

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Adil, Muhammad, Houbing Song, Neeraj Kumar, Mian Ahmad Jan, Amiya Nayak, Ahmed Farouk, and Zhanpeng Jin. "TC-PAA: Deep Learning-Enabled QoS Enhancement Scheme for Cooperative Internet of Vehicles." IEEE Transactions on Vehicular Technology, 2024, 1–13.
<https://doi.org/10.1109/TVT.2024.3396691>.

<https://doi.org/10.1109/TVT.2024.3396691>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

TC-PAA: Deep Learning-enabled QoS enhancement scheme for Cooperative Internet of Vehicles

Muhammad Adil *Member, IEEE*, Houbing Song *Fellow, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, Mian Ahmad Jan, *Senior Member, IEEE*, Amiya Nayak *Senior Member, IEEE*, Ahmed Farouk *Senior Member, IEEE*, Zhanpeng Jin *Senior Member, IEEE*

Abstract—Quality of Service (QoS) plays a pivotal role in numerous delay-sensitive applications that range from general to specific such as the Internet of Medical Things (IoMT), Industrial Internet of Things (IIoT), Unmanned Aerial Vehicles (UAVs), Industrial Automation, and Cooperative Internet of Vehicles (C-IoV), etc. Every application has numerous contributions to human daily life activities, but here in this work, we focused on the C-IoV in the context of QoS metrics. Even though the literature suggested several techniques to address the QoS issues in this emerging technology, but we have not come across a single article that addresses this issue in a cooperative environment, considering the impact of communication congestion and contention by taking into account emergency vehicles and traditional vehicles. Given that, in this paper, we introduce a hybrid framework known as the Traffic Congestion and Priority-Aware Algorithm (TCPAA). This innovative paradigm leverages the capabilities of computer vision, Deep Neural Networks (DNN) and Dijkstra algorithm to strategically incorporate the transmission channels and network entities with an objective to improve the QoS metrics in emergency vehicles. Initially, we developed a dataset with computer vision algorithms "real-time (OpenCV "Background Subtraction") to evaluate and chose the best machine learning algorithms among random forest, support vector machine (SVM), k-means clustering, and DNN. Based on the result statistics, we select DNN, and classified vehicles into two classes: Emergency and traditional vehicles to train the model. Subsequently, we set standard for two type of communications such as regular and prioritized traffic. We incorporate a micro base station (μ BS) in the network for prioritized traffic to facilitate congestion-free communication of emergency vehicles, while the Dijkstra algorithm is used to managed the communication of traditional vehicles. Considering the nature of operation of future autonomous vehicles, we managed most of the decisions processes at the client-side by categorizing the traffic based on the vehicle requirements. Through reliable client-

side management, the high performance and accuracy of TC-PAA underscore its efficiency compared to established field-proven schemes. Adhering to reliability metrics such as latency, packet loss ratio, communication cost, data availability, and traffic priority, the proposed model improves QoS metrics in high-demanding areas of IoV networks.

Index Terms—Internet of Vehicles, Bandwidth Categorization, Quality of Service, Cooperative Intelligent Transportation Systems (C-ITS), Machine learning, Delay sensitive applications, Routing Protocols.

I. INTRODUCTION

In the future Intelligent Transportation Systems (ITS), vehicles will be connected to the Internet and can be controlled from a centralized location. This will allow them to offer various online services such as downloading, streaming, health monitoring, social networking, and many more to their users [1]. This makes the Internet of Vehicles (IoV) even more important, as it combines vehicle ad hoc networks (VANETs) with the Internet of Things (IoT) with the objective of improving the efficiency of this technology [2], [3]. Nevertheless, for IoV networks to be successful, they must meet specific Quality of Service (QoS) requirements to ensure reliable operations for connected vehicles [4], [5]. Thus, it is important to consider the quality of service (QoS) metrics when designing protocols, network topology, and system models for this technology. The literature suggests various ways to improve the communication metrics and address QoS issues of this technology, but advanced solutions tailored to this technology are still needed. IoV relies on Internet connectivity for seamless data sharing, which helps the vehicles to make useful decisions based on collected data and traffic conditions [6], [7]. In terms of network connections, IoV involves extensive data exchange among different components, including vehicle sensors, cluster heads, various base stations, roadside units, and central traffic control centers, which need a reliable communication paradigm [8], [9]. Given that, the objective of QoS optimization in the IoV applications is to facilitate the sharing of precise data within the network in a delay-sensitive environment. This ultimately improves the reliability of these networks, which helps them to achieve more precise results.

Despite stakeholders' relentless efforts in designing new hardware, software, routing protocols, and communication standards, they have struggled to fully meet the new demands of this technology due to its exponential growth in the consumer market. In this paper, we introduce an innovative framework that is known as Traffic Congestion and Priority-Aware

Muhammad Adil is with the Department of Computer Science and Engineering, University at Buffalo, NY 14260, USA. (Email: muhammad.adil@ieee.org)

Houbing Song is with the Department of Information Systems, University of Maryland Baltimore County, USA (Email: h.song@ieee.org).

