PV-Alert: A Fog-based Architecture for Safeguarding Vulnerable Road Users

Aghiles DJOUDI
Pierre and Marie Curie University
aghiles.djoudi@etu.upmc.fr

ABSTRACT

High volumes of pedestrians, cyclists and other vulnerable road users (VRUs) have much higher casualty rates per mile, indicating a strong need of technologies to protect such vulnerable road users.

In order to alleviate the problem, Sensing capabilities of mobile smartphones can be used to detect, warn and safeguard VRUs. Equipped with both WiFi and cellular connection and advanced GPS on-board sensors, Smart-phones can locate VRUs and receives alerts. Unfortunately, GPS location is still plagued with errors that frequently produce inaccurate trajectories.

To overcome this problem, This article presents an innovative map matching algorithm to predict the direction of pedestrians, this algorithm considers the trajectory of the data rather than merely the current position as in the traditional map matching case. The evaluation results indicate that the algorithm is efficient and identify correctly the direction of pedestrians.

In this research we have proposed also an infrastructure-less fog-based architecture where fog nodes process delay sensitive data obtained from smartphones for alerting pedestrians and drivers before sending the data to the cloud for further analysis. Simulation results show that the proposed architecture is able to render alerts in real time when LTE is used to connect smartphones with fog nodes. As a proof of concept, we built a mobile application which send periodically GPS data to the server and receives alerts when there is a high probability of collision.

Keywords

Vulnerable Road Users; Fog Computing; Pedestrian Safety; Low Latency; Crowd sensing

1. INTRODUCTION

Road traffic injuries are one of the leading causes of death globally. According to global status report on road safety 2015 by World Health Organization [19], road traffic injuries claim more than 1.2 million lives each year posing huge impact on health and development. More than half of those deaths are attributed to vulnerable road users (VRUs) which could be pedestrians, cyclists and motorcyclists. Even though traffic accidents and fatalities have decreased greatly over the past few decades, decrease of fatalities among VRUs is much less than all other road users [6]. Pedestrian accidents occur in roads where lines of vision are affected, road intersections, straight roads, and even in pedestrian crossings in both urban and rural areas. In recent days, distracted walking like talking and walking, listening to music or texting and inattention has become an emerging problem to pedestrians due to an exponential growth of use of mobile phones and other smartphones worldwide.

This research aims to take advantage of pervasive existence of smartphones to protect vulnerable road users instead of becoming reason for deaths. Nowadays, many VRUs have their own smartphones and the devices are outfitted with advanced onboard GPS sensors and broadband internet connections.

In this research we propose a fog computing based architecture for VRUs specifically for pedestrians using omnipresent smartphones. The potential of fog computing for intelligent transportation is stated in many literatures including [14]. The proposed architecture is an infrastructure-less solution which uses pedestrians' and drivers' smartphones crowd sensed data to detect their geographical positions and fog nodes to predict a collision risk and send warning if any.

GPS reading together with speed and direction of pedestrians and vehicles is periodically sent to the fog node through wireless connections from mobile devices. Fog node/server intakes the readings and executes pedestrian collision prediction algorithm. If there is any imminent collision it sends warning messages to both pedestrians and drivers.

Summary of our contributions in this research are:

- An advanced map matching algorithm to find the correct direction of the pedestrian.
- A three-tier fog-node based architecture that depends only on the existing infrastructure.
- Implimentation of a mobile application with the collision detection and warning algorithm.

The remainder of this article is organized as follows. Section II elucidates summary of related works. Section III experiment the GPS accuracy of smartphones under various reception conditions. In Section IV, we propose an advanced map matching algorithm to identify the direction of pedestrians. Section V describes our fog computing based architecture and pedestrian collision prediction algorithm. Section VI defines the application requirement as a proof of concept followed by performance evaluation results of LTE in Section VII. Finally conclusions are drawn in Section VIII.

2. RELATED WORKS

Many researches on pedestrian protection mechanisms are conducted to precaution VRUs. Most of these works are infrastructure-based which depends on sensors, cameras, radio tags, road side units, etc. Contemporary researches on pedestrian safety rely on smartphones of road users' together with state-of-the-art technologies to warn them of traffic accidents [18]-[23]. However, still most of this works relies on infrastructures like road side units (RSU), traffic management centers (TMC) and human machine interfaces (HMI) for vehicle-to-pedestrian (V2P) communication and some of them are not reliable, others are not scalable or have high latency.

Traffic safety can be passive or active. Passive safety involves safety countermeasures to mitigate the consequences of an accident once it has happened as much as possible while active one involves taking an action to avoid accidents by detecting, warning and automatic breaking [34], [21]. There are researches that address both types of traffic safeties. For instance, [33] proposed a new approach using state-of-the-art numerical technologies for vulnerable road user safety enhancement. The solution can detect VRUs and provide data to active safety systems to protect accidents and in case of unavoidable accident it puts passive safety structures and systems into operation to mitigate effect of collision. Some other literatures consider all types of VRUs. An urban VRU classification framework using local feature descriptors and hidden markov model have been proposed by [32] to detect and classify pedestrians, bicyclists and motorcyclists. Though the focus is on pedestrian safety, our architecture can also entertain all types of VRU as long as the VRU is equipped with smartphone. Literatures on passive pedestrian protection systems and the earlier works on active pedestrian protection systems that involves cameras, infrared, radar, tags and image processing are discussed in detail in [6].

Recently, to protect distracted users many solutions have also been proposed including designing special traffic lights [9]. The solution which is named the +Lichtlijn is linked to existing traffic lights and emits lighting strip in the pavement. These pedestrian detection mechanisms require an infrastructure and are highly affected by weather and do not work if the pedestrian is not in line of sight or at night. Therefore, researches that depend on V2P communication technologies to overcome the aforementioned problems have got attentions. Smartphones which are becoming causes of many traffic accidents are being used for safeguarding VRUs. Because of their sensing and communication capabilities smartphones are feasible for active safety of VRUs [31]. Smartphone based systems are important to protect VRUs whose line of vision is affected by buildings, trees, parked cars and other hindrances.

