Congestion Control in Vehicular Ad-hoc Networks: A machine learning perspective

[Santosh Kumar Maharana
Research Scholar, Biju Patnaik University of Technology, Odisha
Prashanta Kumar Patra
Dean, SRIC, SOA University, Bhubaneswar, Odisha

Abstract:
Due to the dynamic nature of messages transmitted between vehicle-to-infrastructure (V2I) and Vehicle-to-vehicle(V2V) communications under vehicular ad-hoc networks (VANETs), risk of network congestion increases unexpectedly in vehicular networks. The flooding of beacons in vehicular networks can potentially increase the risk of accidents and other traffic issues which can be developed and left unnoticed due to the channel congestion in vehicular networks. Hence controlling congestion of basic safety messages (BSMs) is a challenging issue and topic of interest for most of the researchers. Our proposed model for controlling congestion in vehicular networks uses a machine learning framework to control and maintain channel busy rate (CBR) in vehicular networks. The present research work is compared with existing machine learning methods for proving its effectiveness.

Keywords- *V2I, V2V, BSMs, CBR, VANET*

1. Introduction

A VANET, which arranges its communication system independently of any other infrastructure, is a MANET's sister. Road side units (RSU) and mobile nodes (MN) make up a VANET [1]. The sensors which are integrated into cars also referred as on board units (OBU), used for data exchange (signal processing) that lead to RSUs are known as mobile nodes. Fixed installation units, or RSUs, act as the MN's communication gateway to servers or the Internet. Although VANET offers a wide range of services, its road safety services—which use Internet-based data exchange to reduce traffic accidents—are the most significant. Vehicles in connection to various other vehicles (V2V) and infrastructure (V2I) communications remain the two classifications of VANET communications [2].

Vehicular ad hoc network is made of numerous movable cars that are capable of communicating with one another without the need for a fixed infrastructure. Through the optimization of traffic flow and a notable reduction in automotive accidents—a serious problem facing modern society—it helps to improve road safety [3].

VANET enables wireless communication within a range of about 1000m utilizing the Dedicated Short-Range Communication (DSRC) Protocol. DSRC utilizes a bandwidth of 75 MHz within the frequency range of 5.85-5.925 GHz, which has been designated by the Federal Communication Commission. It enables cars to establish contact with other vehicles, denoted as (V2V) contact, along with road-side infrastructure, called as V (V2I) communication [4], or an amalgamation of both V2V and V2I. The user's text is available in [5]. In order to create (V2V) communiqué, it remains necessary for cars to be furnished through an On-Board Unit (OBU), Omni-directional antennae, actuators and sensors, as well as a Global Positioning System (GPS). In order to facilitate V2I communication, it is necessary for highways to be equipped with Roadside Base stations or Roadside Units (RSUs). A Roadside Unit (RSU) is a stationary device positioned beside the road to provide specialized services within a designated region, while On-Board Units (OBUs) are mounted in vehicles and consist of integrated transceivers and computing equipment. Both RSUs and OBUs contribute to VANET navigation **as shown in Figure 1.**

