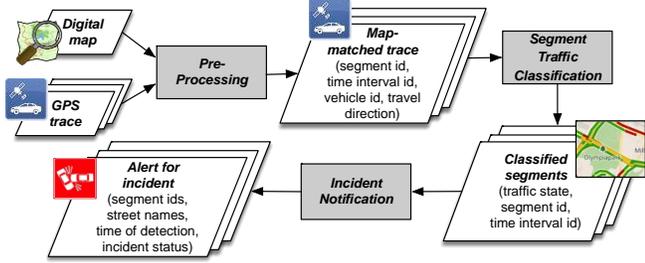


Fig. 1. The IDS architecture.

In the following, a segment of the OSM map is denoted as  $s_j$ , where  $j = 1, \dots, S$ , is the index of the segment, and  $S$  is the total number of segments in the map. A segment  $s_j$  is characterized by a nominal velocity  $v_j^{lim}$  defined according to the speed limit of the portion of road corresponding to the segment, and by a size  $d_j$ , corresponding to the distance between the extremes of the segment. A vehicle is uniquely identified with its id. In addition, in a generic segment  $s_j$ , a vehicle is denoted as  $m_{j,i,t}$ , where  $i = 1, \dots, M_{j,t}$  is the index of the vehicle within the segment  $s_j$ , and  $M_{j,t}$  is the number of vehicles travelling along  $s_j$  in  $T_t$ . Thus, vehicle  $m_{j,i,t}$  is the  $i$ -th vehicle travelling in  $j$ -th segment in  $t$ -th time interval. A GPS trace belonging to a vehicle is a set of information consisting of the unique identifier of the vehicle, and a set of GPS records. A GPS record includes the GPS position (latitude, longitude), the timestamp, and the instantaneous velocity  $v_{j,i,t,h}^{ist}$ , where  $h = 1, \dots, Q_{j,i,t}$  is the index of the GPS record of vehicle  $m_{j,i,t}$ , and  $Q_{j,i,t}$  is the number of positions associated with vehicle  $m_{j,i,t}$  during the observed time interval  $T_t$ . Each vehicle in the segment can be associated with one or more records depending on its speed, which is obviously influenced by the congestion level of the segment (e.g., a vehicle travelling along a blocked segment will send multiple records). In the following, we describe in detail the three modules of our IDS.

#### A. Pre-Processing

First, each GPS position collected from a vehicle is associated with a segment of the digital map. This step requires a routing algorithm to handle incomplete routes in case of jumps or missing GPS points [17]. We employed a modified version of the routing algorithm provided by Graph Hopper [18]. The algorithm generates the shortest path between two non-adjacent segments according to the Dijkstra's algorithm [19], by giving priority to main streets. Then, for each GPS position, the travelling direction (i.e., *forward* or *backward*) of the vehicle in the segment is determined according to an appropriate algorithm we have developed. In summary, the algorithm establishes the travelling direction of a vehicle in a segment by exploiting: i) the information provided by the OSM map regarding the kind of segment (the travel direction is fixed

in unidirectional segments), and ii) the positions (and thus travel directions) of the vehicle in previous samples (actually, the travelling direction on a bidirectional segment is determined by the previous positions of the vehicle).

#### B. Segment Traffic Classification

In this module, the GPS traces collected in the current time interval  $T_t$  are analyzed, and the road segments of the map are classified, by assigning to each segment  $s_j$  a *traffic state*  $TS_t$ , namely, *absent*, *blocked*, or *non-blocked* (indicating, in this latter case, a flowing or light-congested traffic flow).

