ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

# Node Detection and Tracking in Smart Cities Based on Internet of Things and Machine Learning

**Ramyar A. Teimoor[1], Aso Darwesh[2]**

*[1]Department of Computer Science, University of Sulaimani, Sulaymaniyah, Iraq, [2]Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq*

## ABSTRACT

It is essential to know that using technologies in an ethical manner will facilitate human life. Hence, internet of things (IoT) which has been used widely due to developments in information and verbal exchange technologies, with some of its most famous applications are used in the fields of identification, transmission, and health care. In those fields, IoT technologies are used to collect data, recognizing the problem, and proposing a solution for it. Exploiting Global Positioning System (GPS) systems seem to solve the development of node location detection systems. However, concerning a specific target, the accuracy and exactitude of GPS are not enough to satisfy the need for specific location-aware applications. Node detection and tracking system aim is to find an autonomous target using radio frequency identification, IoT tag, GPS location, and k-nearest neighbors (KNN) for detecting the actual zone of the target. We also want to find and predict the best pathway that the node needs to go or want to go through, even to create a system for collecting all the information about nodes and saving it in the cloud and then use it for different purposes. Finding and detection the actual zone of our users are done using KNN algorithm. We use 3NN because that model gets a better result in our dataset, for transmission depending on the users' problem. We are using a new equation to find weights that are integrated with Dijkstra's algorithm. The equation is to calculate the weight between any two nodes using traffic information and image processing for finding a load of the road by counting the number of vehicles inside the image that is collected from our readers. Dijkstra's algorithm is used to find the best path between source and destination using weights between nodes. We tested this algorithm on 11,095 samples on a machine with the following properties; central processing unit (CPU): Core i7 – 4600M at 2.90 GHz (4 CPUs), RAM: 10 GB DDR3, and VGA: Intel HD Graphic 4600–2GB. A set of 7745 nodes used as a training set and 3350 as test sets. The accurate prediction was 3324 nodes, and the false ones were 26 nodes. The accuracy of 3NN algorithm with this configuration is 99.224%, in 11.605 s. After this step, detecting nodes' zone takes only 0.012 s. Using our proposed system, we can detect the actual zone of any user that needs help and track any node that we want. For example, an ambulance in the city and also find the best path for the ambulance or police to travel in from source to destination this idea is used in healthcare but can be used in many other fields such as security, military and information, and communication technologies.

**Index Terms:** Detection, Global Positioning System, Internet of Things, K-Nearest Neighbors, Location, Node Tracking, Radiofrequency Identification

| Access this article online | |
|---|---|
| **DOI:** 10.21928/uhdjst.v3n1y2019.pp30-38 | **E-ISSN:** 2521-4217<br>**P-ISSN:** 2521-4209 |



## 1. INTRODUCTION

Internet of Things (IoT) as a system of reticulate computing devices, mechanical and digital machines, objects, animals or those who have area units with given distinctive identifiers (Unique Identification) and therefore the ability to transfer

**Corresponding author's e-mail:** Ramyar A. Teimoor, Department of Computer Science, University of Sulaimani, Sulaymaniyah, Iraq. E-mail: ramyar.taimur@univsul.edu.iq

Received: 15-04-2019 Accepted: 14-05-2019 Published: 26-05-2019

knowledge over a network while not requiring human-to-human or human-to-computer interaction, become more popular in the future and may control everything [1]. The IoT is recognized collectively of the foremost important areas of future technology and is gaining vast attention from a large varies of industries. The true worth of the IoT for enterprises will be fully accomplished once connected area unit devices are able to communicate with each other [2].

In IoT, we need the THING which is made up of a processor, sensors, and a communication module. This THING when it connects to the internet, it becomes IoT.

In this research, we present a new method for location detection using IoT. The idea is to use radio frequency identification (RFID) tags and machine learning algorithms to find the location of a mobile node and predict the best path to reach the destination.

RFID allows automatic identification and records capture, the use of radio waves, a tag, and a reader, the tag can store further Information than traditional barcodes. The tag incorporates records within the form of the electronic product code. In an international RFID-based object identification machine, three sorts of tags may be used for identification [3].

Passive RFID tags depend on radio frequency strength transferred from the reader to the tag to provide electricity to the tag; now, they may not be battery-powered. This type is used in supply chains, passports, digital tolls, and item-degree tracking [4].

