Mapping of Sensor and Route Coordinates for Smart Cities

Yasir Saleem, Noel Crespi
Institute Mines-Telecom, Telecom SudParis, France
yasir_saleem.shaikh@telecom-sudparis.eu, noel.crespi@mines-telecom.fr

***Abstract*—Over the last decade, the evolution of the Internet of Things (IoT) has resulted in a drastic increase in the development of smart cities, including smart parking and intelligent transportation systems (ITS). Smart cities combine a variety of sensors (such as traffic, parking and weather sensors) deployed within these cities. These sensors are used for various applications, such as transportation, parking and weather forecasting. We propose an approach for the mapping of traffic sensors with route coordinates in order to analyze traffic conditions (*e.g.,* level of congestion) on the roadways. We present an algorithm and provide two illustrative examples that cover all of the possible mapping scenarios. We also evaluate the performance of our proposed approach in terms of sensors' correct detection, missed detection and false detection on the routes. Our work can be used for the development of various smart city applications, such as traffic management and smart parking.**

***Index Terms*—Internet of Things (IoT), smart cities, coordinate mapping, sensor coordinates, route coordinates.**

I. Introduction

In the past few years, the Internet of Things (IoT) has attracted increasing interest in its potential application in a number of domains, including transportation, healthcare, smart cities and the smart grid. The exponential growth in urban populations has resulted in a higher number of cars in cities than ever before, causing traffic congestion and higher pollution levels. A number of IoT solutions to address this problem have been developed for intelligent transportation systems (ITS) through various smart city projects [1], [2], [3], [4].

In smart cities, several types of sensors are installed throughout a city, such as traffic, parking and weather sensors. Traffic sensors, deployed on the roads/streets, monitor the traffic conditions, such as the road load, the number of vehicles and their speed. Such traffic sensors are useful in recommending the least congested route towards a destination point, as well as in distributing the traffic to different routes to reduce the traffic congestion. According to one study, 30% of traffic congestion is the result of drivers looking for parking spots [5]. Therefore, by managing the traffic crowd, we can also efficiently manage parking spots, *e.g.,* the parking system can take the traffic congestion into account through traffic sensors and subsequently can recommend different routes leading to parking spots for those seeking them, thereby minimizing the scenario of high traffic congestion on some routes and low traffic congestion on the other routes. In this manner, we can achieve a more balanced distribution of traffic on all the routes.

However, there is a challenge to achieve the above objective. Generally, most solutions focus on the development of applications assuming that they have readily available information about the mapping of traffic sensors on the routes. These solutions do not focus on the mapping of sensor coordinates into the routes. Inspired by this lacunae, this paper presents a new approach for the mapping of sensor coordinates into the routes. We propose an algorithm for this mapping, utilizing real traffic sensors deployed in the city of Santander, Spain to demonstrate their mapping abilities with four random routes between two points. Since traffic sensors are deployed in large numbers, some traffic sensors may reside outside of the edges of streets (*i.e.,* a small distance from the routes). In order to best incorporate this constraint, our proposed approach considers a deviation margin that allows some flexibility; the main novelty and contribution of this paper. We evaluate the performance of our proposed approach in terms of correct detection, missed detection (non-detection) and false detection of sensors on the routes, showing the effective and significant advantage of our proposed approach. We believe that our proposed approach will be helpful in the development of various smart city applications, such as traffic management and smart parking.

The remainder of this paper is organized as follows. The related work is presented in Section II. Section III presents an algorithm for the mapping of sensors into route coordinates. We provide two illustrative examples covering all the aspects and scenarios of mapping in Section IV. In Section V, we evaluate the performance of our proposed approach, and then conclude the paper in Section VI.

II. Related Work

While much work is being done in IoT for the development of smart cities, that work is mainly focused on the applications, such as smart parking, ITS, traffic management etc. Those studies do not consider the mapping of sensor coordinates into the routes, to the best of our knowledge. For instance, Lau [3] designed a framework that considers user-contributed posts about traffic and road conditions and analyzes driving navigation information from the archived data on online social media. Subsequently, it transmits the collected data to ITS for its dissemination to other drivers. However, this framework considers that raw data collection from sensors and their mapping to the routes are not really within the objectives of those cities. Another work [2] studied the joint prediction of road traffic and parking occupancy in a city by using machine learning techniques. However, it does not work on the mapping of traffic sensors with the routes, instead this technique is mainly focused on applying machine learning for predicting road traffic and parking occupancy by using real-time data. Fernandez et al., [1] studied real traffic and mobility scenarios for a smart city by using real traffic and mobility data gathered in the city of Granada, Spain. Their main purpose was to analyze the collected data using Big Data techniques and subsequently derive useful information. However, they also do not focus on the mapping of sensor coordinates with route coordinates and simply assume the availability of such mapping.

One important observation is that traffic prediction is a very old problem; but we are not working on traffic prediction. We have proposed a novel approach for the mapping of sensor coordinates with route coordinates, which is the basic pre-requisite of traffic prediction.

