IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 14, 2021, accepted June 5, 2021, date of publication June 11, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3088474

A Multimetric Predictive ANN-Based Routing Protocol for Vehicular Ad Hoc Networks

LETICIA LEMUS CÁRDENAS[1], AHMAD MOHAMAD MEZHER[2], PABLO ANDRÉS BARBECHO BAUTISTA[3], JUAN PABLO ASTUDILLO LEÓN[2,4], AND MÓNICA AGUILAR IGARTUA[3]

[1]Centro Universitario del Norte, Universidad de Guadalajara, Guadalajara 44160, Mexico
[2]Electrical and Computer Engineering Department, University of New Brunswick, Saint John, NB E2K 5E2, Canada
[3]Department of Network Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
[4]ABREC, Coordinación de Investigación e Innovación, Quito 170528, Ecuador

Corresponding author: Mónica Aguilar Igartua (monica.aguilar@upc.edu)

This work was supported by the Spanish Government under Research Project sMArt Grid using Open Source intelligence (MAGOS) under Grant TEC2017-84197-C4-3-R. The work of Pablo Andrés Barbecho Bautista was supported by the Secretaría Nacional de Educación Superior, Ciencia y Tecnología (SENESCYT). The work of Leticia Lemus Cárdenas was supported by the Academic Coordination of the University of Guadalajara, México.

ABSTRACT Vehicular networks support intelligent transportation system (ITS) to improve drivers' safety and traffic efficiency on the road by exchanging traffic-related information between vehicles and also between vehicles and infrastructure. Routing protocols that are designed for vehicular networks should be flexible and able to adapt to the inherent dynamic network characteristics of these kind of networks. Therefore, there is a need to have effective vehicular communications, not only to make mobility more efficient but also to reduce collateral issues such as pollution and health problems. Nowadays, the use of machine learning (ML) algorithms in wireless networks are on the rise, including vehicle networks that can benefit from possible data-driven predictions. This work aims to contribute to the design of a smart ML-based routing protocol for vehicular ad hoc networks (VANETs) used to report traffic-related messages in urban environments. We propose a new ML-based forwarding algorithm to be used by the current vehicle holding a given packet to predict which vehicle within its transmission range is the best next-hop to forward that packet towards its destination. Our algorithm is based on a neural network designed from a dataset that contains data records that are captured during simulated urban scenarios. Simulation results show how our ML-based proposal improves the performance of our multimetric routing protocol for VANETs in urban scenarios in terms of packet delivery probability. The performance evaluation of MPANN shows packet losses lower than 20% (and average packet delays below 0.04 ms) for different vehicles' densities, in completely new scenarios but of similar complexity than the Barcelona scenario used to train the model. Even for much more complex scenarios (with narrow curvy streets), our proposal is able to reduce the packet losses in 20% with respect to the multimetric routing protocol as well as the average packet delays in 0.04 ms.

INDEX TERMS Multimetric routing protocol, artificial neural networks, vehicular networks.

I. INTRODUCTION

Today, researchers have access to large amounts of data that can be used to design useful services and applications to be reverted to the users who generated that data. Besides, the current available computation power makes it possible to perform a large amount of tests and experiments to come up with efficient machine learning and neural network models in a short time. In this sense, and during this work, we intend to improve vehicle communications by taking advantage of the big data generated by the vehicles themselves.

Intelligent transport systems (ITS) aim to improve operation and road safety in urban, rural, and highway environments. Nowadays, a lot of information can be collected by computational devices (e.g., sensors, cameras, on-board units) available almost everywhere. This large amount of collected data properly managed with machine learning techniques can be used to develop useful tools to improve the aforementioned ITS. Machine learning (ML) is the application of artificial intelligence to allow computers to predict

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

outcomes and behaviors automatically without the intervention of human beings. One of the main pillars of ML techniques is the ability to manipulate a large amount of data, which today can be easily done because the capacity of computational hardware has increased considerably in recent years.

In this work, we design ML models to improve routing protocols in vehicular networks. The objective is to enhance the selection process of the next-hop candidate node to forward packets towards their destination. This way, vehicles will make forwarding decisions based on predictions according to an ML algorithm. The main contributions of this paper are the following:

- We have created a representative dataset based on the collection of different traffic metrics from VANET simulations in urban scenarios. The five collected metrics that compose the dataset used to train and test our ML-based forwarding algorithm are available bandwidth, distance to destination, vehicles' density, MAC layer losses, and vehicle's trajectory.
  Notice that those metrics are gathered by the vehicles from the beacons periodically interchanged with the vehicles in their neighborhood (i.e., with vehicles within their transmission range). This way, nodes have local knowledge of the VANET, according to the decentralized nature inherent in VANETs.
- We have tested different hyperparameter configurations of the ML model to assist our forwarding algorithm to obtain the best accuracy related to an expected outcome. Specifically, we have trained our model to have the maximum possible packet delivery probability.
- Then, the configuration of the ML-based forwarding algorithm that achieves the highest accuracy will be the one implemented in our multimetric routing protocol for VANETs. Afterwards, the prediction power will be evaluated in VANET scenarios where new data will be used to assess our proposal's performance evaluation.
- At last, we include a performance evaluation to assess the benefits and costs of our approach. The proposed ML-based forwarding algorithm does not incur any additional overhead. We use just the same metrics gathered by the multimetric routing protocol through periodic beacon interchange. Consequently, our algorithm generates the same overhead as the basic multimetric routing protocol used to compare the performance of our approach. Results obtained with our novel ML-based routing protocol are better by far compared to our previous multimedia multimetric map-aware routing protocol (3MRP) [1]. The benefits are better in terms of packet losses at the cost of a slightly higher end-to-end packet delay due to the calculations performed by the ML algorithm.

A. MOTIVATION

The main motivation that led us to design the proposal presented in this work was to design a new scheme for VANET nodes to make efficient and smart forwarding decisions using ML-based techniques. Our contribution lies in the design of ML models considering the specific VANET characteristics in urban scenarios. To attain our goal, we have designed and tested different ML configurations to improve network performance in terms of average packet losses and average end-to-end packet delay. We focus our research work in urban scenarios where vehicles forward warning/reporting messages to an RSU following hop-by-hop V2V and V2I communications.

To the best of our knowledge, there is no equivalent work in the literature on improving multimetric routing protocols for VANETs where hop-by-hop forwarding decisions are based on machine learning predictions. While ML is not a new science, it is currently gaining momentum. ML models learn from past calculations to produce reliable and repeatable results and decisions. Furthermore, as these models are exposed to new data, they can be adapted independently, which can improve a wide diversity of applications.

As we will see in the related works section, some proposals implement predictive mechanisms using machine learning models. These proposals focus primarily on establishing complete data forwarding paths to destination, and they need additional infrastructure or additional signaling overhead for monitoring purposes. To carry out our proposal, we have not needed to add any additional overhead or new infrastructure, maintaining the basic hop-by-hop communication nature of vehicular ad hoc networks.

B. PRECEDENTS

This manuscript is built on the Ph.D. thesis [2] of the first author, Dr. Leticia Lemus Cárdenas, specifically on chapter 7, where she presented her proposals for multimetric predictive routing protocols for VANETs in urban scenarios. In that chapter, she explains the details of her design of ML prediction models to enhance VANET routing protocols using decision tree-based algorithms and using artificial neural networks (ANNs). Therefore, the main results shown in that chapter regarding the ANN-based proposal of a VANET routing protocol have been selected for inclusion in this manuscript, along with a more extensive performance evaluation, deeper discussion of results, and improved explanation of the ANN-based routing protocol.

C. ORGANIZATION

The remainder of the paper is organized as follows: Section II presents some related work. Then, section III describes the contents of the dataset collected from our considered urban scenario. Next, section IV presents our proposal of an ANN-based forwarding algorithm to improve multimetric routing protocols for VANETs. After that, section V presents our approach named multimetric routing protocol based on an ANN (MPANN) for VANETs. Following, section VI describes the simulation scenario where we have implemented our proposal to carry out a performance evaluation, which is shown in section VII. Finally, in section VIII conclusions and future work are presented.

II. RELATED WORKS

In this section, we present state of the art on machine learning solutions proposed to improve VANETs' performance. Our search for related proposals has been focused on works that tackle two aspects: (i) proposals whose strategies to improve the performance of VANETs are supported with ML models; (ii) works that specifically implement ML-based multimetric forwarding strategies to select a next-hop vehicle to forward messages.

(i) Regarding proposals that include ML models in their proposals, it seems clear that the topic in vehicular networks most likely to benefit from ML-based proposals is autonomous vehicles (AVs). Certainly, there are many recent proposals of ML models to assist driving in AVs. For instance, authors in [3] propose a traffic control system based on random-forest (RF) predictions to determine AVs' routes to reduce congestion rates. Their proposal is able to reduce the workload on the human experts who monitor and control AVs. In [4], the authors design and analyze deep learning architectures based on convolutional neural networks (CNN) to enhance semantic image segmentation, which is used in AVs for self-driving. They propose fully convolutional neural (FCN) models, which are trained to obtain the maximum accuracy and lowest training time, keeping precise their FCN model to segment objects.

Authors in [5] propose a ML model to predict vehicles' mobility. Specifically, they propose a centralized routing scheme with mobility predictions in VANETs assisted by a software-defined network (SDN) controller using artificial neuronal networks techniques. The idea is to use the SDN controller to estimate and predict the vehicles' arrival rate measured at road side units (RSU) or at base stations (BS). Their proposal shows benefits in terms of successful transmission probability and average packet delay in both V2V and V2I communications.

Another example is presented in [6], where the authors propose data delivery virtualization using reinforcement learning (RL). First, they evaluate each one of the one-hop link status considering the vehicles' speed, the vehicles' density in the same direction, and the average channel condition. Those metrics are used to select cluster head (CH) nodes that will act as sender nodes. Then, they design a game-theoretical multi-hop routing protocol that includes their RL model. Authors in [7] present a proposal on link prediction by implementing supervised machine learning techniques. A support vector regression is implemented to predict the vehicles' trajectory considering four mobility cases of the vehicle's speeds: high and constant speed, turn case speed, parking or intersection speed, and when the vehicle starts to move. Their proposal predicts link failures using machine learning and the vehicles' trajectory knowledge.

