Smart System for Emergency Traffic Recommendations: Urban Ambulance Mobility

Ayoub Charef[1], Zahi Jarir[2], Mohamed Quafafou[3]
Department of Computer Science, Faculty of Science, Cadi Ayyad University, Marrakech, Morocco[1, 2]
Department of Computer Science, University of Aix-Marseille, Marseille, France[3]

***Abstract*—With the increasing evolution of advanced technologies and techniques such as the Internet of Things, Artificial Intelligence and Big Data, the traffic management systems industry has acquired new methodologies for creating advanced and intelligent services and applications for traffic management and safety. The current contribution focuses on the implementation of a path recommendation service for paramedics in emergency situations, which is one of the most critical and complex issues in traffic management for the survival of individuals involved in emergency incidents. This work mainly focused on the response time to life-threatening incidents, which is an indicator for emergency ambulance services and for recommending a fastest ambulance route. To this end, we propose a hybrid approach consisting on a local approach using machine learning techniques to predict the congestion of different sections of a map from an origin to a destination, and a global approach to suggest the fastest path to ambulance drivers in real time as they move in OpenStreetMap.**

Keywords—Recommendation systems; emergency urban traffic; ambulance mobility; emergency navigation services

I. Introduction and Motivation

Nowadays traffic congestion represents a real major issue for all societies and also an inconvenience for the townspeople because of the increase of population and vehicles in an urban zone. The level of these challenges encountered varies from one region to another depending on road infrastructure resources, road safety education, people's behavior [1], etc. Traffic blockage becomes one of the major and most difficult problems in various urban areas according to the negative effects it induces to the various actors of the road: road users, pedestrians, police officers, which in turn result in various social and economic problems according to [2]. Among these effects and by the observation that the frequency of road accidents and their effects are increasing. more and more every day [3]. Therefore, most of the ministries of equipment and transport over the world have invested in road infrastructures by installing the latest generation technologies and solutions to be able to reduce the level of stress and save the lives of citizens. Moreover, the emergency rescue research is receiving more and more attention. The most important factor in an emergency rescue is rescue timeliness, and often makes a difference between life and death [4].

However, finding an effective response to this challenge is considered as complex and tedious research topic in the literature. Despite the advances and published results of scientific contributions on this topic, this challenge remains open. This complexity is compounded when other constraints or factors must be met, such as the inability to create emergency lanes due to narrow roads, undeveloped urban road infrastructure, etc.

In this paper the proposed approach consists of offering an emergency mobility service with the advantage of continuously recommending to ambulance drivers with the fastest way to reach the destination. The data used comes from road infrastructure sensors, Internet of Things installed in roads, etc. as well as data from previously traveled routes, such as incidents, line changes, travel time, traffic flow, etc. All these data enable us to calculate the speed interval and travel duration for every route taken by an ambulance in response to an emergency. In this contribution, this paper mainly focuses on how to reduce ambulance travel time by addressing the challenges of an efficient ambulance response. To achieve this goal, the proposed approach is twofold: 1) to analyze the mobility of emergency ambulances and identify the main factors preventing their travel, and 2) to model the fastest travel recommendation solution to improve ambulance mobility time by focusing on data collection from roadway infrastructure sensors, archived data on confirmed past travel history.

The remainder of the paper is organized as follows. Section II exposes a related work on path recommendation solutions for Emergency Medical Transportations (EMTs) in emergency states presented in the literature. Section III details the constraints to be considered when proposing an emergency path recommendation solution. Section IV describes the multilayer architecture of the proposed recommendation solution. Section V explains the multi-agents modeling approach, while data and random traffic generation used for an experimentation is exposed in section VI. Section VII presents details of the proposed hybrid approach for predicting the traffic state based on machine learning techniques, and proposing the fastest path to a destination in real-time using Dijkstra algorithm. Finally, Section VIII concludes the paper and presents some future works.

This section starts by exposing some interesting finding published in literature concerning the prediction of travel path for emergency vehicles (EV). Then it focuses on some important machine learning algorithms predicting urban traffic congestion.

II. Literature Review

A. Trajectory Prediction for Emergency Vehicles

Traffic congestion contributes greatly either indirectly or directly to both economic and health losses in countries.

According to [5], other repercussions of traffic jams include pronged time loss, especially during peak hours, mental stress, traffic accidents, air pollution, noise, and disturbance [6]. Therefore, it is difficult for emergency vehicle services to achieve short response times during interventions without using effective advanced solutions, techniques and technologies. In the literature, several approaches have been proposed to reduce ambulance response time. This delay is mainly due to several traffic lights management, uncontrolled intersections, that creates a crowd of vehicles and thus puts ambulances on hold.

The work presented in [7] suggests a solution to make on the traffic lights for ambulance from the start position to the scene of an incident or where a patient is. However, the proposed solution is very restrictive since signalized intersections could present colossal travel delays. So the most generally taken on procedure for EV prioritization is traffic light need control. In the same context, authors in [8] propose a method of controlling traffic lights and performing the aforementioned task to allow the ambulance to cross all intersections without waiting. However, road signs remain an uncontrollable problem.

Authors in [9] propose an Android application with an Internet of Things (IoT) network model for an EV routing using Fuzzy logic-based data fusion. The data fusion approach calculates the exact congestion for a certain place by taking into account crowd inputs and sensory data. However, the suggested system requires the crowdsourced data to be assessed for trust in order to confirm the identity of the crowdsourced user. Moreover, strong communication protocols are required when using IoT in order to reduce the danger of security threats.

In other work, [10] uses radiofrequency sensors to count vehicles, which are equipped with radio frequency identification tags, to propose a dynamic traffic signal as a solution to avoid ambulance blockage. However, the proposed solution is also very restrictive since vehicles should be equipped with radio frequency identification tags.

The work in [11] proposes a unique decentralized approach to geofencing based on radio frequency and global navigation satellite system that allows emergency service vehicles to pass through intersections with the least amount of delays by giving them the right of way green signal. In the same context [12] proposed a cooperative vehicle-road scheduling method that includes a real-time route planning module and a group traffic signal management module.

The approach presented in [13] provides green indicates to help EV to pass through intersections quickly without stopping, and restores the road network to the normal situation as soon as possible by using linear programming to find the shortest green time in each phase after an EV passes the intersection. However, the proposed contribution does not take into account the dynamics the global traffic state and the impact of a scheduling strategy on the ordinary vehicles.

The authors in [14] propose an algorithm for finding the shortest or fastest path. This algorithm based on Ant Colony Optimization with Fuzzy Logic to deal with local path planning for obstacle avoidance by taking into account wind, flow air, and dynamic obstacles. However, the weakness of this approach is that the response time of the algorithm is very slow.

In the same context, [15] and [16] proposed an ASNN-FRR (A traffic-aware neural network for fastest route recommendation) model suggesting a fastest route recommendation. This model uses two powerful predictors for g(·) and h(·) functions of A* algorithm respectively, to find fastest path between two points origin-destination (OD) based on travel time estimation. However, the trajectory recommendation is given initially, the problem here is that the authors do not take into account that urban traffic changes over time.

