Date of publication JUN-30, 2025, date of current version FEB-13, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.2.4013

Packet Error Rate Prediction in VANETs using Bandwidth and Signal-to-Noise Ratio

ETIENNE A. FEUKEU, SUMBWANYAMBE MBUYU
Dept. Electrical and Smart Systems Engineering University of South Africa
Johannesburg/Florida, South Africa 1709

Corresponding author: Etienne A. Feukeu (e-mail: efeukeea@unisa.az.za).

ABSTRACT The Wireless Access in Vehicular Environment (WAVE) aims to facilitate communication between vehicles and infrastructure, primarily for safety improvements. Despite more than a decade of research, significant technological hurdles still remain before this potentially life-saving technology can be seamlessly integrated into modern automobiles. However, the frequent transmission of Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) from each mobile device is the leading cause of broadcast storms, especially considering the limited 10 MHz channel bandwidth. Therefore, to mitigate some of these challenges, making use of the Feed Forward Neural Network (FFNN) and the Levenberg-Marquardt Algorithm (LMA), the current work focuses on designing, developing, and modeling Packet Error Rate (PER) in the Vehicular Ad hoc Network (VANET) environment to enable mobile devices to predict the PER based on such factors as the available Bandwidth (BW) and Signal-to-Noise Ratio (SNR). Following model development and training, rigorous testing and verification are conducted to demonstrate the system's effectiveness and efficiency in predicting PER based on SNR and BW parameters. The resulting evaluation proved that the developed model has the ability to accurately predict PER with an accuracy of 85%.

KEYWORDS Machine Learning; Neural Networks; PER; VANET; WAVE; V2V; V2I.

I. INTRODUCTION

Initiated around a decade ago, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) technologies are integral components of Vehicular Ad-hoc Networks (VANETs), aimed at fostering seamless communication between vehicles and infrastructure elements. Their primary goal is to enhance road safety, alleviate traffic congestion, and ultimately optimize transportation efficiency [1]. These technologies operate within the framework of Cooperative Intelligent Transport Systems (C-ITS), which seek to enable vehicles to communicate with each other to prevent accidents, provide timely warnings to drivers, and streamline traffic flow [2]. Two key features of C-ITS, Turn Assist (TA) and Junction Movement Assist (JMA), serve as vital tools in ensuring safe driving practices. By combining these features, the United States Department of Transportation (US DoT) estimates the potential to prevent a substantial number of accidents annually. Additionally, a study performed by the US DoT showed that the combined implementation of TA and JMA could save up to 600,000 accidents annually [3]. The significance of these technologies is further underscored by findings from the National Highway Traffic Safety Administration (NHTSA), which suggest that a significant portion of motor vehicle crashes could have been averted through the implementation of V2V communication. This assertion was subsequently supported by J. Koon, showing that 615,000 motor vehicle crashes in the year 2020 could have been prevented using V2V communication technology [4]. Globally, there were an estimated 1.19 million road traffic deaths in 2021 [5]. Road traffic accidents have been the leading cause of death and a major contributor to disability; road traffic injuries also impose an enormous economic cost on societies. It was estimated that the global macroeconomic cost of road traffic injuries was over US$1.8 trillion in 2021 [6].

Moreover, the prevalence of human error in a large proportion of road traffic accidents highlights the urgent need for advanced technological interventions like VANET. The promise of VANET technology lies in its potential to significantly reduce fatalities and injuries resulting from road accidents on a global scale. Operating under the Wireless Access in Vehicular Environments (WAVE) standard, vehicles equipped with V2V capabilities can exchange critical information, such as their respective positions and speeds,

within a limited radius. This exchange of data empowers drivers with real-time alerts about potential hazards, including imminent collisions, lane changes, and adverse road conditions. In fact, under the WAVE standard, vehicles have the capability to communicate with each other through V2V, sharing data such as position, speed, and direction. Within a 300-meter radius, vehicles can exchange these signals up to ten times per second [2], providing them with a comprehensive 360-degree "view" of their surroundings and other cars in the vicinity. Drivers can receive alerts from V2V regarding potential crashes, lane changes, rear-end cross traffic, blind spots, and slippery road conditions. This will only be possible if there is a successful message exchange between the involved nodes.

However, the effectiveness of VANET is not without its challenges. Broadcast Storms (BrS) pose a significant threat to communication reliability when the available channel bandwidth becomes overwhelmed by the volume of message transmissions. Although the Channel Busy Ratio (CBR) can be locally estimated [2], there is no direct relation between the CBR and the PER. Undoubtedly, PER estimation becomes a big challenge in such an environment. To mitigate this risk of BrS, the proposed approach involves modeling the Packet Error Rate (PER) estimation in VANET using Machine Learning and Levenberg-Marquardt algorithm. This innovative method utilizes locally computed metrics like the Signal-to-Noise Ratio (SNR) and available Bandwidth (BW) obtained from the CBR to predict the likelihood of successful data transmission.

The significance of this work in the realm of VANET cannot be overstated. By offering a systematic methodology for generating and analyzing data, developing robust neural network models, and addressing complex challenges of the PER estimation, the research contributes substantially to advancing the capabilities of VANET technology. As of the time of writing, no comparable literature exists in the area of PER estimation based on SNR and BW using Machine Learning (ML), making this research a pioneering effort in the field.

The work demonstrates how to:

- Generate a valid dataset from a model simulation.
- Explore and analyze complex data.
- Create a robust NN model from a valid dataset.
- Use NN to Model a complex V2V PER estimation.

The remainder of this work is arranged as follows: a literature review is presented in Section 2. The packet error rate derivation is explained in Section 3. The data generation is discussed in Section 4. The proposed ML algorithm is introduced in Section 5. The model testing and evaluation are shown in Section 6. And finally, the conclusion is provided in Section 7.

II. LITERATURE REVIEW

While the concepts of Packet Error Rate (PER) derivation, calculation, and estimation are not new, to the best of our knowledge, very little work has been done in the area of Vehicular Ad-hoc Networks (VANETs) due to the complexity of their formulation, particularly in the case of the PER estimation model as a function of Signal-to-Noise Ratio (SNR) and available Bandwidth (BW). However, there are a few papers in the literature for PER formulation and derivation. A new Packet Error Rate (PER) degradation model for VANETs was presented in [7]. The study presented a novel method for channel modelling that takes into account the random behaviour of wireless networks in the setting of frequent topology changes and significant node mobility in automotive contexts. The model utilizes Markov-based trace analysis to study the dynamics of packet error rate degradation concerning SNR and Doppler Shift (DS). The study gives a way to fine-tune the model based on actual observations in VANET settings and delivers insights into packet-level channel performance.