In [8] we can see a proposal named reliable self-adaptive routing algorithm (RSAR) to design a Q-learning-based routing protocol. The basic idea is to use a Q-learning algorithm to assess the path from each source's neighbor to destination using continuous interaction with the external environment. With their Q-learning-based algorithm, they register a table with scores (Q) of the node's neighbors, computed from measures of link-availability, link life-time, and distance to destination. Then, the neighbor with the highest Q-value is selected to forward the packet towards destination. Similarly, we find another proposal presented in [9] where the authors introduce a novel routing protocol for urban VANETs called RSU-assisted Q-learning-based Traffic-Aware Routing (QTAR). QTAR learns the road segment traffic information based on the Q-learning algorithm. In QTAR, a routing path consists of multiple dynamically selected high-reliability connection road segments that effectively enable packets to reach their destination. Simulation results show higher average packet delivery ratios and lower average end-to-end delays were obtained with respect to other traffic-aware routing proposals.

(ii) Regarding proposals that consider several metrics in their ML-based forwarding strategies, we have not found many works. Our thorough literature review reveals that an efficient routing protocol that considers crucial metrics for vehicular networks, such as trajectory, speed, delivery ratio, distance, and nodes' density, is urgently needed.

Recently, [10] proposes an algorithm to select the path from source to destination based on Decision Trees (DT) prediction. They measure the availability link duration time by observing three metrics: distance, velocity, and node direction. Its results show that DT presents significant benefits in terms of PDR, path duration time, and average hop count. Nevertheless, although it is a novel proposal based on a supervised learning machine learning model, the range of percentage in their packet delivery rate (PDR) results are 10% and 24% in the best case.

The work [11] proposes a vehicular delay tolerant network (VDTN) routing algorithm based on a bayesian network (BN) model that improves the delivery ratio with a minor forwarding overhead. The nodes in the BN model are the attributes of nodes in VDTN, which are closely related to the forwarding of messages. The selected attributes associated to each candidate forwarding node include its location, movement angle, velocity, delivery level, or ability to transmit messages successfully, among others.

Finally, [12] proposes cluster-enabled cooperative scheduling based on reinforcement learning (CCSRL) to improve the communication efficiency, and reliability of vehicular networks, with the goal of maximizing the information capacity. In particular, they leverage the stability to select a cluster head vehicle to enhance data transmission efficiency. Also, a reinforcement learning-based transmission is further designed to guarantee reliable communication among vehicles. Their proposal considers the distance (between current and possible forwarding vehicles), vehicle stability (computed from the velocity and the movement trend), bandwidth efficiency (computed from the number of data packets forwarded and the probability of successful transmission), velocity, vehicles' density (driving in the same direction as the current vehicle holding the packet), and quality of the channel condition.

The previous works propose several strategies to enhance vehicular communications. Some of them require extra infrastructure or additional signaling overhead to assist the monitoring or to transmit signaling messages. We pose that a vehicle could take a smarter decision to forward a packet if it knew in advance the probability to successfully deliver the packet at destination, regarding each candidate node to be chosen as the next forwarding hop. As far as we are concerned, there is no proposal in the literature that includes a ML-based forwarding algorithm to assist a multimetric routing protocol for vehicular networks that predicts, for each candidate forwarding vehicle, which is the packet delivery ratio at destination. In the following sections, we describe our proposal to choose the best forwarding node in VANETs based on a novel ML-based prediction algorithm.

III. DATASET COLLECTED FROM MULTIMETRIC MEASURES IN VEHICULAR URBAN SCENARIOS

In any ML-based project, the first step prior to develop any model is to collect data, process, and arrange it properly according to the ML model requirements. Furthermore, building a tagged dataset is not easy in practice, as it strongly depends on the type of data to be observed. Most of the available datasets regarding vehicular networks are those related to mobility patterns. Unfortunately, there are not available datasets that deal with data packet flows in VANETs. In consequence, it was firstly necessary to gather V2V and V2I information from a realistic urban scenario designed in our simulation framework built using OMNeT++ [13], Veins [14], SUMO [15] and OpenStreetMap (OSM) [16]. Specifically, our goal was to obtain the probability of successful packet delivery under a wide representative number of different characteristics of the simulation environment.

To achieve our goal, we have carried out a large set of VANET simulations over a representative area of Barcelona, Spain, see Fig. 1. We selected a general and representative area of Barcelona that includes wide and narrow streets as well as an avenue. From each simulation, we have collected the values of five metrics:

- **Available bandwidth**, in the link formed by the vehicle currently holding the packet and each candidate node in the neighborhood (i.e., vehicles within transmission range) to be the next forwarding hop.
- **Vehicle's density**. It is the number of neighbors within the transmission range measured at each candidate as the next forwarding vehicle.
- **Distance to destination**. Distance computed from each neighbor to the packet's destination.
- **Vehicle's trajectory**. It informs about the moving direction and speed towards destination, computed for each candidate node in the neighborhood to be the next forwarding vehicle.

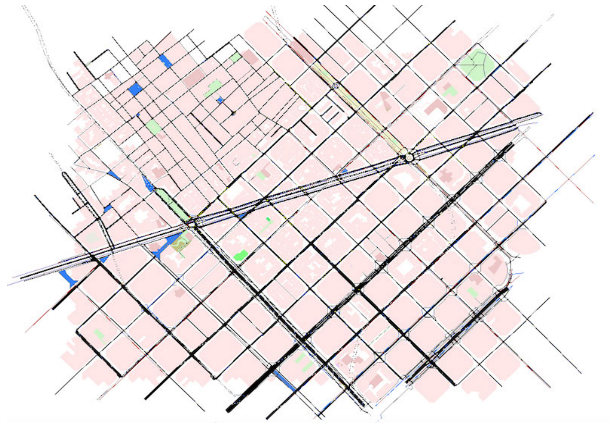


FIGURE 1. Eixample/Gracia district in Barcelona city, Spain. Area = 2300 m × 2100 m. Map extracted from OpenStreetMap [16].

- **MAC losses**. The average percentage of packet losses computed in the MAC layer of the link formed by the current vehicle holding the packet and each one of its neighboring vehicles within transmission range.

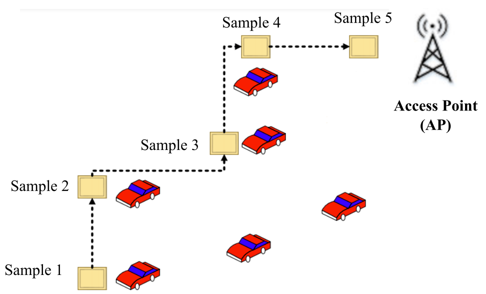
We calculate the five routing metrics listed above from a set of values carried in the beacons that are periodically interchanged by vehicles. Those values are (i) the (v_x, v_y) velocity coordinates of the vehicle, (ii) the MAC layer losses, (iii) the vehicles' density, (iv) the vehicle's location and the (v) T_{idle} (% of time during which the link is not being used). The details are explained in Section V.

In the considered scenario, reporting messages (e.g., regarding traffic statistics or a traffic incident) are forwarded hop-by-hop from a source vehicle to an access point (AP) (see Fig. 2). The AP is located near the right top corner of the map. Vehicles periodically send traffic reporting messages to the AP. This AP's location guarantees that there will be a wide range of possible results collected in the dataset, both from near and far vehicles.

We have designed a multimetric ML-based forwarding algorithm to be included in our multimetric routing protocol for VANETs. The general objective is that vehicles can make the best forwarding decision for each packet transmitted hop-by-hop towards its destination.

To train, test, and validate our ML-based forwarding algorithm, we have collected those five metrics (see columns x_1 to x_5 in Fig. 3) from the already established periodic exchange of beacon messages between neighbor vehicles. Besides, we have also collected if the packet was successfully delivered or not (see column y in Fig. 3). Therefore, to build a representative dataset, it is necessary to get a wide range of results for all possible values in the five considered metrics (see the list above). This way, we relate the five routing metrics, which are the inputs of our ML-based predictive forwarding algorithm to assist routing protocols in VANETs, to the output value of the model (probability of packet successfully delivered), see Fig. 3.

To achieve our goal, we have considered different possible configuration settings in our VANET scenario. Among



Sample	idPacket	X_1 Available bandwidth	X_2 Vehicles' density	X_3 Vehicle trajectory	X_4 MAC layer losses	X_5 Distance to destination
1	1	$X_{1,1}$	$X_{2,1}$	$X_{3,1}$	$X_{4,1}$	$X_{5,1}$
2	1	$X_{1,2}$	$X_{2,2}$	$X_{3,2}$	$X_{4,2}$	$X_{5,2}$
3	1	$X_{1,3}$	$X_{2,3}$	$X_{3,3}$	$X_{4,3}$	$X_{5,3}$
4	1	$X_{1,4}$	$X_{2,4}$	$X_{3,4}$	$X_{4,4}$	$X_{5,4}$
5	1	$X_{1,5}$	$X_{2,5}$	$X_{3,5}$	$X_{4,5}$	$X_{5,5}$

FIGURE 2. Data collection example from our OMNeT++/Veins/SUMO/OSM simulation scenario.

these configuration features, we highlight the following for their notably impact in the performance of the routing protocols: (i) the vehicles' density, ranging from low (20 vehicles/km^2) to high values (300 vehicles/km^2); (ii) the average vehicles' speed in urban roads (20 to 100 km/h); (iii) different positions of the source nodes in the map area. This way, we will have a complete representative dataset that covers the maximum number of possible values in the five considered metrics (available bandwidth, vehicles' density, distance to destination, vehicles' trajectory, and MAC losses). The data collection process requires the following steps:

- Extract one-hop values of the metrics. Obtain the corresponding output value, i.e., the percentage of packets successfully delivered at destination, for each case.
- The dataset is organized in five columns with the values for the five metrics ($x_1, \ldots x_5$) as inputs, and another column with the output values $Y(0, 1)$. This output value means whether the message was received (1) or not (0) at destination.
- Our dataset consists of 6000 records. This dataset size has shown to be enough to get good predictive results in the performance of our proposed routing protocol MPANN, as section VII shows.
- Once we have prepared our dataset, the machine learning models can be trained and tested. The criteria used to determine whether enough data has been collected or not is basically the values of accuracy (training and testing) obtained by the ML-based forwarding decision models. We have created a large number of scenarios with different configuration settings to fully cover the range of values for each routing metric.

x_1 Available bandwidth	x_2 Vehicular density	x_3 Vehicle trajectory	x_4 MAC layer losses	x_5 Distance to destination	y Packet correctly received (1) or not (0)
0.528586	0.198975	0.021809	0.909091	0.065176	0
0.539025	0.198975	0.021089	0.909091	0.02248	1
⋮	⋮	⋮	⋮	⋮	⋮
0.988627	0.367975	0.424266	1	0.340894	1

Samples
Features
Labeled data

FIGURE 3. Appearance of our dataset. Features and label for the training and testing phases in our ML model.