Neeraj Kumar is with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India (Email: neeraj.kumar@thapar.edu/nehra04@gmail.com)

Mian Ahmad Jan is with the Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, 27272, United Arab Emirates. (Email: mjan@sharjah.ac.ae)

Amiya Nayak is with the School of Electrical Engineering Computer Science University of Ottawa, ON, Canada. (Email: nayak@uottawa.ca)

Ahmed Farouk is with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada 83523, Egypt (Email: ahmed.farouk@sci.svu.edu.eg)

Zhanpeng Jin is with the School of Future Technology, South China University of Technology, Guangzhou 511442, China, and the Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260, USA. (Email: zjin@scut.edu.cn)

Corresponding author: Zhanpeng Jin

Algorithm (TCPAA). This innovative paradigm leverages the capabilities of Computer Vision, Deep Neural Networks (DNN) and Dijkstra algorithm to manage the network topology of this technology, and improve the QoS metrics in emergency vehicles during the communication process. Not only does this approach improve QoS metrics, but it also helps in managing traffic in congested areas, indicating its potential utility in emergencies. The model allows participating IoVs to transmit data using predefined traffic classes such as Emergency and traditional vehicles traffic classes. These traffic classes, categorized as high-priority and ordinary traffic classes with different transmission channels. High-priority traffic is handled via a micro base station (μ BS), while Dijkstra algorithm is used to manage ordinary traffic through hop-count communication in the network.

The key contributions of this work are summarized as follow.

- 1) The aim is to design, implement, and evaluate a Traffic Congestion and Priority-Aware Algorithm (TCPAA) for delay-sensitive IoV applications, with the objective to improve the QoS metrics during communication.
- 2) Initially, the primary task is to manage the dataset that could categorize vehicles on the road into two distinct groups such as emergency vehicles and regular vehicles. For this, we created a dataset from real-time video with computer vision algorithms (using OpenCV "Background Subtraction").
- 3) The next task is to chose the most accurate machine-learning (ML) framework for traffic categorization on the client side based on vehicles classification. For this, we examine and evaluate four ML prototypes: SVM, random forest, K-means, and DNN. Based on the result statistics, we selected DNN for categorizing of traffic into two classes such as priority and ordinary traffic.
- 4) Consequently, we distribute the transmission of categorized data across two distinct channels. The first channel, dedicated to priority data, and was managed by a micro base station. Meanwhile, the second channel catered to the transmission of ordinary class traffic, employing the traditional Dijkstra algorithms to ensure effective communication.
- 5) Finally, we conducted an empirical evaluation of our proposed paradigm to validate its efficiency. We performed a comparative analysis that incorporated parameters such as model training, testing accuracy, communication latency, packet loss ratio, communication cost, data availability, and traffic priority and compared them with the existing traditional protocols.

The remainder of the paper is managed as follows. Section II discusses the related work, while Section III presents the proposed system model. Section IV evaluates various comparative machine algorithms, Section V covers the experimental setup and results such as comparative metrics with other schemes, and Section VI concludes the paper.

II. RELATED WORK

In this section, we reviewed the current literature on the Internet of Vehicles (IoV), focusing on the challenges of

dynamic topology, network congestion, and routing protocols, route selection, and mobility management. In [10], the authors discussed the issues of intelligent transportation systems associated with vehicle mobility and urbanization. Moreover, they suggested possible solutions to address them and improve the utilization of this technology in urban areas. Sodhro et al. [11] extended this discussion, and highlighted the research gaps in present literature. Partovi et al. [12] highlighted the new challenges that arise with the emergence of technologies in IoV with an objective to find ways for their redressal. Moreover, the authors emphasized the question of how we can improve the communication and interconnectivity infrastructure of ITS. Hildebrand et al. [13] continued this conversation by emphasizing the importance of blockchain technology in intelligent transportation systems. Moreover, the authors talk about the important aspects of this technology how it can help to address the communication challenges in autonomous vehicles. They also highlighted the ongoing challenges requiring input from industry experts and scientists. Following this, Wen et al. [14] introduced a wideband dual circularly polarized (DCP) antenna for Intelligent Transportation Systems (ITS) to tackle communication challenges. Their model was specifically designed to enhance receiver sensitivity and overall communication quality within an IoV network. Alaya et al. [15] conducted an in-depth review of current video streaming techniques employed in ITS by targeting the enhancement of various QoS and Quality of Experience (QoE) metrics. This study offers valuable perspectives on existing research, and presents a comparison of QoS and QoE that can guide future developments in C-IoV. Kalsoom et al. [16] suggested a 5G-enabled design for Cooperative Driving Vehicles (CDV) using a device-to-device resource allocation mechanism. Miao et al. [17] proposed a traffic light-centered method for QoS provisioning in IoV. The authors leveraged traffic light signals to schedule and manage the traffic with the objective to reduce the average waiting time.

Raj et al. [18] proposed a novel Aerial Intelligent Relay-Road Side Unit (AIR-RSU) platform for ITS. The goal was to improve the network connectivity, and analyze communication stability among Relay-Road Side Units (RRSU) and participating vehicles. Furthermore, they claimed that the AIR-RSU architecture would be useful in the future ITS. However, the model's complexity and high maintenance cost limit its use in practical scenarios. Reference [19]–[21] discussed the traffic management, standardization, and QoS challenges with future research directions. The stakeholder interested to explore this domain are suggested check these references.