III. Mapping of Sensor Coordinates with Route Coordinates

In this section, we present our proposed approach for the mapping of sensor coordinates with route coordinates. The traffic sensors used in Santander, Spain are magneto-resistive sensors that detect the movement and presence of vehicles. They operate at 2.4 GHz frequency band and 250Kbps data rate. They are located at the main entrances of the Santander city and are buried under the asphalt. They measure the main traffic parameters, *e.g.,* road occupancy, vehicle speed, traffic volumes and queue length [6]. For simplicity, we have assumed that the incoming and outgoing traffic on two-way roads is similar. Therefore, the identified traffic sensors on either side of the road represent similar traffic conditions. However, in future extensions of our work, we plan to identify traffic sensors separately on each side of the two-way roads. This assumption is reasonable for our current system as we deployed it for testing purposes in Santander, Spain where the traffic sensors are deployed on one-way streets.

Algorithm 1 presents the mapping of traffic sensors' coordinates with a route's coordinates. The inputs of this algorithm are route coordinates (longitudes and latitudes) which are comprised of starting coordinates (route start longitude $R_{strt,lon}$ and route start latitude $R_{strt,lat}$) and ending coordinates (route end longitude $R_{end,lon}$ and route end latitude $R_{end,lat}$), traffic sensor coordinates (traffic sensor longitude TS_{lon} and traffic sensor latitude TS_{lat}) and finally the deviation margin D. D is a tunable parameter which can be set based on the scenario (*i.e.,* how much is the deviation of the traffic sensors' coordinates from the main route, in general). We will explain deviation margin D in detail in the following discussion.

Part I of the algorithm checks whether the route start longitude $R_{strt,lon}$ is less than or equal to the route end longitude $R_{end,lon}$; a necessary step in order to determine the direction of the route. This part consists of two sub-parts: Part I(a) checks the exact location of traffic sensors on the route, while Part I(b) uses a deviation margin D which gives some flexibility to traffic sensors that are deviated from the main route.

Algorithm 1 Mapping of traffic sensor coordinates with route coordinates.

Input: Route start longitude ($R_{strt,lon}$), route end longitude ($R_{end,lon}$), route start latitude ($R_{strt,lat}$), route end latitude ($R_{end,lat}$), traffic sensor longitude (TS_{lon}), traffic sensor latitude (TS_{lat}), deviation margin (D)
/ Part I */*
if $R_{strt,lon} <= R_{end,lon}$ **then**
    / Part I(a) */*
    if $TS_{lon} >= R_{strt,lon}$ & $TS_{lon} <= R_{end,lon}$ & $TS_{lat} >= R_{strt,lat}$ & $TS_{lat} <= R_{end,lat}$ **then**
        isMatched = true;
    else if $TS_{lon} >= R_{strt,lon}$ & $TS_{lon} <= R_{end,lon}$ & $TS_{lat} <= R_{strt,lat}$ & $TS_{lat} >= R_{end,lat}$ **then**
        isMatched = true;
    end if
    / Part I(b) */*
    if (not isMatched) **then**
        if ($TS_{lon} + D >= R_{strt,lon}$ & $TS_{lon} - D <= R_{end,lon}$) & ($TS_{lat} + D >= R_{strt,lat}$ & $TS_{lat} - D <= R_{end,lat}$) **then**
            isMatched = true;
        else if ($TS_{lon} + D >= R_{strt,lon}$ & $TS_{lon} - D <= R_{end,lon}$) & ($TS_{lat} - D <= R_{strt,lat}$ & $TS_{lat} + D >= R_{end,lat}$) **then**
            isMatched = true;
        end if
    end if
    / Part II */*
else if $R_{strt,lon} > R_{end,lon}$ **then**
    / Part II(a) */*
    if $TS_{lon} <= R_{strt,lon}$ & $TS_{lon} >= R_{end,lon}$ & $TS_{lat} <= R_{strt,lat}$ & $TS_{lat} >= R_{end,lat}$ **then**
        isMatched = true;
    else if $TS_{lon} <= R_{strt,lon}$ & $TS_{lon} >= R_{end,lon}$ & $TS_{lat} >= R_{strt,lat}$ & $TS_{lat} <= R_{end,lat}$ **then**
        isMatched = true;
    end if
    / Part II(b) */*
    if (not isMatched) **then**
        if ($TS_{lon} - D <= R_{strt,lon}$ & $TS_{lon} + D >= R_{end,lon}$) & ($TS_{lat} - D <= R_{strt,lat}$ & $TS_{lat} + D >= R_{end,lat}$) **then**
            isMatched = true;
        else if ($TS_{lon} - D <= R_{strt,lon}$ & $TS_{lon} + D >= R_{end,lon}$) & ($TS_{lat} + D >= R_{strt,lat}$ & $TS_{lat} - D <= R_{end,lat}$) **then**
            isMatched = true;
        end if
    end if
end if
return isMatched;

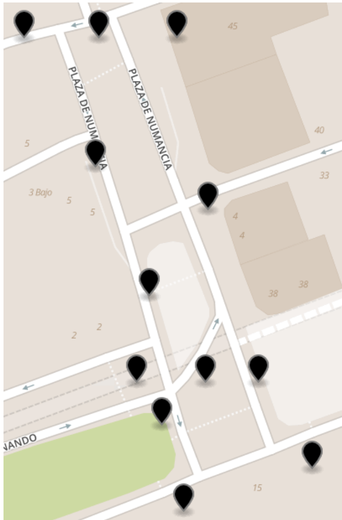

Fig. 1. Irregular deployment of traffic sensors at Santander, Spain.