- Finally, the designed ML-based forwarding algorithm is implemented in our multimetric routing protocol for VANETs. The five routing metrics will be the inputs of the ML model, while the output will be the prediction of the packet successfully delivered or not at destination. This is checked by the vehicle currently holding the packet for each candidate node to be the next forwarding node (see section V-A). The operation of the multimetric routing protocol is described in section V.

We have normalized the collected values for each metric. This is necessary so that the algorithms can operate with them under the same range of values for each metric so that algorithms can fairly evaluate them together. The resulting dataset has the format shown in Figure 3. As additional information, to obtain our dataset (which has 6000 registers), we dedicated about 840 hours of human work and about 3600 hours of simulation using a personal computer (i9, 3,3GHz, 64GB RAM).

Our ML problem is clearly a binary classification problem because the expected outcome has two possible values (0 and 1). Therefore, artificial neural network (ANN) models are suitable options to tackle our goal since they can handle binary data very well, and they have shown an accurate performance in the literature [17]–[19]. Then, the next step is to train ANN classification algorithms and evaluate them through the usual machine learning performance metrics [19], [20]. Finally, the designed ML-based forwarding algorithm will be implemented in our proposal of VANET routing protocol, see section V. Our ML-based multimetric routing protocol will be evaluated with our OMNeT++/Veins/SUMO/OSM simulation framework, see section VI.

In the following sections, the training, validation testing, and implementation phases of our ML-based forwarding algorithm for VANETs are described.

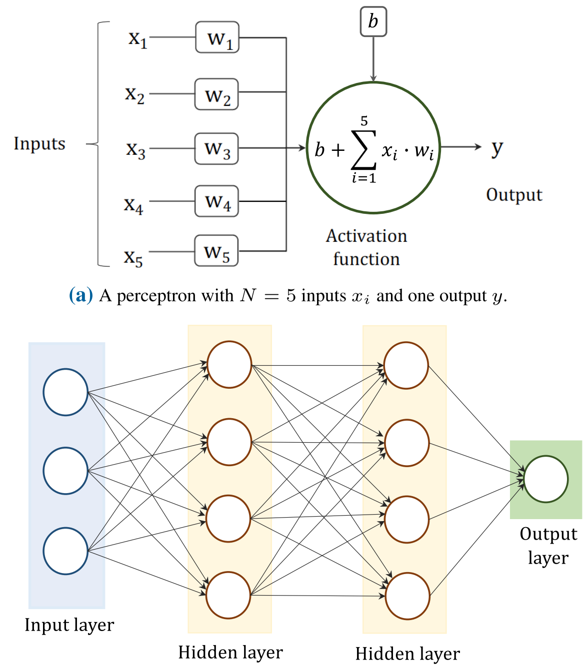
IV. DESIGN OF A FORWARDING ALGORITHM BASED ON AN ARTIFICIAL NEURAL NETWORK

In this section, we summarize the process for training our artificial neural network (ANN) model. The resulting model will be used in our ML-based forwarding algorithm to enhance multimetric routing protocols for VANETs. In [2] we conducted an extensive analysis of various ML models designed for our purpose, with ANN showing the best performance in our scenarios.

In this work, we have trained different models over ANNs with different ANN dimensions (i.e., different number of hidden layers and neurons) to choose the best classifier for our dataset. Our dataset has been collected from a large amount of representative simulations for VANETs in realistic urban scenarios, as was explained in section III. The best classifier has been chosen to balance the trade-off between accuracy, prediction power, ease implementation, and complexity. Our goal is twofold:

- To design an ML-based forwarding algorithm that learns from our representative dataset, see section III. The goal of the ML-based forwarding algorithm is to predict if the packet will be successfully delivered or not for each next-hop candidate node.
- To integrate our ML-based forwarding algorithm in our previously proposed multimedia multimetric map-aware routing protocol (3MRP) [1] for VANETs, as it is explained in section V

After finishing the design and analysis of our ML-based multimetric routing protocol for VANETs, we implemented it in the OMNeT++/Veins/SUMO/OSM simulation framework to carry out a performance evaluation of our proposal in realistic urban scenarios, as it is shown in section VII.



(a) A perceptron with $N = 5$ inputs x_i and one output y.

(b) Feed-forward fully connected neural network.

FIGURE 4. General scheme of an artificial neural network (ANN).

A. PREDICTION MODEL BASED ON A MULTI-LAYER FEED-FORWARD NEURAL NETWORK

Machine learning (ML) defines algorithms that parse data and learn from the dataset to build a model with which we can later predict results. Artificial neural networks (ANN) is a subset of machine learning algorithms classified as deep learning models, which are designed to analyze data with a logic structure like how we humans would draw conclusions [20].

ANNs belong to the set of supervised learning algorithms and are implemented to solve, among others, classification problems. This feature suits well with our problem, where we have a vehicle currently holding a packet that must classify its neighboring vehicles in transmission range to select the best next-hop forwarding vehicle. We have used our collected dataset to generate an ANN model. The ANN model was designed based on the highest accuracy value obtained in the training and testing phases. Next, we describe the steps followed to design our ANN-based forwarding model [21], [22].

1) ANN ARCHITECTURE DESIGN

The ANN model selected corresponds to a feed-forward neural network with unicycle nodes and hidden layers that are fully connected [22]. In this way, each neuron receives input from all neurons from the previous layer, see Fig.4. The *perceptron* is a mathematical model of a biological neuron, see Fig.4a. An ANN is composed of different layers, and how they are configured is related to the features and labels defined in the training dataset. Firstly, the set of *features* (x_i) are our independent variables that form the first layer (*input layer* in Fig.4b). Secondly, the output dependent variable (Y) corresponds to the *output layer*. Finally, the *hidden layers* are not related to the features nor the labels of the training dataset. Therefore, the values of the hyperparameters (i.e., number of hidden layers and number of neurons) will be adjusted. The goal here is to tune those parameters to achieve the highest accuracy for classification problem while keeping a simple structure. We have tested several ANN configurations before obtaining the final design of our ANN model.

Besides, an *activation function* and an *error function* must also be selected. There are different activation functions among which nonlinear functions, such as sigmoid [23] and ReLU [24], are commonly used and widely applied. In particular, ReLU functions are simpler and faster to process. Note that, in a neural network, the hidden layers may require a high number of neurons, where each one must execute an activation function. Therefore, a faster activation function is convenient in the neurons that compose the input and hidden layers.

For the first and hidden layers, we have chosen the well-established Rectified Linear Activation (ReLU) function for, among other characteristics, its computational simplicity and linear behavior [42], [43]. Finally, the sigmoid activation function has been used in the output layer.

The learning process of the ANN designed is described in the following two steps:

- Backpropagation. It corresponds to random output prediction, comparing those predictions with the true output. Thus, *weights* and *bias* (w_i and b in Fig. (4a), respectively) are updated until the predicted output comes closer to the true output. This updating process is executed based on the error function, also named cost function. In our case, we have used the mean square error (MSE), see Eq. (1) [25], as the cost function. This error function measures the difference between the true values (y_i) and the propagated values ($\hat{y}_i$).

$$MSE = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1}$$

The objective is to minimize the MSE. Hence, the smaller the MSE, the more accurate the predictions are. That is, every time the error is calculated, it is required that weight values in the intermediate layers be recalculated as many times as necessary until the error be minimum, tending to zero if possible. To do so, it is needed to use an adjusting function to update the weight values of the intermediate layers. Thus, those values are updated at each epoch of the training stage. The Adam optimizer [26] is an adaptive learning rate optimization algorithm that was designed specifically to train deep neural networks. This algorithm is commonly used for its fast efficiency in deep learning compared to other classical stochastic gradient descent algorithms [27].

- Hyperparameter selection. As already mentioned, our neural network topology is based on feed-forwarding learning, where the layers are fully connected. The characteristics of the input and output layers are based on the features and classes of the collected dataset, respectively, see III. Two important parameters that define the network dimension are: (i) the number of neurons (per hidden layer); and (ii) the number of hidden layers. We have performed different ANN configurations varying the number of hidden neurons and hidden layers (referred to as hyperparameters) to find the best configuration that balances the trade-off between validation metrics and complexity level. In this sense, the validation accuracy and validation loss metrics were calculated for each configuration of number of hidden neurons and hidden layers.

We have computed the validation metric (i.e., the accuracy) of the combination of (1, 2, 3, 5, 10) hidden layers with (5, 10, 32, 64, 128, 256) neurons at each hidden layer. Fig. 5 shows the accuracy of the three best results that we have obtained, including corresponding hyperparameter settings. As it can be seen, the three hyperparameter combinations show similar values of accuracy. However, the most simple model (3 hidden layers, 128 neurons per hidden layer) has the highest accuracy (84%). Note that a more simple neural model means a lower computational cost in terms of number of operations and activation functions in each layer. Therefore, we have selected the configuration model with 3 hidden layers and 128 neurons per hidden layer.

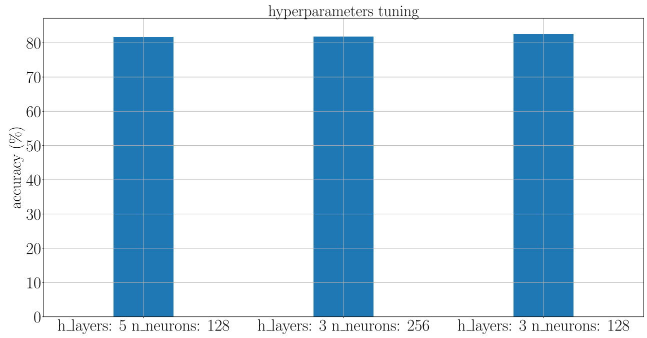


FIGURE 5. Accuracy obtained in our ANN model for different hyperparameter configurations: number of hidden layers and number of neurons per hidden layer.

2) TRAINING PHASE

For the training process of any machine learning algorithm, we have to provide a set of *features* (inputs, five in our case) and one or more *labeled classes* (outputs, one in our case). Note that an important point to take into account is how long (*number of epochs*) it is necessary to train the model. The goal is to train the model long enough to be able to learn the mapping from inputs to outputs, while at the same time, we should avoid overfitting of training data.

Overfitting causes poor performance of ML algorithms. Overfitting refers to a model that models the training data too well to the extent that it negatively impacts the ability of the model to generalize on new data. Therefore, it is important to detect during the training phase of the model at what point the model falls into overfitting. A good practice is to validate the error calculation during the training process by using some validation technique. To avoid overfitting, we have used the *early-stopping validation technique* [28], [29]. The goal is to detect when overfitting starts during supervised training of the model; training is then stopped before convergence to avoid overfitting (early stopping). Usually, the original training dataset is divided into a training dataset and a validation set. Also, the number of training cycles (*epochs*) that the model will repeat through all training samples will be configured.