### 1.1. Active Tags

Lively RFID tags have their own battery supply and can instigate communication with a reader. Active Tag can include outside sensors to monitor temperature, stress, chemical compounds, and other situations. Energetic RFID tags are used in manufacturing, health facility laboratories, and faraway-sensing asset management [4].

Semi-passive RFID tags use batteries to energize the microchip while speaking, while drawing power from the reader. Lively and semi-passive RFID tags are more expensive than passive tags [5].

Passive tags are the most widely used, as they are smaller and less high-priced to enforce. Passive tags have to be "powered up" by the RFID reader before then they can transmit information. Unlike passive tags, active RFID tags have an onboard power supply (e.g., a battery), thereby enabling them to communicate information at all times.

If we want to implement the system in a real city, we must use an active tag and a ultrahigh frequency (UHF) RFID reader because the reading distance is defined primarily based on the size and energy of the antenna. We have the diverse frequency ranges which include low frequency, high frequency (HF), and UHF; with their range covering from few centimeters to 100 m [6]. Hence, by the use of those hardware's we can locate and identify our nodes in any road that we want.

Our proposed system is implemented by creating a small model that contains RFID reader, tag, and Arduino to detect and collect information about nodes or tags. We use passive tags, and a HF RFID reader that uses 13.56 MHz with a reading range between 5 cm and 1 m, to transfer information we create a closed network using 2.4G nRF24L01 Wireless Module w/power amplifier (PA) and low noise amplifiers (LNA) module, it uses the 2.4 GHz band and it can operate with baud rates from 250 kbps up to 2 Mbps. If used in open space its range can reach up to 1000 m (Fig. 1).

### 1.2. Features

1. Voltage: 3–3.6 V (recommended 3.3V) V
2. Maximum output power: +20 dBm
3. Emission mode current (peak): 115 mA
4. Receive mode current (peak): 45 mA
5. Power-down mode current: 4.2 uA
6. Sensitivity 2 Mbps mode in received: −92 dBm
7. Sensitivity 1 Mbps mode in received: −95 dBm
8. Sensitivity 250 kbps mode in received: −104 dBm
9. PA gain: 20 Db LNA gain: 10 Db
10. LNA Noise Fig. 2. 6Db
11. Antenna Gain (peak): 2 Dbi
12. 2 MB rate (Open area): 520 m
13. 1 MB rate (Open area): 750 m
14. 250 Kb rate (Open area): 1000 m.

The module can use 125 different channels, which means that it has the possibility to have a network of 125 independently working modems in one place [7]. Each channel can have up to six addresses, or each unit can communicate with up to 6 other units at the same time for both transmission and receiving, so nRF24L01 can send and receive data and communicate with each other while connected in the network.

### 1.3. K-nearest Neighbors (KNN) Algorithm

KNN is a simple supervised learning algorithm [8] that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has already been used in statistical estimation and pattern recognition since the beginning of the 1970s as a non-parametric technique [9].

In KNN a case is classified by a majority vote of its neighbors, with the case being assigned to the class most common among its KNN measured by a distance function. If $K = 3$, then the case is simply assigned to the class of its nearest neighbors.

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{k} \left( xi - yi \right)^2} \quad (1)$$

We use Euclidean distance, and it should also be noted that the Euclidean distance measures are only valid for continuous variables [10].

The value of $K$ is crucial; the large value gives more precision because it reduces the noise even though there is no guarantee. Choosing the optimal value for $K$ is best done by first inspecting the data. Cross-validation is another way to retrospectively determine a good $K$ value using an independent dataset to validate the $K$ value, historically, the optimal $K$ for most datasets has been between 3 and 7 [11]. In our case, we use 3NN because that model gets a better result in our dataset for detecting the actual zone of any user that needs help after a significant test that we have implied on the algorithm.

**Fig. 1.** 2.4 G nRF24L01 wireless module w/power amplifier long-range transceiver.

## 2. APPLICATION DOMAIN

There are varies application domains which can be compacted by the emergence of IoTs. We categorize the applications into few domains, as in healthcare, Information and Communication Technologies, Transportation, Government, Public Safety and Military, smart home and sensible building, mobile technology, and smart business [12].

In this paper, our focus is on using node detection and tracking based on IoT for domains of health care and track monitoring.