In Part I(a), there are two possible scenarios for the existence of traffic sensor coordinates on the route. Firstly, the traffic sensor's longitude TS_{lon} is greater than or equal to the route's start longitude $R_{strt,lon}$ (*i.e.,* $TS_{lon} \geq R_{strt,lon}$) and less than or equal to the route's end longitude $R_{end,lon}$ (*i.e.,* $TS_{lon} \leq R_{end,lon}$), and the traffic sensor's latitude TS_{lat} is greater than or equal to the route's start latitude $R_{strt,lat}$ (*i.e.,* $TS_{lat} \geq R_{strt,lat}$) and less than or equal to the route's end latitude $R_{end,lat}$ (*i.e.,* $TS_{lat} \leq R_{end,lat}$). Secondly, the traffic sensor's longitude TS_{lon} is greater than or equal to route's start longitude $R_{strt,lon}$ (*i.e.,* $TS_{lon} \geq R_{strt,lon}$) and less than or equal to the route's end longitude $R_{end,lon}$ (*i.e.,* $TS_{lon} \leq R_{end,lon}$), and the traffic sensor's latitude TS_{lat} is less than or equal to the route's start latitude $R_{strt,lat}$ (*i.e.,* $TS_{lat} \leq R_{strt,lat}$) and greater than or equal to the route's end latitude $R_{end,lat}$ (*i.e.,* $TS_{lat} \geq R_{end,lat}$).

However, as presented in Fig. 1, given the irregular deployment of traffic sensors in Santander, Spain, it might be possible that a traffic sensor is slightly deviated from the main route. For this situation, we propose a deviation margin D in order to give some flexibility in the detection of traffic sensors on the route. Hence, in Part I(b) of the algorithm, we have added the deviation margin D to the traffic sensor coordinates when comparing them with the route start coordinates (*i.e.,* while checking whether a traffic sensor's coordinates are greater than the route start coordinates) and subtracted the deviation margin D from traffic sensor coordinates when comparing with route end coordinates (*i.e.,* while checking whether a traffic sensor's coordinates are less than the route's end coordinates). This helps to give some flexibility for the traffic sensor coordinates that are deviated from the straight path of the route. Subsequently, after adding and subtracting the deviation margin D to and from the traffic sensor coordinates when comparing them with start and end route coordinates, respectively, they are compared in a similar manner as presented in Part I(a).

If the route start longitude is greater than route end longitude of route, Part II will be executed. Part II also consists of two sub-parts: Part II(a) checks the exact existence of traffic sensor on the route, while Part II(b) uses a deviation margin D, which gives some flexibility to traffic sensors which are deviated from the main route.

In Part II(a), there are again two possible scenarios for the existence of traffic sensor coordinates on the route. Firstly, the traffic sensor's longitude TS_{lon} is less than or equal to the route's start longitude $R_{strt,lon}$ (*i.e.,* $TS_{lon} \leq R_{strt,lon}$) and greater than or equal to the route's end longitude $R_{end,lon}$ (*i.e.,* $TS_{lon} \geq R_{end,lon}$), and the traffic sensor's latitude TS_{lat} is less than or equal to the route's start latitude $R_{strt,lat}$ (*i.e.,* $TS_{lat} \leq R_{strt,lat}$) and greater than or equal to the route's end latitude $R_{end,lat}$ (*i.e.,* $TS_{lat} \geq R_{end,lat}$). Secondly, the traffic sensor's longitude TS_{lon} is less than or equal to the route's start longitude $R_{strt,lon}$ (*i.e.,* $TS_{lon} \leq R_{strt,lon}$) and greater than or equal to the route's end longitude $R_{end,lon}$ (*i.e.,* $TS_{lon} \geq R_{end,lon}$), and the traffic sensor's latitude TS_{lat} is greater than or equal to the route's start latitude $R_{strt,lat}$ (*i.e.,* $TS_{lat} \geq R_{strt,lat}$) and less than or equal to the route's end latitude $R_{end,lat}$ (*i.e.,* $TS_{lat} \leq R_{end,lat}$).

If the traffic sensor is not identified on the route in Part I, Part II uses a deviation margin D to check the existence of traffic sensors on the route. The reason for using a deviation margin is explained above in the description of Part I(b) of the algorithm. However, in contrast to Part I(b), in Part II(b), the deviation margin D is subtracted from the traffic sensor's coordinates when comparing them with the route's start coordinates (*i.e.,* while checking whether a traffic sensor's coordinates are less than the route's start coordinates), and added to the traffic sensor's coordinates when comparing with the route's end coordinates (*i.e.,* while checking whether a traffic sensor's coordinates are greater than the route's end coordinates). Subsequently, after subtracting and adding the deviation margin D from and to the traffic sensor's coordinates when comparing them with the start and end route coordinates, respectively, this part is compared in a similar procedure in Part II(a).