By collecting and analyzing social media data and using new technologies of granular computing that transforms big data collections into granular information, [17] proposes a fastest path optimization model to incorporate the impact of traffic events and generate the optimal routing strategy. However, the obtained results show that the strategy is good for public transport and not for the emergency vehicles.

To help EVs such as ambulance, fire engine, and police vehicles to greatly benefit from accurate path prediction systems, [18] and [19] suggested a ventilation network model to automatically assess and determine potential escape routes. A variant of Dijkstra's algorithm is used to forecast intricate pathways between any two sites across a model. However, the proposed model uses data from the begin and not take in consideration frequent traffic changes.

The proposed approach in [4] offers a data-driven system for precisely predicting the path taken by an ambulance moving with blue lights and sirens in response to an emergency situation. This technique calculates the typical response time of an ambulance to an emergency call using historical data, and then estimates the travel time from these speeds for each route. Finally, it finds the quickest path between any beginning and finishing points using a common graph-theoretic approach based on Hidden Markov model. However, historical data can only be used to make predictions; in contrast, emergency situations need using real-time data.

The work of [20] uses the technique of multiple classifiers with maximum vote to predict the route to the destination with the highest degree of accuracy and select the best path to reach the destination as quickly as possible. Three algorithms—the artificial neural network (ANN), the k-nearest neighbors (KNN), and the support vector machine (SVM) are the foundation of this system. As classifiers, which can't foresee the quickest path between two points. By and by, the real travel season of a similar route can be exceptionally irregular at various times or simultaneously of day on various days, because of the blockages from vehicles.

B. *Traffic Congestion Prediction*

Several interesting contributions focus on traffic congestion prediction problems to avoid traffic jams and minimize travel time. The objective of this subsection is to present some of them in order to highlight the most used techniques.

Table I shows the comparison of Proposed Approach with Existing works in Literature.

TABLE I. COMPARISON OF PROPOSED APPROACH WITH EXISTING MODELS

Reference	Methodology	Prediction studies	Target domain	Prediction by area	Prediction on intersection	Runtime prediction for EVs	Historical Data /Runtime Data
[20]	ANN And KNN	Machine learning	predict congestion	✓	✗	✓	✓
[21]	Genetic Algorithms	Probabilistic reasoning	analayse short-term congestion.	✓	✗	✗	✓
[22]	Genetic Algorithms	Probabilistic reasoning	predict congestion at desired point.	✗	✓	✗	✗
[24]	particle swarm optimization	Probabilistic reasoning	predict the urban traffic congestion	✓	✗	✗	✗
[25]	KNN, SVM, Hybrid Model	Machine learning	short-term volume prediction	✗	✗	✓	✓
[26]	Extended Kalman filter	Machine learning	Traffic Congestion Estimation	✗	✗	✗	✗
[27]	neural network	Machine learning	predict congestion	✓	✗	✗	✗
[28]	Bayesian network	Probabilistic reasoning	congestion probability	✗	✗	✗	✗
[29]	deep learning	Machine learning	traffic flow	✓	✗	✗	✗
[33]	fuzzy logic inference	Probabilistic reasoning	Traffic Flow Detection	✗	✗	✓	✓
[34]	Evolutionary Crisp Rule Learning	Probabilistic reasoning	predict congestion level	✗	✗	✓	✗
[36]	Fuzzy logic evaluation	Probabilistic reasoning	Traffic pattern determination	✗	✗	✗	✗
[37]	SRHTCP (fuzzy logic)	Machine learning	predict congestion level	✗	✗	✗	✓
[38]	Artificial neural network	Machine learning	T. Congestion state	✗	✗	✓	✓
[39]	Regression model	Machine learning	Traffic congestion score	✗	✗	✗	✗
The Proposed Solution	Sliding Window	Machine learning	predict traffic congestion level	✓	✓	✓	✓

The approach suggested in [21] that the combination of a Genetic Algorithm and the Cross Entropy method for the optimization of a Parallel Hierarchical Fuzzy Rule-Based System would help to develop a synergy between exploration and exploitation to improve the system's parameters.

The authors in [22] designed an approach based on hierarchical fuzzy rule based system and genetic algorithms to build traffic congestion prediction systems from a large number of input data. However, this approach does not take into consideration many parameters that will have to be obtained in real time.

In the same context [23] and [24] estimate and predict traffic congestion in large-scale urban regions using a fuzzy comprehensive evaluation method. This method involved mapping the floating vehicle in sample points using data from its route. However, the authors focus on the traffic prediction on the roads and don't take in account the intersections where the congestion is more interesting to manage.

In the paper [25], a hybrid model of SVM and KNN is proposed as a short-term prediction technique for highway exiting traffic. However, the problem is that the traffic data are not strictly periodic. For example, in the normal day the peak hours could vary from 4:30pm to 6:00pm, but on weekdays usually happen in afternoon. In the same context, the work presented in [26] consists on creating a multimodal Big data fusion framework for traffic congestion prediction. This work uses a highway traffic dataset to predict traffic travel times using data mining predictive and Extended Kalman filters. However, the weakness of the approach lies in the non-use in the traffic data used of certain important qualified parameters such as time and area.

The works proposed in both [27] and [28] are based on the Bayesian Network analysis approach to model the probabilistic dependency structure between causes of congestion on a particular road segment and analyzing the probability of traffic congestion given various roadway condition scenarios. However, the obtained results show that the both approaches take a lot of time for making prediction, and thereafter cannot be considered as real time solution.

In [29], offers a fusion deep learning model that takes spatial-temporal correlation into account in order to address the issue of predicting urban road traffic flow and increase prediction accuracy.

Referring to the literature, this work can summarize that the challenge of emergency traffic recommender systems is still an open problem and the proposed approaches are not complete to be used in such situations. Firstly, the majority of these evoked contributions attempt to forecast traffic flows for a specific area of the urban road network and are mainly based on the variations of the time series specific to this area. However, they must take into account the entire region, including origin and destination, as a whole. Also, the prediction response must be provided at runtime and continuously. In addition, most traffic jam prediction are for road segments and not for intersections, despite the fact that most of the time traffic jams start from intersections.

This motivates us to propose an emergency traffic recommender system combining optimal path problems and real-time congestion prediction to find the fastest route from an origin to a destination. This system is based on a flexible architecture having the advantage to collect useful information gathered from road infrastructure, traffic tracking, etc., and to process and provide real-time congestion prediction using machine learning algorithms and in particular classification algorithms. The main reason for choosing classification algorithms is that their processing time is lower than deep learning algorithms according to [30], and they also meet the goal of reaching a fastest path.

III. Research Methodology

The objective of the current contribution is to propose an emergency traffic recommendation solution having the advantage of proposing fastest routes from an origin to a destination while avoiding as much as possible that ambulances get stuck in traffic jams during their travels. The implementation of such an emergency traffic recommendation system requires massive empirical data to build the most appropriate and fastest path for ambulance drivers. Therefore, this paper proposes a flexible architecture that has the advantage of collecting all the required data from the environment and also of processing them in real time to respond to the travel emergency.