In order to avoid false alarms and manage potential outliers, we take into account a minimum number  $M_{j,t}$  of vehicles travelling along the segment  $s_j$  in  $T_t$ . Indeed, if  $M_{j,t}$  is below a certain threshold, i.e.,  $M_{j,t} < M_{min}$ , the number of vehicles is not sufficient to reliably detect a relevant traffic congestion due to an incident. On the other hand, a possible traffic congestion condition will be detected whenever a sufficient amount of vehicles will accumulate in the segment. Hence, for the aim of this paper, i.e., incident detection, we take into account only the case  $M_{j,t} \geq M_{min}$ , and the special case of a complete absence of traces ( $M_{j,t} = 0$ ). Thus,

- if  $M_{j,t} = 0$ , we set  $TS_t = absent$ ;
- if  $M_{j,t} \geq M_{min}$ , we i) compute the average velocity  $\bar{v}_{j,i,t}$  of each vehicle  $m_{j,i,t}$  in  $s_j$  as  $\bar{v}_{j,i,t} = \sum_{h=1}^{Q_{j,i,t}} v_{j,i,t,h}^{ist} / Q_{j,i,t}$ ; ii) sort the velocities  $\bar{v}_{j,i,t}$  of the vehicles; iii) compute the median value  $\tilde{v}_{j,i,t}$ ; iv) set  $TS_t = blocked$  if  $\tilde{v}_{j,i,t} < v_{block}$ , otherwise we set  $TS_t = non-blocked$ .

In the experiments, based on heuristic considerations, we set  $v_{block} = 3$  km/h adapting the suggestion in [13] to the urban roads of Pisa. The value of  $M_{min}$  was chosen to reduce the influence of personal drivers' behaviors on the decision about the traffic state of the segment. E.g., with  $M_{min} = 1$ , a driver stopping the vehicle to drink a coffee would produce a traffic state erroneously set to *blocked*. We verified that  $M_{min} = 4$  is a choice that guarantees to limit the effect of personal drivers' behaviors, and to determine a state even if the traffic is not intense. At the end of the classification, the system has associated with each segment of the map, for the considered time interval, a traffic state. The segments having  $TS_t = blocked$  will undergo the *Incident Notification* module.

#### C. Incident Notification

This module appropriately generates an *alert for incident* exploiting: i) a temporal analysis of the information associated with the segments in  $N$  time intervals preceding the current one, ii) a spatial analysis of the traffic condition in segments adjacent to the considered one, and iii) an analysis of the vehicles stuck in the queue. An *alert for incident* indicates the spatial extension of the event, i.e., the segment (or the list of segments) associated with the incident event. The segments are initially expressed as a set of ids, and are then converted to the city street name by using the OSM tool *Nominatim* for reverse geocoding. An *alert report* contains the list of alerts related to the current time interval. More precisely, since this module uses the information related to  $N+1$  time intervals (the  $N$  previous intervals and the current one), the first alert report will be sent at least after  $N+1$  time intervals, while the following ones will be sent at the end of each time interval, i.e., every 2

minutes. In the experiments, we set  $N = 2$  so as to achieve a good trade-off between reliability and responsiveness.

By exploiting spatial and temporal considerations, the system is able to identify possible incidents, if specific conditions are met. In fact, when an incident occurs, the related traffic congestion typically lasts for a quite long time, and the vehicles stuck in the queue caused by the incident are almost the same in subsequent time intervals. The incident can block the whole roadway or it can block only part of it, allowing the transit of vehicles in its free portion. In the former case, we will typically observe a “queue with head” (Fig. 2), i.e., an almost total absence of GPS traces ahead of the incident, and a queue of vehicles behind the incident, which tends to grow with time. In the latter case, it is extremely hard to detect an incident [15] and distinguish it from a mere traffic congestion condition. Thus, in this paper, we will consider only incidents blocking the whole roadway. In addition, we do not take into account other anomalous conditions (e.g. roadworks, bad weather, and unexpected events) that may affect the traffic flow.