A closed-form equation of the packet loss probability, outage probability, packet collision probability at the MAC layer, and packet error probability at the physical layer were all explored numerically to examine the performance of VANETs in [8]. The study investigated a VANET model for several fading channels, including Nakagami-m, Weibull, Rician, and Rayleigh channels, while accounting for the minimal safety distance between vehicles. There was further discussion of the variables impacted by many cars, vehicle speed, contention window size, packet size, and distances between vehicles. Based on the results of the simulation research, it was evident that under various fading channels (Nakagami-m, Weibull, Rician, and Rayleigh channels), the packet error and loss probability are consistently lower than those without the minimum safety distance. In [9], authors adopted the cluster-based approach to analytically model VANETs. Closed-form expressions of packet loss probability for different fading models (i.e., Rayleigh, Rician, Nakagami-m, and Weibull channels), which consider both MAC and Physical Layer characteristics, were provided. The derived expressions also encompassed the effect of decoding failure and packet collision based on SNR and back-off timer, respectively. Using extensive simulations, the results characterized the impact of channel conditions, vehicle speed, and contention window on the performance of the cluster-based VANETs.

In [10], the authors conducted a comparative analysis between a Realistic Packet Error model [11] and a Basic Error Model, the latter relying solely on antenna sensitivity. Simulations were performed on capacity channels of 6, 12, and 27 Mbps, aligning with the IEEE standard recommendation 802.11p [12] for reduced packet error rates. The findings suggested no statistically significant difference between the two packet error models for low and medium-rate channels. However, a notable distinction emerges in high-capacity channels (27 Mbps) due to the inability of high antenna sensitivity to mitigate interference effects, even in scenarios with low data traffic. It is emphasized that while configurations with high sensitivity and basic error models may yield reliable results in some cases, such outcomes are not guaranteed. A heuristic function employing regression methods to calculate the packet loss rate (PLR) in a vehicle network, which considers varying vehicle density, transmission speed, and transmission power values, was introduced in [13]. Regression algorithms were tested on a variety of urban and highway situations utilizing realistic traffic scenarios simulated with Omnet++, Sumo, and Veins environments. A PLR dataset was created from the resultant transmissions from connected cars, which were then used as test and training data for several regression methods. Based on numerical results, the Catboost regression approach predicts PLR with the least amount of error when compared to other methods. With the help of the proposed heuristic, PLR may be directly determined given a particular set of transmission parameters. This allows system designers to choose solutions that meet PLR criteria or change

the transmission parameters to lower PLR below the desired level.

A novel approach for enhancing reliable vehicle-to-vehicle (V2V) communication across diverse environments was explored in [14]. Performance metrics, including time to collision, bit error rate (BER), packet error rate (PER), average throughput (Mbps), beam selection probability, and computational complexity factors, demonstrate the proposed system's excellence through extensive experiments and simulations compared to traditional methods. The approach achieves notable improvements, with an average throughput of 1.7 Mbps, a 0.1 reduction in BER, and near 100% beam power efficiency at a computational factor of 34. These results signify the efficiency and robustness of the proposed method, highlighting its potential for advancing V2V communication reliability in various scenarios.

Even though a large number of previous studies were assessed on various platforms, the majority of the suggested models could be very sophisticated, making them difficult to execute and assess in actual VANET circumstances. Furthermore, the accuracy of the results might be impacted by some models' reliance on assumptions that might not always hold true in real-world VANET setups. In order to avoid requiring difficult mathematical derivation, the current study aims to create a Neural Network (NN) model that might assist in quickly estimating the PER as a function of SNR and BW in VANETs.

III. PACKET ERROR RATE DERIVATION

Before deciding on the Machine Learning (ML) algorithm to be used, it is imperative to understand the nature of the problem at hand. The model consists of estimating the Packet Error Rate (PER) as a function of the Signal-to-Noise Ratio (SNR) and the available channel Bandwidth (BW) in a vehicular environment. In essence, PER is used in communication systems to measure the dependability and quality of data delivery. It shows the proportion of packets that are incorrectly received among all the packets that are transmitted. PER is often expressed as a fraction or as a percentage.

During transmission, packets may become damaged or lost due to various reasons, including noise, interference, fading, or congestion. One metric that aids in our comprehension of how these errors affect system performance is the PER. Based on the PER derivation theory under the Wireless Access in Vehicular Environment (WAVE) [15], assuming that all mobiles are equipped with Global Positioning System (GPS) receivers and sensors, from which the state of other neighbour vehicles (e.g., location, speed, direction, and acceleration) can be easily obtained [16].

Before calculating the PER, it is important to first calculate the Packet collision probability (Pcol) in a MAC layer separately, followed by the Packet Error probability (Perr) in the physical layer. If τ is used to express the probability that a given node transmits after the backoff time counter reaches zero [16], then it can be defined as

$$\tau = \frac{2p}{(CW-1)p+2}, \tag{1}$$

where p is signified the freezing of the backoff counter when the channel is detected idle, and CW is the contention window size. Then the probability of collision subjected by the transmitted packed over the VANET with N-1 neighbours can be expressed as

$$P_{col} = 1-(1-\tau)^{N-1}. \tag{2}$$

Additionally, the average SNR received at the distance R under the impacted path loss [17] can be defined as

$$\bar{\gamma}(R) = \frac{G_r G_t \lambda^2 P_t}{N_0 (4\pi)^2 R^{\alpha}}, \tag{3}$$

where G_r and G_t are the antenna gain for the receiver and transmitter and both values are equal to one.

$$\lambda = \frac{c}{f_c}, \tag{4}$$

where λ is signal wavelength (c=3x108m/s is speed of light and fc=5GHz is carrier frequency). All the vehicles have the same transmitted power P_t and N_0 is Additive White Gaussian Noise power (AWGN). Parameter α is denoted as the path loss exponent.

During transmission a packet is divided in an M block size and each block is also affected by the vehicular mobility.

$$M = \frac{L_p}{T_{coh} log_2(1-\gamma_{th})}, \tag{5}$$

where L_p is the packet size and γ_{th} is the SNR threshold, T_{coh} is the coherence time which is presented by

$$T_{coh} = \sqrt{\frac{9}{16\pi f_D^2}}, \tag{6}$$

where f_D is the Doppler Shift which is calculated as

$$f_D = \frac{v}{\lambda} cos\theta = \frac{v}{c} f_c cos\theta, \tag{7}$$

where v is the receiver velocity towards the transmitter in the direction of motion, c is the speed of light, f_c is the carrier frequency, θ is the arrival angle of the received signal relative to the direction of motion. Now, considering the Rayleigh fading channel in our model, the PDF of SNR under Rayleigh fading channel [18] is expressed as

$$f(\gamma) = \frac{1}{\bar{r}(R)} exp\left(-\frac{\gamma}{\bar{\gamma}(R)}\right). \tag{8}$$