Srinidhi et al. [22] continued the discussion and proposed a Hybrid Energy Efficient and QoS Aware (HEEQA) algorithm. This algorithm was constructed from a combination of Quantum Particle Swarm Optimization (QPSO) and an improved Non-dominated Sorting Genetic Algorithm (NSGA), with an objective to enhance the QoS of ITS networks. Zhu et al. [23] introduce a virtual infrastructure, utilizing an AI-based framework to enhance the "intelligence" of the IoVs. According to the authors, this setup allows the rapid development of AI-based transportation systems on computers, just like real-world transportation systems. Even though the authors

assert its effectiveness, we have reservations about how well the simulated results would match real-world deployments. This is due to various external and internal factors that could impact communication parameters. Bi et al. [24] proposed an incentive scheme for ITS to improve the QoS metrics. Ni et al. [25] devised a novel algorithm aimed at enhancing the QoS metrics within ITS, utilizing the intersection control units to manage traffic within vehicular ad hoc networks. The authors utilized a mutual exclusion approach, which facilitates negotiation among vehicles at intersection points and hand over processes. Moreover, a variety of innovative solutions to this problem have been discussed in references [26]–[29] which could help to address the issue at hand. Despite the numerous efforts of researchers and involved stakeholder, QoS within Cooperative Internet of Vehicles (C-IoV) remains an active field of inquiry, because of the continuous advancement. Therefore, this technology urgently requires a robust QoS paradigm that would not only address QoS impediments effectively but also maintain reliable communication metrics such as end-to-end delay, priority traffic, communication cost, throughput, and packet loss ratio.

III. PROPOSED METHODOLOGY

In this section, we discuss how our proposed framework works. However, before starting the detail discussion. First, we need to set the stage for the question why we selected Deep Neural Network (DNN) over other comparable machine learning algorithms such as random forest, support vector machine (SVM), and k-means, etc. Following our explanation for this choice, we evaluated each algorithm using the same dataset to ensure the reliability of our experimental results, which we will discuss in the upcoming section. Despite these measures, we incorporated an additional network component in the form of a micro base station, designed specifically to handle priority class traffic as categorized by the DNN. At this point, we are moving forward to the proposed paradigm.

A. Data Classification based on Deep Neural Network

In this section, we discuss how the DNN is managed to learn, classify, and work. Let's assume that the incoming signal of raw data (D) is amplified according to the Nyquist criterion because we will be implement this model in the C-IoV networks. Therefore, we want to consider the possible scenarios of real network traffic. Following this discussion, the n^{th} output of the matched filter over an observation period (O_{period}) of length N can be simplified as follows:

$$O_{period} = Tx_{ran(n)} \times (e^{j(\theta_0 + \frac{\pi f_0 n}{N})} \times \alpha) + \omega \quad (1)$$

In equation 1, the incoming signal of length N = 0, 1, 2, 3, 4, . . . , N - 1, whereas the signal amplitude is symbolized with α , and phase offset and normalized frequency offset are generalized with θ and f respectively. Furthermore, we used $Tx_{ran(n)}$ to denote the random transmitted signal with a non-Gaussian noise ($\omega(n)$). To maintain generality in the model, we assumed α , θ , and $Tx_{ran(n)}$ as constants. However, to tackle the uncertainty condition of noise, we incorporated a

time-correlated, non-Gaussian noise condition into the model. And we assume that $\omega(n)$ follows the Gaussian mixture model (M_m) [30], with the probability density function (PDF) represented as P_f :

$$P_f(\omega(n)) = \sum_{m=0}^M \frac{\lambda_m}{\pi \sigma_m^2} \exp\left(-\frac{|\omega(n)|^2}{\sigma_m^2}\right) \quad (2)$$

In equation 2, σ_m^2 signifies the variance of the m^{th} noise component, while the term λ_m is used to symbolize the proportion of noise generated from the m^{th} Gaussian component. Given that, each λ_m value ranges from 0 to 1. In addition, we assumed the use case of only one Gaussian component such as K = 1, then the noise would be $\phi(n) \sim CN(0, \sigma_0^2)$. Similarly, the time-related noise produced in Equation 2 using the autoregressive AR(P) filter, which can be generalized as $a(i)_{i=1}^{Pro}$, when it comes to sequential generation, and could be managed as follows:

$$\omega(n) = - \sum_{i=1}^{Pro} [a(i) \times (n - i)] \times \omega + e(n) \quad (3)$$

For classification purposes, we employed posterior probability (P_{pro}) with a maximum criteria adhering to the Bayes' decision rule. Given that, we assumed, a dataset (D) is made up of N classes, which is partitioned into two groups: $G = (G_r) = [(G_{r_0}, G_{r_1})]$. In this context, a decision is needed to be taken in the form of designated groups. For this, we utilize the undermentioned formula with the posterior probability P_{pro_i} .

$$\hat{P}_{pro_i} = \arg \max_{P_{pro_i} \in G} P_{pro_i}(D|G_r) \quad (4)$$

In equation 4, the posterior probability P_{pro_i} represents the probability of a particular event with prior probability (Pe_{pro_i}), occurring given the observed data D and the hypothesis G_r .

To visually represent this process, we added figure 1 in the paper.