For each segment  $s_j$  having  $TS_t = \text{blocked}$  in the current time interval, a spatiotemporal analysis is performed, taking into account segments adjacent to  $s_j$ , and the traffic condition in  $N$  previous time intervals. First, with the aim of defining the spatial extension of the event, we iteratively check the state of all the segments adjacent to  $s_j$ , in both in the travel directions, and that of the segments adjacent to these, until we find a segment with *non-blocked* state. We consider two different types of traffic event: 1) restricted to a single segment, or 2) involving a set of segments. More precisely,

1) if the event is restricted to segment  $s_j$ , i.e., segments adjacent to  $s_j$  have traffic state *non-blocked* or *absent*, we perform a temporal analysis of the state of  $s_j$  in  $N$  previous time intervals, to check if the event is relevant or not:

- a) if  $s_j$  had traffic state *blocked* in all the  $N$  previous time intervals, the event is considered to be relevant, as it has blocked the traffic for quite a long time. Thus, to distinguish an *incident* from a blocked traffic condition, we check whether the vehicles stuck in the queue are the same, or have changed, in consecutive time intervals:
  - i) if a percentage  $P$  of vehicles present in the segment in  $N$  previous time intervals are the same, an *alert for incident* in  $s_j$  is sent;

2) if the event involves a set of segments around  $s_j$ , the segment (or the series of segments) immediately behind  $s_j$  has traffic state *blocked*, and:

- a) if the segment ahead of  $s_j$  has traffic state *absent*, an *alert for incident* for the set of segments around  $s_j$  is directly sent (“queue with head” condition in Fig. 2);
- b) otherwise if the segment ahead of  $s_j$  has traffic state *non-blocked*, we check if an incident is occurring to consider the common situation in a city in which the traffic seems regular due to an access point (e.g., a side street, a parking) ahead of  $s_j$ . Hence, we check the ids of vehicles stuck in queue as explained earlier. If the condition in i) is met on the set of segments, the system sends an *alert for incident* for the set of segments around  $s_j$ .

As regards parameter  $P$ , high values of  $P$  would reduce the accuracy and low values would increase the number of false alarms. In the experiments,  $P$  was set with the aim of obtaining

the best trade-off between high detection rate and low false alarm rate. Theoretically, the value of  $P$  should be 100% since, in case of a traffic block, the same vehicles will remain in the same positions for a while. Actually, it could occur that one or more vehicles decide to make a U-turn or to opt for another route, taking a cross street. For these reasons, we set  $P = 90\%$ .

At the end of processing, the *Incident Notification* module generates an alert report, which describes the alerts found in the analysis. Fig. 2 depicts how the system works: after an incident has occurred, the users registered to the system are directly notified with the information about the incident, without the need to access official traffic news channels. Thus, they are able to avoid the roads involved in the incident.

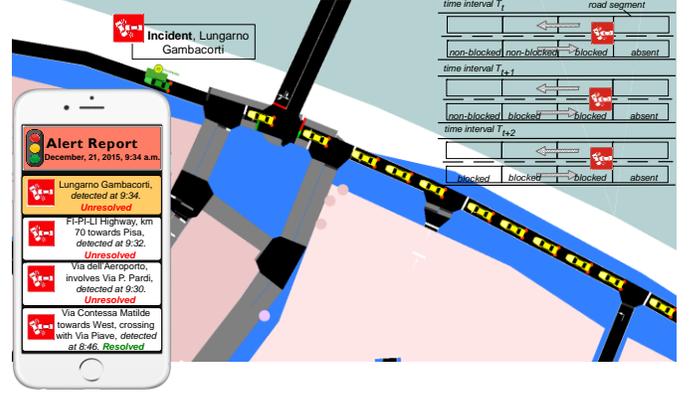


Fig. 2. The “queue with head” condition after a simulated incident, and the corresponding alert on the user smartphone.