V2P communication prototype has been developed using 3G wireless network and WLAN to deter possible collisions by giving alarm to both pedestrians and vehicles [22]. The authors have developed an algorithm that estimates the collision risk between pedestrians and vehicles and tested the prototype at T intersection. However, the system is not scalable to apply in different road scenarios and to accommodate more road users.

In an architecture proposed by [18], information generated by vehicles' and cyclist's mobile devices is sent over heterogeneous communication architecture and processed in a central server i.e. cloud server which generates messages that are shown on the drivers' and cyclists' human-machine interface.

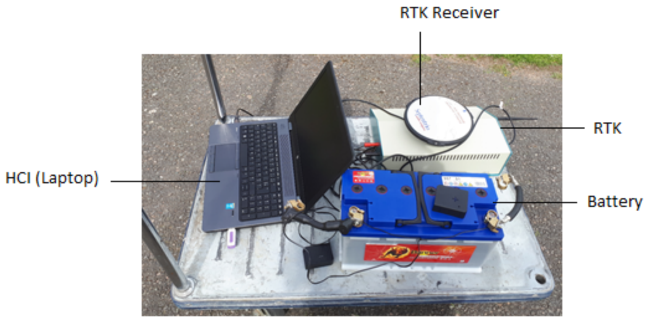


Figure 1: RTK receiver equipment

The cooperative intelligent transportation system proposed allows the use of vehicles as mobile sensors that share their positions, speed, and direction in form of floating car data with the VRUs warning each other about their locations so that they can take appropriate maneuver to avoid collisions. The system is primarily designed for cyclists and it relays on infrastructures like RSUs, TMC and HMI.

Our work differs from aforementioned related works in that it is based on a fog computing architecture that take advantage of geographically distributed fog servers in order to collect crowd sensed data from pedestrians and vehicles, predict collision risk and dispatch warning messages to road users and vehicles. As mobiles have limited capacity, a scalable architecture with low latency is mandatory. PV-Alert (Pedestrian-Vehicle alert), meets these characteristics and doesn't require special infrastructures except existing ones like users' mobiles, wireless connections and fog servers.

3. GPS ACCURACY

3.1 Experimental setup

Literatures have investigated the potential of smartphones as mobile sensors for active safety systems that are devoted to protecting vulnerable road users such as pedestrians, cyclists and motor cyclists. However, GPS readings of smartphones have to be improved in order to precisely predict and avoid traffic accidents between vehicles and vulnerable road users. There are a number of factors that affect accuracy of GPS receivers. Weather condition, obstructions, noise and interference are some of those factors that results in inaccuracy of GPS readings by delaying GPS radio signals. Even the most accurate GPS readings inaccuracy is 5m to 50m. This is inaccurate to use for safety applications like VRU collision prediction and avoidance. Even though GPS readings of latest smartphones have increased, we have worked towards improvement of GPS reading improvements.

The results indicate that the horizontal accuracy is improved by matching the trajectory of the pedestrian to one of the 8 directions. Even though, as the result shows lateral deviations are still too high, the accuracy is good enough for many active safety applications.

Our experiment consists in evaluating GPS location error of different positions with smartphones and RTK receivers (see Figure 1). The experimental design assess the relative positions of receivers situated at different known distances, according to known geographical directions, and under var-

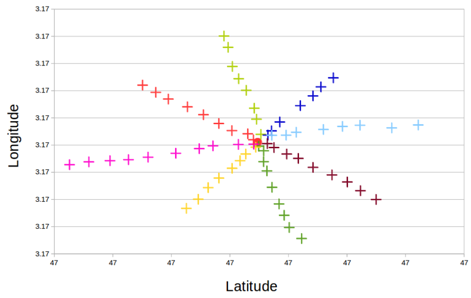
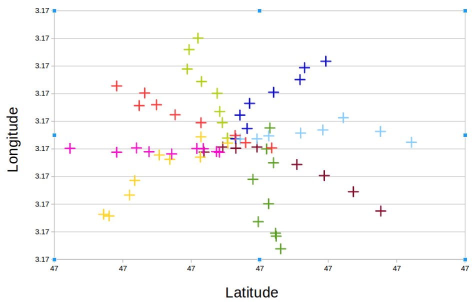
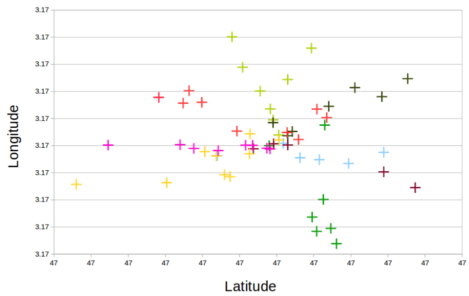


Figure 2: Measurements taken by RTK receiver



(a) Measurements taken on a sunny day



(b) Measurements taken on a cloudy day

Figure 3: Measurements taken in urban area

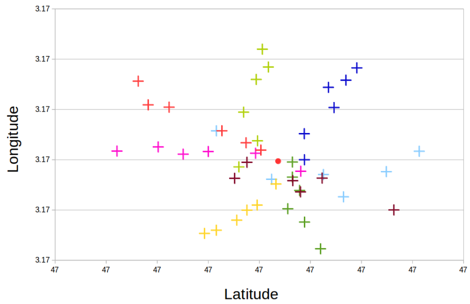
ious reception conditions. The results are intended to validate the principles of using GPS in detection collision algorithm used in the V2P (Vehicle-To-Pedestrians) project.