The proposed architecture uses techniques to collect, filter, store and process the data in real-time, and machine learning algorithms to analyze and predict the fastest path in real time throughout the mobility.

To build this architecture, we started by understanding, identifying, and analyzing all parameters involved in the emergency traffic recommendation process before the modeling phase. The necessary parameters we have identified mainly belong to two main categories: static and dynamic parameters. Static parameters refer to data that do not change during the period of travel, such as infrastructure data, traffic signals, road structures, red light scheduling, etc. While the dynamic category focuses on variables whose values change over time, such as traffic trips.

Traffic congestion prediction has two essential steps of forecast model development and data collection. Each step of the process is important and, if performed incorrectly, could have an impact on the results or consequences of the next steps. After data collection, data processing assumes an imperative part to prepare the training and testing datasets, and help to predict the traffic flow as well as an estimated traffic congestion state. In addition, proposed traffic datasets are divided into two categories: stationary data and probe data. Additional categories for fixed information includes sensor and fixed cameras. Probe data includes data collected from GPS or applications installed on vehicles, traffic parameters we have identified mainly belong to two main categories: static and dynamic parameters. Static parameters refer to data that do not change during the period of travel, such as infrastructure data, traffic signals, road structures, red light scheduling, etc. While the dynamic category focuses on variables whose values change over time, such as traffic trips, data collected by IoT connected objects, or sensors, driver behaviors, traffic flow, etc. Fig. 1 summarizes the components and parameters taken into account by the proposed architecture.

In order to propose a flexible model for emergency traffic services for the case of ambulance mobility, this paper was interested in the use of multi-agent system (MAS). That is more adapted to this case and meets our needs perfectly. Moreover, it has the ability to hold the properties of distributed systems, to meet the required coordination which consists in organizing the cooperation between the connected objects by sharing knowledge and relying on the capabilities and knowledge of each entity, and to adapt to artificial intelligence techniques. For all these reasons, in this paper, we have opted for SARL (Multi-agent programming language of system) as an agent programming language because of its robustness, the multitude of features offered, etc. the proposed architecture is shown in Fig. 2.

The Fig. 3 represent the main layers of the emergency traffic recommender system concern the multi-agent system module, a data management module represented by a NoSQL solution (mongoDB in this work), an extraction and decision module for traffic management, and a mobility services module that represents the interface of the recommendation system offering to the ambulance drivers real-time positioning information, the fastest path between a departure point and destination, etc. This last module interacts with the multi-agent system module, which through the agents, allows to manage each physical object (sensors, IoT, etc.). These agents are permanently connected to the road infrastructure. Based on the information collected from the physical infrastructure, this module is able to compose the required event or task using a coordination process and send it to the navigation services module.

The Traffic Management Extraction and Decision module provides the overall decision on the appropriate measure of mobility, fluidity in the segments and nodes to traverse to reach a destination from an origin. This decision is based on historical urban traffic states, data collected in real time by the multi-agent system. To implement the proposed solution, we propose a hybrid approach: 1) a local approach dedicated to traffic prediction based on a learning algorithm and 2) a global approach to propose the fastest path from an origin to a destination based on the output of the local approach.

The aim of the next section is to detail the proposed MAS module.

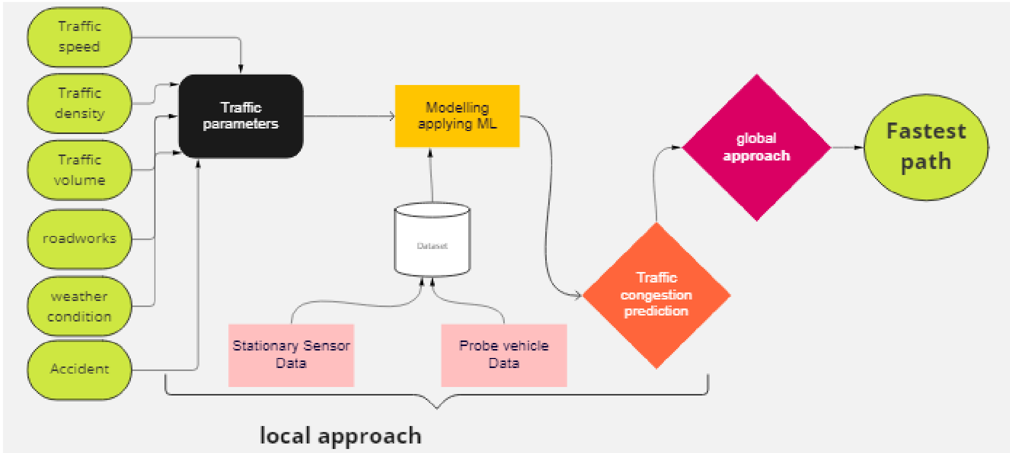

Fig. 1. The Proposed Architecture for Emergency Traffic Recommendation System.

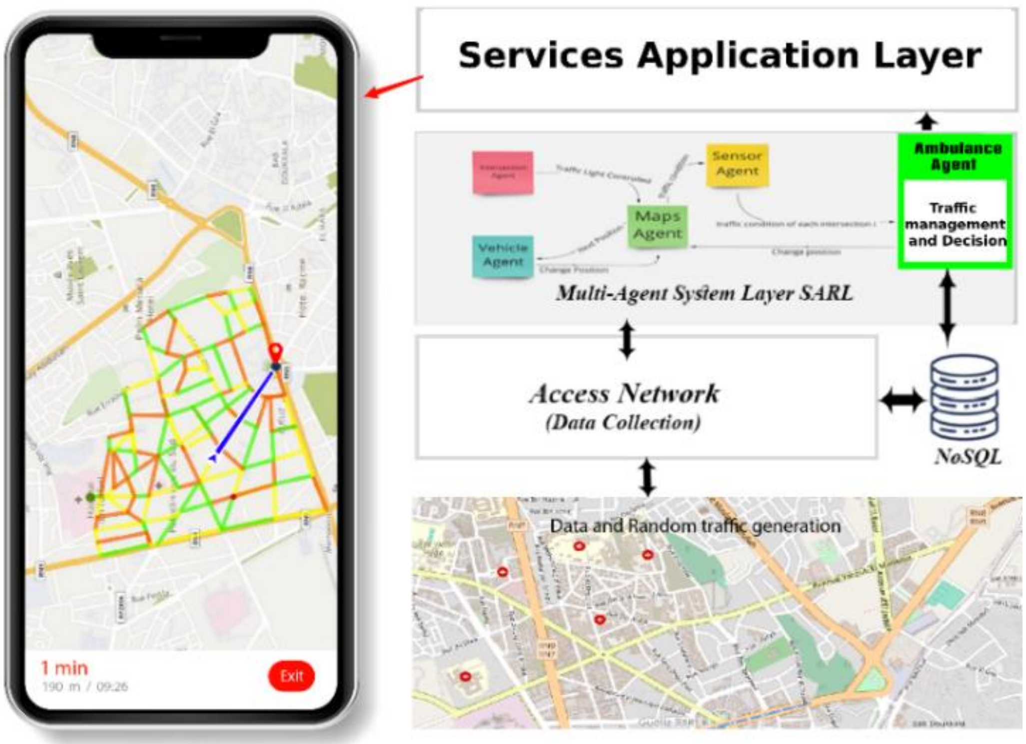

Fig. 2. An Architecture of Multi-Agent with SARL.