## IV. EXPERIMENTAL ANALYSIS

### A. Experimental Setup and GPS Data Simulation

The traffic simulator SUMO (Simulation of Urban Mobility) [20] was employed to simulate the GPS traces of about 50,000 vehicles traveling in the city of Pisa, Italy, on an area corresponding to about 70 km<sup>2</sup>. We took into account two different scenarios, i.e., working day and holiday. To reproduce as close as possible the real traffic condition in the city, we exploited the typical traffic conditions in working day and holiday, extracted from Google Traffic [21], and the real-world data collected from smartphone of voluntary drivers within a specific data collection campaign. We wish to point out that the simulation of data is a common practice in the literature [1][2], and it was necessary, as real-world data collected by voluntary drivers are typically insufficient, due to privacy or permission issues. We generated 20 different routes of vehicles crossing the city in 10 different time intervals, thus obtaining about 200 routes per scenario. The routes taken into account contain mainly urban roads, but they contain also freeway roads. The GPS traces were sampled every 30 seconds from 6 a.m. to 11 p.m. Then, we selected the most significant routes and time intervals to simulate 24 fake incidents per scenario (with durations of 15, 20, and 30 min.), in different times and places. We performed several simulations with duration from 15 to 40 minutes. Each simulation takes into account on average 3 incidents on different areas of the city at different times. On average, about 1000 vehicles were considered at each instant in the simulations. Regarding the incident duration, we have experimentally found that incidents with duration shorter than

10 min. are hardly detected by the system and can be easily mistaken with congestion situations caused, e.g., by traffic lights. Further, an incident blocking the whole roadway typically takes more than 10 minutes to be resolved.

### B. Incident Detection Results

The performance of our system was evaluated using the following measures: i) *Detection Rate* (DR),  $DR = \#cdi/\#i$ , being  $\#cdi$  the number of correctly detected incidents, and  $\#i$  the number of simulated incidents; ii) *Miss Rate* (MR),  $MR = (\#i - \#cdi)/\#i$ ; iii) *Number of false alarms* (NFA), the number of erroneously detected incidents; iv) *Precision*,  $Prec = \#cdi/(\#cdi + NFA)$ . In addition, we evaluated the *Mean Time to Detect* (MTD), that is, the interval between the time the incident occurred and the time of its detection.

In both the scenarios, we correctly detected 22 incidents out of 24, obtaining  $DR = 91.6\%$ ,  $MR = 8.3\%$ ,  $Prec = 88\%$ ,  $NFA = 3$ . Regarding MTD, we got values equal to 6.9, and 6.63 min., for the workday and the holiday scenarios, respectively. The two missed incidents may be due to the presence of several access points along the road interested by the incident. In fact, from the access points new vehicles can enter the road and join the queue. We have verified, however, that in the case of the missed incidents, the system indicates anyway that a congestion event (*blocked* traffic) is occurring on the road. Regarding the NFA, we wish to point out that, although the heavy simulation performed, only 3 false alarms were generated by the system per scenario.

In the following, we compare our results with those achieved by some of the IDSs recalled in Section II. We observe that, although these results are obtained using different datasets and methods, they are comparable with the ones obtained by our IDS. In addition, we detect incidents in a more complex context (whole city), and by employing less information. In [1], [2], [15] and [11], the authors obtain  $DR = 78.7\%$  and  $MTD = 2s$ ,  $DR = 100\%$  and  $MTD = 70s$ , and  $DR = 92\%$  and  $MTD = 106s$ ,  $DR = 90\%$ , respectively. The best performing model for the urban scenario in [12] achieves  $DR = 87.31\%$  and  $MTD = 98.61s$ . Regarding the MTD performance we achieved worse results than the above-mentioned approaches. We have to consider, however, that the time to detect an incident strongly depends on the current traffic condition and on the average distance between the incident location and the starting point of the vehicles' routes. Further, our system uses GPS data coming from a wider area, involves a larger number of road segments and vehicles, and uses less information (it does not need, e.g., vehicle sensor probes, fixed sensors, historical data, weather data). In addition, we wish to point out that incident detection in urban road networks is a more challenging task than incident detection on freeways [11], as urban and freeway road networks have different traffic flows due, e.g., to different speed limits and access points. More in detail, lower speed limits and the presence of many access points, intersections, or urban-specific elements (e.g., traffic lights, pedestrian crossings) may hide the occurrence of an incident.

## V. CONCLUSIONS

We have presented an IDS able to detect road incidents

from real-time GPS data. The IDS assigns to each road segment a traffic state, based on the presence of GPS traces, and generates appropriate *alerts* indicating the incident area. The experiments performed using GPS data simulated in Pisa, Italy, show a DR of 91.6% and an MTD shorter than 7 minutes.