Note that although we can combine the two '*if*' conditions into one within each sub-part, we have presented them separately for a better understanding, as well as a better differentiation using the examples (presented in Section IV). We have combined them in our implementation in order to reduce the processing load.

IV. Examples

For better understanding of the algorithm, we present two illustrative examples to demonstrate the operation of the algorithm for the mapping of traffic sensor coordinates with route coordinates. The first example demonstrates the scenario in which the traffic sensor lies exactly within the routes (*i.e.,* there is no deviation), while the second example demonstrates the scenario in which traffic sensors are slightly deviated from the route, and hence we will use deviation margin D to detect such traffic sensors on the route.

A. Example 1 (no deviation)

In this section, we demonstrate the first scenario in which traffic sensor lies exactly within the route without any deviation.

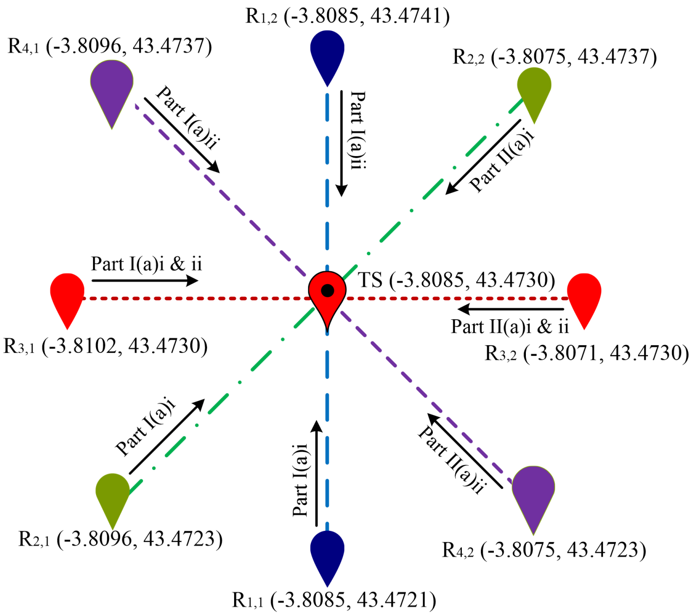

Fig. 2. An example of coordinates mapping having no deviation. The lines originating from and terminating at map markers, represent the routes. The arrowhead lines represent the direction of the route (*i.e.,* starting and ending points) and the "Part xyz" above the arrowhead lines corresponds to the matching parts mentioned in Algorithm 1.

Fig. 2 presents four different routes covering all the possible scenarios of coordinates mapping with different colors and line patterns. The traffic sensor which is our main point of interest to be detected on the route is located at the center (*i.e.,* the red map marker with black dot in the middle). For all the coordinates of map markers in the figure, the first part represents the longitude and the second part represents the latitude. For example, TS (-3.8085, 43.4730) shows the traffic sensor having longitude = -3.8085 and latitude = 43.4730. Similarly, $R_{1,1}$(-3.8085, 43.4721) represents point 1 (can be either starting or ending point) of route 1 having longitude = -3.8085 and latitude = 43.4721.

Let us start with the route in blue color $R_{1,1} \rightarrow R_{1,2}$ by considering $R_{1,1}$ (-3.8085, 43.4721) as starting point and $R_{1,2}$ (-3.8085, 43.4741) as ending point. Here, the route start longitude (-3.8085) is equal to route end longitude (-3.8085), so it matches Part I of Algorithm 1. Within Part I, it satisfies the first condition of Part I(a), *i.e.,* traffic sensor longitude (-3.8085) is equal to the route start longitude (-3.8085) and is also equal to the route end longitude (-3.8085), and traffic sensor latitude (43.4730) is greater than route start latitude (43.4721) and is less than route end latitude (43.4741). Hence, the traffic sensor is identified within the route. In a similar manner, the green route $R_{2,1} \rightarrow R_{2,2}$ with $R_{2,1}$ (-3.8096, 43.4723) as starting point and $R_{2,2}$ (-3.8075, 43.4737) as ending point also fulfills the first condition of Part I(a). For the route in red color $R_{3,1} \rightarrow R_{3,2}$ with $R_{3,1}$ (-3.8102, 43.4730) as starting point and $R_{3,2}$ (-3.8071, 43.4730) as ending point, it meets both conditions defined in Part I(a) of the algorithm. Hence, the traffic sensor is identified within the route by fulfilling both conditions of Part I(a). The purple route $R_{4,1} \rightarrow R_{4,2}$ with $R_{4,1}$ (-3.8096, 43.4737) as starting point and $R_{4,2}$ (-3.8075, 43.4723) as ending point, as well as blue route $R_{1,2} \rightarrow R_{1,1}$ with $R_{1,2}$ (-3.8085, 43.4741) as starting point and $R_{1,1}$ (-3.8085, 43.4721) as ending point, fulfill the second condition of Part I(a).