### 2.1. Health Care

In some cases, patients do not even have enough time to get to a hospital. One of the most obvious and popular applications of health care is sending an ambulance to transfer the patients to a hospital for treatment, so what is the best process for find the patient's location at the briefest time and how we can track an ambulance in a city [13]. The idea is to find the patient's location and guide the ambulance to take the shortest and/or fastest way.

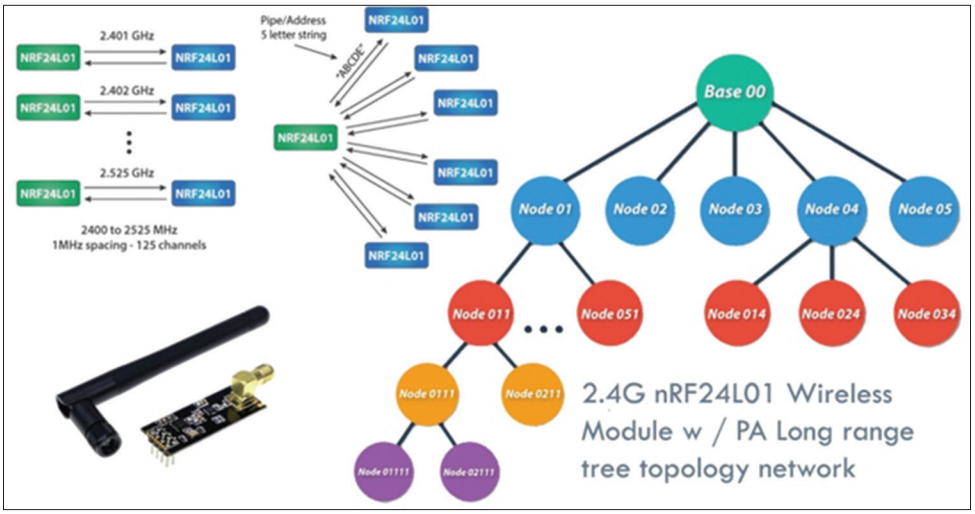


**Fig. 2.** nrf24l01 tree topology network [7].

### 2.2. Tracking and Monitoring

Real-time monitoring through connected devices can save lives, with real-time monitoring of the condition in place by means of a smart device connected to a system or smartphone app; connected devices can collect information and other required data and use a communication technique to transfer the collected information to a data center [14].

The IoT device in an ambulance can collect and transfer data: Like blood pressure, oxygen and blood sugar levels, weight, and electrocardiograms [15].

### 2.3. End-to-end Connectivity and Affordability

IoT enables interoperability, machine-to-machine communication, information exchange, and data movement that make service delivery effective [16].

### 2.4. Data Assortment and Analysis

Vast amount of data that an IoT device sends in a very short time due to their real-time application is hard to store and manage if the access to the system is unavailable. Even for field providers to acquire data originating from multiple devices and sources and analyze it manually is a tough bet [17].

## 3. PROBLEM STATEMENT AND AIM OF USING NODE DETECTION AND TRACKING

Node tracking is used in many fields such as transportation [18] and military security [19]. Among the techniques used in this field, we focused on IoT using RFID technology for its simplicity and its performance in this field.

In this research, we use IoT and machine learning algorithms to detect nodes in a closed network. Then, find the actual zone of any node in a smart city.

After that, we suggest the best pathway to any destination depending on distance, speed, time, and traffic. Finally, we predict the path of any IoT node using machine learning techniques.

## 4. METHODOLOGY

At the beginning, we should determine the location; in our case, its Iraq-Sulaymaniyah Governorate. Then, data collection about this area must be collected (Fig. 3).

We divided this city into 42 zones (Fig. 4) depending on the police stations, classified areas, main roads, and hospitals. Then, we group zones and decrease them into 27 zones (Fig. 5). To simplify and improve our dataset.

At this step, we work only on 27 zones, due to the mentioned reason above, for each zone, we collect 100–500 samples depending on the size of the area. The idea of choosing this region is for it has most of the hospitals and police stations inside it, availability of more accurate data comparing to other zones, and it is the most active part of Sulaymaniyah in terms of traffic and connected roads (Fig. 6).

After determining our zones, for each sample in any zone, we get longitude, latitude, zone code, nearest hospital, and police station that is responsible for that zone and the nearest Fire Stations is an important attribute we collect for each zone, so for training set and testing set we need all attributes mentioned above. We use Google earth for samples collection. Our system implemented by java and some data structure algorithm is used for getting better performance.