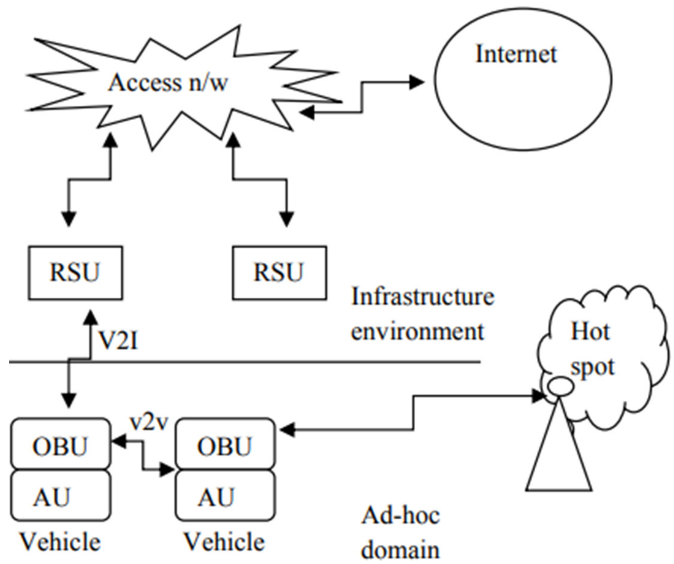

Figure.1 Architecture of VANET

Through video monitoring and sensors, Vehicular network allows vehicles and RSUs to transmit a range of data, such as the length of traffic bottlenecks and safety information to prevent accidents[6]. When this kind of information is disclosed, drivers receive an alert or warning message about potential hazards. This will contribute to a notable decrease in auto accidents. [7] The principles, traits, difficulties, and prerequisites of VANET security were elucidated by Hassnaa et al. [8]. They provided information on security protocols and architecture. They described the known attack types and their defences, as well as the various problems and technological difficulties pertaining to VANET security. The primary aspects of VANET, including its architecture, functionality, and simulation tools for simulating its protocols and applications, were provided by Saif et al. [9]. On the other hand, Lakshmanaprabu SK et al. [10] provided an overview of the difficulties that now exist as well as possible uses. They covered medium access control techniques, routing strategies, hardware and spectrum concerns, as well as security and privacy issues related to VANETs. Applications and significance of VANETs in intelligent transportation networks.

2. Literature Survey- Channel Congestion control in VANET

VANETs have evolved quickly for supporting diversified traffic, supporting a wide range of applications, especially security-related applications such as advance notifications for collisions, violation messages for traffic-signals, applications including systems for providing weather information, nearest/ cheapest restaurant location etc. Various safety specific applications depend on event-driven ‘alert’ notifications, as well as regular BSMs containing critical data, like speed, positioning and direction of the nearest vehicles participating in traffic. Due to the limited channel capacity, and the fact that BSMs need to be delivered reliably (in order to maintain an appropriate level of awareness), reliable BSM delivery has proven to be a challenge for VANETs. Hence, multiple research work have been conducted emphasizing congestion control algorithms for BSMs in recent years.

Common parameters used to control channel congestion are message transmission rate (MTR) [11] and power (or a mix of both) [12-13]. Apart from them parameters like carrier sense threshold [14] and data rate [15] are also utilized to control congestion. Machine learning helps to improve performance in a variety of tasks in different industries, including finance, healthcare, etc. [16] in most of recent research work. Machine Learning based congestion control techniques are also utilized for providing solution of different challenges in VANets [17].

Machine Learning (ML) solutions for VANets have been applied in various contexts, including misbehavior detection (MDA) [18], multi-hop broadcast protocols (MUPs) [19], DDoS attack detection (DDoS) [20] and delay minimization routing (LMR) [21]. In recent years, several papers are suggested Machine Learning-based solutions for congestion control and load balancing of heavily used VANET communications. For example, in Paper [22], Locally Located and Centralized ML-CC was proposed as a solution to the congestion at the intersections. The ML-CC strategy was based on K-means clustering.

According to this model, Roadside Traffic Units (RSU's) are installed at intersections to collect all the traffic transmitted by different nodes within their area of coverage. Paper [23] used deep-reinforcement learning and solved problem of resource allocations in vehicular communications, using unicast as well as broadcast methods. Primarily it is based on resource allocation and the appropriate message selection for onward transmission. Paper [24] used reinforcement learning for controlling power utilization in vehicles and rate adaption by radio access network downlink for cellular networks.

Similarly Tang J et al. [25] proposed an optimized beacon-rate selection using function approximation with different policies. RL-CDCA , a multi-spatial Reinforcement Learning based dynamically changing channel assignment on MAC layer was proposed[7]. Here nodes share their individual reward among other nodes adjustment of their channel selection decisions.

Developed protocols for MAC for efficient use of channel sharing in wireless communication were also proposed [9], [11]. Due to unreasonable number of research activities regarding RL are conducted on vehicular communications, the current work presents a specific RL methodology for congestion control in VANET.