To make this experiment we used an Android smart-phone equipped with GPS on-board sensors, PC for recording receiver positions from RTK receiver and one tape for measuring ground distances.

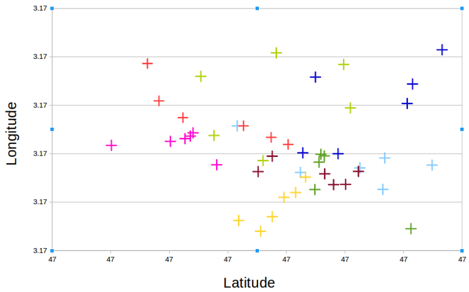
We test the GPS accuracy in 2 different areas (Urban and Rural area) and in two different weather conditions (Sunny and Cloudy day). First, We put the Android smartphone at an initial point O of known coordinates. From the point O and a direction D0 to D0+315°, and in 9 different distances, from 1m to 16m. For each 72 pairs (D, d), we take the position provided by the smartphone and RTK receiver (Latitude ϕ ° and Longitude λ °).

3.2 Results exploitation and solutions

To get GPS errors we calculate the distance between the position of the smartphone and the position of RTK receiver. Results shows that GPS location is more accurate in rural



(a) Measurements taken on a sunny day



(b) Measurements taken on a cloudy day

Figure 4: Measurements taken in rural area

area and in a sunny day than in urban area with the same weather condition, this is due to physical obstacles such as buildings and non-physical obstacles such as noise and interference with other signals. The results shows also that the GPS accuracy is higher when the whether condition is good, other parameter can also influence GPS accuracy such as the strength of the signal received and the quality of GPS on-board sensors of the smart-phone used.

Research in map-matching algorithms, which are at the heart of any navigation systems, has been focused on how to match newly obtained GPS data with road networks [8, 7, 16]. However, due to GPS errors, as explained above, even advanced map-matching algorithms sometimes are unable to find the correct road segment [20, 15]. As a less costly alternative approach to hardware-based approaches, such as differential GPS (DGPS), a software-based approach is explored in this work to reduce GPS horizontal errors. The proposed artificial neural network ANN is trained by the results of previous experiment and is used to improve the horizontal accuracy of the proceeding GPS points.

Our work in this paper explores a software-based approach to increase the correct segment identification percentage of map-matching algorithms by improving the accuracy of raw GPS points without installing additional hardware or considering external data.

4. CHAIN-CODE-BASED MAP MATCHING

Finding the correct segment is the key issue for map matching. Typically two main parameters, distance and direction deviation, are coupled to solve this problem.

The chain-code-based map matching approach was chosen because instead of computing the precise angle to represent a trend of movement, which is traditionally used, a discrete

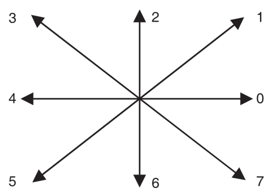


Figure 5: Eight-direction chain code

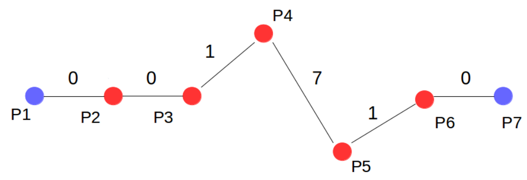


Figure 6: Digital map with GPS data

eight-direction chain code, as shown in Figure 5, can be considered in order to reduce noise from GPS due to random movements of pedestrians. Moreover, when pedestrians walk in one direction at relatively low speed, GPS data is often plagued with errors that frequently produce inaccurate trajectories. To overcome this problem, the chain-code-based map matching algorithm considers the trajectory of the data rather than merely the current position as in the typical map matching algorithms.

4.1 Eight-Direction Chain Code

Since the angle between any two adjacent links on most intersections is usually greater than 45°, we use an eight-direction Chain Code 0, 1, 2, 3, 4, 5, 6, 7 to represent eight direction interval on the counterclockwise direction as shown in Figure 5.

The directions of contour boundaries are coded with integer values k = 0, 1,..., K-1 in the counterclockwise direction starting from the direction of the positive x-axis.

In order to find the closet segment to GPS tracking points, we use the difference between the directions chain codes (D) and the trajectory chain code of a user (Dcc) to show the extent of consistency of direction among them. With this, Dcc is defined as follows:

Δ = |Chain-Code (pedestrian) – Chain-Code (direction)|

$$Dcc = \begin{cases} \Delta & \Delta < 4 \\ (|\Delta - 8|) \ mod \ 4 & otherwise \end{cases} \quad (1)$$

With this definition, discrete chain codes (Dcc) represent the angle differences between the 2 chain-codes. This representation not only eliminates noise within short-distance moving, but also is computationally fast for real-time navigation.

Furthermore, GPS position fixes are less reliable at a speed of less than 3.0 m/s. In such cases, in order to reduce the uncertainty of the direction under pedestrian users' control, the algorithm invokes a four-step Dcc between a user's trajectory and direction segments rather than only taking a one-step Dcc.

In Figure 7, P1, P2,..., P7 show the same GPS trajectory as the one in Figure 6.

With a GPS data such as P2 , Di (i = 1, 2, 3) could be calculated as the perpendicular distance to segments. When the first step is complete, the chain code from P2 to P3 is calculated, which is 0. Since heading directions are more meaningful than each-step directions, after two steps, the chain code from P2 to P4 is calculated, which is 1. Similarly, after four steps, the chain code from P2 to P5 is obtained as 1 and the chain code from P2 to P6 is obtained as 0. Therefore, Dcc between P2 P3 and AD is |0-0|, Dcc between P2 P4 and AD is |1-0|, Dcc between P2 P5 and AD is |1-0| and Dcc between P2 P6 and AD is |0-0|.