The above described routes match Part I of the algorithm, while the remaining routes match Part II of the algorithm which we are going to present next. The green route $R_{2,2} \rightarrow R_{2,1}$ with $R_{2,2}$ (-3.8075, 43.4737) as starting point and $R_{2,1}$ (-3.8096, 43.4723) as ending point matches Part II of the algorithm because start route longitude (-3.8075) is greater than end route longitude (-3.8096). Within Part II, it matches the first condition of Part II(a), *i.e.,* traffic sensor longitude (-3.8085) is less than route start longitude (-3.8075) and greater than route end longitude (-3.8096), and traffic sensor latitude (43.4730) is less than route start latitude (43.4737) and greater than route end latitude (43.4723). Similar to the red route in Part I(a), the red route $R_{3,2} \rightarrow R_{3,1}$ with $R_{3,2}$ (-3.8071, 43.4730) as starting point and $R_{3,1}$ (-3.8102, 43.4730) as ending point fulfills both conditions defined in Part II(a) of the algorithm. Finally, the purple route $R_{4,2} \rightarrow R_{4,1}$ with $R_{4,2}$ (-3.8075, 43.4723) as starting point and $R_{4,1}$ (-3.8096, 43.4737) as ending point matches the second condition in Part II(a).

Hence, the traffic sensor is successfully identified within all the routes $R_{1,1} \rightarrow R_{1,2}$, $R_{2,1} \rightarrow R_{2,2}$, $R_{3,1} \rightarrow R_{3,2}$, $R_{4,1} \rightarrow R_{4,2}$, $R_{1,2} \rightarrow R_{1,1}$, $R_{2,2} \rightarrow R_{2,1}$, $R_{3,2} \rightarrow R_{3,1}$ and $R_{4,2} \rightarrow R_{4,1}$. This example also verifies the correct operation of Algorithm 1 without deviation margin D.

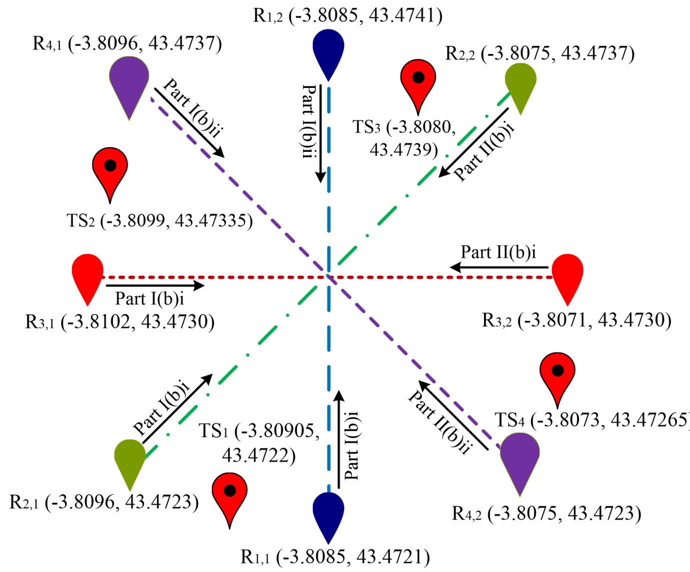

Fig. 3. An example of coordinates mapping with deviation. The lines originating from and terminating at map markers, represent the routes. The arrowhead lines represent the direction of the route (*i.e.,* starting and ending points) and the "Part xyz" above the arrowhead lines corresponds to the matching parts mentioned in Algorithm 1.

B. Example 2 (with deviation)

In this section, we demonstrate the second scenario in which traffic sensors do not lie within the exact routes, rather they are deviated from the routes.

Similar to previous figure, Fig. 3 presents four different routes covering all the possible scenarios with different colors and line patterns. Unlike Fig. 2, due to deviation, we cannot have a single traffic sensor which can cover all the possible scenarios, therefore, we have presented four traffic sensors (*i.e.,* TS_1, TS_2, TS_3 and TS_4) to be detected. Each traffic sensor, located between the two routes, is used for both routes separately to check its existence on the route by using deviation margin D. For instance, traffic sensor TS_1 is used for routes $R_{1,1} \rightarrow R_{1,2}$ and $R_{2,1} \rightarrow R_{2,2}$, TS_2 is used for routes $R_{3,1} \rightarrow R_{3,2}$ and $R_{4,1} \rightarrow R_{4,2}$, TS_3 is used for routes $R_{1,2} \rightarrow R_{1,1}$ and $R_{2,2} \rightarrow R_{2,1}$, and TS_4 is used for routes $R_{3,1} \rightarrow R_{3,2}$ and $R_{4,1} \rightarrow R_{4,2}$. For all the coordinates of map markers in the figure, the first part represents the longitude and the second part represents the latitude. For example, TS_1 (-3.80905, 43.4722) shows the traffic sensor 1, having longitude = -3.80905 and latitude = 43.4722. Similarly, $R_{1,1}$(-3.8085, 43.4721) shows point 1 (can be either starting or ending point) of route 1 ($R_{1,1} \rightarrow R_{1,2}$) having longitude = -3.8085 and latitude = 43.4721. Note that in this example, we have set deviation margin $D = 0.0006$.