4. Machine learning approaches for congestion control in Vehicular Ad-hoc networks:

4.1 Data Collection: Traffic Scenario Creation from OpenStreetMap

SUMO, also known as Simulation of Urban Mobility, is a software tool used for the simulation and management of extensive road networks. Its purpose is to accurately replicate road traffic scenarios. Initially, a specific area is chosen on a tangible map and loaded into SUMO to generate the traffic scenario. Subsequently, this scenario is merged with a machine learning model.

This work employs the use of Open Street Map (OSM), a digitized street map, to precisely identify the boundaries and intersections of roads, resulting in very accurate results. OSM, a globally collaborative open-source platform, is created by a community of mappers to create street-level maps. Users have the ability to search for and choose specific regions, as well as different street features like lanes, primary roads, highways, sidewalks, railroads, underground passages, traffic signals, bridges, and junctions. In order to streamline the simulation process, the city map, which includes the main roadways, is extracted from OSM and converted into a map format. An OSM file is a text file encoded in XML format that contains information on nodes, streets, and tags (object properties).

Figure 2 presents a flowchart that shows the process of generating SUMO configuration files using the city map obtained from OSM. The process of creating a map may be outlined in a step-by-step manner as follows:

1 The map.osm file from Open Street Map was imported for the selected geographic region.
2 The map.net.xml file was generated using the NETCONVERT command-line program.
3 The route file map.rou.xml was generated using the randomTrips.py Python tool.
4 The PLOYCONVERT command-line application was then used to produce the map.poly.xml file, integrating further data from typemap.xml.
5 Afterwards, the SUMO configuration file (map.sumocfg) was created using the previously stated files.
6 The SUMO configuration file encapsulates the traffic scenario that has been developed.

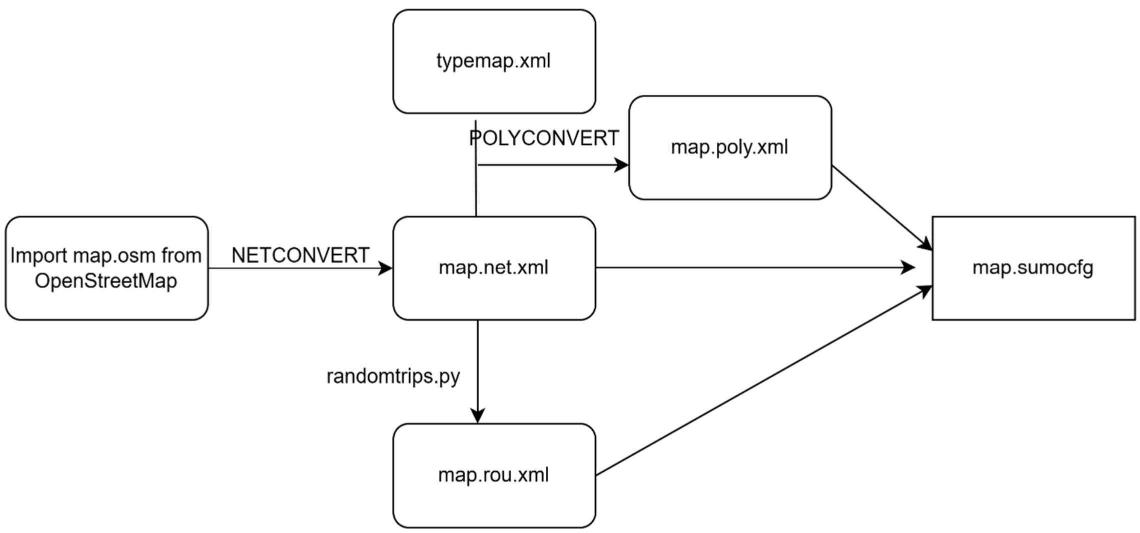

Figure. 2 Creating SUMO configuration file for real world traffic

The SUMO simulator often uses a default routing method called Dua Router, which is a specialized functionality inside SUMO that creates a vehicle-routing file (map.rou.xml). In order to use Dua Router, it is necessary to create a road network (map.net.xml) and define traffic needs, such as trip and flow definitions. The demand definitions, obtained from the origin and destination edges, assist in calculating the most efficient path for dynamic user assignment (DUA). However, when using machine learning with SUMO, the default routing approach is replaced with a machine learning algorithm.