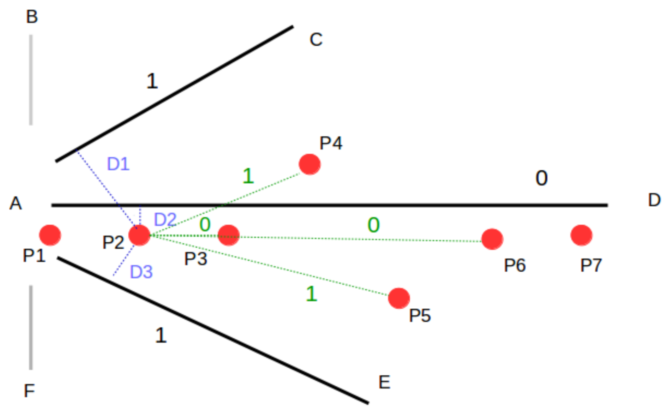


Figure 7: Example of chain-code-based map matching

4.2 Chain-Code-based Map Matching Technique

In this algorithm, distance and direction are coupled to find the best direction of the pedestrian. First, the algorithm uses perpendicular distances from GPS data to each direction segment candidate and direction difference between user's trajectory and each direction segment to select the correct direction.

As shown in Figure 7, in order to identify which segment a GPS point, such as P2 in Figure 6, is most possibly mapped onto, both distance and direction movement are calculated to be weighted scores. All the segment candidates, close to P2 , are a link set "AC", "AD", "AE"; the distances from P2 to these links are D1, D2, D3, four-step Dcc calculations between trajectory and these segment candidates are taken as described in the previous section. Next, among all the segment candidates, the direction segment with the highest matching evaluation will be chosen. In this case, segment "AD" would be considered the best selection based on map matching evaluation, and as a consequence, P2 is determined to be projected to "AD".

4.2.1 *Linear Model*

Distance and difference in direction are two determining factors to calculate matching results in order to identify the correct direction. Linear model is built on the linear relationship between matching result (M) and evaluation parameters including Dcc and Distance (D).

The linear evaluation equation used in this work is as follows:

$$V_{ij} = W_{ij} * D_{ij} + \sum_{i=0}^{4} W_{ijm} * Dcc(Step[i+m], D[j]) \quad (2)$$

$$M_{ij} = 1/V_{ij} \tag{3}$$

where:

- i is the indexing number of GPS points.
- j is the indexing number of direction segments.
- D_{ij} is the distance from the i^{th} GPS point to the j^{th} segment.
- Dcc(Step[i+m], D[j]) is the four-step Dcc with m going from 1 to 4.
- W_{ij} is the weight of D_{ij}.
- W_{ijm} is the weight of Dcc.

For the i^{th} GPS point, M_{ij} is used to calculate the total weight assigned to the j^{th} candidate link. The link with the highest M_{ij} is selected as the correct link for GPS point i. Therefore, the larger M_{ij} , the smaller is V_{ij} . In Equation 2, the total weighting score can then be obtained by summing up the individual scores, including weighting distance and four-step weighting Dcc.

4.2.2 Non-linear model

Radial Basis Functional Neural Network:

The structure of RBF networks usually has three layers. Each hidden unit in the network has two parameters: a center u_j and a width σ_j associated with it. The output of each hidden unit depends only on the radial distance between the input vector and the center parameter for that hidden unit. The response of each hidden unit is scaled by its connecting weights W_{kj} to the output units and then summed to produce the overall network output:

$$y_k(x) = \sum_{i=1}^{M} W_{kj}\theta_j(x) + W_{ko} \tag{4}$$

where the Gaussian activation function for RBF networks is given by:

$$\theta_j(x) = exp\bigg(- \frac{||x - u_j||^2}{2\sigma_j^2}\bigg) \tag{5}$$

where x is the dimensional input vector with elements x_i, u_j is the vector determining the center of the basis function, y_k(x) is the output of RBF neural network.

There are two steps to train a RBF neural network: the first step determine the parameters of the basis functions through unsupervised training using only the input data set and the second determine weights W^{kj} using both input and output data (hidden units are activated using an input pattern and the weights to the output layer are then modified to produce the desired output for the given input). Once all the parameters are produced by training in a RBF network, the neural network model could be used to compute the output based on new input data.

By definition in Equations (4.5) and (4.6), the corresponding non-linear evaluation equation used in this work is defined as follows:

$$M_{ij} = \sum_{k=0}^{n} \alpha_k * exp\bigg(- \beta \; \frac{||x_{ij} - u_k||^2}{2\sigma_j^2}\bigg) \tag{6}$$

$$V_{ij} = 1/M_{ij} \tag{7}$$

$$x_{ij} = \begin{bmatrix} D_{ij} \\ Dcc(Step[i+1], D[j]) \\ Dcc(Step[i+2], D[j]) \\ Dcc(Step[i+3], D[j]) \\ Dcc(Step[i+4], D[j]) \end{bmatrix} \tag{8}$$

- x_{ij} is the input vector.
- α_k is the weight of the k^{th} output value in hidden layer.
- β is the weight matrix of input vector in hidden layer.

As the output, M_{ij} is used to calculate the total weight assigned to the j^{th} candidate link for the i^{th} GPS point. The direction with the smallest V_{ij} is viewed as the correct direction.

4.3 Validation

To validate the chain-code-based map matching algorithm, the process of chain-code-based map matching is implemented and trained on 5 directions starting from the direction of the positive x-axis and tested on the 3 remaining directions of 8-directions chain-code as shown in Figure 5. In the following testing, GPS points in 8 directions were collected by walking using Samsung galaxy smartphone, RTK receiver is used to identify the correct direction and compare it with the output of chain-code-based map matching algorithm.