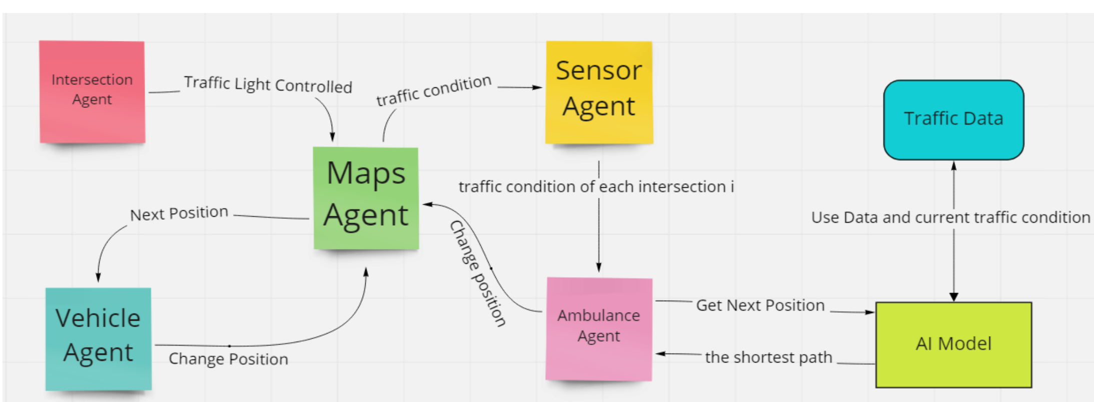

Fig. 3. The Proposed Architecture for Emergency Traffic Recommendation System.

IV. Multi-Agent System Modelling

To recommend the shortest time-based path in the case of an emergency from a departure point to an arrival point for an ambulance driver, the proposed approach was interested in the use of multi-agent systems, we have opted for SARL as an agent programming language that provides a set of concepts for the development of holonic systems (agents composed of other agents, with possibly new capacity) and based on events for communication between agents.

The proposed MAS contains five categories of agent:

- The maps agent, which is responsible for the simulation of traffic on the map, the input and the output of the agent vehicle, and also communicates with the other agents, for example giving the current coordinates to the ambulance agent.
- The sensor agent, responsible to collect data for each section on the map. For each section, we have two sensor agents positioned at the entry and exit of the section. The aim of this agent is to calculate the flow and the density of the vehicles on each section.
- The Intersection agent that takes control of the traffic light and priority of an intersection.
- The vehicle agent.
- The ambulance agent which represents the ambulance driver.

Table II summarizes all functions and events used to communicate between agents.

The agent ambulance needs to make a decision at each intersection from the origin to destination. So to choose the right direction it will communicate with the map agent to get its current position after that communicate with all sensor agents in each section of the road from its position to the arrival point to get the traffic condition for each section at instant T and used to predict the traffic condition at T+Δt to find uncongested road segments that will allow it to reach the arrival point.

According to fluid mechanics, the flow rate q is equal to the product of the concentration k and the velocity of the flow v, the average velocity for a traffic flow at time t and point x is defined by the ratio between the flow rate and the density such that:

$$\mathrm{v}(x,t) = \frac{q(x,t)}{k(x,t)} \tag{1}$$

Therefore, the proposed methodology was adopted to predict the flow and density of each section on the map and then calculate the average speed as a function of time to cross each different section. The main idea of this study is to identify similar patterns from the historical data of density and flow, and as an input real time data to predict the next state of the Traffic.

TABLE II. Agent Types Used in Proposed Multi-agent Systems

Agent	Event	Functions
Intersection	TrafficLightEvent: for each ΔT the Intersection agent send the traffic light condition to the maps agent	control the traffic light and priority of intersection
Vehicle	SendPositionEvent: for each ΔT the vehicle agent sends their position to the maps agent: to get permission to move to the next position.	move from intersection to other
Maps	After receiving the request from the vehicle agent to change their position, depending on the capacity of the red light or the priority the agent maps to verify the request, if possible it emits ChangePositionEvent: to give permission to the vehicle agent to change his position. In addition, the maps agent is able to detect every vehicle entering or leaving on section j, and it sends this information to the input sensor agent j and output sensor agent j.	simulation of traffic
Sensor	after receiving the request from the ambulance agent, it uses the information received from the maps agent to calculate the flow and density and send SendEtatTrafficEvent to respond to this request.	calculate traffic flow, density and speed
Ambulance	GetEtatTrafficEvent : send message to Sensor Agent to get (flow and density) of each edges of maps.	is VehicleAgent with some exception (move from an origin to destination, always have a priority in intersection)

V. Data and Random Traffic Generation

To evaluate the proposed approach and given the lack of empirical data, we were interested in randomly generating data representing vehicles, on road sections and intersections, that move randomly with a constant speed. The main properties of this traffic generation are presented as follows.

- Initialization at t=t0: Assign to each intersection a number of vehicles NV.
- For each intersection I at time t with the number of vehicles NVi(t), assign to each exit section of intersection I a random percentage XTi(t) % of NVi(t).
- The exit sections of an intersection I correspond to the entry sections at other intersections J1, J2, ...Jn.
- the following equations are presented by [31].
- The traffic flow is the number N of vehicles passing a period Δt at point x,

$$Q_{\Delta t}(x) = Q(x, t \rightarrow t + \Delta t) = \frac{N}{\Delta t} \tag{2}$$

- The density K is the number M of vehicles between x and x+Δx at time t.

$$K_{\Delta x}(t) = K(x \rightarrow x + \Delta x, t) = \frac{M}{\Delta x} \quad (3)$$

- These two equations (1) and (2) allow us to calculate the time needed to go from an intersection I to an intersection J.

An example of traffic visualization after vehicle generation is shown in Fig. 4

A. Data Mining

It incorporates choice interaction as information determination, pre-handling as information cleaning from missing qualities, copies, inconsistencies, and incorrect information, dimensional reduction transformation using 0–1 ranges for normalizing, and choice interaction as information determination. By using the formula below, it is very likely possible to determine data normalization.

$$V^i = \frac{v - min_a}{max_a - min_a}(new\_max_a - new\_min_a) + new\_min_a \quad (4)$$

Where:

- vi = updated data following normalization.
- V =normal data (without normalization).
- new_maxa = the new maximum value is 1.
- The new mina variable's maximum value is 0.
- maxa = the column's highest value.
- mina is the column's lowest value.