## REFERENCES

- [1] S. Kamran and O. Haas, "A multilevel traffic incidents detection approach: Identifying traffic patterns and vehicle behaviours using real-time GPS data," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, Istanbul, Turkey, 2007, pp. 912–917.
- [2] D. Srinivasan, J. Xin and R.L. Cheu, "Evaluation of adaptive neural network models for freeway incident detection," *IEEE Trans. on Intell. Transp. Syst.*, vol. 5, no. 1, pp. 1–11, 2004.
- [3] P.S. Castro, D. Zhang and S. Li, "Urban traffic modeling and prediction using large scale taxi GPS Traces," in *Proc. 10th Int. Conf. Pervasive*, Newcastle, UK, 2012, pp. 57–72.
- [4] G. Anastasi, M. Antonelli, A. Bechini, S. Brienza, E. D'Andrea, D. De Guglielmo, P. Ducange, B. Lazzzerini, F. Marcelloni and A. Segatori, "Urban and social sensing for sustainable mobility in smart cities," in *Proc. IFIP/IEEE Int. Conf. Sust. Internet and ICT for Sustainability (SustainIt)*, Palermo, Italy, 2013, pp. 1–4.
- [5] K. Boriboonsomsin, M. Barth, W. Zhu and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1694–1704, 2012.
- [6] D.B. Work and A.M. Bayen, "Impacts of the mobile Internet on transportation cyberphysical systems: Traffic monitoring using smartphones," in *Proc. Nat. Workshop Res. High-Confidence Transp. Cyber-Physic. Syst.: Automotive, Aviation and Rail (HCTCPS)*, Washington, DC, 2008, pp. 18–20.
- [7] J. Yoon, B. Noble and M. Liu, "Surface street traffic estimation," in *Proc. 5th ACM Int. Conf. Mobile Syst., Appl. and Serv.*, 2007, pp. 220–232.
- [8] The Smarty project. <http://www.smarty.toscana.it/>.
- [9] E. D'Andrea, P. Ducange, B. Lazzzerini and F. Marcelloni, "Real-Time Detection of Traffic from Twitter Stream Analysis," *IEEE Trans. on Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2269–83, 2015.
- [10] G. Pan, G. Qi, W. Zhang, S. Li and Z. Wu, "Trace analysis and mining for smart cities: Issues, methods, and applications," *IEEE Comm. Magazine*, vol. 51, no. 6, pp.120–126, 2013.
- [11] H. Dia and K. Thomas, "Development and evaluation of arterial incident detection models using fusion of simulated probe vehicle and loop detector data," *Information Fusion*, vol. 12, pp. 20–27, 2011.
- [12] B. Ghosh and D.P. Smith, "Customization of automatic incident detection algorithms for signalized urban arterials," *J. of Intelligent Transp. Systems*, vol. 18, no. 4, pp. 426–441, 2014.
- [13] R.-P. Schäfer, K.-U. Thiessenhusen, E. Brockfeld and P. Wagner, "A traffic information system by means of real-time floating-car data," in *Proc. 9th ITS World Congr.*, Chicago, IL, 2002.
- [14] P. Mohan, V.N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones," in *Proc. 6th ACM Conf. Embedded Network Sensor Systems*, New York, NY, 2008, pp. 323–336.
- [15] C. Basnayake, "Automated traffic incident detection with GPS equipped probe vehicles," in *Proc. 17th ION GNSS Int. Tech. Meeting Satellite Div. Inst. Nav.*, Long Beach, CA, 2004, pp. 741–750.
- [16] Open Street Map. [www.openstreetmap.org/](http://www.openstreetmap.org/)
- [17] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, *Global Positioning System. Theory and Practice*. Springer-Verlag Wien, 2013.
- [18] Graph Hopper Route Planner. [www.graphhopper.com/](http://www.graphhopper.com/)
- [19] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [20] D. Krajzewicz, G. Hertkorn, C. Rössel and P. Wagner, "SUMO (Simulation of Urban MOBility)," in *Proc. 4th Middle East Symp. Sim. and Modelling*, 2002, pp. 183–187.
- [21] Google Traffic. <https://www.google.it/maps/>