Fig. 6 shows the evolution of the error function (MSE) throughout the epochs in our ANN. Here we can see the training error (blue line) and the validation error (orange line) around each training cycle (epoch). In training, the error value decreases as the number of repetitions increases (see blue line). However, its real performance to unseen data (see orange line) converges and cannot decrease any more. As it can be seen, the validation error converges after 200 epochs. It indicates that the model can no longer accurately predict beyond that number of repetitions. Therefore, we are overfitting the model for epoch>200.

By using the early-stop technique, the training will stop when the validation loss does not decrease anymore to

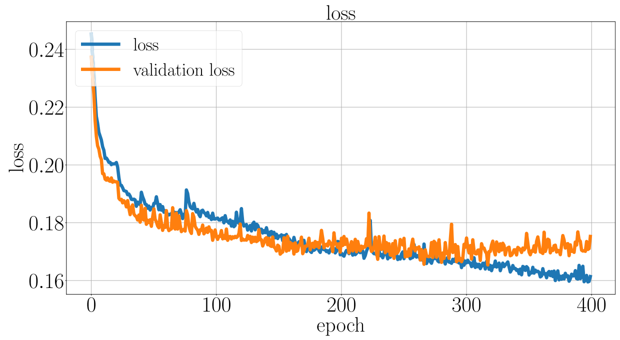


FIGURE 6. An example of overfitting detection during the training phase. Suitable number of epochs = 200, to avoid overfitting.

avoid overfitting the model. Thereby, we obtain a sufficient number of epochs that our model needs to reach the minimum possible MSE till convergence. The building, training, and validation of our ANN model have been implemented using the Keras [30] tool. We have obtained the final ANN model with this tool, ready to be implemented in our OMNeT++/Veins/SUMO/OSM simulation framework.

The configuration parameters used in the training phase are described in Table. 1. This table shows that we have used a dataset split ratio of 80% for training, 10% for validation, and 10% for testing. For further details on the design of our ANN model, we refer the reader to [2].

TABLE 1. Set of configurations parameters for the ANN modeling, training, validation and testing phases.

	Parameter	Value
Model architecture	Neural network type	Feed-forward
	Layer type	Dense
	Input and intermediate activation function	ReLU
	Output activation function	Sigmoid
	Number of neurons per hidden layer	128
	No. of input layers	1
	No. of hidden layers	3
	No. of output layers	1
Learning process	Loss function	Mean square error (MSE) Eq. 1
	Optimizer	Adaptive moment estimation (Adam) [26]
	Evaluation metric	Accuracy
Training model	Training split	80% of the dataset
	Validation split	10% of the dataset
	Testing split	10% of the dataset
	Number of epochs	200
	Validation method	Early stopping

3) VALIDATION PHASE

To validate the final design of our ANN model, we have obtained the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC), shown in Fig. 7. The ROC curve plots the relation between true positive rate (TPR) and false positive rate (FPR). The AUC is a parameter that tells us how much the model can distinguish between classes, i.e., two classes in our case ("1" means packet successfully delivered and "0" means the contrary). The AUC evaluates the goodness of a model. It takes values between 0.5 (bad model with no class separation capacity) and 1 (excellent model with a good measure of separability). Thereby, the test with the largest area under the curve is preferable. In our case, the AUC has a value of 0.887, which means that there is an 88.7% probability that the diagnostic made to predict a true positive value is more correct than that of a randomly chosen false positive. Note that the ROC represented in Fig. 7 corresponds to the ANN configuration that obtained the highest accuracy value in the training phase, see section IV-A2.

Concluding, we have designed an ANN model for our collected dataset, see section III. The ANN has 1 input layer, 3 hidden layers, 1 output layer (i.e., ANN with 1/3/1) and 128 neurons at each hidden layer. Fig. 8 shows the scheme of the final ANN architecture that we have designed following the previous steps. In Fig. 8, we can see the following information:

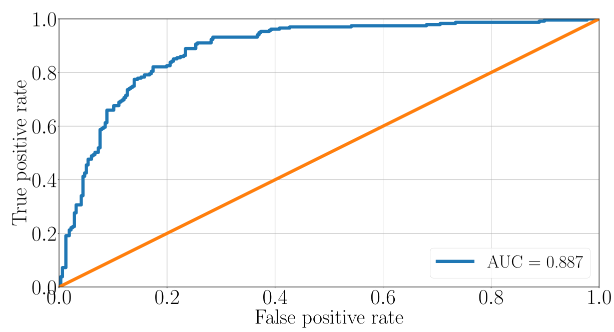


FIGURE 7. ROC curve and area under the ROC curve (AUC) of our ANN model once trained.

- S: The output of our ANN. We have 2 classes: "1" means packet successfully delivered at destination and "0" means the contrary.
- i: It represents the input number, $1 \leq i \leq 5$. In our case we have five features: distance to destination, trajectory, vehicles' density, available bandwidth and MAC losses.
- j: It represents the neuron number, $1 \leq j \leq 128$.
- k: It represents the layer number, $1 \leq k \leq 5$ (1 input layer, 3 hidden layers and 1 output layer).
- x_i: They represent the inputs to the first layer (input layer): distance to destination, trajectory, vehicles' density, available bandwidth and MAC losses.
- $w_{i,j}^{(k)}$: Weights used in the ANN model. They represent the connections between layers and show the relevance

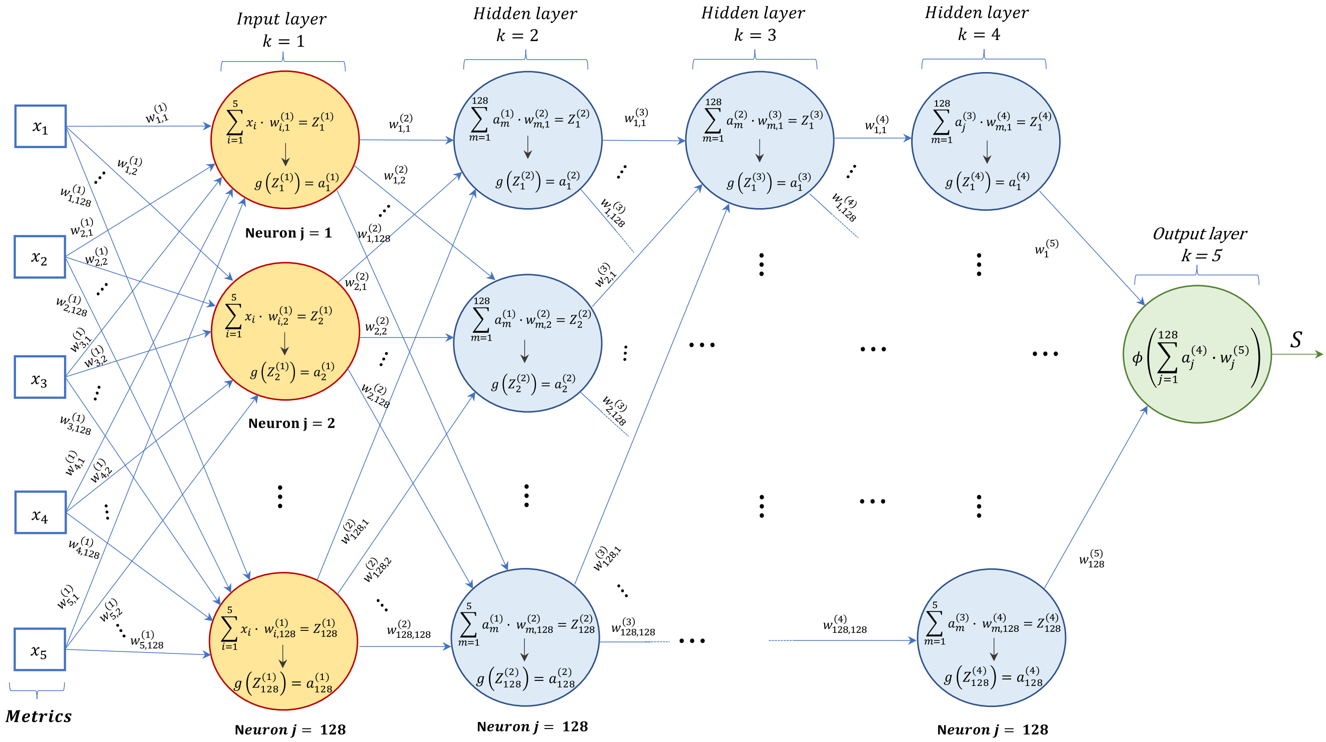


FIGURE 8. Scheme of our designed ANN model to be included in the forwarding algorithm of VANET routing protocols.

of a particular neuron. $w_{i,j}^{(k)}$ is the weight correspondent to input i, neuron j, layer k.

- $\sum_{i=1}^{5} x_i \cdot w_{i,j}^{(1)} = Z_j^{(1)}$, $1 \leq j \leq 128$: It is the sum of weighted inputs at the input layer.
- $\sum_{m=1}^{128} a_m^{(k-1)} \cdot w_{m,j}^{(k)} = Z_j^{(k)}$, $1 \leq j \leq 128$, $2 \leq k \leq 4$: It is the sum of weighted inputs at the hidden layers.
- $g(z_i^k) = a_j^{(k)}$: They represent the activation functions at the hidden layers. Also, we have used the ReLu function, as is recommended in the hidden layers due to its good performance and fast operation.
- $\phi(\sum_{j=1}^{128} a_j^{(4)} \cdot w_j^{(5)})$, $1 \leq j \leq 128$, $1 \leq k \leq 4$: It represents the activation function at the output layer. We have used the Sigmoid function [23].

Our ANN-based forwarding algorithm will assist the node in the vehicular network currently holding the packet to make the best forwarding decision. That is, which is the best next-hop vehicle to forward the packet towards its destination. Therefore, our ANN-based forwarding algorithm has been designed to maximize the probability of successfully delivering the packet at its destination.

So far, we have finished with the design of our ANN-based forwarding algorithm. The next step is to include our ANN model in the forwarding algorithm to assist vehicles to make the best forwarding decision. Specifically, we will include our ANN-based forwarding algorithm in our previous proposal of VANET routing protocol named multimedia multimetric map-aware routing protocol (3MRP) [1], as it is explained in section V.