4.3.1 Evaluation of Linear Map Matching Models

For evaluation purposes, we employed different forms of linear models in matching evaluation.

The distance, four Dcc after step one, two, three are the five influencing factors in matching evaluation equation. Their linear relationship was formulated as Equation 2. After normalization for these five variables, five estimated weight parameters meet the condition as follows:

$$W_{ij} + \sum_{m=1}^{4} W_{ilm} = 1 \tag{9}$$

After testing different combinations, the four weight parameters are given as:

$$W_{ij} = 0.4, W_{ij1} = 0.1, W_{ij2} = 0.1, W_{ij3} = 0.2, W_{ij4} = 0.2$$

4.3.2 Evaluation of Non-Linear Map Matching Models

We used a RBF neural network to build a non-linear evaluation model. As mentioned in Section 4.2.2, the RBF neural network considered in this work has three layer, including 5 input layer, 132 hidden layer and 1 output layer. In the training stage, 132 pairs of parameters, with input vectors and output values, are calculated and normalized as a sample data set for training the RBF neural network. The training performance is completed after several iterations, when

Input[0] Distance (m)	Input[1] Dcc[step1]	Input[2] Dcc[step2]	Input[3] Dcc[step3]	Input[4] Dcc[step4]	Output
0.2	0	0	0	1	0.0078
0.7	0	0	0	2	0.3864
1.3	2	2	2	5	1.6598
2.8	1	2	2	3	17.271
4.9	3	4	4	5	557.87
6.2	1	1	1	2	921.43

Table 1: Map matching evaluation using RBF neural network

		Linear model		Non-linear model	
Scenarios	Number of GPS point	Correct link identification (%)	Average time (ms)	Correct link identification (%)	Average time (ms)
1	30	98.4 %	4.3	98.3 %	21.1
2	30	93.4 %	4.4	92.6 %	19.3
3	24	94.4 %	3.2	91.4 %	15.7
4	24	87.4 %	3.5	79.7 %	14.2

Table 2: Comparing the linear model and the non-linear model resulrs

error gets close to 0. This means that we could use this trained model to perform map matching evaluation.

Table 1 shows different output values according to different input vectors for a GPS point and all relevant segment candidates. It could be drawn that the first pair of input parameter results in the smallest output value in the list, which means that the corresponding segment provides the closest match to the GPS data, and thus will be identified as a selected link for the GPS data.

4.3.3 Performance

To evaluate the impact of testing environment in our map matching algorithm, we used 4 scenarios to test our 2 models: the first scenario was in urban area and in a sunny day, the second was in the same area but in a cloudy day, the third scenario was in rural area with a good weather conditions and the fourth scenario was in a rural area with a worse whether conditions as shown in Figures 3 and 4 in the previous section. For each scenario, three data sets of three different directions were processed to validate the map matching algorithm. Linear models and non-linear models are separately applied to analyze their performances in making map matching decisions.

Table 2 shows the performances of map matching by using the linear model. Compared with those implemented by other map matching algorithms for car navigation (see [15] for example), these results indicate that the chain-code-based map matching algorithm is able to achieve a better performance in terms of correct identifications using a combination of distance and direction.

Testing results show that the linear model identify more than 93% of correct link in the 3 first scenarios. Meanwhile, in the case of scenario 4, we realize that the performance is lower then 90% due to GPS errors, two directions cannot be distinguished if their distances are below the GPS error range which is generally assumed to be a 5 m or more.

The result leads to the conclusion that linear evaluation is more advantageous to implement than non-linear evaluation in terms of time performance. Compared to the linear model, map matching results in the non-linear model show slightly lower correct link identification. Therefore, in order to meet the need of real-time navigation, the linear model is preferred as the model for the map matching decision.

5. VURNERABLE ROAD USERS APPLICATION: PEDESTRIAN WARNING SYSTEM

In this section we first present proposed architecture and its components. Next, description of the algorithm including its flow chart is outlined.

5.1 System Architecture

The pictorial representation of the new architecture is shown in Figure 8. The architecture has three layers with three corresponding components crowd, fog node and cloud server.

a) A Crowd which refers to pedestrians and drivers performs the opportunistic type of crowd sensing of longitude, latitude, speed, and direction using their smartphones. Just like [5] and [18]. After appending time stamp the data is sent to fog node every second. Note that the minimum cooperative awareness message (CAM) frequency for VRU applications is set to be one second by European Telecommunication Standards Institute (ETSI). Though the architecture can work with lower frequencies, we have taken this threshold value in order to save smartphones' energy. More actions are taken to save Smartphone's energy. Pedestrians who are moving away from vehicle roads and which are not in the proximity of roads are excluded from sending the CAM message every second. Furthermore, even though smartphone owners are near roads their mobiles send data to fog node only when there is any potential risk of collision and they are moving. For instance, if a pedestrian is walking in a road and if there is no car in nearby then his mobile will not send position data to save energy consumption.

b) Fog node is another important component of PV-Alert. It has a responsibility of receiving CAM messages sent from mobile devices and executing the algorithm proposed. If any risk of collision is anticipated, the node sends alert message in real time to both pedestrian and vehicle as accidents are usually due to driver's error and/or pedestrian's carelessness. Fog nodes may be connected to smartphones using radio access networks (RAN) which could be WLAN, WiMAX or cellular networks such as LTE. The length of road segment covered by a single node is dependent on the communication technology used, it varies from hundreds of meters to kilometers. Moreover, fog nodes performs perturbation and aggregation of the collected data before sending it to the cloud. The nodes are extended to cloud server using core networks.

c) The third layer comprises of cloud server. Its responsibility in this architecture is performing aggregated analysis on data received from fog nodes for further use in traffic analysis and decision making. Edge location of fog nodes in bus stations, supermarkets, and road side buildings make fog computing an ideal solution for latency sensitive applications like traffic safety [4].