Thereafter, the outage probability P_{out} is defined as the probability when the average SNR of the received signal $\bar{\gamma}(R)$ is less than the SNR threshold γ_{th}. If and only m=1 (m is the level Nakagami-m fading) then the P_{out} of Rayleigh fading channel [19] is defined as

$$P_{out} = \int_0^{\gamma_{th}} f(\gamma) d_\gamma = 1 - exp\left(-\frac{\gamma_{th} N_0 (4\pi)^2 R^{\alpha}}{G_r G_t \lambda^2 P_t}\right). \tag{9}$$

Then the Packet Error probability (P_{err}) for an M block of

packets is defined as

$$P_{err} = 1 - exp\left(-M\frac{\gamma_{th}N_0(4\pi)^2R^{\alpha}}{G_rG_t\lambda^2P_t}\right). \quad (10)$$

Therefore, if total N packets are transmitted over the medium, then the PER is defined as

$$PER = N\frac{P_{err}}{N} = P_{err}. \quad (11)$$

IV. DATA GENERATION

Before embarking on data generation, the communication channel model needs to be explored to clarify and ascertain whether the model was developed in a valid VANET environment.

A. CHANNEL MODEL

The MATLAB-based channel model utilized in this research proved to be effective in simulating the V2V wireless channel environment [20]. In this model, the communication begins with the transmission of the Medium Access Control (MAC) Protocol Data Unit (MPDU) from the transmitter, which is then passed to the physical layer. At the physical layer, the MPDU is encapsulated to create the Physical Layer Convergence Protocol (PLCP) Service Data Unit (PSDU), which is further used to generate waveform packets transmitted over the wireless channel. The wireless channel is structured using the V2V channel model compliant with IEEE802.11p [21], enhanced with Additive White Gaussian Noise (AWGN).

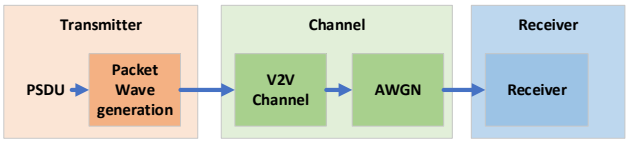

Figure 1. V2V channel model.

This configuration operates at a sampling rate of 10 MHz, taking into account the desired V2V communication delay profile. Two delay profiles are employed: Non-Line of Sight (NLOS) for urban communication and Line of Sight (LOS) for highway communication. At the receiver end, tasks include packet detection, coarse carrier frequency offset estimation and correction, symbol timing, and fine carrier frequency offset estimation and correction. The Long Training Field (L-LTF) is extracted from the synchronized received waveform, followed by OFDM demodulation and initial channel estimates acquisition. Channel tracking, if enabled, updates channel estimates per symbol using decision-directed channel tracking [22]. If disabled, initial channel estimates from L-LTF are used for the entire packet duration. The non-HT Data field is then extracted from the synchronized received waveform, and the PSDU is recovered using the extracted data field along with the channel estimates and noise power estimate.

B. PER DATA GENERATION

Using the described channel model and the simulation parameters depicted in Table 1, data were generated by simulating the packet error rate (PER) as a function of the signal-to-noise ratio (SNR), data size, and the channel bandwidth. Simulations were performed for both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios, with and without channel tracking.

Table 1. Simulation Parameters

Simulation parameters	Value
Number of frames	10000
Frame length	600 Byte
Channel Sampling Rate	10 MHz
MCS	BPSK 1/2
Channel type	V2V
Channel bandwidth	5, 10, 20 MHz
SNR	2 to 40 dB
Channel Delay Profile	LOS, NLOS
Maxi number of errors	1000

For each SNR point, multiple packets were transmitted, and the PER was calculated. The following processing steps occurred for each packet:

a) A protocol data unit (PDU) was created and encoded to form a single packet waveform.

b) The waveform was transmitted through the channel.

c) The long-training field (L-LTF) was extracted from the synchronized received waveform. The L-LTF was OFDM demodulated, and initial channel estimates were obtained.

d) Channel tracking was activated for the first set of simulations and deactivated for the second set. The PDU was recovered using the extracted data field, the channel estimates, and the noise power estimate.

After transmitting 10,000 frames of 600 bytes each at each SNR point and a specified channel bandwidth (5, 10, or 20 MHz), Figures 2 and 3 were generated for the LOS and NLOS scenarios with and without channel tracking.

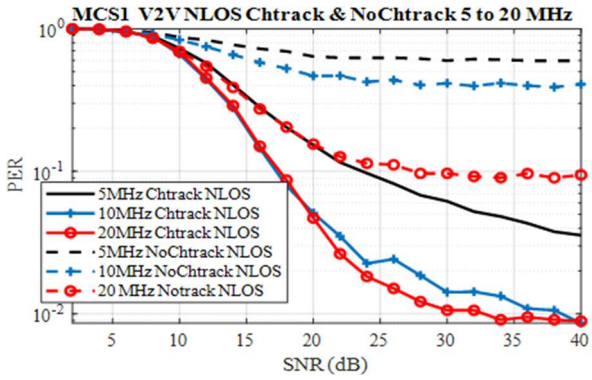

Figure 2. Data generation with channel tracking.

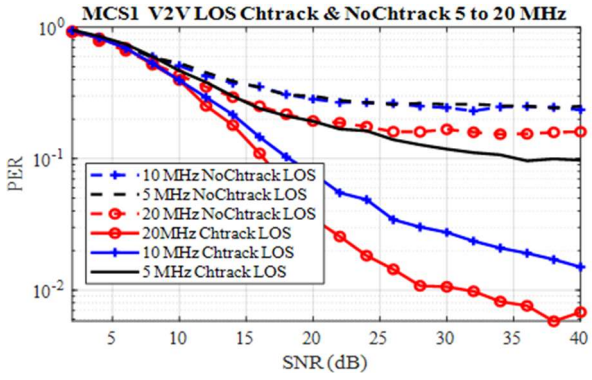

Figure 3. Data generation with No channel tracking.

It is noteworthy that the default VANET channel BW is set at 10 MHz, this default value made it difficult to execute a

simulation with a bandwidth (BW) variation of 1 to 10 MHz which was our region of interest, based on the V2V communication model suggested by MATLAB [20]. In MATLAB, the communication model's default BW values, were set to 5, 10, 20, or other higher numbers, which made it difficult for us to change the BW to the desired value. Developing our desired model began with overcoming this obstacle.

It is clear from looking at both numbers that raising the SNR level lowers the PER. Fig. 2 shows that beginning with SNR values of 30 dB and 38 dB, respectively, a PER of less than 10^{-2} may be attained for data production with channel tracking, especially with BWs of 20 and 10 MHz. Fig. 2 shows that all situations performed well below the PER of 10^{-1} for SNRs larger than 30 dB. Nonetheless, the 5 MHz, LOS with channel tracking is the least effective performance in this figure.