V. MULTIMETRIC ROUTING PROTOCOL BASED ON AN ARTIFICIAL NEURAL NETWORK (MPANN) FOR VEHICULAR AD HOC NETWORKS

Once we have designed and validated our ANN-based forwarding algorithm to help vehicles make the best forwarding decision, we have to implement it in a routing protocol for VANETs. We have chosen our previous proposal named multimedia multi-metric map-aware routing protocol (3MRP), which shown to improve the performance compared to other proposals in terms of average percentage of packet losses and average end-to-end packet delay [1]. 3MRP uses five routing metrics: available bandwidth, distance to destination, vehicles' density, MAC layer losses and vehicle's trajectory. Vehicles periodically update those metrics regarding their neighbors in transmission range using the periodical interchange of beacons. Then, the vehicle currently holding a packet computes a multimetric score for each candidate to be the next-hop forwarding vehicle. The neighbor with the highest multimetric score is chosen to forward the packet in the next hop, and this is repeated hop-by-hop to forward the packet towards its destination.

A. MPANN ROUTING ALGORITHM

In this work, we substitute the forwarding criteria of 3MRP by our ML-based forwarding algorithm, which was introduced in section IV-A1. Our proposal will assist the current vehicle holding the packet to take the best forwarding decision so that it chooses the neighbor that gives the highest chance to successfully deliver the packet to destination. Our approach is called **multimetric routing protocol based on an artificial neural network (MPANN)** for VANETs.

The node currently holding the packet computes the five normalized metrics for each neighbor within transmission range [2]. Vehicles transmit three values (vehicle's velocity, MAC layer losses and vehicles' density) in their beacons periodically interchanged with their neighbors. The calculation of the five routing metrics (distance to destination, trajectory, vehicles' density, available bandwidth and MAC losses) computed from those values carried in the beacons, is explained in [31] and [1]. In the following, we will just summarize the process.

- The basic **distance** metric refers to the distance from each next forwarding candidate node to destination, where lower distances are preferred [2].
- The **trajectory** is defined as the prediction of the future node's position, whose calculation allows the source node to know if the candidate vehicle will be closer or going away from destination. For this case, nodes that move towards destination are preferable [1].
- The **available bandwidth** considers the idle time of the wireless link formed between the current vehicle holding the packet and each candidate next-hop node [31].
- **Vehicle density** is the number of vehicle registered in the neighbor's list. Nodes with a higher vehicle density value are better rated, up to a limit above which the vehicle density is too high to increase collisions [1].
- **MAC layer losses** are computed locally at the node itself, and it is used as a kind of local network feedback. A node with lower losses rate in the MAC layer, means that it is a better candidate to forward the message, so it will be better scored in that metric [31] [32].

Afterwards, the vehicle uses our ML-based forwarding algorithm to predict the output (i.e., the packet will be delivered or not at destination) for each neighbor. With this information, the node will arrange its neighbors' list and will make the best forwarding decision, as we will explain next.

Algorithm 1 describes the sorting process of the neighbors list applying our designed ANN model. First of all, a variable i is initialized with $i = 1$, and it will be incremented until it reaches a value equal to the size of the neighbors list (N). Two pointers are pointing to the two first locations of the neighbors list (lines 2 and 3). Then, the five metrics fields of the current and the previous node in the list are taken as inputs to the ANN model and we obtain their corresponding outputs val_1 and val_2, respectively (see lines 6 and 7). Afterwards, the outputs val_1 and val_2 are compared and we put the neighbor with the highest value at the top of the list. The process is repeated till the end of the list. Once the list is arranged, the node currently sending the packet can find the best next forwarding node at the top of the list. Recall that if at the

Algorithm 1 Sorting Neighbor List Using Our ANN Designed Model (Fig.8) to Select the Best Candidate Node in the Forwarding Message Process

```
   Requirement: neighbors list, the five metrics as ANN
                models' inputs, ANN model
   Set up      : i = 1; curr = head;
1  while i <= N do
2      tmp = curr;
3      curr = curr → next;
4      read current;
5      if (curr! = NULL) then
6          curr (available bandwidth, density of neighbors,
           trajectory of node, distance to destination and
           MAC losses) → ANN model = val1;
7          tmp (available bandwidth, density of
           neighbors,trajectory of node, distance to
           destination and MAC losses) → ANN model=
           val2;
8          if (val1 < val2) then
9              head=tmp;
10             tmp=curr;
11             curr=head;
12         end
13     end
14     i++;
15 end
```

top of the list we find values close to 1, this means that the node currently holding the packet has good candidate nodes to choose the next forwarding node of the packet.

Notice that MPANN does not incur in any additional overhead compared to 3MRP, as it is shown in Section VII. The reason is that the ANN model included in the MPANN forwarding algorithm uses the same five routing metrics than 3MRP as input features.

VI. DESCRIPTION OF THE SIMULATION SCENARIOS

We have configured a simulation scenario in our OMNeT++/Veins/SUMO/OSM simulation platform to carry out a performance evaluation of the MPANN proposal presented in this work. We assume that vehicles know their position and have road map information. We also assume that nodes periodically interchange geographic coordinates, vehicles' density, and vehicle's speed with each neighbor node within their transmission range, using hello messages (beacons) sent once per second. When a node receives a hello message, it updates its neighbors' table. Vehicles calculate and normalize a set of five routing metrics regarding each neighbor, see Section III. The five considered metrics are available bandwidth, vehicles' density, distance to destination, vehicle's trajectory, and MAC layer losses.

In the simulation scenarios, there is an access point (AP) located near the upper right corner of the map. Vehicles periodically send traffic report messages to the AP.

FIGURE 9. **Urban area of Berlin city. Map extracted from OpenStreetMaps (OSM) [16]. Simulation area 2500 m × 2500 m.**

Once we have implemented the ML-based forwarding algorithm (described in Section IV) in the MPANN (described in Section V) routing protocol, we have evaluated its performance by means of two simulation tests. All the city maps have been generated from the OpenStreetMaps (OSM) [16].

We have organized our simulations in two groups:

Test 1. The first test corresponds to the performance evaluation of MPANN in the same scenario from which the dataset to train and validate the ANN forwarding model was collected (see Fig. 1). This is a general urban area of Barcelona city with a usual oreography for an urban scenario, i.e., it has wide, narrow, regular, and irregular streets. The simulation area is of 2300 m × 100 m. We compare the performance of MPANN to the well-known GPSR [33] as a reference and our former proposal 3MRP [1].

Test 2. In a second test, we evaluate our proposal MPANN in other urban scenarios different than the scenario used to train the designed ML-based forwarding model. We will see how adaptable our approach is to new scenarios with unseen data and evaluate how meaningful and generalized our dataset is. This way, we can see how flexible our designed ML-based forwarding model is and assess if it can perform well in other scenarios different from the one used to train the model. For that, we have configured two new city maps with different characteristics in our simulation framework:

a. An urban area of Berlin city, see Fig. 9. It is similar in size as the Barcelona scenario used to generate the dataset (see Fig. 1), but with a more complex oreography of the streets. Simulation area is of 2500 m × 2500 m.

b. An urban area of Rome city, see Fig. 10. The area is larger than the Barcelona scenario, with a more complex oreography of the scenario, with a lot of irregular streets. Simulation area is 3500 m ×2400 m.

The simulation settings of the tests carried out in Barcelona, Berlin and Rome, are depicted in Table 2.

FIGURE 10. **Urban area of Rome city. Map extracted from OpenStreetMaps (OSM) [16]. Simulation area 3500 m × 2400 m.**

TABLE 2. **Simulation parameters used to evaluate the performance of the ML-based forwarding-decision models.**

Parameters	Values description
Vehicles' density	50, 100, 150, 200 vehicles/km^2
Simulation area:	2300 m × 2100 m (Barcelona), 2500 m × 2500 m (Berlin) and 3500 m x 2400 m (Rome)
Transmission range:	340 m
Data transmission rate	6 Mbps
Source type	Constant bit rate (CBR)
Reporting message size:	259 Bytes
Beacon size:	155 Bytes
Bandwidth	10 Mbps
MAC standard	IEEE 802.11p
Routing protocols	GPSR [33], 3MRP [1], MPANN (see section V)
Simulation time	100 s
Runs (95% confidence intervals)	5 repetitions per point
Simulation tools	OMNeT++ [13] /Veins [14] /Sumo [15]/OSM [16]

In Section VII we will show results for four vehicles' densities, ranging from very sparse VANETs (50 vehicles/km^2) where it is hard to find vehicles around, and connectivity is low, to very congested VANETs (200 vehicles/km^2) where the chance of packet collision is high. In this way, we model different VANET scenarios, including cases with (i) low traffic as we can see during the night or the weekend, (ii) normal traffic flows that happen during weekdays, and also (iii) congested situations during traffic jams.

VII. PERFORMANCE EVALUATION OF MPANN UNDER DIFFERENT URBAN SCENARIOS

In this section, we will carry out a performance evaluation of our proposal MPANN under the different VANET scenarios described in section VI. The performance evaluation of each considered routing protocol is done in terms of average percentage of packet losses and average end-to-end packet delay. In addition, we have also measured the run-time to perform

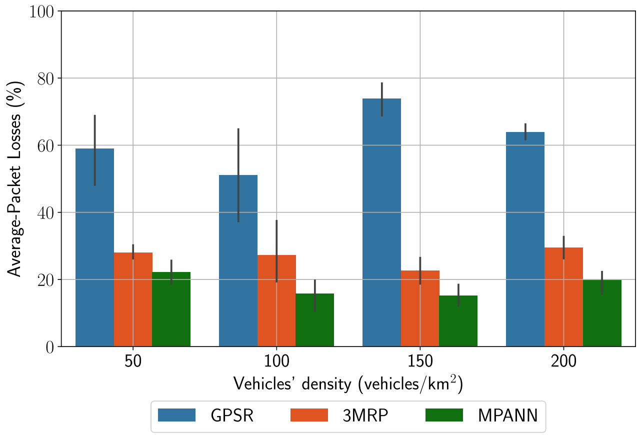


FIGURE 11. **Average percentage of packet losses for different vehicles' densities (5 repetitions/point, 95% CI).**

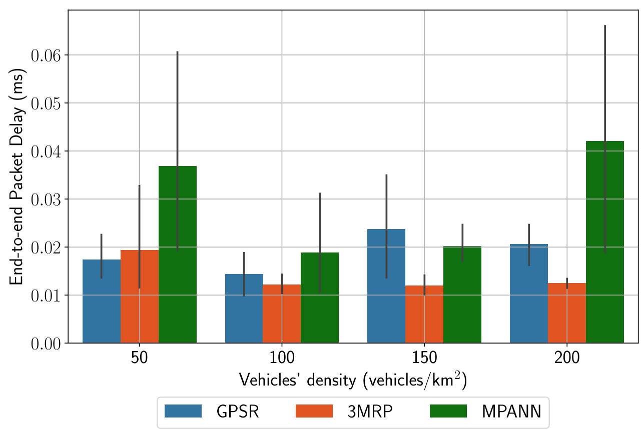


FIGURE 12. **Average end-to-end packet delay for different vehicles' densities (5 repetitions/point, 95% CI).**

the simulations, the computational cost (in terms of time), and the overhead. In this way, we will be able to analyze the pros and cons of our proposals, weighing the benefits and costs incurred. In the following, we show the simulation results of the two tests described in Section VI.