5.2 Algorithm Description

At fog level, we developed a pedestrian collision prediction algorithm in order to anticipate collisions and send alerts to pedestrians and drivers. The algorithm can be applied to any road scenarios.

In initialization step, pedestrians' and drivers' mobiles are subscribed/connected to the server. Smartphones send

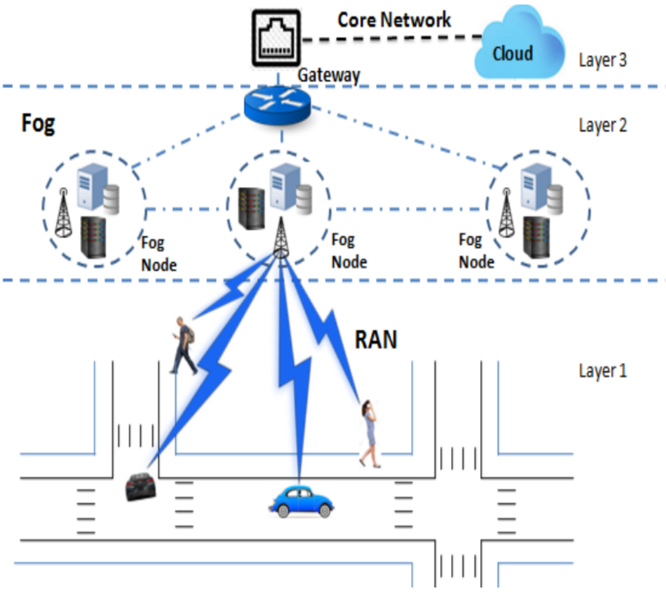


Figure 8: Architecture of PV-Alert

data to the server every second (maximum CAM beacon frequency set by ETSI). The next step is identifying intersecting pedestrian and vehicles using the sensed data, road information and the source, destination as well as path of each vehicle. Intersecting pedestrians and vehicles are those which cross each other posing collision risk. The most crucial module is collision risk prediction module. Its duty is estimating potential dangers of collisions between pedestrians and vehicles at particular instant of time. For this purpose, minimum information exchange distance (D_{min}) is computed using formula (10) which is taken from [1].

$$D_{min} = V_{veh} * (t_p + t_r + t_{tx} + t_c) + GPS_{veh} + GPS_{ped} \quad (10)$$

Where,

- V_{veh} = is velocity of a vehicle
- t_p = is time for perception and it is 0.83s [1]
- t_r = is time for reaction and it is 0.15s for touch [2]
- t_{tx} = is transmission delay
- t_c = is time for computation of algorithm
- GPS_{veh} and GPS_{ped} are GPS errors of vehicles and pedestrians positions.

Next, actual distance from vehicle to pedestrian crossing is calculated. Actual distance is simply calculated using distance formula as locations of the two parties are known and as we have a database storing road information. Based on D_{min} and actual distance it can be determined whether there is an imminent collision or not. If actual distance is less than D_{min} then the pedestrian is in a collision risk region and warning message is sent to both driver and pedestrian. The algorithm runs indefinitely in iteration in fog node. In order to evaluate the architecture in simulated environment the algorithm is implemented and installed in a representation of fog node.

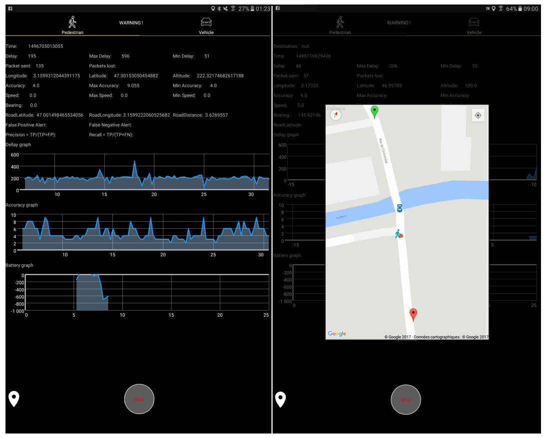


Figure 9: Detail of the user interface

5.3 System Overview

As a proof of concept we developed a mobile application in the Android system of the Samsung Galaxy and implement the collision detection algorithme on the server in a Linux system with JAVA. The scenario that can be implemented is smaller than the simulated one in terms of area and number of vehicles and pedestrians, but it can be really useful to obtain data about the application and the communication reliability between pedestrians and vehicles that are equipped with different communication technologies. To be able to test our application and locate the pedestrian, it is necessary to broadcast their position to the server. the server decide whether the situation is credible or not and sends alerts when there is a height probability of collision.

As it is shown at Figure 9, before running the application, users can act as pedestrians or drives so they have to choose between this two different modes of operation. The Android application for pedestrians periodically provides information about the location and direction of the pedestrian to the server, and shows a view of the road he is riding with the location of the nearby vehicles. Thus, the pedestrian will be able to place himself/herself in a better position and be ready for being overtaken by a car. The mobile application for drivers permanently collects information on the position and kinematics of the vehicle and periodically transmits them over Wi-Fi interface using UDP. Wifi communication network is used in order to intercommunicate all the agents involved in V-Alert: pedestrian (that act as Vulnerable Road Users), vehicles, and the central server known as fog computing.

6. COMMUNICATION EVALUATION

6.1 Simulation setup

The main objective of our research is to enable drivers & vehicles avoid collision by notifying each of them trough their mobile devices. Mobile smartphones could communicate with the fog node using Wi-Fi, WiMAX or cellular connections like LTE. Different works already show that both Wi-Fi [1] and cellular [3], [11] connections can be used for traffic safety applications. Our simulations are done for LTE.