In Figure 1, we illustrate the mapping function (f) in various neural network layers, accompanied by the activation function and hyperparameters. Typically, this paradigm adapts to the characteristics of incoming data samples, conforming to the set classification classes. Subsequently, we trained the DNN classifier using the considered dataset with 'S' samples. Furthermore, we utilized the real (R_{part}) and imaginary (I_{part}) parts of these samples, denoted as S(N), where N signifies the total number of samples.

$$S(N) \Rightarrow (R_{part}, I_{part}) = \text{two-dimensional vector} \quad (5)$$

Following Equation 5, we can confirm the following benefits. The R_{real} and I_{part} parts of the data contains the in-phase and out-of-phase raw data signals respectively, which generally applies to the same independent distribution for a single data input signal. This combination encapsulates the information that the input samples carry, exhibiting universal features relevant to both priority and ordinary classes. As

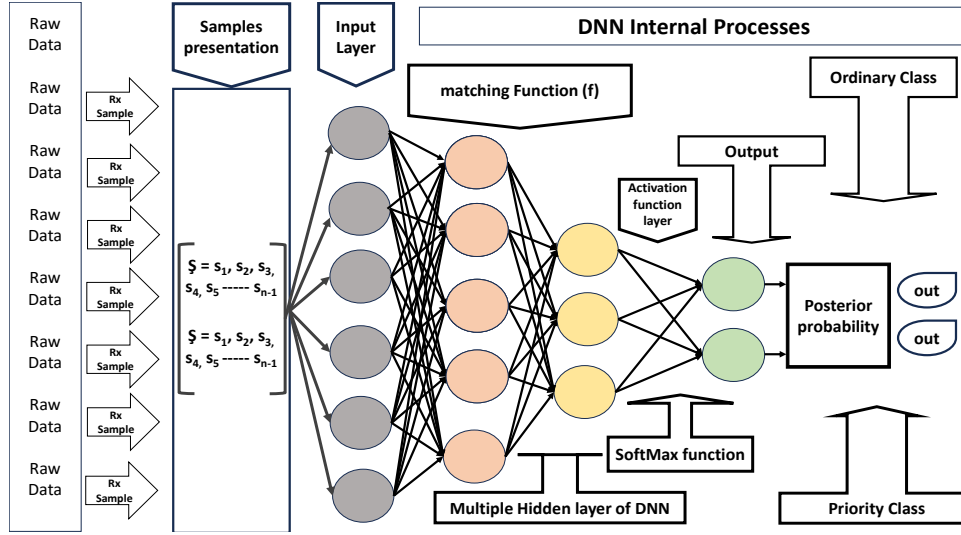


Fig. 1: Architectural design of the proposed system model

illustrated in Figure 1, the sequence of incoming data samples obtained in a single observation can be generalized as follows:

$$S = \begin{bmatrix} s_1, s_2, s_3, s_4 & \cdots & \cdots & s_{(N-1)} \\ s_1, s_2, s_3, s_4 & \cdots & \cdots & s_{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ s_1, s_2, s_3, s_4 & \cdots & \cdots & s_{(N-1)} \end{bmatrix}^T \quad (6)$$

Likewise, we have enlightened the internal steps detailing how a DNN was constructed and how it learns the a posteriori probability function, denoted as P_{pro} , from the input matrix (M_m) of a length (L), where L is a member of the set $1, 2, 3, 4, 5, \dots, L_{n-1}$. Firstly, we mapped the received data samples $S \in S_{N-1}$ to an L -dimensional vector V in the DNN as follows:

$$f : S \in S_{n-1} \rightarrow V \in \mathbb{S}^{(V)}L \quad (7)$$

In equation 7, the function f is determined by an input denoted as S , where $S \in S_{n-1}$. The function f then transforms S into an output V , where $V \in \mathbb{S}^{(V)}L$. This means that f takes something from one group, represented by S_{n-1} , and produces something in another group, which is generalized as $\mathbb{S}^{(V)}L$. The vector V resulting from this process is interpreted with the extracted attributes and passed through a softmax activation (SF_{max}) function to obtain an L -dimensional output vector.

$$V_y = \frac{\exp(V)}{\sum_{i=1}^L \exp(v_i)} \quad (8)$$

According to Equation 8, we can observe that the sum of all elements of V_y equals one for the input data samples, i.e., $\sum_{i=0}^{L-1} V_{y_i} = 1$, and $V_{y_i} \geq 0$ for $i = 1, \dots, L$. Each V_{y_i} can be interpreted as a probability, corresponding to the modeled a posteriori probability associated with the classification scheme of M_m , i.e., $V_{y_i} = \hat{P}(M_m; SF_{max}, \theta|s)$. To efficiently process the incoming dataset samples (S), we employed a classification

matrix (M_{m_j}) and a one-shot vector ($V_{shot} \in S_{L_j}$). In this one-shot vector, the j^{th} element of V_{shot} is set to 1 and 0. For the optimization of θ value, we managed the cross-entropy function (f_{CE}) \Rightarrow equation 9.