By using the TraCI library, the SUMO simulator may be controlled, and establishing a connection to this library facilitates the implementation of a deep learning technique [17]. Python is used as the programming language to control both TraCI and machine learning in this work. PyCharm, an adequate integrated development environment (IDE) for Python

programming, enables the seamless integration of machine learning model into SUMO.

4.2 Methodology

The machine learning approach uses the following steps:
Step 1: Traffic prediction using Machine learning classifiers
Step 2: Channel Busy Rate (CBR) prediction using regression
Step 3: Implementing machine learning methods

4.3. ML Classifiers used for traffic prediction

Traffic prediction algorithms include Naive Bayes, logistic regression [13], KNN, Decision tree classification and Random-forest classification.

Four factors affect journey time: day, time, weather, and temperature. Machine learning algorithms must classify the traffic situation from 1 to 5 from the input parameters of day, zone, weather, and temperature.

In order to forecast the dependent data variable, the Logistic Regression model examines a number of independent variables [8]. This parametric classification model predicts categorical output using a set number of parameters and input data. Equation (1) fits Sigmoid curves.

$$Sigmoid(x) = \frac{1}{(1+e^{-x})} \tag{1}$$

Classification with K-NN: K-NN is a nonparametric method that makes no data assumptions [9]. The closest neighbours of an unknown variable are denoted by K. In order to find the shortest distance from unknown data, KNN uses equation (2) to compute all possible Euclidean distances.

$$ED = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

Naive Bayes: A fast and efficient classification algorithm [10]. Bayes' theorem is used according to equation (3), assuming that the attributes are independent.

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3}$$

Decision Tree classification: A method that uses trees to do regression and classification. Internal nodes stand for dataset attributes, and leaf nodes for decision rule results [6]. By answering yes or no, questions divide the tree into subtrees. This process starts by choosing root characteristics and continues to branch out until it reaches the leaf node. Because trimmed trees are smaller, faster prediction calculations are feasible.

Random Forest (RF) classification: Applied to the tasks of regression and classification. It is possible to increase predictability by using the mean and a large number of decision trees applied to subsets of the dataset, as shown in Figure 3 [9]. The output is predicted by RF using tree consensus. The accuracy and lack of overfitting are both improved by RFs that have more trees. Compared to other algorithms, random forest learns quicker, even when making precise predictions.

4.4. Predicting CBR using Regression

Using regression analysis, a dependent (target) and independent (s) variable can be predicted and linked. One or more independent variables are utilized to forecast a continuous target variable. CBR, is a variable that is influenced by factors such as the density of vehicles, the size of the packets, and the total number of packets transmitted.

1) Lasso regression: The process reduces parameters in order to punish complexity [11]. L1 regularization is used to induce penalties in LASSO (4). The penalty is equal to the absolute value of the coefficient.

LASSO = residual error +λ∗ (sum of the absolute errors)

$$= \sum_{k=1}^{n} (Y_k - \sum x_{kl}\beta_l)^2 + \lambda * \sum_{l=1}^{p} |\beta| \tag{4}$$

λ is known as regularization parameter, The optimal value of λ can be determined with cross-validation techniques, such as k-fold cross-validation; this approach finds the λ value that minimizes the mean squared error or other performance metrics. A higher λ value applies more regularization. As λ increases, model bias increases while variance decreases. This is because as λ becomes larger, more coefficients β shrink to zero.

2) Ridge regression: Penalty is L2 regularization, where lambda is multiplied by the squared magnitude of loss function coefficients. Ridge model coefficients will not diminish like LASSO [11]. Ridge = residual error + λ∗ sum of coefficient squares.

$$= \sum_{k=1}^{n}(Y_k - \sum x_{kl}\beta_l)^2 + \lambda * \sum_{l=1}^{p}|\beta^2| \quad (5)$$

3) Elastic net regression: In this approach, LASSO is appended to Ridge regression and suppresses the influence of several features without eliminating all features [12].

$$L=\sum_{k=1}^{n}(Y_k - \sum x_{kl}\beta_l)^2 + \lambda * \sum_{l=1}^{p}|\beta_l{}^2| + \lambda * \sum_{l=1}^{p}|\beta_l| \quad (6)$$

5. Implementation of machine learning methods

The average Channel Busy Rate (CBR) was calculated for Naive Bayes, logistic regression [15], KNN, Decision tree classification and Random-forest classification.