### 4.1. First Step

First Step must provide the dataset to the system for training and testing to get a result and finding the actual zone of each node that the system can recognize.

### 4.2. Second Step

Number of instances in our dataset is 11,095 samples for 27 zones, we use 70% of our samples for training and 30% for testing (Fig. 7) and depending on these samples and using machine learning our system can find the actual zone for each of the users because Sulaymaniyah Governorate at least has 10 billion coordinates which is a large number amount of data. We take only 11,095 coordinates, and we can give the same results. We use two cross-validated methods for sampling, Hold-out and bootstrap sampling, but Hold-out has better accuracy than bootstrap in our work.

### 4.3. Third Step

We use KNN to find actual zone (Euclidian distance) is used to calculate the distance between two points.

### 4.4. Fourth Step

After building a model, we must test it in the real world, for that reason using two approaches we can test our system, one of them is IoT tag and RFID reader by detecting node and sending tag information and their location to our system for finding the zone and other attributes. Or for the second mechanism the user send information to the system through a mobile application using SMS that contain critical information for finding that the user and their information are valid to


**Fig. 3.** Iraq-Sulaymaniyah governorate.

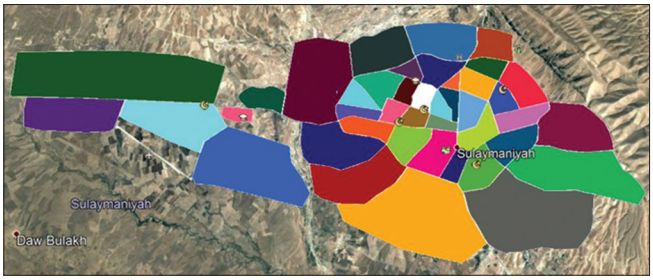


**Fig. 4.** Sulaymaniyah 42 zones.

**Fig. 5.** Case study inside city.

our system or not, the information that the user send contain longitude, latitude, phone number, and type of the help that s/he needs like the need for an ambulance or police or Firefighter. After determining the problem, we go to the next step.

### 4.5. Fifth Step

For sending an ambulance or police or Firefighter to the user, our system must find the best and shortest path to those users that need help. we create a closed network node detection in our city, so using the user information system the algorithm can detect the source and the destination targets, then calculate all possible ways from source to destination depending on these factors: Distance, speed, Traffic in that path and roads load because we already know that in some roads may have a lot of traffic in some periods such as in the morning and evening (start and end of a workday).

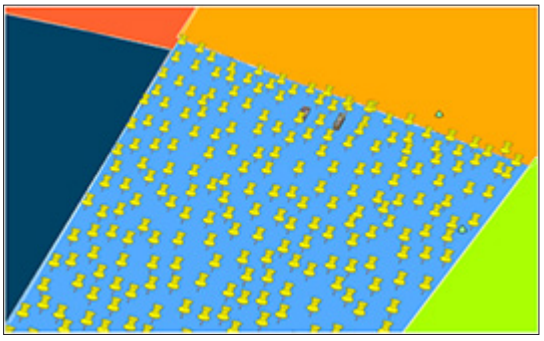

**Fig. 6.** Training sample.

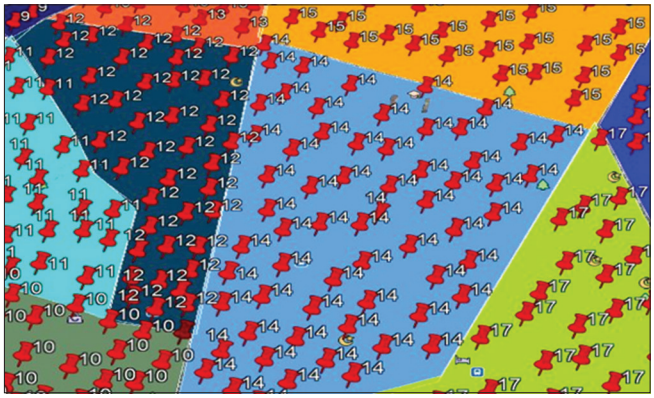

**Fig. 7.** Testing sample.

In our proposed system, we created a new algorithm for finding weights between nodes, and we can say that it is an improvement in finding weights between two readers or nodes in smart cities. Using (equation no. 2), we can calculate and find weights between any two nodes depending on (distance, speed, traffic, and some other parameters) and give that weight to Dijkstra's algorithm [20] to find the best path between source and destination.