A. TEST 1: PERFORMANCE EVALUATION IN THE SAME SCENARIO USED TO TRAIN THE ANN MODEL. THE BARCELONA SCENARIO

This set of simulations were carried out in the same simulation scenario (see Fig. 1) used to collect the dataset with which we trained the ANN-based forwarding model. Of course, now we have used different simulation seeds, so the simulations are different, although the simulation settings (map, area, street layout ...) are the same.

1) EVALUATION OF THE AVERAGE PERCENTAGE OF PACKET LOSSES AND THE AVERAGE END-TO-END PACKET DELAY

Fig. 11 shows the average percentage of packet losses for the three routing protocols for VANETs analyzed: GPSR [33], 3MRP [1], and MPANN (sec. IV-A). We can see that GPSR, which uses distance to destination as single routing metric, in general offers the highest packet losses, around 62% to 80%. We can see that 3MRP [1] shows notably better results than GPSR, with packet losses around 22% to 30%. MPANN shows a significant improvement is obtained for all vehicles' densities. With MPANN, the average packet losses are below 21% for all vehicles' densities, and MPANN is able to outperform 3MRP for all vehicles' densities, around 18% to 36%. Thus, although 3MRP and MPANN use the same five routing metrics to make the forwarding decisions, the ML-based forwarding algorithm included in MPANN helps to better choose the next forwarding vehicle. Notice that the ANN forwarding algorithm included in our proposal MPANN predicts which candidates will be able to deliver the packet at destination, as it was explained in Section IV-A.

Fig. 12 shows the average end-to-end packet delay for the same evaluated routing protocols for diverse vehicles' densities. We can see that the highest delays correspond to our routing protocol MPANN. The main reason is that since MPANN is able to lose less packets than 3MRP, many of those packets followed longer paths and got to destination, consequently increasing the average end-to-end packet delay. Notice that the ML-based predictive models included in MPANN do not consume any significant amount of time, as it is shown in section VII-A3. Nevertheless, that increment in delay experience with MPANN is very small (maximum 40 ms) compared to 3MRP. This small increase in delay pays off given the notably reduction in packet losses with MPANN.

To further assess the pros and cons of our proposals, we show in the following subsections the evaluation of the three routing protocols in terms of running time and computational cost.

2) EVALUATION OF THE SIMULATION RUNNING TIME

The running time represents the time that each routing algorithm needs to complete a running simulation for a specific configuration of the simulation settings, i.e., time to obtain a point in each figure. OMNeT++ [13] is an event-based network simulator, and the time spent for each simulation depends on the number of participating nodes. For each node that forwards the message, a set of operations must be executed. The running time increases as the number of nodes increases. Also, the running time increases as the network area increases since longer forwarding paths are potentially needed to reach destination (number of hops potentially would increase).

Fig. 13 shows the running time taken for each routing protocol to perform 100 simulation seconds, i.e., simulation time to obtain a single point in the figures of the performance evaluation. In that figure, we show the average of 10 independent simulations, including a CI of 95%. For this experiment, we have considered an intermediate vehicular density of 100 $vehicle/km^2$. Simulations were carried out in a server with the features presented in Table 3.

Fig. 13 shows that GPSR [33] experimented the highest running time, about 12 hours per point on average. This protocol selects the forwarding nodes based only on the

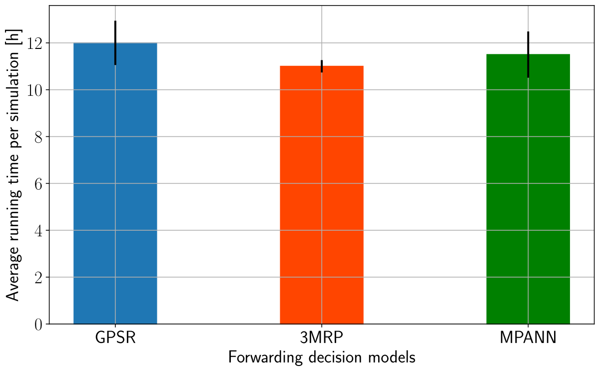


FIGURE 13. Computer running-time consumed to finish a 100 sec-simulation in OMNeT++/Veins/SUMO/OSM. Vehicles' density = 100 vehicles/km^2, 95% CI.

TABLE 3. Hardware features of the server used to run the simulations needed to obtain Fig. 13.

Server	Ubuntu 16.4 LTS
RAM	251.4 GiB
Processor	Intel® Xeon(R) Gold 6138
Base clock frequency	2.00 GHz × 47
OS type	64 bits

distance metric, and when it switches to the perimeter routing mode, the delay can notably increase. With this perimeter mode, long routes to the destination node (AP) can potentially be taken; and even in the worst case, routing loops may appear. Therefore, there are more events to be processed, which means a higher running time. On the other hand, 3MRP [1] requires less running time (11 hours per point) than GPSR. Although 3MRP performs more operations since it handles five routing metrics (distance to destination, vehicle's trajectory, vehicles' density, available bandwidth, and MAC losses), the message forwarding is more efficient. 3MRP selects better forwarding nodes at every hop towards destination, resulting in a lower hop count, and thus fewer nodes have to process the messages. Finally, we can also see in Fig. 13 that MPANN (11h 30m) took half an hour more than 3MRP (11 hours). The additional delay incurred by MPANN is due to the architecture of the ANN algorithm, due to the operations done to evaluate each neighboring node that entails a slight extra consumption of time.

Nonetheless, this computer running time is just a researcher's worktime needed to assess any new proposal's performance. Once the performance evaluation has been analysed, what really matters would be the increase in computational cost that implementing our proposal would cause in real vehicles. Given the impossibility of using real smart vehicles in our experiments, we present the computational cost in terms of time required to compute our ML-based proposal in the next section. As we will see, this computational time is really insignificant: The computational cost incurred by our proposal MPANN is below 0.2 msec, which is irrelevant considering the improvements obtained in terms of percentage of packet losses and average end-to-end packet delay.

TABLE 4. Hardware specifications of the personal computer used to obtain the computational cost in terms of time.

Computer	macOS Sierra Version 10.12.6
Processor	Intel i7
Base clock frequency	2.9 GHz
Memory	8 GB 1600 MHz

3) COMPUTATIONAL COST, IN TERMS OF COMPUTATIONAL TIME

In this subsection, we have calculated the computational cost in terms of time of our ML-based forwarding algorithm, designed using an ANN as it was explained in Section IV. This metric is frequently used to measure the efficiency and viability of a designed algorithm. To calculate the computational cost, we have analyzed the time incurred by each one of the forwarding algorithms used in the routing protocols. In addition, we have carried out two experiments to obtain the computational time of our proposals using Matlab: (i) A first experiment varying the vehicles' density, and (ii) a second experiment making the vehicles' density vary throughout time. For these two experiments, we used the hardware described in Table 4. In both experiments, we generate a traffic of 2 packets/sec. In this way, we are able to replicate in Matlab an equivalent sequence of packets transmitted that vehicles would process in our OMNeT++/Veins/SUMO/OSM simulation framework.

(i). **First experiment**. We have generated random values (in the range [0, 1]) in Matlab for each normalized routing metric (i.e., distance to destination, trajectory, vehicles' density, available bandwidth, and MAC losses). Vehicles have a transmission range of 340 meters. We have considered the same four different values of the vehicles' density as in our simulations with OMNeT++/Veins/SUMO/OSM, i.e., 50, 100, 150, and 200 vehicles/km^2.
Fig. 14 shows the computational cost (in terms of time) of the evaluated routing protocols for VANETs. MPANN shows the highest computational cost, with an increment of 0.02 msec with respect to 3MRP and GPSR. However, notice that this value represents an insignificant time amount compared to the end-to-end average packet delays shown by ANN in the range 20-40 milliseconds, as it was depicted in Fig. 12. In Fig. 14 we can see that the maximum computational time used by the CPU is 0.028 msec when the vehicles' density is the maximum considered (200 vehicles/km^2) since the number of operations is maximum too. The delay increment is due to the cost of the additional operations included in the ANN forwarding algorithm.

(ii). **Second experiment**. In this second Matlab test, we have set the number of neighbors per vehicle to be a random value in the range [1, 50]. This range of number of vehicles in OMNeT++/Veins/SUMO/OSM would correspond with a vehicles' density defined in the range [3, 138] vehicles/km^2 for a transmission range

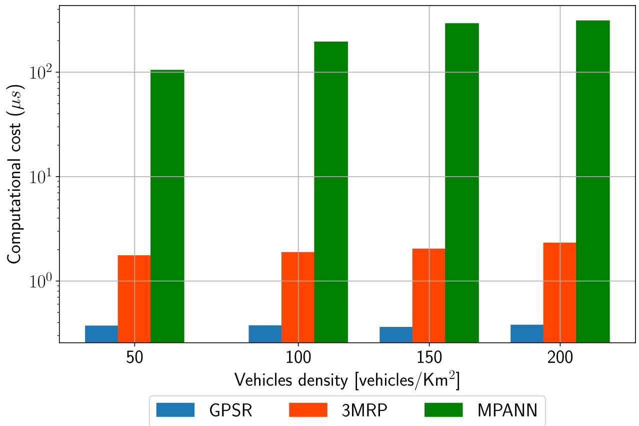


FIGURE 14. Computational cost (in time) of our ML-based forwarding algorithm measured in Matlab, for four vehicles' densities.