B. Algorithm for Congestion Coloring in Sections and Intersections

The upstream and downstream movements of road segments in a road network frequently have an impact on the traffic conditions of those segments. For instance, congestion frequently starts on one or more road segments and expands to other road segments over time. Fig. 5 illustrates congestion by assigning colors to road segments; the red, yellow, and green lines represent congested, slightly congested, and uncongested road segments, respectively.

To visualize changes in congestion on a map, we propose to color the sections and intersections according to the following two rules:

Rule 1 — CIj is the maximum capacity of the intersection J, so if the value of NVj(t) approaches CIj then it will have congestion. Three classes of congestion are considered by this experiment: Class 0 for NVj(t)<= CIj/3 (green coloring), Class 1 for CIj/3< NVj(t) <= 2CIj/3 (yellow coloring) and Class 2 for 2Cj/3< NVj(t) <= CIj (red coloring).

Rule 2 — CTk is the maximum capacity of the section K, so if the value of XVk(t) approaches CTk then it will have congestion. Three classes of congestion are considered in this experiment: Class 0 for NVk(t)<= CTk/3 (green coloring), Class 1 for CTk/3< NVk(t) <= 2CTk/3 (yellow coloring) and Class 2 for 2CTk/3< NVk(t) <= CTk (red coloring).

The idea is to exploit the fact that traffic jams move, dissipate or form from one section to another or from one intersection to another [32]. In order to recognize these congestions and show them on a map, we are interested in the sliding window algorithm to predict the state of a section exiting an intersection I at time t based on the traffic history of the entry sections of the same intersection I at time t+Δt.

For a more efficient recommendation system, it is mandatory that the prediction time of the traffic parameters is as fast as possible. In other words, the faster path recommendation must be present before the ambulance arrives at the next intersection.

The data of this simulated history is stored in a MongoDB database and comes from the traffic flows (number of cars per two minute) randomly generated on 84 intersections and 300 sections on the map of the city Marrakech in Morocco. These collected data which correspond to about 1000 hours of execution of the traffic generation algorithm generated more than 30000 elements for each section.

Fig. 4. Example of Traffic Generation between Intersections and Sections.

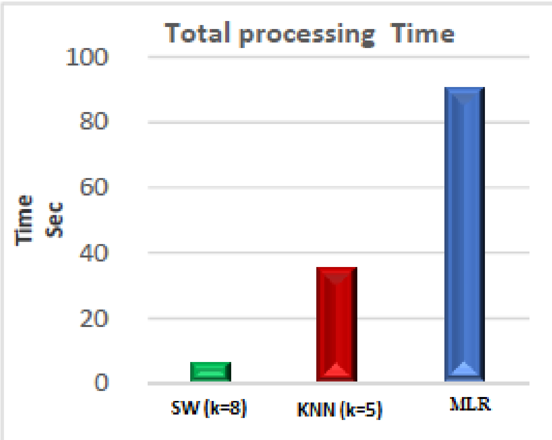

Fig. 5. The Processing Time of SW, KNN and MLR.

VI. Proposed Methodology based on Hybrid Approach

Recall that the proposed approach is based on hybrid approach: 1) a local approach dedicated to traffic prediction based on a machine learning algorithm and 2) a global approach to propose the fastest path from an origin to a destination.

A. *Local Approach: Traffic State Prediction using Machine Learning*

The traffic conditions on the same road throughout time exhibit some degree of regularity and consistency. As a result, future road traffic circumstances can be reliably predicted using an accurate comparison of historical and current road traffic conditions [40] and [41].

To identify the more suitable machine learning model for the proposed local approach, this work presents in the following subsections a comparison between the sliding window model, multivariate linear regression model and K-nearest Neighbor model. So we first construct a representative high-capacity historical database. Next, we define the elements of the model, including the value of K sliding window, Finally, a similar traffic flow matching to the present actual time and the observation data from the history database are retrieved to predict the traffic flow at the next time based on the observed values of the input and the search mechanism.

1) Sliding window algorithm: This algorithm has the advantage to mathematically model the aforesaid defined dependencies. Therefore, it is able to make the relationship between the historical data and previous k min data to predict the future traffic conditions [42].

The reason for applying the sliding window matching is that there is always a slight variation in traffic conditions that may depend on the variation of the last few minutes and also a dependence between the traffic conditions that persist in the current day and those of previous days [33] and [35]. Therefore, the previous days of the same time are checked to find similar traffic conditions. The sliding window is a good technique to capture the variation that could correspond to the real time variation.

The algorithm is as follows:

1. The matrix “TC” of the last k min for current day’s width size k×i, i is the number of the sections of intersection I.
2. Consider the matrix "PY" of n min data of size n i.
3. From the matrix "PY," create k sliding windows, each of size n*i, as W1, W2, W3, and Wn-k-1.
4. With the matrix "TC" as ED1, ED2, ED3,..., EDn-k-1, represent the Euclidean distance of each sliding window.
5. Select matrix Wi as:

W = Corresponding Matrix (Min.(EDi))

$\forall i \in$ [1, n-k-1]

6.

(i) Calculate the variation vector, abbreviated "VC," for the matrix "TC" of dimension k-1*1.

(ii) Calculate the variation vector for matrix "W" with dimension k-1*1as "VP" for WC.

(iii) Mc =Mean(VC)

(iv) Mp =Mean(VP)

(v) Estimated Variation “V” = (Mc + Mp)/2

(vi) To determine the anticipated traffic situation in t+Δt, add "V" to the existing traffic condition at t.

The performance of the Sliding Window is affected by the chosen number of neighborhoods (the parameter K). Table III shows the method used to determine the K sliding windows (k=8 in this case).

Algorithm.1 Sliding window

Input: hisdata[][] , runtimeData[][].

Output: pValue : prediction Value.

```
min = infinity
for i in (0, hisdata.length) do
  sum = 0
  for j in (0, runtimeData.length) do
      for k in (0, runtimeData.length) do      d      Euclidean distance
          sum = sum+(runtimeData[j][k] - hisdata[j+i][k]) *
                    (runtimeData[j][k] - hisdata[j + i][k])
          sum = √ sum
      end for
  end for
  if sum >= min then
      min =sum
      for k in (0, runtimeData.length) do
          pValue ← pValue + hisdata[i+b][hisdata[i].length−1]
              +hisdata[i][hisdata[i].length-1]
      end for
  end if
end for
return pValue=pValue/runtime.length*2.
```

The steps to calculate the Longest Substring k sliding windows are explained below:

- Selection of beginning data as the system's basic knowledge base.

- find the k sliding windows.
- The preparation of the training data in the form of the criteria's worth of fresh information about which the status is unknown
- Evaluation of each training set's state in accordance with predetermined rules to provide system knowledge.
- Based on each sample's training data and testing data, a distance calculation is made for each sample. the Euclidian equation:

$$d(x,y) = \sqrt{\sum_i^n (x_i - y_i)^2} \quad (5)$$

- determining the testing data's status based on the average of K samples of training data from sliding windows.