Considering the number of beacons received by different machine learning classifiers considering 500 vehicles in a highway of 4 lane X 4Km dimension the average Channel Busy Rate (CBR) was calculated, with average simulation time of 100s. Figure 3 represents the comparison of CBR for each of the classifiers. Similarly Figure 4 represents the comparison of CBR for each of the classifiers.

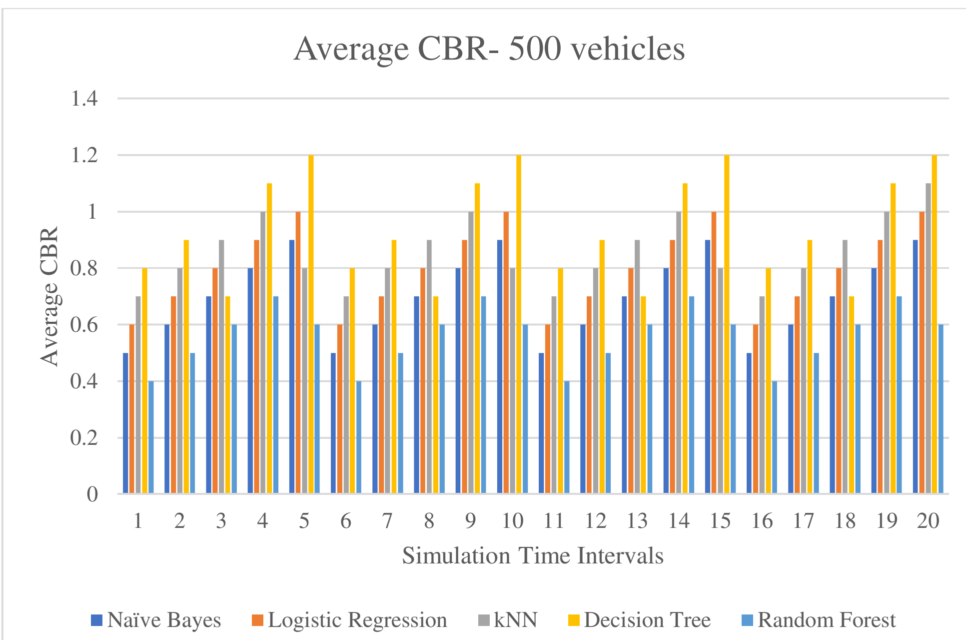

Figure. 3 Comparison of average CBR for ML classifiers- 500 vehicles

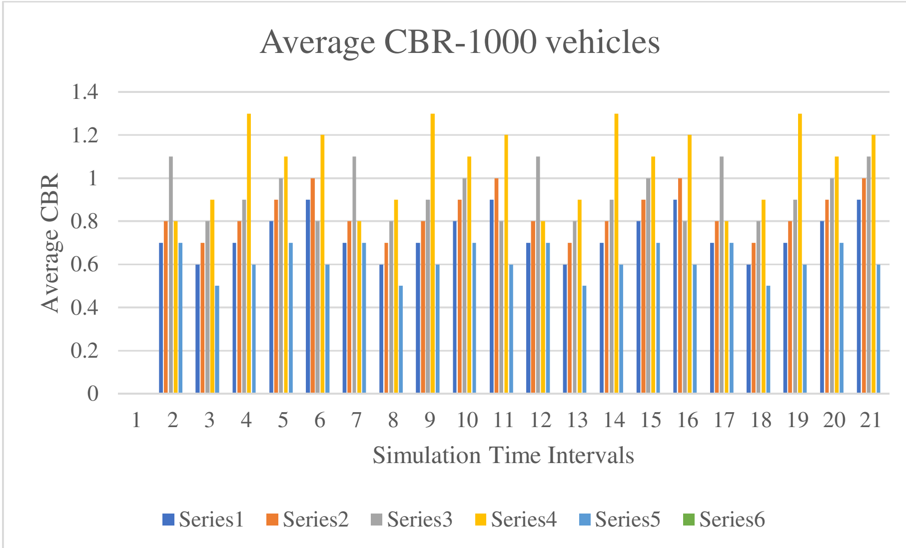

Figure. 4 Comparison of average CBR for ML classifiers- 1000 vehicles

6. Conclusion

In this research, we design Machine Learning based framework for controlling vehicular congestion and evaluate the system using changing traffic flow models. The findings show that, Random-forest method exhibits minimum average CBR among all other machine learning methods. The average CBR for random forest shows a range between 0.5-1 which can effectively handle and avoid congestion. Our upcoming research will concentrate on employing additional channel metrics like BER and IPD, and utilizing deep neural networks to handle live action and real-world state space.

References