$$T = \frac{D}{S} + \left(Tr * Te * N * Z\right) + C \qquad (2)$$

Where:

$T$ = Time $_{\text{weight in (s)}}$

$D$ = Distance $_{\text{ab (m)}}$

$S$ = Speed $_{\text{ab (m/s)}}$

$T_r$ = Traffic $\begin{cases} 0 \\ 1 \end{cases}$

$T_e$ = Traffic time

$N$ = No. side

$C$ = Car count in image

$T_c$ = Traveled car in Traffic ~ 30

$$Z = \frac{C}{Tc} \quad \begin{cases} 1 \,; 0 \le Z > 1 \\ \left[\frac{C}{Tc}\right] \ge 1 \,; floor\ function \end{cases}$$

In the algorithm above, we have ($C$ = Car count in image), when it comes to deep learning-based object detection algorithm, there are three primary object detectors you'll encounter:

- Regions-convolutional neural network (R-CNN) and their variants, including the original R-CNN, Fast R- CNN, and Faster R-CNN
- Single Shot Detector
- YOLO.

We use YOLO v3 [21] in our algorithm, in particular, YOLO trained on the Common Objects in Context (COCO) dataset [22]. The COCO dataset consists of 80 labels we use only those that are related to vehicles.

We are implementing YOLO v3 by two powerful languages python and java in our algorithm we created a specific java file that can send parameters and run the python file and return the number of detected vehicles inside the images and send it to our algorithm (Fig. 8).

Number of images is equal to the number of RFID readers in that path; this process is repeated every 1 min because we need updated data from an updated image from any node.

We already know that the problem of image processing is memory consumption, we separated the work into two parts; the process of counting the number of vehicles in an image is separated from another process such as finding the actual zone and detecting the best path each of them run on a separate machine. A web service is used to communicate between them.

### 4.6. The Dijkstra's Algorithm [20]

#### *4.6.1. Step 1: Initialization*

- Assign the zero weights value to node s, and label it as permanent (The state of node $s$ is [0, $P$]).
- Assign to every node a weights value of ∞ and label them as temporary (The state of every other node is [∞, $t$]).
- Designate the node $s$ as the current node.

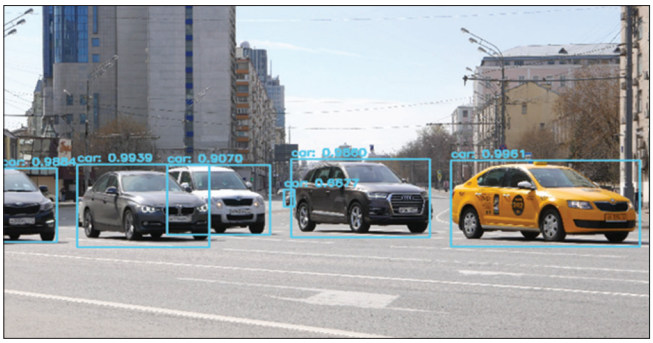


**Fig. 8.** Vehicle detection in image using YOLO [21].

#### *4.6.2. Step 2: Weights value update and current node designation update let i be the index of the current node*

1. Find the set J of nodes with temporary labels that can be reached from the current node i by a link (i, j). Update the weights values of these nodes. For each $j \in J$, the weights value $d_j$ of node j is updated as follows new $d_j = \min\{d_j, d_i + c_{ij}\}$ where $c_{ij}$ is the cost of link (i, j), as given in the network problem.
2. Determine a node j that has the smallest weights value dj among all nodes $j \in J$, find j* such that min $j \in J$ $d_j = d_{j*}$
3. Change the label of node j* to permanent and designate this node as the current node.

#### *4.6.3. Step 3: Termination criterion*

If all nodes that can be reached from node s have been permanently labeled, then stop – we are done. If we cannot reach any temporary labeled node from the current node, then all the temporary labels become permanent – we are done. Otherwise, go to Step 2 [7].

Using the above algorithm, the system can find the best path from source to destination.