$$optim(f_{CE}, \theta) = - \sum_{i=1}^L v_i \log(v_{y_i}) \quad (9)$$

To explore this further, let's assume that the incoming data sample S is generated from the j^{th} classification scheme of the matrix (M_{m_j}). In this case, the classifier has two options for V_j : 1 for priority traffic and 0 for ordinary traffic. Given this, we can replace V_{y_i} with its physical meaning (phy_m) and posterior probability, i.e., $P_{pro}(M_{m_j}; f_{CE}, \theta|s)$, to transform the loss function (f_{loss}).

$$\begin{aligned} optim(f_{CE}, \theta) &= - \sum_{i=1}^L v_i \log \hat{P}_{pro}(M_{m_i}; phy_m, \theta|s_0, s_1) \\ &= - \log \hat{P}_{pro}(M_{m_j}; phy_m, \theta|s_0, s_1). \end{aligned} \quad (10)$$

In accordance with Equation 10, we minimized the loss function f_{loss} , which corresponds to $P_{pro}(M_{m_j}; phy_m, \theta|s)$, a quantity derived from the matrix M_{m_j} and S . Likewise, the match criterion is established using set parameters to facilitate the classification decision. And this can be simplified as follows (equation 11):

$$\hat{M}_{m_i} = \arg \max_{M_{m_i} \in M} \hat{P}(M_{m_i}; phy_m, \theta|s) \quad (11)$$

To summarize, the primary goal during the training phase was to accurately classify traffic based on predetermined labels, using a posterior probability decision process for priority and ordinary traffic. The harmonization of this training approach with decision criterion ensures an efficient and precise classification process. Furthermore, for a more comprehensive understanding, we have included Algorithm 1 in the paper.

Algorithm 1 Steps: How DNN Process Data in our work

Require: Accurate data classification of two classes

```

1: Start
2: Input  $\Rightarrow$  Raw-data
3: samples  $\Rightarrow S = s_1, s_2, s_3 \dots s_{n-1}$ 
4: Data Classification  $\leftarrow$  feature matching
5: Predefined classes  $\leftarrow$  two: 0 and 1
6:   If (traffic class  $\in$  1)
7:     then
8:       Categorize  $\leftarrow$  priority class
9:     Else
10:      Categorize  $\leftarrow$  ordinary class
11:   End If
12: return Two traffic classes: Priority and ordinary

```

Important Note: *In this work, we responded to the potential questions that are practical, as well as those we anticipate from reviewers and practitioners in this field.*

Question Number 1

Why did we choose the DNN model in presence of other machine learning algorithms?

(We answered this question in section-IV:) Nevertheless, it's important to note that the selection and optimization of a model is a tedious task and requires significant effort, which we have done. Yet, we aim to further extend the applicability of this model to practical applications of C-IoV technology, with the objective of enhancing its productivity while maintaining the trust of all involved stakeholders.

Question Number 2

Is this the contribution of this work?

Answer: No, our approach involved applying the output of the DNN to the Internet IoV network, with the objective of enhancing the communication metrics. This is particularly crucial for vehicles such as police cars, fire brigades, ambulances, and other vehicles that require the sharing of information in a delay-sensitive environment. Given that, we compared our results with those from traditional routing protocols to verify the reliability and effectiveness of our paradigm."

IV. COMPARISON OF RIVAL ALGORITHMS: JUSTIFICATION FOR OUR SELECTION

In this section, we discuss and compare three machine learning algorithms—K-Means [31], Random Forest [32], and Support Vector Machine (SVM) [33]—utilizing the experimental results to justify our selection of the Deep Neural Network (DNN) for this work. We assessed and evaluated the performance of each algorithm using a new dataset created from traffic videos. This helped us justify why we chose DNN over the other three algorithms.

A. Classification Accuracy of the K-Means Algorithm

In this section, we briefly explore the k-Means clustering method, an unsupervised learning technique, in the context of our proposed model. Our objective is to validate why this model is not as effective as our selected DNN approach. Initially, we categorized the incoming traffic of the dataset into different classes (k clusters) utilizing the k-Means algorithm, where k represents two pre-defined traffic classes. The following steps outline this process. Subsequently, the k-Means algorithm chooses k random classes from the incoming training dataset to serve as the centroids of the clusters (i.e., $Cl_1, Cl_2, Cl_3, Cl_4, \dots, Cl_k$). These clusters are then classified into two groups: priority traffic cluster Cl_{prio} and ordinary traffic cluster Cl_{ordi} . For each incoming chunk of the training instance X, we compute the Euclidean distance (ED) from X to each cluster centroid Cl_i , where i ranges from 1 to k. Next, we identify the nearest cluster centroid Cl_q to X, which is the arithmetic mean (AM) of the instances in the cluster. We repeat this process until the centroids of clusters $Cl_1, Cl_2, Cl_3, Cl_4 \dots Cl_k$ to stabilize or minimize the sum of squared errors (SSE).

$$SSE = \sum_{i=1}^n (X_i - C_q)^2 \quad (12)$$

Here in equation 12, X_i represents the i^{th} term of data point, and Cl_q is the centroid of the cluster to which the i^{th} data point is assigned. For every test Z, we calculated the Euclidean distance $ED(Cl_i, Z)$, where i ranges from 1 to k, with an objective to find the cluster Cl_r closest to the Z. Following this, we classified Z as either priority or ordinary traffic class using the Threshold rule (TR). The rules for classifying a test Z instance as a part of cluster Cl_r using the Threshold method would be summarized as follows:

$$Z = \begin{cases} 1, & \text{if } Pr(\omega_{k \rightarrow 1} | 1_{Cl_q} | Z \in Cl_r) > \Theta; \\ 0, & \text{otherwise.} \end{cases}$$