Only the 20 MHz, LOS scenario, beginning at an SNR value of 28 dB and higher, was able to obtain a PER lower than 10^{-1} in the event of data transmission without channel tracking, as shown in Fig. 3.

Every other scenario outperformed the 10^{-1} PER during the whole simulation run. Even though it was crucial to examine the response in both figures, only Fig. 2 data will be taken into consideration for the remainder of the study because most V2V communication instances function effectively with channel tracking.

C. DATA EXPLORATION

Table 2 displays our interest-specific data selections. The three parameters in this table are called PER, SNR, and BW.

Table 2. Dataset

	Dataset							
PER	0.932	0.992	0.690	. . .	. . .	0.009	0.037	0.015
SNR	2	4	6	. . .	. . .	36	38	40
BW	20	5	10	. . .	. . .	20	5	10
Sample	1	2	3	. . .	. . .	118	119	120

Every parameter has 120 samples, which correspond to all the data points that were collected to create Fig. 2. As of right now, a cursory glance at the table only reveals variations in each parameter's values collected throughout the whole range of sample sizes.

Fig. 4 was built to assist comprehend the association between each relevant parameter in order to provide a greater understanding of the data displayed in this table.

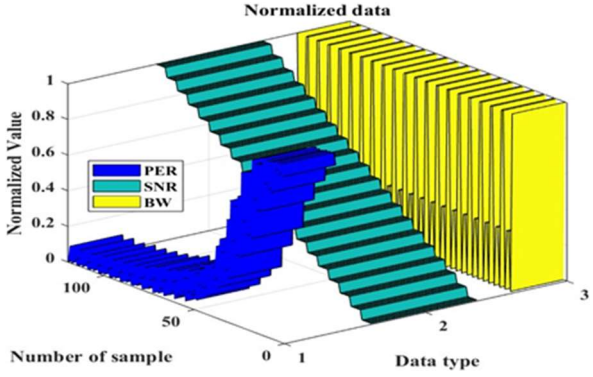

Figure 4. Data exploration

When examining the relationship between BW and PER in Fig. 4, it is evident that all three bandwidths (5, 10, and 20 MHz) remain constant while the PER changed from zero to its maximum.

Incidentally, this observation demonstrates that despite the constant values of BW, there seems to be no direct relation with the variation of PER. Analyzing the correlation between SNR and PER, it can be observed that at higher SNR values, there is a lower PER, and when the SNR starts dropping, the PER begins to rise at a certain corresponding ratio. In this case, there is a sort of direct relationship between SNR and PER. Lastly, examining the connection between SNR and BW in the same figure, the BW values remain constant across the entire SNR range. These BW values exhibit no discontinuities and remain constant. This suggests that the BW is fully indifferent to variations in SNR.

Further exploration of the dataset histogram as depicted in Fig. 5 shows that our data does neither follow positively skewered nor sparse distribution. This representation therefore proves that in order to create a good model, a data preprocessing is highly required. The resultant data analysis makes it evident that certain parameters are wholly independent of one another, and this is the point at which model building is essential. The ability to see a fluctuation in BW from 1 to 10 MHz, regardless of our generated data, which only has default values of 5, 10, and 20 MHz as required by MATLAB [20], is one of the main issues that we want to solve with the proposed model response.

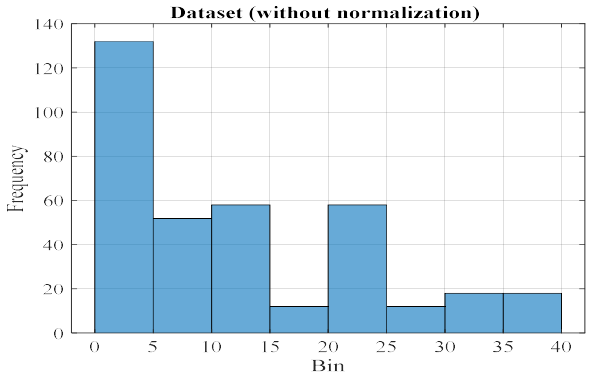

Figure 5. Dataset Histogram.

Understanding the significance of creating a PER estimating model becomes more obvious after looking at the intricacy of the PER derivation as indicated by equations 1 through 11. Additionally, it is seen in Table 2 and the data exploration picture in Fig. 4 that three of our parameters (SNR, BW, and PER) seem to depend on each other very little or not at all. Therefore, if an analytical model were to be derived, this would make it more complicated. These days, complicated issues may be represented using one or more Artificial Intelligence (AI) system algorithms, thanks to scientific developments. The AI subsystem's machine learning (ML) approaches include a variety of algorithms designed to tackle various kinds of issues. Given the intricacy of our issue, the Neural Network (NN) will be the best option in this instance. Generally speaking, NNs are able to learn and model complex and nonlinear relationships between inputs and outputs, draw conclusions and generalizations, find hidden patterns, correlations, and predictions, and model very volatile data and variances needed to forecast unexpected events [23]. This

means that the model will consist of two input layers, one hidden layer with ten neurons, and one output layer as we are dealing with a single target or label (PER) prediction based on input parameter values or features (SNR and BW).

V. PROPOSED MACHINE LEARNING MODEL

The proposed ML model presented in Fig. 6 describes all main components of our approach. The model is made of combination of a Neural Network (NN) and the Levenberg-Marquardt Algorithm.

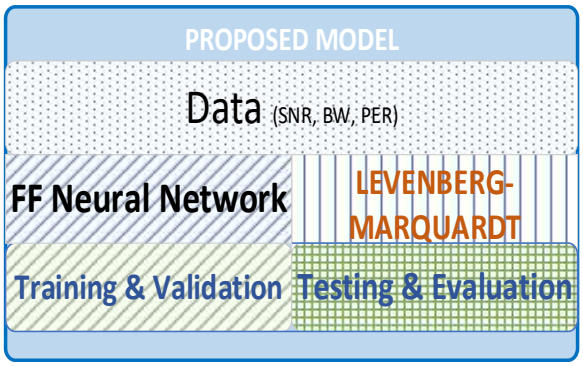

Figure 6. Proposed ML Model.

In order to construct a robust model, the data is first pre-processed, then explored to determine if any recognized patterns exist. Subsequently, the explored data is fed into the proposed model which then trained, validated, tested, and evaluated. Each component of this proposed model is extensively described and elaborated in the subsequent subsections.