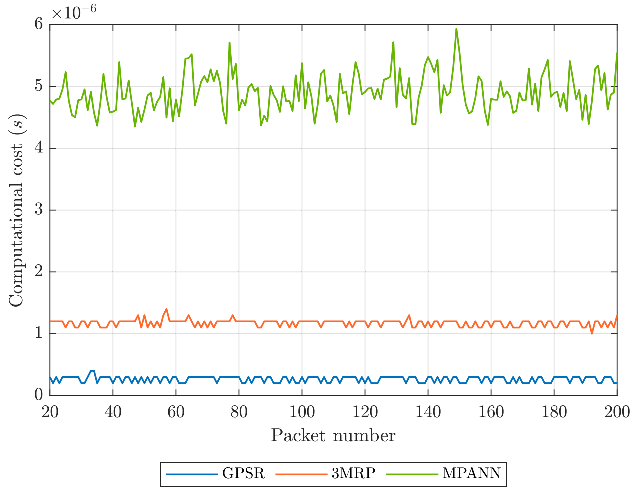


FIGURE 15. Computational cost (in time) of our ML-based forwarding algorithm measured in Matlab, for each incoming packet measured in a general node.

of 340m. In Fig. 15 we can see that MPANN produces a computational time of 48 μsec, whereas, for 3MRP and GPSR, it is around 0.8μsec and 12μsec. Even MPANN needs 36 more times computational time than the others, 48 μsec is still very small compared to the arrival time between consecutive packets so that packets can be processed quickly without any problem. Notice that for a packet flow rate of 6 Mbps and a packet size of 259 bytes (see Table 2), the average arrival time between consecutive packets is 345.33 μsec/packet.

In summary, using Matlab we emulate the same sequence of operations done over each packet and for each neighbor node as we would have in the OMNeT++/Veins/SUMO/OSM simulation framework. Then, in Matlab it is easy to calculate the computational time dedicated per packet by the different forwarding algorithms that operate in GPSR, 3MRP, and MPANN. Then, we can calculate the total computational time necessary to process all the packets sent in a 100-sec simulation. This way, we can assess the trade-off between benefits (shown

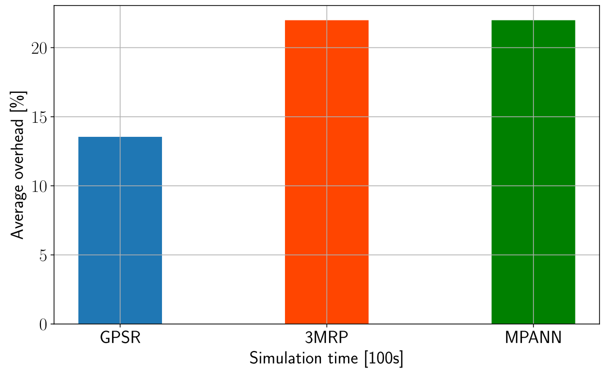


FIGURE 16. Overhead incurred by MPANN compared to GPSR [33] and 3MRP [1].

in section VII-A1) and costs in terms of computational time. Effectively, the computational time needed is very low, and it rewards the benefits of MPANN in terms of average percentage of packet losses (see Fig. 11) and average end-to-end packet delay (see Fig. 12).

4) OVERHEAD

Fig. 16 shows the overhead incurred by each one of the routing protocols analyzed in this performance evaluation. Although GPSR [33] has the lowest overhead (14%), it produces the highest percentage of packet losses, see Fig. 11 and Fig. 12, respectively. On the other hand, the other routing protocols (3MRP and MPANN) increase the overhead to 22% (i.e., a relative 57% higher amount). The reason is that those two routing protocols use additional fields (whose size is from 2 to 8 bytes) in the hello messages to carry the values needed to calculate the routing metrics (in 3MRP) and inputs (in MPANN). Notice that our proposal MPANN uses the same additional fields included in our former protocol 3MRP [1], so there was no need to increase the overhead. Those fields are updated by each vehicle at the moment of sending the current hello message (or beacon). Those additional fields carry an 8-byte field with the (v_x, v_y) velocity coordinates of the node, an 8-byte field for the MAC layer losses value, and a 4-byte field for the vehicles' density value. The calculation of the five routing metrics (available bandwidth, vehicles' density, distance to destination, vehicle's trajectory, and MAC losses) computed from those values carried in the beacons was explained in Section V.

The forwarding algorithm uses these routing metrics at the vehicle currently holding the packet to make the forwarding decision and choose the next node to forward the packet. Besides, MPANN does not produce any additional overload compared to 3MRP since it uses the same beacon structure already established in 3MRP [1] to gather the values to compute the inputs (i.e., the same routing metrics as 3MRP) for the ML-based forwarding algorithm. Although our prediction model includes a small extra cost in terms of packet delay (delay in MPANN is maximum 20 msec. higher than in 3MRP, according to Fig. 12), the benefits obtained in terms of packet losses for MPANN are significantly notable,

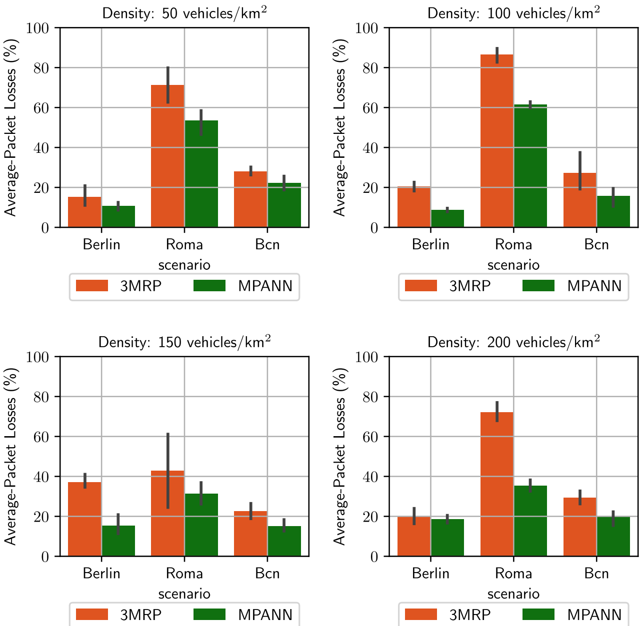


FIGURE 17. Average percentage of packet losses for MPANN vs. 3MRP [1] in the 3 city maps considered (5 repetitions/point, 95% CI).

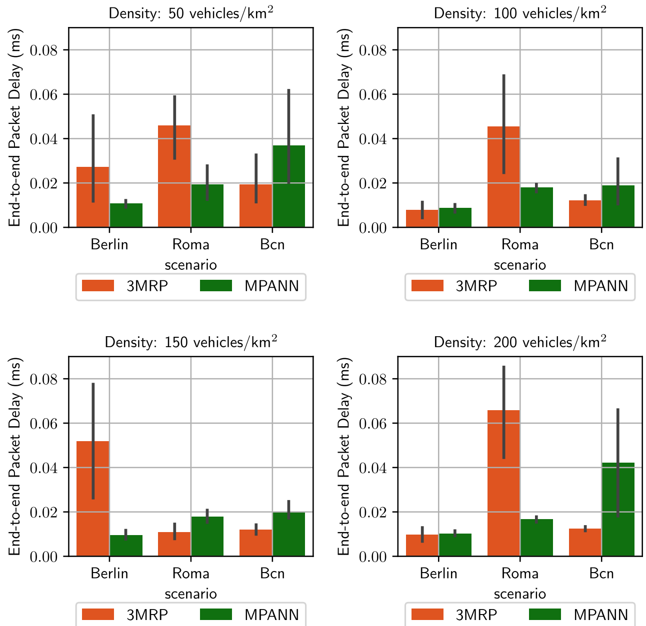


FIGURE 18. Average end-to-end packet delay for MPANN vs. 3MRP [1] in the 3 city maps considered (5 repetitions/point, 95% CI).

as it is shown in Fig. 11. This improvement is achieved in MPANN without having introduced any additional overhead with respect to 3MRP (see Fig. 16).

B. TEST 2: PERFORMANCE EVALUATION OF THE PROPOSED MPANN ROUTING PROTOCOL IN OTHER CITY MAPS

In this second test, we have considered two different VANET scenarios where we have evaluated both routing protocols: our proposal MPANN (presented in Section VII) and our previous proposal 3MRP [1]. These scenarios are different from the Barcelona scenario (Fig. 1), which was the scenario used to gather the dataset with which we trained and validated our ANN forwarding model, see Section IV). In this last phase of our performance evaluation, we want to assess the flexibility of our proposals over other city maps with different characteristics. We have evaluated the performance of both routing protocols in an area of Berlin (Fig. 9 (similar in size but more complex in street layout) and in an area of Rome (Fig 10) (larger in size and much more complex in street layout). Our goal is to validate if the prediction model of our ML-based forwarding algorithm can classify data correctly in different city maps (different from the map from which we took the data to train the ANN model).

Fig. 17 shows the average percentage of packet losses for both MPANN and 3MRP [1] routing protocols for the three different city maps and four vehicles' densities. As it can be seen, our MPANN routing protocol performs better in the three urban scenarios, even in the more complex city map of Rome. Notice that for both routing protocols, the highest percentage of packet losses corresponds to the Rome scenario, since the considered area of Rome is more complex with narrow and one-way streets. In Rome, the percentage of packet losses with MPANN is around 55% for a sparse vehicles' density, and it improves for the 150 vehicles/km^2 density to around 30%, since the network connectivity improves. Overall, we see that MPANN outperforms 3MRP in around 5-20% in the Rome results.

Additionally, MPANN also improves 3MRP over the Berlin map for the four vehicles' densities. MPANN achieves average packet losses from 8% for 100 vehicles/km^2 to a maximum of 18% for 200 vehicles/km^2.

Fig. 18 shows the average end-to-end packet delay for both MPANN and 3MRP [1] routing protocols for the three city maps and different vehicle densities. On the one hand, for the same map used to create the dataset (Barcelona city area), delays with MPANN are higher than with 3MRP [1] in around 18 to 40 milliseconds. On the other hand, however, this slight increase in delay pays off since MPANN offers a lower percentage of packet losses than 3MRP, in around 5-20% according to Fig. 17.

In the Berlin city area, which has a similar street layout as the Barcelona city area, MPANN shows a similar average end-to-end packet delay for the four vehicles' densities, of around 10 msec 3MRP gets more variable delays, from 10 msec for 100 and 200 vehicles/km^2 to 50 msec for 150 vehicles/km^2. In the Rome scenario, MPANN shows average packet delays under 20 msec for the four densities, whereas 3MRP offers a maximum delay around 60 msec for 200 vehicles/km^2.

Overall, we can conclude that our ANN-based proposal of a routing protocol for VANETs called MPANN shows a good adaptation in a quite similar Berlin scenario. However, a lower prediction power is presented in the Rome scenario.

The reason is that the model does not have enough experience with an urban configuration so complex as the one of the Rome city. Therefore, although the model can face new scenarios, its prediction power is not so good when the city layout differs a lot from the city map used to create the dataset. Anyway, we can also notice that our proposal MPANN outperforms 3MRP in terms of packet losses and packet delay for most cases.