Testing results for training data 70% and testing data 30% the confusion matrix result with sliding window method can be seen in Table III.

TABLE III. SLIDING WINDOW TESTING RESULTS FOR TRAINING DATA 70% AND TESTING DATA 30%

K value	Accuracy	Precision	Recall	Classification Error	Mean squared Error
5	83.85%	83.22%	84.93%	16.15%	0.157
6	84.49%	83.75%	85.60%	15.51%	0.143
7	84.57%	84.30%	86.23%	15.43%	0.115
8	85.98%	85.69%	88.45%	14.02%	0.104
9	84.22%	85.08%	85.45%	14.78%	0.121

2) Multivariate linear regression: An effective way of modeling and prediction to investigate the quantitative relationship between the dependent and independent variables is to create a multivariate linear regression model utilizing historical data. Assume that there are m independent variables stated as Xtj (j = 1,..., m), the sample size is n, and the dependent variable is expressed as yt at t. Then, the multivariate linear regression model can be written as follows.

In case of multivariate linear regression, results are depending on multiple input values. The parameters used in the implementation are: m-row for the number of training elements (the number of vehicles in each section at time t), n-column for the number of features (the number of section and intersection).

Algorithm.2 Multivariate linear regressioN

Input: data set D={(x1,y1) ,(x2,y2) …(xn,yn)} at t

Output: θ=[w1,w2…..wd,b]

1: Define the loss function $J(\theta)=1/2n\ \Sigma^n(\theta^T xi\text{-}yi)^2$
2: Generating the randomly θ^0
3: **repeat**
4: $\theta^k = \theta^{k-1} - a(\partial J(\theta)/\partial \theta)$ where θ^k is the value of k^{th} iteration and
a is the iteration Step.
5: until $J(\theta^{k+1})\text{-}J(\theta^k) < t$
6: return 0

3) KNN: K Nearest Neighbor calculation falls under the Supervised Learning class and is utilized for grouping (most usually) and regression. It is a flexible calculation additionally utilized for ascribing missing qualities and resampling datasets. As the name (K Nearest Neighbor) proposes it thinks about K Nearest Neighbors (Data highlights) anticipate the class or continuous value for the new Data point.

KNN is a non-parametric supervised learning technique, that allows to classify the data point to a given category with the help of training set. The parameters used in the implementation are: K = 5 (after the test result), the number of vehicles of each intersection and road and the traffic states (0: free-flowing traffic/1: semi-congested/2: congested).

Algorithm.3 KNN

D: the set of learning objects.

Z(I)=(NV1, NV2..NVk,t) the vector of real-time values, such that NVIj represents the number of vehicles in the input section j of intersection I at time,t) with j belongs to 1...k

CG is the set of traffic states (0: free-flowing traffic/1: semi-congested/2: congested)

L class used to label the objects:

(CG, the number of vehicles in the exit edge of the intersection I at time t+Δt)

Input: Z(I)

Output: z=(CG, number of vehicles in the edge) ∈ L, class of z

1: **for** object y in D **do**
2: calculate d(z, y), the distance between z and y
3: **end for**
4: select N of D, the set (neighborhood)of the k nearest training
objects for z.
5: cap – max is the maximum capacity of the output
section of the intersection I
6: returns = Σ y∈N NV (t)/cap - max

B. Comparison between Multivariate Linear Regression, KNN and Sliding Window

Table IV compares the performance values that sliding window, multivariate linear regression, and KNN provide in three separate simulation tests: 70:30, 50:50, and 30:70. When compared to Multivariate Linear Regression and KNN, Sliding Window's performance metrics of accuracy, precision, recall, classification error, and mean squared error yield better results. Accordingly, Sliding Window's prediction is more accurate than both of these methods.

Fig. 5 shows the comparison of the processing time of the three algorithms. The result shows that the processing time of SW, KNN and MLR is 13, 38, 87 seconds respectively.

TABLE IV. COMPARISON ANALYSIS OF SLIDING WINDOW, KNN AND MULTIVARIATE LINEAR REGRESSION

Sim Test	Method	Accuracy	Precision	Recall	Classification Error	Mean squared Error
70 :30	SW	85,98%	85,69%	88,45%	14,02%	0,104
	KNN	84.44%	84.41%	84.2%	15,56%	0,124
	MLR	77,25%	72,08%	83,03%	22,75%	0,345
30 :30	SW	87,38%	88,84%	89,30%	12,62%	0,061
	KNN	87,13%	88,44%	88,56%	12,87%	0,083
	MLR	76,57%	72,21%	82,46%	23,43%	0,339
30 :70	SW	81,94%	78,05%	86,07%	18,06%	0,179
	KNN	81,30%	76,89%	84,77%	18,70%	0,196
	MLR	56,13%	54,23%	66,89%	43,87%	0,463

C. Global Approach: Prediction based on the Traffic Flow Coefficient

To determine the fastest path between a starting point and all other intersections of a graph (map), we propose as a first contribution to adapt Dijkstra's algorithm to take into account the traffic flow parameter. The value of this parameter is computed according to the real time traffic density, the traffic history, the road condition (public works, ...), etc. This value is also continuously calculated from the data collected at the moment by the road infrastructure sensors in order to find the fastest path from a starting point and to exclude the road with a high degree of congestion. Before presenting the equation (7) used to determine the time needed to cross each road, in [31] present the equations it refers to.

The density K is expressed as the number of vehicles per unit length (veh/km or veh/m).

$$k = \frac{TO}{L+l} \tag{6}$$

where TO is the occupancy rate, L is the length of the assumed vehicles and l is the length of the detection loop, tell that Q is the traffic flow.

$$Q = K \times V \tag{7}$$

Thus, the velocity v is expressed as follows.

$$V = \frac{Q}{TO} \times (L+l) \tag{8}$$

Finally, the travel time Ti for edge i is calculated as follows.

$$Ti = \frac{Vi}{Di} \tag{9}$$

The algorithm 2 used follows the following steps:

1. Initialize all roads with a high degree of "infinite" congestion, in other words vehicles cannot move. Moreover, the ambulance is initially at the starting point at time T0.
2. Activate the starting point (algorithm 4 line 4)
3. Calculate the temporary travel time of all intersections adjacent to the current intersection by adding their times (algorithm 4 line 7).
4. If the calculated time of an intersection is less than the current time, update the time and set the current intersection as the antecedent. This step is also called updating and is the central idea of Dijkstra. (algorithm 4 line 9-19)
5. Set the intersection with the minimum temporary time as active. Mark its travel time as permanent. (algorithm 4 line 10)
6. Repeat steps 3 to 5 until there are no more intersections with a permanent time, whose neighbors still have a temporary travel time.