Herein, "1" and "0" denote the priority and ordinary traffic classes, respectively. The term $Pr(\omega_{k \rightarrow 1} | 1_{Cl_q} | Z \in Cl_r)$ signifies the probability of instances within the " Cl_r ", while " Θ " symbolizes the predetermined threshold level. For the evaluation, we set the threshold level at 0.5 to classify the traffic into two distinct classes: 1 and 0. Consequently, we set the Bayes Decision Rule as follows:

$$Z = \begin{cases} 1, & \text{if } Pr(\omega_{k \rightarrow 1} | 1_{Cl_q} | Z \in Cl_r) > P(\omega_{.5} | Z \in Cl_r) \\ 0, & \text{otherwise} \end{cases}$$

1) Statistical Analysis of K-Means Results: In this segment, we have explored the evaluation process of the K-Means algorithm by considering our desired dataset. Initially, we copied the necessary dependencies and clean up the data. Next, the features in the data are normalized by dividing them by 255. Consequently, we split the dataset into a training set and a validation set, with 80% and 20% respectively, using a random seed of 42 to ensure reproducibility. Thereafter, we enabled the k-means algorithm to store the inertia (or loss) of the model on the training and validation sets, for different

numbers of clusters. For this, we tested 5 to 15 clusters to measure and evaluated that how internally coherent the clusters are. For each number of clusters, the fits method of the K-means algorithm during the training data computes the inertia on the training set, transforms the validation data to cluster-distance space, and computes the inertia on the validation set. The results garnered from this process for training losses and validation losses are comprehensively visualized in Figure 2.

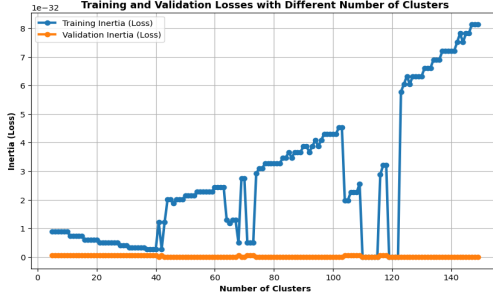


Fig. 2: Visual Representation of Training and Validation Results Using K-Means Algorithm

B. Classification accuracy of Random Forest Algorithm

In this subsection, we talk about the random forest algorithm that had been evaluated as a comparative algorithm, while choosing the DNN paradigm for this work. In order to substantiate the usefulness of the Random Forest Algorithm (RFA), we used the considered dataset as input data with the two output to design a model that will classify the traffic into two classes such as priority and ordinary traffic (For Emergency vehicles and Traditional vehicles). Initially, the RFA predicts the output value based on the inputs. Following this, it seeks to understand the relationship between the input variables and the output.

Let's consider a dataset (D) that can be defined as $D = (x_i, y_i), i = 1, 2, 3, 4, \dots, n-1$, where $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip})$. The objective is to build a model f' such that $Y' = f'(X)$ for a set of random variables $X = (X_1, X_2, X_3, \dots, X_p)$ and Y . We used \hat{f} for predicting the value of the response variable based on the predictors: $\hat{y}_0 = \hat{f}(x_0)$, where $x_0 = (x_{01}, x_{02}, x_{03}, \dots, x_{0p})$ trees. This signifies that the estimated response \hat{y}_0 is selected by the function \hat{f} applied to the predictor vector x_0 . In classifying dataset D, we adopted the Gini Classification criteria in our experimental process. We formulated D as follows: $D = N_L \sum_{k=1}^K p_{kL_C} (1 - p_{kL_C}) + N_R \sum_{k=1}^K p_{kR_C} (1 - p_{kR_C})$, where p_{kL_C} represents the proportion of class k in the left node and p_{kR_C} denotes the proportion of class k in the right node to ensure the categorization of data into two traffic classes. This process is illustrated in Figure 3 for a more comprehensive understanding of the data classification method.

1) Statistical Analysis of Random Forest Algorithm Results::

In this section, we have evaluated the Random Forest algorithm on the considered dataset to analyze the results.