The fog node is placed in proximity of eNodeB. Our simu-

Parameter	Value
General Parameters	
Packet Size	1KB
Simulation Time	120s
CAM Frequency	1Hz
SUMO Parameters	
Vehicle Speed	10-80kmph
Pedestrian Speed	5kmph
Simulation Area	3000m x 60m
Scenario	Refer Figure 10
LTE	
TxPower(eNB)	25dB
TxPower (UE)	15dB
Downlink bandwidth	30MB
Uplink bandwidth	25MB

Table 3: Simulation parameters

lation is different from evaluations of other similar works in that our test includes large number pedestrians and vehicles. Such simulation is very important, since it is very difficult to run on real environment.

Our architecture is evaluated using discrete-event open network simulation environment ns-3 and microscopic, multi-modal traffic simulation tool SUMO. Both tools are most widely used by many researchers since they are open source and are being actively supported. Most important simulation parameters applied are outlined in Table 3.

The first step in conducting our simulation is to get mobility trace file from SUMO. We obtain the file after setting all required SUMO parameters for a specific simulation scenario and running the simulation. Default arrival rate of SUMO is used for the traffic. The mobility trace file is then used by ns-3 to create node mobility. We have considered the following two performance metrics:

- Round Trip Delay time (RTD): the time period from sending CAM beacon to the fog node to receiving a warning message in case of anticipated accident.
- Packet Delivery Ratio (PDR): the average ratio of packets received by fog node and smartphones to the total packets sent to fog node (from smartphones) and to smartphones (from the fog node).

RTD is computed using the following formula:

$$RTD = T_{sp-fn} + T_c + T_{fn-sp}$$

Where,

- T_{sp-fn} is end-to-end delay from smartphones to fog node.
- T_{fn-sp} is end-to-end delay from fog node to smartphones.
- c is computation time of the algorithm.

We used the following equation to calculate PDR:

$$PDR = 100 * \frac{1}{2} * (\frac{P_{Rec-sp}}{P_{Gen-fn}} + \frac{P_{Rec-fn}}{P_{Gen-sp}})$$

Where,

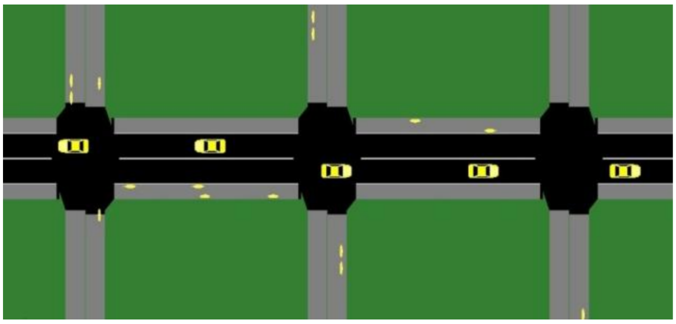


Figure 10: Simulation scenario[1]

- P_{Gen-sp} is total number of packets generated by smartphones.
- P_{Rec-sp} is total number of packets received by smartphones.
- P_{Rec-fn} is total number of packets received by fog node.
- P_{Gen-fn} is total number of packets generated by fog node.

6.2 Simulation Scenario

The road scenario considered is a two lane straight road with many pedestrian roads crossing it and pedestrians' line of vision is affected by buildings, trees, parked cars and other hindrances see Figure 10. The reason for why we have chosen this scenario is vehicles speed is high in straight road and pedestrians coming out of smaller intersecting roads are highly susceptible to traffic accidents due to affected line of vision, distraction and inattention.

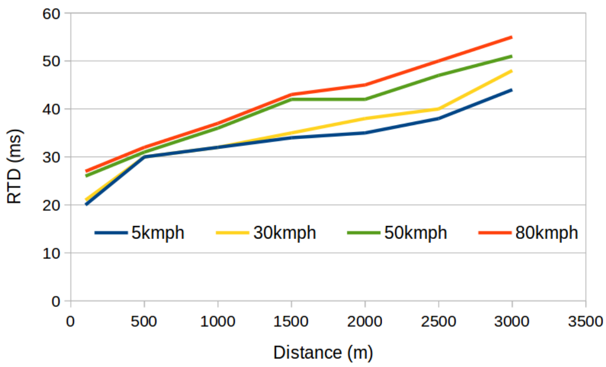
Moreover, severity of a straight road crash is 1.7 (0.9 - 3.2) times more serious or fatal outcome than on non-straight roads [12]. Even though many researches focus on crossing roads [1] and T roads [18], literatures reveal high percentage of road accidents in straight roads; 80% [9], 89.8% [10], and 93% [13]. Pedestrian accidents are even common in pedestrian crossings [17].

6.3 Simulation Results and Discussions

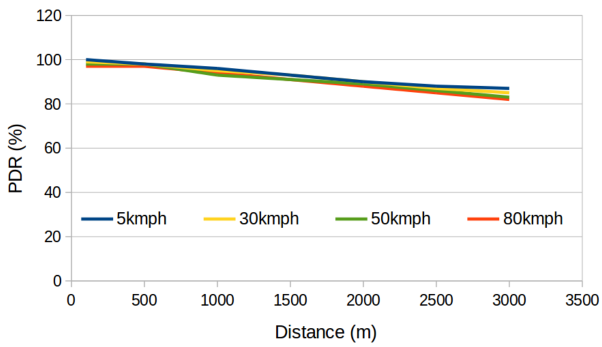
Experimental result of round trip time delay and packet delivery ratio when an LTE connection is used by smartphones to communicate with the fog node is given in Figure 11 and 12. Round trip time delay increases from 20ms to 60ms as the nodes move away from the fog node, as shown in Figure 11a. The delay fulfils the application requirement and we can notice that the difference between fast-moving vehicles and slow-moving vehicles is not significant due to high mobility support of LTE [28], [29]. The same is true for the PDR as shown in Figure 11b. PDR exceeds 80% for all distances and the difference among vehicles moving at different speeds is insignificant. However, there is a tendency of PDR decrease as speed and distance increase.