Let us start with the route in blue color $R_{1,1} \rightarrow R_{1,2}$ by considering $R_{1,1}$ (-3.8085, 43.4721) as starting point and $R_{1,2}$ (-3.8085, 43.4741) as ending point. The traffic sensor TS_1 (-3.80905, 43.4722) will be checked using deviation margin D whether it exists or not in the route $R_{1,1} \rightarrow R_{1,2}$. Here, the route start longitude (-3.8085) is equal to route end longitude (-3.8085), so it matches the Part I of Algorithm 1. Within Part I, it does not satisfy any of the condition in Part I(a). For instance, the traffic sensor longitude (-3.80905) is less than route start longitude (-3.8085) and is also less than route end longitude (-3.8085). This condition is not fulfilled which is the preliminary part of both conditions within Part I(a), therefore the status 'isMatched' is still 'false' and so, it goes to Part I(b). Here, the traffic sensor TS_1 longitude (-3.80905) plus deviation margin $D = 0.0006$ (-3.80905 + 0.0006 = -3.80845) is greater than route start longitude (-3.8085), and traffic sensor longitude (-3.80905) minus deviation margin $D = 0.0006$ (-3.80905 − 0.0006 = -3.80965) is less than route end longitude (-3.8085). The traffic sensor TS_1 latitude (43.4722) plus deviation margin $D = 0.0006$ (43.4722 + 0.0006 = 43.4728) is greater than route start latitude (43.4721), and traffic sensor TS_1 latitude (43.4722) minus deviation margin $D = 0.0006$ (43.4722 − 0.0006 = 43.4716) is less than route end latitude (43.4741), therefore it matches the first condition in Part I(b) of Algorithm 1. Hence, the traffic sensor TS_1 is identified within the route $R_{1,1} \rightarrow R_{1,2}$ by using deviation margin D.

In a similar manner, for traffic sensor TS_1 (-3.80905, 43.4722), the green route $R_{2,1} \rightarrow R_{2,2}$ with $R_{2,1}$ (-3.8096, 43.4723) as starting point and $R_{2,2}$ (-3.8075, 43.4737) as ending point, as well as for traffic sensor TS_2 (-3.8099, 43.47335), the red route $R_{3,1} \rightarrow R_{3,2}$ with $R_{3,1}$ (-3.8102, 43.4730) as starting point and $R_{3,2}$ (-3.8071, 43.4730) as ending point, both also fulfill the first condition of Part I(b). For traffic sensor TS_2 (-3.8099, 43.47335), the purple route $R_{4,1} \rightarrow R_{4,2}$ with $R_{4,1}$ (-3.8096, 43.4737) as starting point and $R_{4,2}$ (-3.8075, 43.4723) as ending point, as well as for traffic sensor TS_3 (-3.8080, 43.4739), the blue route $R_{1,2} \rightarrow R_{1,1}$ with $R_{1,2}$ (-3.8085, 43.4741) as starting point and $R_{1,1}$ (-3.8085, 43.4721) as ending point, both fulfill the second condition of Part I(b). Hence, the traffic sensor TS_1, TS_2, TS_2 and TS_3 are identified within the routes $R_{2,1} \rightarrow R_{2,2}$, $R_{3,1} \rightarrow R_{3,2}$, $R_{4,1} \rightarrow R_{4,2}$ and $R_{1,2} \rightarrow R_{1,1}$, respectively, using deviation margin D.

The above routes match Part I of the algorithm, while the remaining routes fulfill Part II of the algorithm which we are going to present next. For traffic sensor TS_3 (-3.8080, 43.4739), the green route $R_{2,2} \rightarrow R_{2,1}$ with $R_{2,2}$ (-3.8075, 43.4737) as starting point and $R_{2,1}$ (-3.8096, 43.4723) as ending point matches Part II of the algorithm because route start longitude (-3.8075) is greater than route end longitude (-3.8096). Within Part II, it does not match any condition within Part II(a), rather it matches the first condition of Part II(b), *i.e.,* traffic sensor TS_3 longitude (-3.8080) minus deviation margin $D = 0.0006$ (-3.8085 − 0.0006 = -3.8086) is less than route start longitude (-3.8075), and traffic sensor TS_3 longitude (-3.8080) plus deviation margin $D = 0.0006$ (-3.8085 + 0.0006 = -3.8074) is greater than route end longitude (-3.8096). The traffic sensor TS_3 latitude (43.4739) minus deviation margin $D = 0.0006$ (43.4739 − 0.0006 = 43.4733) is less than route start latitude (43.4737) and traffic sensor TS_3 latitude (43.4739) plus deviation margin $D = 0.0006$ (43.4739 + 0.0006 = 43.4745) is greater than route end latitude (43.4723). Hence, the traffic sensor TS_3 is identified within the route using the deviation margin D. Finally, for traffic sensor TS_4 (-3.8073, 43.47265), the red route $R_{3,2} \rightarrow R_{3,1}$ with $R_{3,2}$ (-3.8071, 43.4730) as starting point and $R_{3,1}$ (-3.8102, 43.4730) as ending point fulfills the first condition of Part II(b), while for the same traffic sensor TS_4 (-3.8073, 43.47265), the purple route $R_{4,2} \rightarrow R_{4,1}$ with $R_{4,2}$ (-3.8075, 43.4723) as starting point and $R_{4,1}$ (-3.8096, 43.4737) as ending point fulfills the second condition of Part II(b).