Algorithm.4 Dijkstra

Input: current-Point: the current ambulance junction, destination,graph of time at T0 *d* travel time for (each section i and each node j), from the prediction Algorithm.
Output: path[] : the fastest path.

```
 1: while (current- Point! = destination) do
 2:     min = infinity

 3:     for vertex v in Graph do
 4:         time[v] =infinity              // initial travel time from current-
            Point to vertex v is set to infinity
 5:         previous[v] =undefined  // Previous node in fastest
            path from current-Point
 6:     end for
 7: time[current-Point]=0  // travel time from current-Point to
    current-Point
 8:     Q = set   // the set of all nodes in Graph
 9:     while sum >= min do
10:         u =  node // node in Q with fastest time[ ]
11:         remove u from Q
12:         for neighbor v of u do //where v has not yet been
            removed from Q.
13:             alt = time[u] + timebetween(u; v)
14:             if  alt ¡ time[v] then
15:                 time[v]  = alt
16:                 previous[v] = u
17:             end if
18:         end for
19:     end while
20: fastest path = previous[]  //  the ambulance moves to the
    next point
21: current = Next-Point // from fastest-path
22: Graph =prediction- travel - time(T 0 + t)   d graph of
    time at T0+t
23: end while
```

VII. EMERGENCY TRAFFIC RECOMMENDER SYSTEM IMPLEMENTATION

To implement the proposed emergency traffic recommender system, this paper opted for the agent-oriented programming language SARL, the DBMS MongoDB, and the JavaScript language to build the emergency navigation service for paramedics based on OpenStreetMap.

Fig. 6. Update of the Recommended Path to the Destination.

Fig. 6 illustrates the different stages of movement of the two ambulances A, B. The first ambulance A (red dot) uses the proposed recommendation system to avoid congestion, while the second ambulance B (blue dot) uses classical Dijkstra's algorithm to find the fastest path between the starting point and the destination.

TABLE V. COMPARISON BETWEEN THE PERFORMANCES OF THE THREE METHODS ON DIFFERENT TIMES OF THE SIMULATION

simulation time	Method	distance	time of trips	Fastest Path time
2:00 pm	KNN	1.5 km	449 s	449 s
	MLR	1.7 km	483 s	
	SW	1.5 km	449 s	
2:40 pm	KNN	1.7 km	413 s	413 s
	MLR	1.7 km	413 s	
	SW	1.7 km	413 s	
3:00 pm	KNN	1.7 km	285 s	285 s
	MLR	1.7 km	285 s	
	SW	1.7 km	285 s	
3:20 pm	KNN	1.7 km	435 s	419 s
	MLR	1.5 km	428 s	
	SW	1.9 km	419 s	

The Table V compares the performances of congestion prediction that sliding window, multivariate linear regression, and KNN provide in different moments of the simulation. The result shows that the Sliding Window's prediction is better than both of these methods. Sliding Window has the advantage to make the ambulance avoiding traffic congestion every time, and taking different way to destination by using the fastest way based on the distance and traffic congestion.

Based on the results obtained, we are currently working on improving the proposed equation (8) to include more detailed parameters such as road condition (public works, presence of accidents, ...), type of vehicle crowd, speed of the vehicle crowd, etc.

VIII. CONCLUSIONS

The objective of this contribution is to propose a solution for traffic recommendation in emergency situations in order to avoid ambulances to be stuck in traffic jams at the time of trips and interventions. For this purpose, we proposed an architecture based on machine learning techniques for classification and collection of the required information to predict the fastest path to ambulances. Given the lack of empirical data, this paper proposed a random traffic generation algorithm to experiment with the recommendation solution. Based on the generated dataset, we proposed a hybrid approach composed of a local approach to visualize the state of congestion on the map of the city of Marrakech under OpenStreetMap, and a global approach to suggest the fastest path as the ambulance moves. The local approach relies on the Sliding window technique to predict the congestion of sections in three classes: congested, slightly congested and not congested sections. While the global approach implements the Dijkstra algorithm using the multi-agent system modeling language SARL.

The proposed recommendation solution predicts the fastest path for emergency situation from an origin to a destination. Since these recommendations may or may not be taken into consideration by drivers, we are currently integrating drivers experience interaction with the recommender system to build more efficient emergency traffic recommender system.

REFERENCES

[1] P. Zhao, Z.Yu, Investigating mobility in rural areas of China: Features, equity, and factors, Transport Policy 94 (2020) 66–77.

[2] L. Changle, Y. Wenwei, M. Guoqiang, X. Zhigang, Congestion Propagation Based Bottleneck Identification in Urban Road Networks, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 69, NO. 5, MAY 2020.

[3] H. Hammad, M. Ashraf, F. Abbas, H. Bakhat, A. Qaisrani, M. Mubeen, S. Fahad, M. Awais, Environmental factors affecting the frequency of road traffic accidents: a case study of sub-urban area of Pakistan, 11674–11685 (2019). https://doi.org/10.1007/s11356-019-04752-8.

[4] M. Poulton, A. Noulas, D. Weston., G. Roussos, Modelling metropolitan-area ambulance mobility under blue light conditions. DOI 10.1109/ACCESS.2018.2886852, IEEE Access.

[5] M.S. Ali, M. Adnan, S. Muhammad Noman, B. Fazal Abbas."Estimation of traffic congestion cost-A case study of a major arterial in karachi," Procedia Engineering, vol. 77, pp. 37–44. https://doi.org/10.1016/j.proeng.2014.07.030.

[6] A.M. De Souza, C.A. Brennand, R.S. Yokoyama, E.A. Donato, E.R. Madeira, L.A. Villas, (2017). Traffic management systems: A classification, review, challenges, and future perspectives. International Journal of Distributed Sensor Networks 2017, Vol. 13(4).

[7] M. Weisfeldt, L. Becker, Resuscitation after cardiac arrest: a 3-phase time-sensitive model, JAMA, vol. 288, no. 23, pp.3035–3038.

[8] K. Athavan, G. Balasubramanian, S. Jagadeeshwaran , N. Dinesh, (2012). Automatic ambulance rescue system. IEEE. second international conference on advanced computing & communication technologies. IEEE Transactions on Vehicula,Vol. 69, NO. 5.

[9] V. Servizi, C. Pereira, MK. Anderson, A. Nielsen, Transport behavior-mining from smartphones: a review, European Transport Research Review; Heidelberg Vol. 13, N° 1, (Dec 2021). DOI:10.1186/s12544-021-00516-z.

[10] S. Vasuki, G. Ruthvik, (2015). Automated traffic signal for hassle free movement of ambulance. IEEE international conference on electrical, computer and communication technologies. .doi: 10.1109/ICECCT.2015.7226114.

[11] P. Rosayyan, S. Subramaniam, S. Ganesan, Decentralized Emergency Service Vehicle Pre-Emption System Using RF Communication and GNSS-Based Geo-Fencing . IEEE Transactions on Intelligent Transportation Systems, 1–10. doi:10.1109/tits.2020.3007671.

[12] L. Ding, D. Zhao, Z. Wang, G. Wang, C. Tan, L. Fan, H. Ma, Learning to Help Emergency Vehicles Arrive Faster: A Cooperative Vehicle-Road Scheduling Approach. arXiv:2202.09773. https://doi.org/10.48550/arXiv.2202.09773.