## 5. RESULTS

We use 3NN because it gives better results, as explained in Table 1. We divide our 11,095 samples into two-part training and testing by two famous sampling techniques in cross-validate sampling hold-out and bootstrap in the table below you can see the result of accuracy to find the actual zone of detected node.

In each sample, we have a lot of attributes, but the important ones are longitude and latitude. Using KNN we calculate and find the 3, 5, and 7 nearest neighbors of each sample after that we find the means absolute error for all tests, as shown

**TABLE 1: KNN results for both hold-out and Bootstrap**

| Result: 70% of samples for trine and 30% for test | | | |
|---|---|---|---|
| **Type of sampling** | **3 NN accuracy** | **5 NN accuracy** | **7 NN accuracy** |
| Hold-out | 99.22388% | 99.07462% | 98.98507% |
| Bootstrap | 98.70793% | 98.73798% | 98.97836% |

KNN: K-nearest neighbors

in Table 1 we have best accuracy (99.22388%) in hold-out at 3NN, the mean absolute error is 0.00776 which shows that our dataset and our implemented algorithm have a good result.

When a node has been detected, we calculate the zone based on their nearest neighbors as described in the third step. After that, we determine the destination (fourth step) then suggest the best path (Fig. 9). We describe the closed network area inside our city, the entire RFID reader's node connected to each other. Each reader has a specific number, location, pathway that is connect to other nodes that node has traffic or not, nearest (hospital, police stations, and Firefighter), so if any reader detects the specific node, alert the system by sending some packet through our Wi-Fi network for notifying it that the node is here now, and system must choose the best path for it to travel to its destination in the first example the result is the best case because we do not have any traffic (Fig. 9).

In the second example, we have traffic between those two nodes (Fig. 10).

At any point in the city, we can collect new information and set new order when the node is recognized by a RFID reader so the system can detect a new path if eventually, some accident happens such as traffic accident or cartography like fire at some building. Hence, the system finds the best path depending on all attributes that we have mentioned before.

Hence, our work can be updated at any time so that the reader detects the tag and calculates the new information for tracking and detecting the path that the node must travel.