[1] Subramaniam M, Rambabu C, Chandrasekaran G, Kumar NS. A Traffic Density-Based Congestion Control Method for VANETs. Wireless Communications and Mobile Computing. 2022; 2022:1–14. Available from: https://doi.org/10.1155/2022/7551535.

[2] Ngo TT, Huynh-The T, Kim DS. A novel VANETs-based traffic light scheduling scheme for greener planet and safer road intersections. IEEE Access. 2019; 7: 22175–22185. Available from: https://doi.org/10.1109/ACCESS.2019.2891250.

[3] Jindal V, Bedi P. An improved hybrid ant particle optimization (IHAPO) algorithm for reducing travel time in VANETs. Applied Soft Computing. 2018; 64: 526–535. Available from: https://doi.org/10.1016/j.asoc.2017.12.038.

[4] Ata A, Khan MA, Abbas S, Khan MS, Ahmad G. Adaptive IoT Empowered Smart Road Traffic Congestion Control System Using Supervised Machine Learning Algorithm. The Computer Journal. 2021;64(11):1672–1679.Available from: https://doi.org/10.1016/j.jksuci.2018.10.011.

[5] Zheng H, Chang W, Wu J. Traffic flow monitoring systems in smart cities: Coverage and distinguishability among vehicles. Journal of Parallel and Distributed Computing. 2019;127: 224–237. Available from: https://doi.org/10.1016/j.jpdc.2018.07.008.

[6] Khatri S, Vachhani H, Shah S, Bhatia J, Chaturvedi M, Tanwar S, et al. Machine learning models and techniques for VANET based traffic management: Implementation issues and