A. NEURAL NETWORK MODEL

A Neural Network (NN), which is modelled after organic nerve systems, uses fundamental parts that function in parallel to integrate processing at several levels. An input layer, one or more hidden layers, and an output layer make up the network. There are many nodes, or neurons, in each layer. Because the nodes in each layer get all of the outputs from the layer before it, there is connectivity between neurons at different levels. Usually, each neuron is given a weight that might fluctuate during the learning process. These weights automatically change during training in line with a preset learning rule until the artificial neural network completes the specified job. A decrease or increase in weight causes a variation in the neuron's signal intensity [19]. Because of its single hidden layer topology, the neural network used in this study as depicted in Fig. 7 is also known as a Feedforward Shallow Neural Network (FFNN). The SNR and BW vectors are connected to the input layers, I1 and I2, respectively. There are ten neurons in this hidden layer, numbered 1 through 10, and the output layer, denoted by 'y,' generates the estimated PER. It should be noted that the middle-layer neurons use a sigmoid activation function, while the upper layer uses a Rectified Linear Unit (ReLU) activation function, providing convenience in error computation and loss minimization through the use of chain rules and the backpropagation approach. The sigmoid function is crucial because the step function includes only flat segments, lacking a gradient for Gradient Descent to navigate, whereas the sigmoid function has a well-defined nonzero derivative everywhere, enabling progress at each step. Conversely, although the ReLU (z) function is continuous and not differentiable at z = 0, it performs effectively and is computationally efficient.

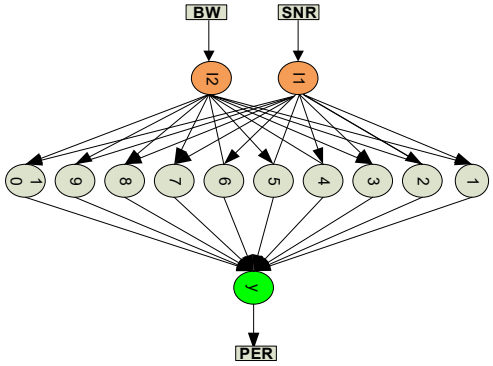

Figure 7. Feed Forward Neural Network.

The next stage is choosing the optimization algorithm after the neural network has been constructed. To achieve speedy convergence, the 'Levenberg-Marquardt technique (LMA)' was adopted as an effective optimization technique. The initial description of the Levenberg-Marquardt method can be found in [24], and a discussion of its use in neural network training can be found in [25], which covers pages 12 through 19 of [26]. It is noteworthy that this algorithm is the most effective method for training feedforward neural networks of modest size, usually up to several hundred weights. The unique quality of LMA is that it combines the Gauss-Newton method with steepest descent to get consistently optimum results. LMA operates as an iterative technique, pinpointing the minimum of a function expressed as the sum of squares of nonlinear functions. Its remarkable performance has established it as a standard method for addressing nonlinear least-squares problems. Additional insights into the proposed optimization algorithm (LMA) can be found in references [24], [25], and [26].

B. LEVENBERG-MARQUARDT ALGORITHM

Based on the proposed Neural Network (NN), the mathematical exploration of each neuron analysis proved that the optimal point will only be achieved when the error or loss is adequately minimized. The weight and bias variables are adjusted according to the Levenberg-Marquardt method, and the backpropagation algorithm is used to calculate the Jacobian matrix of the performance function with respect to the weight and bias variables. Based on the mentioned iterative processes, the NN estimation model will be well trained. The proposed neural network has a single output, and therefore, for each training dataset there is an error. The error vector for each training dataset will be:

$$e^T = (e_1, e_2, \ldots \ldots e_N)\,. \tag{12}$$

If E stands for the whole error, then the network activity function is defined as below and the purpose of the training process which consists of adjusting the weight vectors to minimize E is described by the subsequent equations.

$$E(w) = \frac{1}{2}\sum_{q=1}^{N}\left(y_q - \hat{y}_q\right)^2 = \frac{1}{2}\sum_{q=1}^{N} e^2\,, \tag{13}$$

where y_q is the actual value and $\hat{y}_q$ is the estimated value. The

weight adjustment used in the LMA is performed by the following equation:

$$\Delta w = (J^T J + \mu I)^{-1} J^T e \,, \tag{14}$$

where the Jacobian "J" matrix can be defined as follows:

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial x_1} & \frac{\partial e_1}{\partial x_2} & \cdots & \frac{\partial e_1}{\partial x_n} \\ \frac{\partial e_2}{\partial x_1} & \frac{\partial e_2}{\partial x_2} & \cdots & \frac{\partial e_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_N}{\partial x_1} & \frac{\partial e_N}{\partial x_2} & & \frac{\partial e_N}{\partial x_n} \end{bmatrix}. \tag{15}$$

Then let

$$X^T = (x_1, x_2, \ldots, x_n) = \left(w_{1,1}^1, w_{1,2}^1, \ldots, w_{s^1,s^0}^1, w_{1,1}^2, \ldots, w_{1,s^{M-1}}^M\right), \tag{16}$$

where the subscript "s" defined the number of hidden layers. Thereafter, the Jacobian matrix element is calculated using the chain rule method as follows:

$$[J]_{q,l} = \frac{\partial e_q}{\partial x_l} = \frac{\partial e_q}{\partial w_{ij}^m} = \frac{\partial e_q}{\partial \hat{y}_q} \cdot \frac{\partial \hat{y}_q}{\partial w_{ij}^m} = -\frac{\partial \hat{y}_q}{\partial w_{ij}^m}. \tag{17}$$

Thus, it becomes possible to calculate $J_{q,l} = -\frac{\partial \hat{y}_q}{\partial w_{ij}^m}$.

Let us assume that the neural network has a hidden (h) layer and an output layer (o) numbered as m =1,2, ... where the layer number 1 is the hidden layer "h" and layer number 2 is the output layer "o", then by substitution, m = h,o, ...The weight vector of the network is then represented as follows:

$$w^T = \left(w_{1,1}^h, w_{1,2}^h, \ldots, w_{s^h{}_R}^h, w_{1,1}^o, \ldots w_{1,s^h}^o\right), \tag{18}$$

then the Jacobian matrix is expressed by equation 19 as

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_N}{\partial w_1} & \frac{\partial e_N}{\partial w_2} & & \frac{\partial e_N}{\partial w_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial e_1}{\partial w_{1,1}^h} & \frac{\partial e_1}{\partial w_{1,2}^h} & \cdots & \frac{\partial e_1}{\partial w_{s^h,R}^h} & \frac{\partial e_1}{\partial w_{1,1}^o} & \cdots & \frac{\partial e_1}{\partial w_{1,s^h}^o} \\ \frac{\partial e_2}{\partial w_{1,1}^h} & \frac{\partial e_2}{\partial w_{1,2}^h} & \cdots & \frac{\partial e_2}{\partial w_{s^h,R}^h} & \frac{\partial e_2}{\partial w_{1,1}^o} & \cdots & \frac{\partial e_2}{\partial w_{1,s^h}^o} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial e_N}{\partial w_{1,1}^h} & \frac{\partial e_N}{\partial w_{1,2}^h} & & \frac{\partial e_N}{\partial w_{s^h,R}^h} & \frac{\partial e_N}{\partial w_{1,1}^o} & & \frac{\partial e_N}{\partial w_{1,s^h}^o} \end{bmatrix}, \tag{19}$$