Figure 12. shows the impact of the number of vehicles and pedestrians on delay and packet delivery ratio at different distances. Due to the high mobility support of LTE, we can observe that the speed does not significantly impact

[1]Smaller dots in the figure represents pedestrians

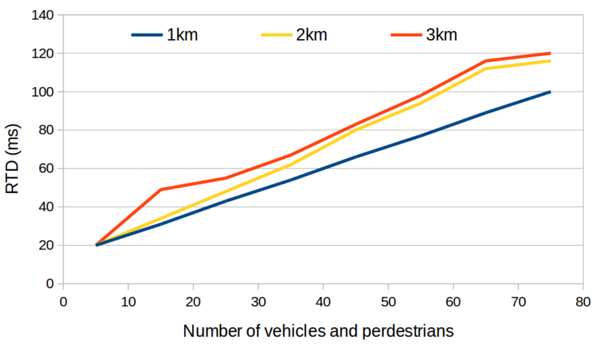


(a) RTD vs. distance

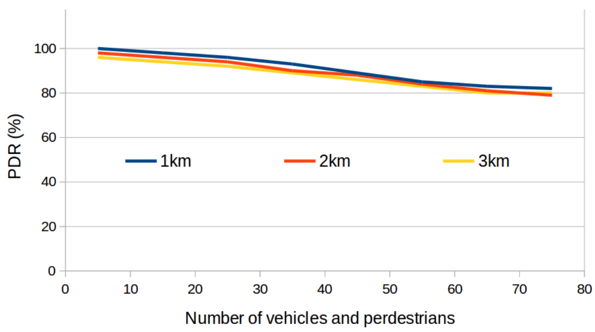


(b) PDR vs. distance

Figure 11: LTE Connection with the fog node



(a) RTD vs. number of vehicles and pedestrians



(b) PDR vs. number of vehicles and pedestrians

Figure 12: LTE Connection with the fog node

the delay and PDR. Therefore, the experiments were realized for different average distances and a velocity of vehicles that does not exceed 80kmph in urban scenarios. At average distance of 1KM, all nodes' delay is below the threshold value; however, as vehicles and pedestrians average distance increase (2KM and then 3KM), the delay increases due to the signal attenuation combined with network congestion. In Figure 12a, if the number of vehicles is 55 or below, the system satisfies delay requirement at any distance. PDR is more affected by the number of nodes than distance as depicted in Figure 11b. The average PDR of uplink and downlink is above 80% for all the distances though it gradually decreases as the number of nodes increases. Moreover, we have noticed that downlink PDR is higher than uplink due to downlink's higher bandwidth.

7. REFERENCES

[1] J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi, and J. E. Naranjo. Vehicle to pedestrian communications for protection of vulnerable road users. pages 1037–1042, 06 2014.

[2] backyardbrains.com. Experiment: How fast your brain reacts to stimuli. `https://backyardbrains.com/experiments/reactiontime`, 2016.

[3] M. Bagheri, M. Siekkinen, and J. K. Nurminen. Cellular-based vehicle to pedestrian (v2p) adaptive communication for collision avoidance. pages 450–456, Nov 2014.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. pages 13–16, 2012.

[5] K. David and A. Flach. Car-2-x and pedestrian safety. 5:70–76, 03 2010.

[6] Europa.eu. 2016 road safety statistics: What is behind the figures. `http://europa.eu/rapid/press-release_MEMO-17-675_en.htm`, 2017 [Accessed: 20 – Apr- 2017].

[7] M. Hashemi and H. A. Karimi. A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48:153 – 165, 2014.

[8] M. Hashemi and H. A. Karimi. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal of Intelligent Transportation Systems*, 20(6):573–590, 2016.

[9] E. T. S. Institute. Intelligent transport systems (its); vehicular communications; basic set of applications; definitions. 2009.

[10] R. C. J Knowles, L Smith and E. Delmonte. Analysis of police collision files for pedestrian fatalities in london. 2006.

[11] S. Kato, M. Hiltunen, K. Joshi, and R. Schlichting. Enabling vehicular safety applications over lte networks. pages 747–752, Dec 2013.

[12] C. B. Keum. Analysis of road traffic crashes and injury severity of pedestrian victims in the gambia. 2016.

[13] D. Kouabenan and J. Guyot. Analysis of road accidents on pedestrian crossings caused by speeding. pages 119–126, 2004.

[14] A. Munir, P. Kansakar, and S. U. Khan. IFCIoT: Integrated Fog Cloud IoT Architectural Paradigm for

Future Internet of Things. *ArXiv e-prints*, Jan. 2017.

[15] M. A. Quddus, W. Y. Ochieng, L. Zhao, and R. B. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7(3):157–167, Dec 2003.

[16] M. Ren and H. A. Karimi. A chain-code-based map matching algorithm for wheelchair navigation. 13:197–214, 04 2009.

[17] J. Rokytova and M. Sklenar. Study of the causes of pedestrian accidents by severity. October 2002.

[18] C. Sugimoto, Y. Nakamura, and T. Hashimoto. Development of pedestrian-to-vehicle communication system prototype for pedestrian safety using both wide-area and direct communication. pages 64–69, March 2008.

[19] Who.int. Global status report on road safety 2015. `http://www.who.int/violence_injury_prevention/road_safety_status/2015/en`, 2016 [Accessed: 15-Jan- 2017].

[20] L. Zhao, W. Y. Ochieng, M. Quddus, and R. Noland. An extended kalman filter algorithm for integrating gps and low cost dead reckoning system data for vehicle performance and emissions monitoring. 56, 05 2003.