challenges. Peer-to-Peer Networking and Applications. 2021;14(3):1778–1805.
[7] D'andrea E, Marcelloni F. Detection of traffic congestion and incidents from GPS trace analysis. Expert Systems with Applications. 2017; 73:43–56. Available from: https://doi.org/10.1016/j.eswa.2016.12.018.
[8] Jobaer S, Zhang Y, Hussain MAI, Ahmed F. UAV-Assisted Hybrid Scheme for Urban Road Safety Based on VANETs. Electronics. 2020;9(9):1499. Available from: https://doi.org/10.3390/electronics9091499.
[9] Rashid MM, Datta P. Performance Analysis of Vehicular Ad Hoc Network (VANET) Considering Different Scenarios of a City. International Journal of Computer Applications. 2017;(10):162. Available from: https://doi.org/10.5120/ijca2017913329.
[10] Lakshmanaprabu SK, Shankar K, Rani SS, Abdulhay E, Arunkumar N, Ramirez G, et al. An effect of big data technology with ant colony optimization-based routing in vehicular ad hoc networks: Towards smart cities. Journal of Cleaner Production. 2019; 217:584–593. Available from: https://doi.org/10.1016/j.jclepro.2019.01.115.
[11] Rath M, Pati B, Pattanayak BK. Mobile agent-based improved traffic control system in VANET. 2019. Available from: https://link.springer.com/chapter/ 10.1007/978-981-10-8797-4_28.
[12] Rizwan A, Karras DA, Dighriri M, Kumar J, Dixit E, Jalali A, et al. Simulation of IoT-based Vehicular Ad Hoc Networks (VANETs) for Smart Traffic Management Systems. Wireless Communications and Mobile Computing. 2022; 2022:1–11. Available from: https://doi.org/10.1155/2022/3378558.
[13] Elhoseny M, Shankar K. Energy efficient optimal routing for communication in VANETs via clustering model. 2020. Available from: https://doi.org/10.1007/978-3-030-22773-9.
[14] Qureshi KN, Abdullah AH, Kaiwartya O, Iqbal S, Butt RA, Bashir F. A Dynamic Congestion Control Scheme for safety applications in vehicular ad hoc networks. Computers & Electrical Engineering. 2018;72: 774–788. Available from: https://doi.org/10.1016/j.compeleceng.2017.12.015.
[15] Jain R. A congestion control system based on VANET for small length roads. 2018. Available from: https://doi.org/10.48550/arXiv.1801.06448.
[16] Liu X, Jaekel A. Congestion Control in V2V Safety Communication: Problem, Analysis, Approaches. Electronics. 2019;8(5):540. Available from: https://doi.org/10.3390/electronics8050540.
[17] Mohanty A, Mahapatra S, Bhanja U. Traffic congestion detection in a city using clustering techniques in VANETs. Indonesian Journal of Electrical Engineering and Computer Science. 2019;13(3):884. Available from: https://ijeecs.iaescore.com/index.php/IJEECS/article/view/13156.
[18] Sharma S, Pandey R. Accident detection, avoidance and prevention using intelligent transportation system. International Journal of Computer Applications. 2018;182(7):10–12. Available from: https://www.ijcaonline.org/archives/volume182/number7/sharma-2018-ijca-917639.pdf.
[19] Ravikumar K, Vishvaroobi T. Congestion control in vehicular ad hoc networks (VANET) using meta-heuristic techniques. International Journal of Computer Science Trends and Technology. 2017;5(4):66–72. Available from: http://www.ijcstjournal.org/volume-5/issue-4/IJCST-V5I4P12.pdf.

[20] Abdelatif S, Derdour M, Ghoualmi-Zine N, Marzak B. VANET: A novel service for predicting and disseminating vehicle traffic information. International Journal of Communication Systems. 2020;33(6): e4288. Available from: https://doi.org/10.1002/dac.4288.
[21] Mallah RA, Quintero A, Farooq B. Distributed Classification of Urban Congestion Using VANET. IEEE Transactions on Intelligent Transportation Systems. 2017;18(9):2435–2442. Available from: https://doi.org/10.48550/arXiv.1904.12685.
[22] Choe C, Ahn J, Choi J, Park D, Kim M, Ahn S. A Robust Channel Access Using Cooperative Reinforcement Learning for Congested Vehicular Networks. IEEE Access. 2020; 8: 135540–135557. Available from: https://doi.org/10.1109/ACCESS.2020.3011568.
[23] Ullah A, Yaqoob S, Imran M, Ning H. Emergency Message Dissemination Schemes Based on Congestion Avoidance in VANET and Vehicular FoG Computing. IEEE Access. 2019;7: 1570–1585. Available from: https://doi.org/10.1109/ACCESS.2018.2887075.
[24] Kothai G, Poovammal E, Dhiman G, Ramana K, Sharma A, Alzain MA, et al. A New Hybrid Deep Learning Algorithm for Prediction of Wide Traffic Congestion in Smart Cities. Wireless Communications and Mobile Computing. 2021; 2021:1–13. Available from: https://doi.org/10.1155/2021/5583874.
[25] Tang J, Yang L, Liu S, Liu W, Wang M, Wang C, et al. Caps-LSTM: A Novel Hierarchical Encrypted VPN Network Traffic Identification Using CapsNet and LSTM. Science of Cyber Security. 2021; p. 139–153. Available from: https://link.springer.com/chapter/10.1007/978-3-030-89137-4_10.