consequently,

$$J = [J^h \quad J^O]. \tag{20}$$

For each training data point (each pattern), we have an error, and for each error per network weight ($w^T = [w^h \quad w^o]$), one row of the Jacobian matrix is formed. Therefore, for each q-th training data, we will have:

$$J(q) = [J^h(q) \quad J^O(q)]. \tag{21}$$

Now let w^k be a vector of the composed weights obtained at the k-th iteration, then the weights vector w^{k+1} of the (k+1)-th iteration can be expressed as

$$w^{k+1} = w^k + \Delta w. \tag{22}$$

The formula to calculate the weight increment of the neural network is obtained after solving the extreme values of the second order Taylor formula of E(w) by the least square method as follows:

$$\Delta w = -(J^T(w)J(w) + \mu I)^{-1} J^T(w) e(w), \tag{23}$$

where I is the identity matrix and μ the learning rate.

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as

$$H = J^T J, \tag{24}$$

and the gradient can be computed as follows:

$$g = J^T e, \tag{25}$$

where J is the Jacobian matrix that holds the first derivatives of the network errors with respect to the weight and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique [26] that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$X_{k+1} = X_k - \Delta w = X_k - (J^T J + \mu I)^{-1} J^T e. \tag{26}$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes a gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm. Further and detailed explanation of the employed LMA can be found in the work proposed by some authors in [27, 28].

In summary, the specific steps needed to predict and select the right PER based on the SNR and the BW are as follows:

1) Start by choosing the initial parameters for the training error Ɛ, μ_0, θ and the weight w^0. Then let k=0 and $\mu = \mu_0$.

2) Then the error output and network index function $E(w^k)$ are calculated using equation 13.

3) The Jacobian matrix J(w) of the partial derivative e(w) is calculated using equation 19.

4) Calculate the weight increment of the network Δw using equation 23.

5) Then the error output and network index function $E(w^k)$ are calculated using equation 13.

6) If $E(w^k) < \varepsilon$ go to step 8. Else calculate w^{k+1} and $E(w^{k+1})$ using equation 22 and 13, respectively.

7) If $E(w^{k+1}) < E(w^k)$, let k=k+1, $\mu = \frac{\mu}{\theta}$, go back to step 2. Otherwise $w^{k+1} = w^k$, then μ= θμ and go to step 4.

8) Stop.

VI. MODEL TESTING AND EVALUATION

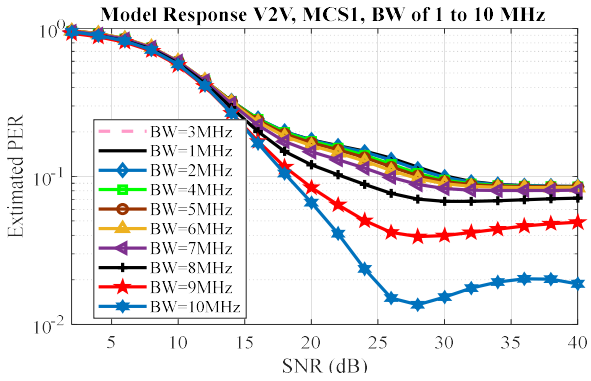

Figure 10. Estimated PER response from the model.

To investigate the model's response in terms of SNR and PER, Figures 10 and 11 were generated. Fig. 10 illustrates the estimated PER response from the model, while Fig. 11 shows a magnified version of the PER response.

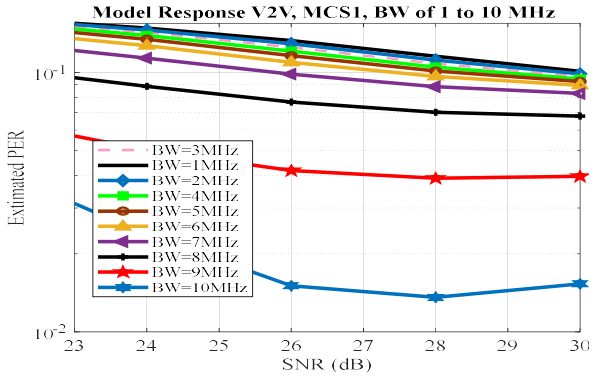

Figure 11. Magnified PER response from the model.

Figures 10 and 11 were generated for channel bandwidths ranging from 1 to 10 MHz. This range is crucial because initially, due to default constraints in MATLAB for V2V communication channels, it was practically infeasible to compute PER for bandwidths of 1 to 4 and 6 to 9 MHz, as MATLAB's default bandwidth values are 5, 10, 20 MHz, etc. However, it was imperative to explore the response across the entire range of 1 to 10 MHz to understand how bandwidth variation affects PER from 1MHz to the dedicated channel bandwidth of 10 MHz for V2V communication. This figure comprises 10 curves, each corresponding to a specific bandwidth with SNR ranging from 2 to 40 dB. The analysis of this figure reveals minimal variation in PER for bandwidths less than 7 MHz when SNR exceeds 30 dB. However, the magnified view in Fig. 11 clearly shows noticeable differences in estimated PER from 1 to 10 MHz within the SNR range of 23 to 30 dB. Additionally, when the bandwidth exceeds 7 MHz, a significant difference in PER estimation becomes apparent. This observation aligns with communication theory fundamentals, indicating that higher bandwidth leads to lower PER.

Now in view to compare the predicted response of the generated model in Fig10 with that used to create the model (actual) in Fig. 2, Figures 12,13 and 14 were computed.

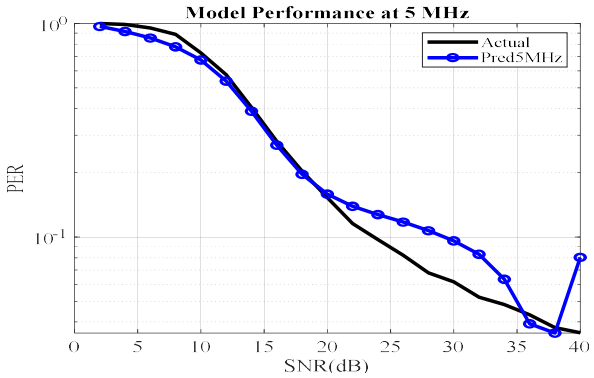

Figure 12. Model Performance at 5 MHz.