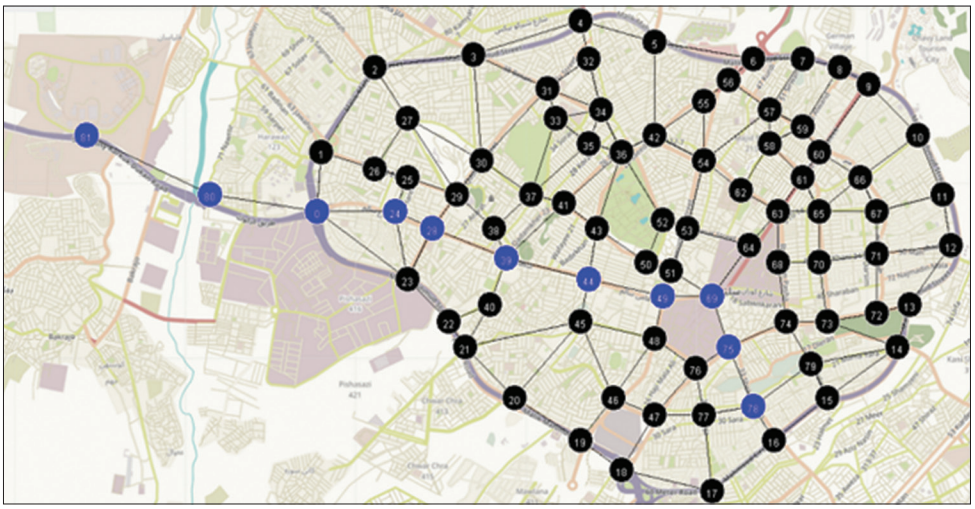

**Fig. 9.** Best case path detection.

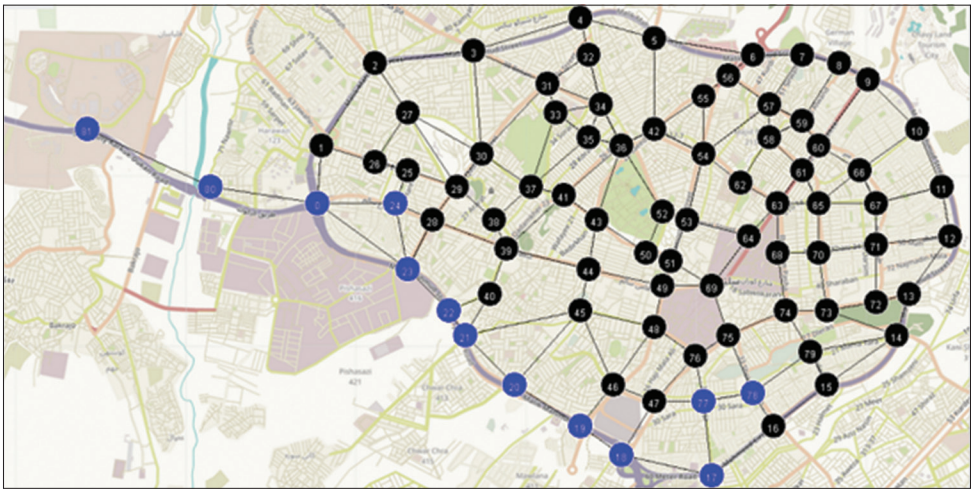

**Fig. 10.** Path detection depending on all attributes.

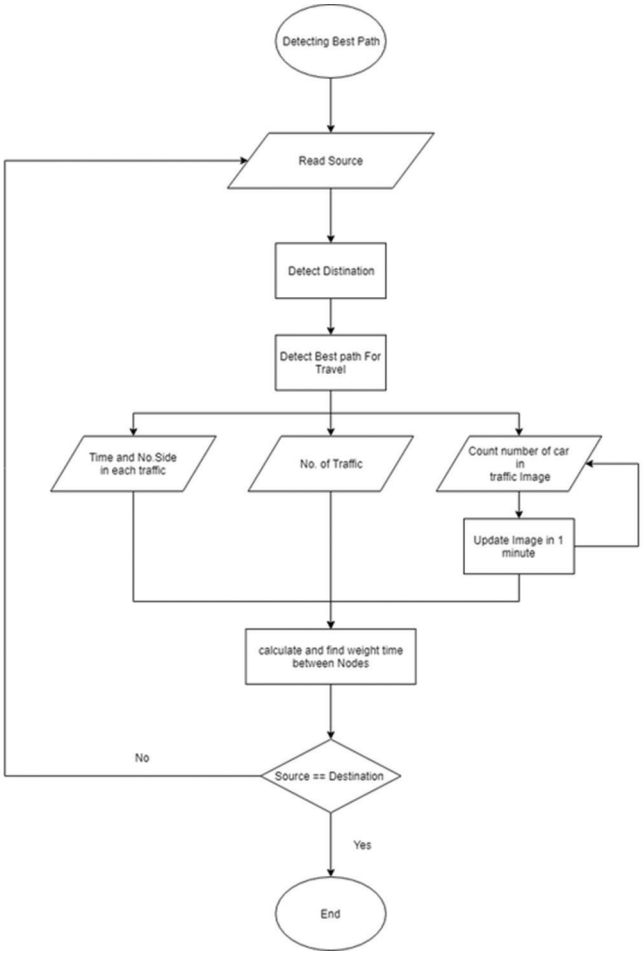


**Fig. 11.** Path detection flowchart.

Seeing the flowchart above (Fig. 11) describes all steps from source to destination for finding the best path. At any reader that detects a specified node, those steps above are done and send new instruction to the node. Like ambulance, police, and firefighters to travel and also they can send the notification before each interconnection of roads and ask the system to determine the path for it and send an instruction to them.

All the detection and information collected from readers and the system are automatically saved in an encrypted manner in a database on the server.

## 6. CONCLUSION

The character of IOT has changed the entire world. This has been finished through the help of introducing IoT devices, smartphones, and character automated collaborators within the health area, security, learning, traveling, and transmission. The framework we propose is fit for following self-governing node inside a closed network area depends on RFID innovation. In the end, using this system, we can detect any node at any position in our city and find the actual zone for it and choose the best path from the source node to the destination node depending on the type for transmission, for example, if the problem is to provide health care, the system detects the best path from the source to the nearest hospital or if the problem is to provide police services, detect the best path to the nearest police or a fire station. It calls the nearest firefighters station. We have tested our algorithms on a prototype implemented on a surface of $1 \times 1$ m$^2$; the results obtained are excellent. Our goal is to implement the proposed system in a real city and validate the algorithm on real data. Using this system and dataset can help other researchers work in other areas that can help cities use technologies to improve their lives.