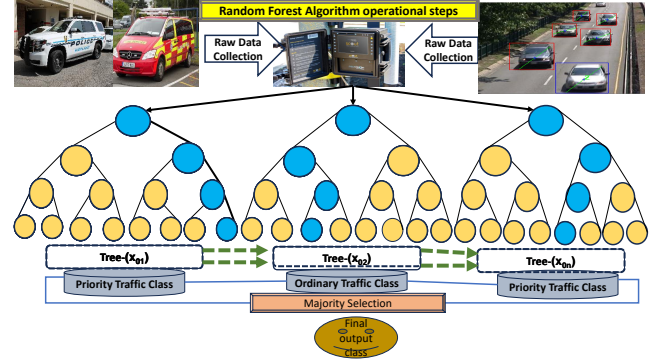


Fig. 3: Operational Steps and Tree Generation in the Random Forest Algorithm

Initially, we defined the features for input data and target variables for classification. Next, we transformed the target variable into a binary format, where values less than 0.5 are labeled as 0, and values greater than or equal to 0.5 are labeled as 1 (Emergency vehicles and Traditional vehicles). The dataset was then split into training and validation sets, with a train and test size of 0.8 and 0.2, respectively. For reproducibility, we used the random state of size 42. Next, we initialized empty lists to store accuracy and loss data for different numbers of trees in the Random Forest Classifier. For each value of n (number of trees), we trained a Random Forest Classifier. The accuracy of the classifier on the training set was calculated and stored. Similarly, the accuracy of the validation set was calculated to assess the model's reliability. We also computed the log loss on both the training and validation sets and saved the results. Finally, we presented the results graphs of training accuracy and validation accuracy followed by training losses and validation losses in Figures 4 and 5.

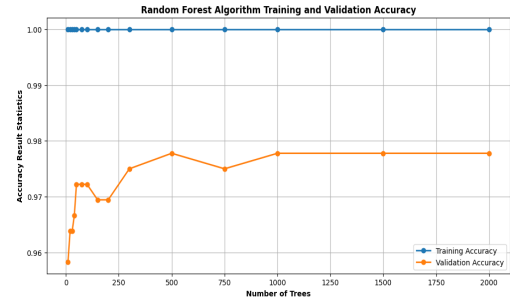


Fig. 4: Random Forest Algorithm Training and Validation Accuracy Statistical Analysis

C. Support Vector Machine

Support Vector Machine (SVM) is a widely used machine learning algorithm for binary classification that maps data from a lower-dimensional space to a higher-dimensional space using a kernel function, allowing for linear separation of complex problems [34]. In this segment, we aim to categorize the created dataset into two classes using SVM by taking advantage of its excellent classification capabilities. However, before diving into the technical details, let's familiarize the

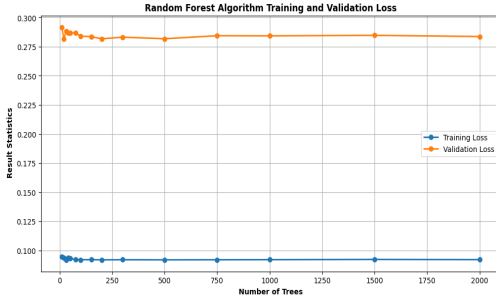


Fig. 5: Random Forest Algorithm Training and Validation Losses Statistical Analysis

reader with how SVM works in the context of this study. Given that, we categorized the considered dataset in terms of traffic classes. These traffic classes are classified into priority and ordinary traffic classes, i.e., PC_{tr} and OC_{tr} . Furthermore, the identical classification metrics $CM \in (Y_1, Y_2, PC_{tr}, OC_{tr})$ demonstrate the classification of data (traffic classes) in a time slot (ΔT). Here, Y_1 and Y_2 represent the output of the model. To effectively employ the confusion matrix (CM) as multi-classifiers — CM_1, CM_2, CM_3 , and CM_4 — we fed each sub-matrix into a SVM for training. This yielded four classifiers: $CL_{00,01}, Y_{00,10}, CL_{00,11}$, and $Y_{01,10}$. We then utilized $CL_{00,01}$ and $Y_{00,10}$ (Y_1, PC_{tr}) and $Y_{00,11}, Y_{01,10}$ (Y_2, OC_{tr}) as sub-metrics in the SVM algorithm. This allowed the SVM to prioritize classifiers under $CL_{00,01}$ and $CL_{00,11}$, represented by 1 as a priority traffic class. Conversely, the SVM recognized classifiers under $CL_{00,11}$ and $CL_{01,10}$ as an ordinary traffic class, which is represented by 0. The primary goal of the SVM is to identify an optimal hyperplane using a set of classifiers, thereby categorizing data according to predefined parameters.

A linear function, which is symbolized by (ω) initially classifies data into defined sub-matrices. Given that, we assumed an optimization problem where data, such as $(CL_{00,01}, CL_{00,10})$, is linearly separable. A slack variable support in applying this to incomplete linearly separable models. The hypothesized hyperplane is as follows:

$$CM = TR_{SVM}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2) \quad (13)$$

In equation 13, $TR_{SVM}^{\Delta T}$ represents the traffic categorization time slot with linear function (ω) such as PC_{tr} and OC_{tr} , followed by biased vector (B_{SVM}) to allocate the required channel i.e. Y_1 or Y_2 in the hyperplane. Thus, the decision function (D_f) can be formalized as below:

$$D_f(\omega)(Y) = TR_{SVM}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2) \quad (14)$$

Now, we have set the steps for traffic categorization for the collected data such as $(PC_{tr}, OC_{tr}) \in T_R$ (received data (D)). Thus, we can write it as below:

$$\begin{aligned} D_f(\omega)(PC_{tr}) &\geq 0 \text{ if } CM = 1 \\ D_f(\omega)(OC_{tr}) &\leq 0 \text{ if } CM = 0 \end{aligned} \quad (15)$$

Next, we have considered the distance metric, which is symbolized by (D_s) for the geometric margin. Hence, the traffic class in the hyperplane can be expressed as below:

$$D_s = \frac{CM(TR_{SVM}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2))}{\|T_{R_{SVM}}\|} \quad (16)$$