Figures 12 and 13 represent the performance of the model at 5 MHz and 10 MHz bandwidths, respectively. The 5 and 10 MHz bandwidths were chosen because those were the only default MATLAB values available to create the model initially. In both Figures 12 and 13, “Actual” represents the recorded data used to create the model, whereas “Pred5MHz" and "Pred10MHz” represent the currently predicted values generated by the created model.

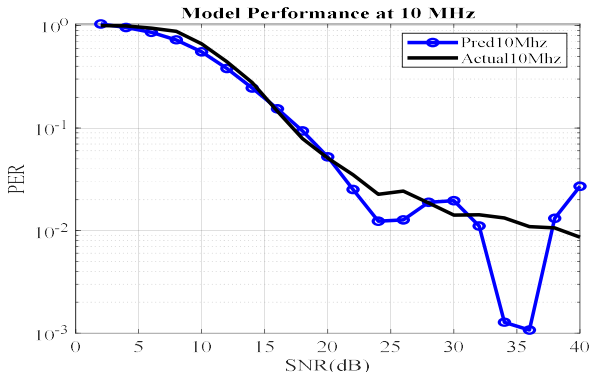

Figure 13. Model Performance at 10 MHz.

Looking at Fig. 12, it can be observed that the predicted value aligns well with the actual data, despite some small discrepancies when the SNR becomes greater than 20 dB. This discrepancy is acceptable because it indicates that the predicted value is slightly lower than the actual value, indicating underperformance of the model in this SNR range. The analysis of Fig. 13 shows that the model performs well when the SNR is below 20 dB. However, from 20 dB upward, the predicted response slightly overperforms the actual data. Moreover, it is important to know the magnitude of these discrepancies in order to understand the tolerance or margin of error embedded in our model. To quantify these discrepancies, Fig. 14 was computed. In Fig. 14, the error or discrepancy curves of both 5 MHz and 10 MHz bands are depicted. The 5 MHz bandwidth is represented in black, while the 10 MHz bandwidth is displayed in blue.

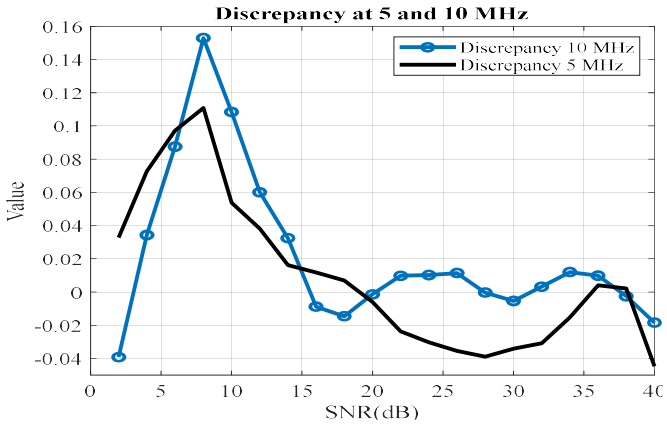

Figure 14. Performance Discrepancy.

The analysis of Fig. 14 reveals that contrary to the observations in Figures 12 and 13, the maximum error occurs below the 20 dB SNR value, with the minimal error occurring when the SNR is beyond 20 dB. This is justified by the fact that, using a logarithmic scale in PER, the curves indicate higher PER values at lower SNR ranges, and a slight discrepancy in this range is also amplified. This underscores the importance of computing Fig. 14. Additionally, it can be observed that the maximum error occurs between the 5 dB to 15 dB SNR range in both bandwidths. In general, it is noticeable that the maximum discrepancy of the entire model is 15%, as depicted in Fig. 14. Based on this value, we can confidently deduce that the developed model is capable of providing a valid PER prediction with a maximum tolerance of 15%. Therefore, it can be concluded that the developed model is 85% accurate at all times.

VII. CONCLUSIONS

This study introduces a model designed to predict the Packet Error Rate (PER) in V2V communication. The prevalence of broadcast storms in VANETs, caused by frequent transmissions, can rapidly deplete channel resources, hindering communication. To optimize transmission decisions and reduce the risk of packet loss, this study proposes a model that predicts PER based on factors like Signal-to-Noise Ratio (SNR) and available bandwidth derived from the Distributed Congestion Control (DCC). The computation of PER is inherently intricate, influenced by various factors that are difficult for individual devices to measure directly. Consequently, this study aims to streamline this process by developing a PER prediction model using Machine Learning (ML) techniques. The model is then developed and validated through a combination of data simulation, analysis, and testing. This ensures the model effectiveness and efficiency in determining PER based on SNR and available bandwidth in the vehicular network. The evaluation demonstrates that the developed model has the ability to predict PER with 85% accuracy.

This work's primary contributions include:

- Generating a valid dataset through model simulation.
- Exploring and analyzing complex data.
- Constructing a robust Neural Network (NN) model from the validated dataset.
- Utilizing NN to model intricate V2V PER estimation.

References

[1] U. S. Department of transportation, *Vehicle Safety Communications Project V Final Report*, U.S.: Rep. DOTHS 810591 Nat. Highway Traffic Safety Admin, 2006.

[2] C-ITS Platform, Final report, January 2016. [Online]. Available at: https://transport.ec.europa.eu/document/download/9aa3fd20-c439-4843-bf1d-3d3a1ec6be42_en?filename=c-its-platform-final-report-january-2016.pdf.

[3] Global Infrastructure Hub, "Vehicle to Vehicle (V2V) connectivity", 05 02 2024. [Online]. Available: https://cdn.gihub.org/umbraco/media/3176/1-vehicle-to-vehicle-v2v-use-case.pdf.

[4] J. Koon, "Competing V2V Technologies Emerge, Create Confusion," 05.01.2023. [Online]. Available at: https://semiengineering.com/competing-v2v-technologies-emerge-create-confusion/.

[5] NHTSA (US DoT), "Driver Assistance Technologies," 05.02.2024. [Online]. Available at: https://www.nhtsa.gov/vehicle-safety/driver-assistance-technologies#nhtsa-in-action.

[6] WHO, "Global status report on road safety 2023," Geneva; 2023. Licence: CC BY-NC-SA 3.0 IGO, 05 02 2024. [Online]. Available at: https://iris.who.int/bitstream/handle/10665/375016/9789240086517-eng.pdf?sequence=1.

[7] P. Fazio, M. Tropea and F. De Rango, "A novel PER degradation model for VANETs," *Proceedings of the IEEE Communications Letters*, vol. 19, no. 5, pp. 851-854, 2015. https://doi.org/10.1109/LCOMM.2015.2399294.

[8] F. Zeng, C. Li and H. Wang, "Performance evaluation of different fading channels in vehicular ad hoc networks," *Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea, 2018, pp. 356-361. https://doi.org/10.1109/ICTC.2018.8539400.