To further improve these results, we should include a larger amount of registers in our dataset, taken from diverse cities with different street layouts, and then train the ANN-based model again. Alternatively, another option should be to train a small set of different models, each one for a type of city. This way, we would have a model for cities like Barcelona or Berlin, another model for cities like Rome, and so on. These are goals pointed out in the future work that will follow our research work.

VIII. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel ML-based multimetric routing protocol for VANETs whose forwarding algorithm is based on machine learning decisions. Our proposal is named **multimetric predictive artificial neural network-based (MPANN)** routing protocol.

We have created a dataset from a large number of simulations in an urban scenario, recording five routing metrics (distance to destination, vehicle path, vehicle density, available bandwidth, and MAC losses) calculated from the data exchanged between nodes through periodic hello messages. The dataset collected has been used to train different learning algorithms using traditional machine learning performance metrics (accuracy, ROC, AUC). Then, we have chosen the ML model that showed the best performance, an artificial neural network named **ANN-based forwarding model**.

To evaluate our proposal MPANN, we have used two tests for four different vehicles' densities: (i) A first test in the same city map with which the dataset was created, i.e., a 2300m $\times$ 2100m area of Barcelona; (ii) a second test using two different city maps: a similar area of Berlin, and a larger and more complex area of Rome. Our goal was to measure the level of flexibility and adaptation attained by our designed ML-based forwarding algorithm in the MPANN routing protocol. We have compared our MPANN learning model to the well-known GPSR [33], and to our previous proposal of multimetric routing protocol for VANETs name 3MRP [1]. Results show improvements up to 20% of packet losses and up to 0.04 ms (60%) in average end-to-end packet delay, see Fig. 17 and Fig. 18.

We have also evaluated the performance of our proposal in terms of *simulation run-time*, *computational cost* (in terms of time) and *overhead*. MPANN shows a slight increment around 1.5 msec in the computational time with respect to GPSR and 3MRP. However, these values are minimal compared to the end-to-end average packet delays shown by ANN in the range 20-40 ms, see Fig. 12. Our proposal MPANN does not add any extra overhead compared to 3MRP, since it uses the same header fields in the beacon messages as 3MRP [1] to carry the values to compute the five routing metrics needed in the ANN model.

Finally, we can conclude that our designed MPANN routing protocol adapts very well in scenarios with similar streets layout as the Barcelona scenario. However, we also conclude that it would still be advisable to learn more with new data from other different scenarios to generalize it further. This way, we would obtain a model able to adapt to most of the kind of city environments and street layouts. Accordingly, as future work, we will extend our dataset with new values from new scenarios using heterogeneous city maps with very different street layouts and orography. Notice that it would just be necessary to conduct a large amount of new simulations and process the data properly to include more registers in the dataset. We would then follow the same procedure designed in this work to train and validate a new version of the ANN-based forwarding algorithm. Finally, an evaluation of the performance in different types of cities will show the accuracy of results concerning the current MPANN version.

REFERENCES

[1] A. M. Mezher and M. Aguilar Igartua, "Multimedia multimetric map-aware routing protocol to send video-reporting messages over VANETs in smart cities," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10611–10625, Dec. 2017.

[2] L. Lemus Cárdenas. (Jul. 2020). *Ph.D. Thesis: Enhancement of Vehicular Ad Hoc Networks Using Machine Learning-Based Prediction Methods*. [Online]. Available: http://hdl.handle.net/10803/67002

[3] S. Lee, Y. Kim, H. Kahng, S.-K. Lee, S. Chung, T. Cheong, K. Shin, J. Park, and S. B. Kim, "Intelligent traffic control for autonomous vehicle systems based on machine learning," *Expert Syst. Appl.*, vol. 144, Apr. 2020, Art. no. 113074.

[4] C. C. Kaymak and A. Uçar, *A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving*. Cham, Switzerland: Springer, 2019, pp. 161–200.

[5] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3967–3979, Apr. 2019.

[6] C. Wu, T. Yoshinaga, Y. Ji, and Y. Zhang, "Computational intelligence inspired data delivery for vehicle-to-roadside communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12038–12048, Dec. 2018.

[7] M. Laroui, A. Sellami, B. Nour, H. Moungla, H. Afifi, and S. B. Hacene, "Driving path stability in VANETs," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[8] D. Zhang, T. Zhang, and X. Liu, "Novel self-adaptive routing service algorithm for application in VANET," *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 1866–1879, May 2019.

[9] J. Wu, M. Fang, H. Li, and X. Li, "RSU-assisted traffic-aware routing based on reinforcement learning for urban vanets," *IEEE Access*, vol. 8, pp. 5733–5748, 2020.

[10] F. H. Kumbhar and S. Y. Shin, "DT-VAR: Decision tree predicted compatibility-based vehicular ad-hoc reliable routing," *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 87–91, Jan. 2021.

[11] J. Wu, Y. Guo, H. Zhou, L. Shen, and L. Liu, "Vehicular delay tolerant network routing algorithm based on Bayesian network," *IEEE Access*, vol. 8, pp. 18727–18740, 2020.

[12] Y. Xia, L. Wu, Z. Wang, X. Zheng, and J. Jin, "Cluster-enabled cooperative scheduling based on reinforcement learning for high-mobility vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12664–12678, Nov. 2020.

[13] *OMNeT++ Discrete Event Simulator*. Accessed: Jun. 2021. [Online]. Available: https://omnetpp.org

[14] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[15] *SUMO-Simulation of Urban Mobility*. Accessed: Jun. 2021. [Online]. Available: http://sumo.sourceforge.net

[16] S. Coast. *OpenStreetMap (OSM)*. Accessed: Jun. 2021. [Online]. Available: https://www.openstreetmap.org

[17] C. L. Patrick, *Artificial Neural Networks: Application*. Norderstedt, Germany: Books on Demand, 2011.

[18] R. R. Nadikattu, "The supremacy of artificial intelligence and neural networks," *Int. J. Creative Res. Thoughts*, vol. 5, no. 1, pp. 950–954, 2017.

[19] R. Reed and R. J. MarksII, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1999.

[20] B. Levander, *Machine Learning Book: Learn Machine Learning, Deep Learning, Data Science, and More!: Machine Learning Beginners*. USA: Independently Published, 2021.

[21] M. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.

[22] N. Karayiannis and A. N. Venetsanopoulos, *Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications*, vol. 209. Cham, Switzerland: Springer, 2013.

[23] S. Sivanandam and S. Deepa, *Introduction to Neural Networks Using MATLAB 6.0*. New York, NY, USA: McGraw-Hill, 2006.

[24] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks*, vol. 10. New York, NY, USA: Springer, 2017, pp. 3–978.

[25] S. Raschka, *Python Machine Learning*. Birmingham, U.K.: Packt, 2015.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[27] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: https://arxiv.org/abs/1609.04747

[28] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, Aug. 2007.

[29] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping and non-parametric regression: An optimal data-dependent stopping rule," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 335–366, Jan. 2014.

[30] A. Eldeeb, E. Bursztein, F. Chollet, H. Jin, and Q. S. Zhu. *Keras, Deep Learning for Humans*. Accessed: Jun. 2021. [Online]. Available: https://github.com/kerasteam/keras

[31] C. Sarr, C. Chaudet, G. Chelius, and I. Guérin-Lassous, "A node-based available bandwidth evaluation in IEEE 802.11 ad hoc networks," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 21, no. 6, pp. 423–440, 2006.

[32] C. Tripp-Barba, L. Urquiza-Aguiar, M. Aguilar Igartua, D. Rebollo-Monedero, L. de la Cruz Llopis, A. M. Mezher, and J. Aguilar-Calderón, "A multimetric, map-aware routing protocol for VANETs in urban areas," *Sensors*, vol. 14, no. 2, pp. 2199–2224, Jan. 2014.

[33] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2000, pp. 243–254.

LETICIA LEMUS CÁRDENAS received the degree in telecommunication engineering from the University of Guadalajara (UdeG), México, in 2007, and the M.Sc. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2011 and July 2020, respectively. She has been an Associate Professor with the Department of Knowledge Foundations, North University Center, University of Guadalajara, since 2012. Her research interests include vehicular *ad hoc* networks, machine learning, deep learning, electric vehicles, and autonomous vehicles in urban environments.

AHMAD MOHAMAD MEZHER received the M.S. degree in signals and systems from the Central University of Las Villas, Santa Clara, Cuba, in 2011, and the Ph.D. degree in network engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2016. He was a recipient of a FI-AGAUR Fellowship from the Generalitat of Catalunya and the Social European Budget. He currently holds a McCain Postdoctoral Fellowship in innovation with the Electrical and Computer Engineering Department, University of New Brunswick (UNB). His research interests include smart grid communications, vehicular *ad hoc* networks, data privacy, and load forecasting.

PABLO ANDRÉS BARBECHO BAUTISTA received the degree in electronic engineering from the University of Azuay, Cuenca, Ecuador, in 2012, the M.Sc. degree in communication networks from the Pontificia Universidad Católica del Ecuador, in 2017, and the M.Sc. degree in telecommunication engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, where he is currently pursuing the Ph.D. degree in network engineering. His research interests include design and performance evaluation of routing protocols for vehicular *ad hoc* networks (VANETs). His research activity focuses on the integration of electric vehicles and autonomous vehicles in VANETs and their interaction with smart grids and smart cities.

JUAN PABLO ASTUDILLO LEÓN received the degree in electronics engineering from the Universidad Politècnica Salesiana, Cuenca, Ecuador, in 2012, the specialization degree in telecommunications technologies and the master's degree in telecommunications engineering from the Universidad de Buenos Aires, Buenos Aires, Argentina, in 2014 and 2015, respectively, and the Ph.D. degree from the Networking Department, Universitat Politècnica de Catalunya, Barcelona, Spain, in 2020. He was an Assistant Professor with Universidad Politècnica Salesiana, in 2016. His research includes wireless mesh networks, wireless mobile *ad hoc* networks, smart grids, vehicular networks, machine learning, and deep learning.

MÓNICA AGUILAR IGARTUA received the M.Sc. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1995 and 2000, respectively. She is the author of more than 25 journal articles, has chaired several conferences and is a member of the editorial board of the *Ad hoc Networks* journal. Her research interests include design and performance evaluation of multimetric routing protocols to distribute multimedia services over vehicular *ad hoc* networks (VANETs). Lately, her research activity focuses on the integration of electric vehicles and autonomous vehicles in vehicular networks and their interaction with smart grids and smart cities.