In summary, traffic sensors: TS_1 in routes $R_{1,1} \rightarrow R_{1,2}$ and $R_{2,1} \rightarrow R_{2,2}$, TS_2 in routes $R_{3,1} \rightarrow R_{3,2}$ and $R_{4,1} \rightarrow R_{4,2}$, TS_3 in routes $R_{1,2} \rightarrow R_{1,1}$ and $R_{2,2} \rightarrow R_{2,1}$, and TS_4 in routes $R_{3,2} \rightarrow R_{3,1}$ and $R_{4,2} \rightarrow R_{4,1}$ have been successfully detected which complies and verifies the successful operation of Algorithm 1 for identifying deviated traffic sensors on the routes using deviation margin D.

V. Performance Evaluation

In this section, we evaluate the performance of our proposed approach for the mapping of sensors coordinates into route coordinates.

Fig. 5 shows the deployment of traffic sensors at Santander, Spain. For performance evaluation, we randomly selected two locations which serve as starting and destination points. Subsequently, we plotted four possible routes using different colors between these two points by using Brouter offline routing engine [7] (a third-party application) and applied our proposed algorithm to map traffic sensors into these routes which is presented in Fig. 6. This figure provides an overview of mapping of traffic sensor coordinates into the route coordinates. We will provide the detailed analysis of mapping using and without using deviation margin D in this section.

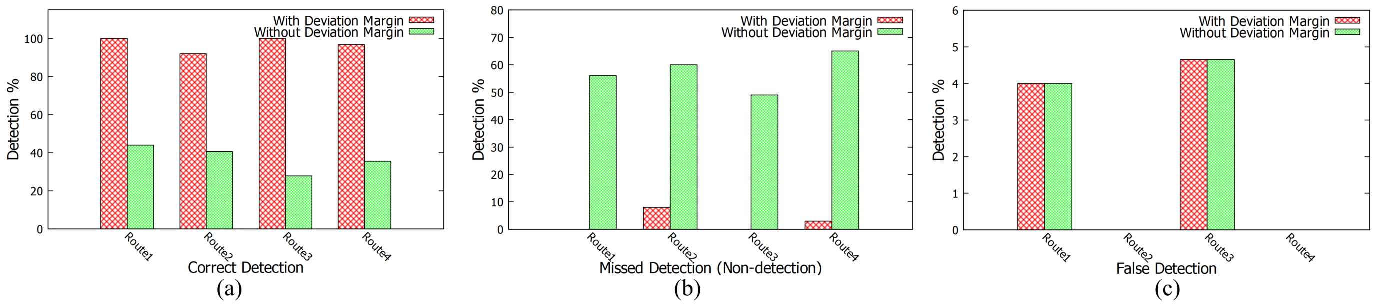

Fig. 4. (a) Percentage of correct detection of sensors on the routes, (b) percentage of missed detection (non-detection) of sensors on the routes and (c) Percentage of false detection of sensors on the routes.

Fig. 5. Deployment of traffic sensors at Santander, Spain.

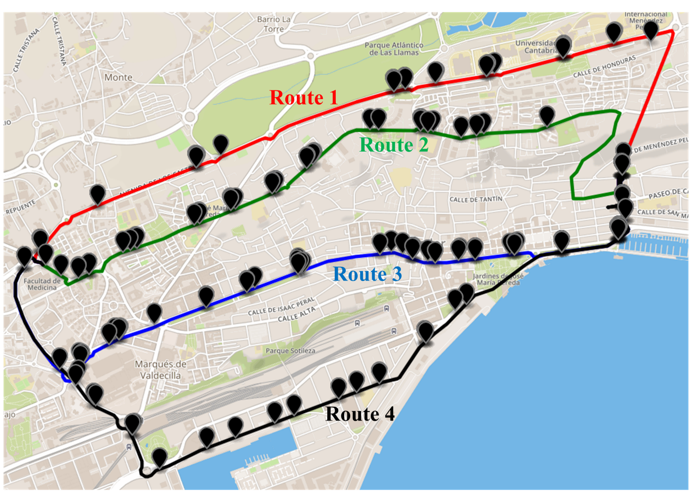

Fig. 6. Mapping of traffic sensors coordinates into routing coordinates.