## REFERENCES


1. D. Wu and P. Lorenz. "The internet of things for smart cities: Technologies and applications". *IEEE Network*, vol. 33, pp. 4-5, 2019.
2. I. Lee and K. Lee. "The internet of things (IoT): Applications, investments, and challenges for enterprises". *Business Horizons*, vol. 58, no. 4, pp. 431-440, 2015.
3. F. Wang and T. Su. "Indoor tracking by rfid fusion with IMU data". *Asian Journal of Control*, vol. 21, no. 2, pp. 1-10, 2019.
4. A. A. R. Green. "*RFID Pet Monitoring and Identification System with RFID Tag Access Hatch*". Universiry of West London, London, p. 21389570, 2019.
5. R. Tesoriero, J. A. Gallud, M. D. Lozano and V. M. R. Penichet. "Tracking autonomous entities using RFID technology". *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 650-655, 2009.
6. "*Types of Rfid Systems*", 2018. Available from: https://www.impinj.com/about-rfid/types-of-rfid-systems. [Last accessed on 2019 Feb 20].
7. Dejan. *"How To Build an Arduino Wireless Network with Multiple NRF24L01 Modules How to mechatronics*", 2019. Available from: https://www.howtomechatronics.com/tutorials/arduino/how-to-build-an-arduino-wireless-network-with-multiple-nrf24l01-modules. [Last accessed on 2019 Mar 10].
8. R. Agrawal. *Integrated Parallel K-Nearest Neighbor Algorithm*. Springer, Singapore.
9. M. A. Jabbar, B. L. Deekshatulu, and P. Chandra. "Classification of Heart Disease Using K Nearest Neighbor and Genetic Algorithm". *Procedia Technology*, vol. 10, pp. 85-94, 2013.
10. S. Buczkowska, N. Coulombel and M. De Lapparent. *A comparison of Euclidean Distance, Travel Times, and Network Distances in Location Choice Mixture Models*. Networks and Spatial Economics, Berlin, 2019.
11. R. Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". *Proceeding IJCAI'95 Proceedings of the 14th International Joint Conference on Artificial Intelligence*. vol. 2, pp. 1137-1143, 2013.

12. M. H. Asghar, A. Negi and N. Mohammadzadeh. “Principle application and vision in Internet of Things (IoT)”. *International Conference on Computing, Communication and Automation*, pp. 427-431, 2015.
13. A. Sharma and R. Kumar. “Service-level agreement energy cooperative quickest ambulance routing for critical healthcare services”. *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3831-3848, 2019.
14. R. Pi and K. Zeb. “Healthcare Monitoring System and Transforming Monitored Data Into Real Time Clinical Feedback Based on Iot Using”. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1-6, 2019.
15. A. Subasi, M. Radhwan, R. Kurdi and K. Khateeb. “IoT Based Mobile Healthcare System for Human Activity Recognition”. *2018 15th Learning and Technology Conference*, pp. 29-34, 2018.
16. B. Y. J. A. Mccann, G. P. Picco, A. Gluhak, and K. H. Johansson. “Connected things connecting Europe”. *Communications of the ACM*, vol. 62, no. 4, p. 46.
17. M. A. V. Paul, T. A. Sagar, S. Venkatesan and A. K. Gupta. *Impact of Mobility in IoT Devices for Healthcare*. Springer, Cham, 2018.
18. N. C. Soe, T. Lai, L. Thein and T. Aung. “GPS Tracking and Traffic Monitoring System in Urban Transportation”. *2018 IEEE 7th Global Conference on Consumer Electronics*, vol. 29, no. 4, pp. 804-805, 2018.
19. B. Iyer. “IoT enabled tracking and monitoring sensor for military applications”. *International Journal of Systems Assurance Engineering and Management*, vol. 9, no. 6, pp, 1294-1301, 2018.
20. R. T. Michael and T. Goodrich. “*Algorithm Design: Foundation, Analysis and Internet Example*”. John Wiley, New York, pp. 341-345, 2006.
21. J. Redmon. “*YOLOv3: An Incremental Improvement*”. Cornell University, New York, 2018.
22. J. Redmon, S. Divvala, R. Girshick and A. Farhadi. “*You Only Look Once: Unified, Real-Time Object Detection*”. Cornell University, New York, 2016.