[9] F. Jameel, Faisal, M. A. A. Haider and A. A. Butt, "Performance analysis of VANETs under Rayleigh, Rician, Nakagami-m and Weibull fading," *Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, Islamabad, Pakistan, 2017, pp. 127-132. https://doi.org/10.1109/C-CODE.2017.7918915.

[10] C. Tripp-Barba, L. Urquiza-Aguiar, J. Estrada, J. A. Aguilar-Calderón, A. Zaldívar-Colado and M. A. Igartua, "Impact of packet error modeling in VANET simulations," *Proceedings of the 2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST)*, Ota, Nigeria, 2014, pp. 1-7. https://doi.org/10.1109/ICASTECH.2014.7068133.

[11] F. Abrate, A. Vesco, and R. Scopigno, "An analytical packet error rate model for WAVE receivers," *Proceedings of the 2011 IEEE Vehicular Technology Conference (VTC Fall)*, Sept. 2011, pp. 1–5. https://doi.org/10.1109/VETECF.2011.6093093.

[12] IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," pp. 1–2793, 2012.

[13] O. N. Koç and E. Maşazade, "Packet loss rate prediction for vehicular networks with regression methods," *Proceedings of the 2022 30th Signal Processing and Communications Applications Conference (SIU)*, Safranbolu, Turkey, 2022, pp. 1-4. https://doi.org/10.1109/SIU55565.2022.9864805.

[14] T. H. Ahmed, J. J. Tiang, A. Mahmud, C. Gwo-Chin, D. T. Do, "Evaluating the performance of proposed switched beam antenna systems in dynamic V2V communication networks," *Sensors*, vol. 23, 6782, 2023. https://doi.org/10.3390/s23156782.

[15] F. Zeng, C. Li and H. Wang, "Performance evaluation of different fading channels in vehicular ad hoc networks," *Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea, 2018, pp. 356-361. https://doi.org/10.1109/ICTC.2018.8539400.

[16] Q. Yang, J. Zheng, and L. Shen, "Modeling and performance analysis of periodic broadcast in vehicular ad hoc networks," in Proc. IEEE GLOBECOM, Houston, TX, USA, pp. 15, 2011.8 VOLUME XX, 2017

[17] F. Jameel, Faisal, M. A. A. Haider and A. A. Butt, "Performance analysis of VANETs under Rayleigh, Rician, Nakagami-m and Weibull fading," *Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, Islamabad, Pakistan, 2017, pp. 127-132. https://doi.org/10.1109/C-CODE.2017.7918915.

[18] T. A. Shanmugasundaram and A. Nachiappan, "Impact of Doppler shift on the performance of RS coded non-coherent MFSK under Rayleigh and Rician fading channels," *Proceedings of the 2013 International Conference on Human Computer Interactions (ICHCI)*, Chennai, 2013, pp. 1-5. https://doi.org/10.1109/ICHCI-IEEE.2013.6887819.

[19] MathWorks, "What Is Neural Network? 3 things you need to know," 15 05 2023. [Online]. Available at: https://www.mathworks.com/discovery/neural-network.html.
[20] The MathWork, inc, 1994-2024. 802.11p Packet Error Rate Simulation for a Vehicular Channel. [Online]. Available at: https://www.mathworks.com/help/wlan/ug/802-11p-packet-error-rate-simulation-for-a-vehicular-channel.html.
[21] IEEE Std 802.11p-2010: IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments, IEEE, New York, NY, USA, 2010.
[22] J. A. Fernandez, D. D. Stancil and F. Bai, "Dynamic channel equalization for IEEE 802.11p waveforms in the vehicle-to-vehicle channel," *Proceedings of the 2010 48th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, 2010, pp. 542-551. https://doi.org/10.1109/ALLERTON.2010.5706954.
[23] J. Kaur, "Deep learning vs machine learning vs neural network - What's the difference?", 17.11.2022. [Online]. Available at: https://www.xenonstack.com/blog/deep-learning-vs-ml-vs-neural-network.
[24] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431-441, 1963. https://doi.org/10.1137/0111030.
[25] M. T. Hagan and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1999. https://doi.org/10.1109/72.329697.
[26] M. Hagan, H. Demuth and M. Beale, *Neural Network Design*, Boston, MA: PWS Publishing, 1996.
[27] E. A. Feukeu and M. Sumbwanyambe, "Using neural network and Levenberg–Marquardt Algorithm for link adaptation strategy in vehicular ad hoc network," *IEEE Access*, vol. 11, pp. 93331-93340, 2023. https://doi.org/10.1109/ACCESS.2023.3309870.
[28] E. Feukeu, & S. Mbuyu, "Machine learning algorithm for a link adaptation strategy in a vehicular ad hoc network," *Inteligencia Artificial*, vol. 26, issue 72, pp. 146–159, 2023. https://doi.org/10.4114/intartif.vol26iss72pp146-159.

***DR ETIENNE ALAIN FEUKEU** is a senior lecturer at the University of South Africa (UNISA). He is a member of IEEE and a senior member of SAIEE. He has completed his PhD in 2015 from Tshwane University of Technology and postdoctoral studies from the Vaal University of Technology, University of Cape Town, and the University of South Africa. He has a dual Master in telecommunication, Electrical and Electronic Systems with ESIEE of Paris. He has also received several research awards from National Research Foundation of South Africa. He was also a research fellow at Aalborg University. His field of expertise include wireless network, analysis, development, and design of link adaptation algorithms in WLAN, MIMO systems, Vehicular Ad Hoc Networks (VANETs), doppler shift mitigation, congestion management, Artificial Intelligence (machine learning), Internet of Things (IoT) and Renewable Energies.*

***PROF. MBUYU SUMBWANYAMBE** is a professor at the University of South Africa and is a member of IEEE, SAIEE, and international association of engineers (IAENG). He is the recipient of UJ-commonwealth scholarship. Prof. Mbuyu Sumbwanyambe has a MEng in Electrical and Electronic Engineering from the University of Johannesburg in South Africa and a PhD in Engineering Management from the same University. He has a wide range of industry experience, which he gained in South Africa and Zambia, working as consultant project engineer and a researcher at the University of Johannesburg. He is the head of postgraduate and research in the Department of Electrical and Mining Engineering. He has held workshops with the government on the value of ICTs for social and economic development. He is keen promoter of collaboration between industries, the university, and other universities in the SADC region. Prof. Sumbwanyambe has a multi-disciplinary background and has varied research interests, which include Risk management, Application of Game theory in telecommunication, digital communication, resource management, regulation and policy issues, telecommunication and ICT engineering, economic and social ICT delivery, Energy efficiency and Demand side management and data mining, and information systems evaluation. He has reviewed several papers for IEEE and Taylor and Francis Journals, as well as acted as chair and invited panelist at numerous international conferences. He has contributed to book chapters and published them extensively in journal and conference proceedings.*