[13] L. Zhong, Y. Chen, A Novel Real-Time Traffic Signal Control Strategy for Emergency Vehicles. IEEE Transactions on Intelligent Transportation Systems, 19(1), 13–24. doi:10.1109/tits.2016.2625324.

[14] Dimitrios V.Lyridis, An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints, 2021, Ocean Engineering, Vol: 241, 109890, https://doi.org/10.1016/j.oceaneng.2021.109890.

[15] R. Chen, C. Gotsman, Efficient fastest-path computations for road maps. Comp. Visual Media 7, 267–281 (2021). https://doi.org/10.1007/s41095-021-0211-2.

[16] W. Sun, C Zhu, L. HuiL, ASNN-FRR: A traffic-aware neural network for fastest route recommendation. Geoinformatica (2021). https://doi.org/10.1007/s10707-021-00458-7.

[17] Li, X., Zhou, J. & Pedrycz, W. Linking granular computing, big data and decision making: a case study in urban path planning. Soft Comput 24, 7435–7450 (2020). https://doi.org/10.1007/s00500-019-04369-6.

[18] C.M. Stewart, S.M. Aminossadati, M.S. Kizil, EMERGENCY EGRESS PATHWAY PREDICTION USING VENTILATION MODELS. The Australian Mine Ventilation Conference 2017, No: Stewart_089.

[19] S. Ye, (2012), A reliability-based approach of fastest routes planning in dynamic traffic network under emergency management situation. International Journal of Computational Intelligence Systems, 4(6), 1224–1236. doi:10.1080/18756891.2011.9727871 10.1080/18756891.2011.9727871.

[20] J. Cyriac, K.S. Vijula Grace, (2020). A New Approach for Accurate Path Prediction Using Multiple Prediction System. https://doi.org/10.1016/j.matpr.2020.03.599.

[21] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, A. Perallos, (2016) "A hybrid method for short-term traffic Congestion forecasting using genetic algorithms and Cross entropy," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2.

[22] X. Zhang, E. Onieva, A. Perallos, E. Osaba, V. C. S. Lee, (2014) "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 127.

[23] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, B. Zhang, (2016). "Urban traffic congestion estimation and prediction based on floating car trajectory data," Future Generation Computer Systems, vol. 61, pp. 97–107.

[24] Q. Yang, J. Wang, X. Song, X. Kong, Z. Xu, B. Zhang, (2015) Urban traffic congestion prediction using floating car trajectory data, in Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, pp. 18–30, Springer, Zhangjiajie, China.

[25] X. Wang, K. An, L. Tang, X. Chen, (2015), "Short term prediction of freeway exiting volume based on SVM and KNN," International Journal of Transportation Science and Technology, vol. 4, no. 3, pp. 337–352.

[26] T. Adetiloye, A. Awasthi, (2019), "Multimodal big data fusion for traffic congestion prediction," Multimodal Analytics for Next-Generation Big Data Technologies and Applications, Springer, Berlin, Germany, pp. 319–335.

[27] X. Liu, Y. Feng, Q. Wang, H. Zhang, X. Wang, (2014), "Prediction of urban road Congestion using a bayesian network approach," Procedia—Social and Behavioral Sciences, vol. 138, no. C, pp. 671–678.

[28] J. Kim, G. Wang, (2016), "Diagnosis and prediction of traffic Congestion on urban road networks using bayesian networks, «Transportation Research Record: Journal of the Transportation Research Board, vol. 2595, no. 1, pp. 108–118.

[29] Y. Zheng, C. Dong, D. Dong, S. Wang, (2021), Traffic Volume Prediction: A Fusion Deep Learning Model Considering Spatial–Temporal Correlation. https://doi.org/10.3390/su131910595.

[30] A. Aslam, E. Curry, Investigating response time and accuracy in online classifier learning for multimedia publish-subscribe systems. Multimed Tools Appl 80, 13021–13057 (2021). https://doi.org/10.1007/s11042-020-10277-x.

[31] C.Buisson, (2010), Lesort J Comprendre le trafic routier. Methods and calculus. CERTU, 111p, 2010, Coll. References, 978-2-11-098892-8.

[32] L. Manman, L. Jian. S. Jiahui, T. Qiang, (2019), Day-to-Day Evolution of Traffic Flow with Dynamic Rerouting in Degradable Transport Network. https://doi.org/10.1155/2019/1524178.

[33] C. Wentao, W. Jie, (2019), Research on traffic flow congestion based on Mamdani fuzzy System, https://doi.org/10.1063/1.5090755.

[34] E. Onieva, P. Lopez-Garcia, A.D. Masegosa, E. Osaba, A. Perallos, (2016), A Comparative Study on the Performance of Evolutionary Fuzzy and Crisp Rule Based Classification Methods in Congestion Prediction, https://doi.org/10.1016/j.trpro.2016.05.368.

[35] J. Zaki, A. Ali-Eldin, S. Hussein, S. Saraya, F. Areed, (2019), "Time aware hybrid hidden Markov models for traffic Congestion prediction," International Journal on Electrical Engineering and Informatics, vol. 11, no. 1, pp. 1–17.

[36] J. Wang, Y. Mao, J. Li, Z. Xiong, W. Wang, (2015), Predictability of road traffic and Congestion in urban areas, PLoS One, vol. 10, no. 4, Article ID e0121825.

[37] F. Tseng, J. Hsueh, C. Tseng, Y. Yang, H. Chao, L. Chou, Congestion prediction with big data for real-time highway traffic. 2018, IEEE Access, 1–1. doi:10.1109/access.2018.2873569.

[38] K. Nadeem, T.Fowdur, (2018), Performance analysis of a real-time adaptive prediction algorithm for traffic congestion, Journal of Information and Communication Technology, vol. 17, no. 3, pp. 493–511.

[39] L. Jiwan, H. Bonghee, L. Kyungmin, J. Yang-Ja, A prediction model of traffic congestion using weather data," in Proceedings of the 2015 IEEE International Conference on Data Science and Data Intensive Systems. Lee, J., Hong, B., Lee, K., & Jang, Y.-J. (2015). A Prediction Model of Traffic Congestion Using Weather Data. 2015 IEEE International Conference on Data Science and Data Intensive Systems. doi:10.1109/dsdis.2015.96.

[40] S. Azimi, R. Delavar, A. Rajabifard, (2017). Multi-agent simulation of allocating and routing ambulances under condition of street blockage after natural disaster. The International Archives of the Photogrammetry, Remote sensing and spatial Information Sciences, Volume XLII-4/W4.

[41] S. Ibri, M. Nourelfath, H. Drias, (2012). A multi-agent approach for integrated emergency vehicle dispatching and covering problem, engineering applications of artificial intelligence, Volume 25, Issue 3.

[42] P. Kapoor, S. Singh Bedi, Weather Forecasting Using Sliding Window Algorithm, Hindawi Publishing Corporation ISRN Signal Processing Volume 2013, Article ID 156540, 5 pages http://dx.doi.org/10.1155/2013/156540.