In our performance evaluation, we use the value of deviation margin $D = 0.00006$ which is selected after doing some experiments. We used Brouter offline routing engine [7]. Brouter is an offline and online routing engine which is built upon Open Street Maps (OSM) [8]. It calculates routes using OSM and elevation data. It is available as offline engine, Android application as well as a web service. Its unique features include freely configurable routing profiles, completely offline operation, advanced routing algorithm with elevation consideration, alternative route calculations, support of nogo and via points, and consideration of long distance cycle routes. We created a script that takes traffic sensors data (including coordinates) deployed at Santander, Spain from NGSI context broker [9] using REST API. These traffic sensors are part of WISE-IoT [10], an H2020 EU-KR project. The script takes the list of route coordinates in JSON format as input which are generated using Brouter routing engine. Finally, the script uses our mapping algorithm (see Algorithm 1) to map traffic sensors' coordinates into the list of route coordinates and provides the output in GeoJSON format that can be directly imported into http://geojson.io to see the mapping into user-friendly visual interface (as can be seen in Fig. 6) and then we analyse the mapping of sensors on the routes manually.

We used three performance metrics for the evaluation.

- *Correct detection* is the percentage of correctly detected traffic sensors on the routes.
- *Missed detection (non-detection)* is the percentage of missed detections (or non-detection) of traffic sensors on the routes, *i.e.,* traffic sensors that exist on the route but are not detected by the algorithm.
- *False detection* is the percentage of false detection of traffic sensors on the route, *i.e.,* the traffic sensors, that lie outside the route.

Fig. 4(a), (b) and (c) present the evaluation results of percentage of correct detection, missed detection and false detection of traffic sensors into the four routes presented in Fig. 6 by using and without using the deviation margin D. Fig. 4(a) shows that by using deviation margin, D, we achieve almost 100% detection of traffic sensors on all the four routes. However, on the other hand, without using deviation margin D, the detection of traffic sensors on the four routes is very low, *i.e.,* even lower than 50%. Similarly, as shown in Fig. 4(b), by using deviation margin D, the percentage of missed detection is very low, *i.e.,* lower than 10%, while without using deviation margin D, the percentage of missed detection is very high, *i.e.,* between 50% to 70%. Finally, as shown in Fig. 4(c), by using deviation margin D, the percentage of missed detection is very low, *i.e.,* lower than 10%, while without using deviation margin D, the percentage of missed detection is very high, *i.e.,* between 50% to 70%. This proves the effectiveness and significant advantage of using deviation margin D for the mapping of sensors coordinates into the routes. In summary, the results related to the correct detection, missed detection and false detection of traffic sensors into route coordinates proves the effectiveness of our proposed algorithm which takes into account the deviation margin D.

VI. Conclusion

We have proposed an approach for the mapping of sensor coordinates into route coordinates by introducing a deviation margin to provide sensors with the flexibility to deviate from the main route. We presented an algorithm along with two illustrative examples that cover all of the scenarios of mapping coordinates. We evaluated the performance of our proposed approach in terms of correct detection, missed detection and false detection. The results prove the efficiency and efficacy of our deviation margin feature. Our proposed approach will certainly be advantageous for new developments in smart cities as they map the deployed sensors along their routes.

VII. Acknowledgment

This research was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 723156, the Swiss State Secretariat for Education, Research and Innovation (SERI) and the South-Korean Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MISP) (No. R7115-16-0002).

References

[1] A. Fernandez-Ares, A. Mora, M. Arenas, P. Garcia-Sanchez, G. Romero, V. Rivas, P. Castillo, and J. Merelo, "Studying Real Traffic and Mobility Scenarios for a Smart City using a New Monitoring and Tracking System," *Future Generation Computer Systems*, vol. 76, pp. 163–179, 2017.

[2] A. Ziat, B. Leroy, N. Baskiotis, and L. Denoyer, "Joint Prediction of Road-Traffic and Parking Occupancy over a City with Representation Learning," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 725–730.

[3] R. Y. Lau, "Toward a Social Sensor Based Framework for Intelligent Transportation," in *IEEE 18th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM*, 2017, pp. 1–6.

[4] C. T. Barba, M. A. Mateos, P. R. Soto, A. M. Mezher, and M. A. Igartua, "Smart City for VANETs using Warning Messages, Traffic Statistics and Intelligent Traffic Lights," in *IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 902–907.

[5] D. C. Shoup, "Cruising for Parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

[6] Smart santander. [Online]. Available: http://www.smartsantander.eu/index.php/testbeds/item/132-santander-summary

[7] Brouter offline routing engine. [Online]. Available: http://brouter.de/brouter/offline.html

[8] Open street maps. [Online]. Available: https://www.openstreetmap.org

[9] Traffic sensors at santander, spain. [Online]. Available: https://mu.tlmat.unican.es:8443/v2/entities?limit=1000&type=TrafficFlowObserved

[10] Worldwide interoperability for semantics iot (wise-iot). [Online]. Available: http://wise-iot.eu/en/home/