In order to define, the distance between traffic vector to hyperplane, we have $D'_s = (\min_s, D_s)$, where the maximization margin for the given scenario would be equivalent to the following formula:

$$\max_{T_{R_{SVM}}, B_{SVM}} (D'_s) = \max_{T_{R_{SVM}}, B_{SVM}} (\min_{D_s} (D_s)) \quad (17)$$

It is noteworthy that when $T_{R_{SVM}}$ and B_{SVM} scale up, the value of D_s in the hyperplane remain the same. Thus, the value of $T_{R_{SVM}}$ and B_{SVM} can be selected with an appropriate scale, which will satisfy the hyperplane such as $T_{R_{SVM}}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2) \geq 1$ for $CL_{00,01}, CL_{00,10} = Y_1, PC_{tr}$. Now, at this stage, the D_f will satisfy the following two conditions:

$$\begin{aligned} D_f(\omega)(PC_{tr}) &\geq 1 \text{ if } CM = 1 \\ D_f(\omega)(OC_{tr}) &\leq 1 \text{ if } CM = 0 \end{aligned} \quad (18)$$

According to equation 18, when the data (traffic) $T_{R_{SVM}}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2) \geq 1$, we can get $D_s = \frac{1}{\|T_{R_{SVM}}\|}$. In addition, equation 17 demonstrates that the minimization and maximization functions are equivalent for margin function, which can be simplified as $(\|\min_{T_{R_{SVM}}}\|^{-1})$ and $(\|\max_{T_{R_{SVM}}}\|^{-2})$ respectively. So, we can write the optimization problem O_p in the presence of equation 17 as below:

$$O_p = (\min_{T_{R_{SVM}}, B_{SVM}}) \left(\frac{1}{2} \|\min_{T_{R_{SVM}}}\|^{-1} \right) \quad (19)$$

$$\forall CM = T_{R_{SVM}}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2) \geq 1$$

Furthermore, we utilized the soft margin SVM (SM_{SVM}) to allow those classifications that can be effectively managed in the event of triggering situation. However, there is always the probability of misclassification (M_q) in any model. Therefore, the soft margin SVM with the probability of M_q is formalized as below:

$$SM_{SVM} = \min_{T_{R_{SVM}}, B_{SVM}} \left(\frac{1}{2} \|\min_{T_{R_{SVM}}}\|^{-1} + M_q \sum_{D_s} \Psi_{ds} \right) \quad (21)$$

Followed by the consequent equation.

$$\begin{aligned} SM_{SVM} &= TR_{SVM}^{\Delta T}(\omega)(PC_{tr}, OC_{tr}) \\ &+ B_{SVM}(Y_1, Y_2) \geq 1 - \Psi_{ds} \quad (\text{where } \Psi_{ds} \geq 0) \end{aligned} \quad (22)$$

The slack variable is symbolized with notation Ψ in equation 22, which is further defined as $\Psi = \max(0, 1 - T -$

$R_{SVM}^{\Delta T} \omega(PC_{tr}, OC_{tr}) + B_{SVM}(Y_1, Y_2)$ to keep generality in the model. In addition, the penalty parameter is symbolized as M_q , where the value of M_q is set as a constant $Pn_q = 0$, to avoid extra computation. Furthermore, variation in the value of Pn_q means less or more errors in the proposed model. Given that, the optimization dilemma of equations 19 is resolved through the Lagrange multiplier method, which is specifically used to eliminate these constraints. The Karush-Kuhn-Tucker situation is adopted to transform the original dilemma into a twofold dilemma, where the sequential optimization algorithm is used to solve the expected problem. Finally, we obtained the traffic vector ($T_{R_{SVM}}^*$) and bias vector for segregated classes (Y_{SVM}^*) through the SVM. Thus, D_f can be rewritten as:

$$D_f(\omega)(Y) = T_{R_{SVM}}^* (\omega)^{\Delta T} (\omega)(PC_{tr}, OC_{tr}) + B_{SVM}^*(Y_1, Y_2) \quad (23)$$

Similarly, during the decision process, when the value of $D_f(\omega)(Y^{test})$ equals 1, then the data is prioritized; however, if it equals 0, then it is treated as ordinary.

1) Statistical Analysis of Support Vector Machine Results:

In this section, we have discussed how the performance of the SVM model is evaluated and which factors are taken into consideration for this evaluation. First, we loaded the dataset, and transformed the target values into binary labels using a predetermined threshold parameter. We partitioned the data into an 80% and 20% split for training and validation, respectively, ensuring a stratified split based on the target labels. We employed the Linear.SVC model from sklearn.svm and adjusted the cost parameter 'C' over a range from 0.0001 to 10000, with some noise to examine its impact on model performance. For each 'C' value, we trained the model, saved its coefficients, and made predictions on both training and validation sets. Finally, we computed the accuracy scores for these predictions to assess the performance of the model under varying cost parameters. Additionally, we determined the accuracy and validation losses to provide comprehensive insights into our model's performance, thereby setting the foundation for our DNN implementation. Moreover, the results obtained during simulation are shown in Figures 6 and 7.

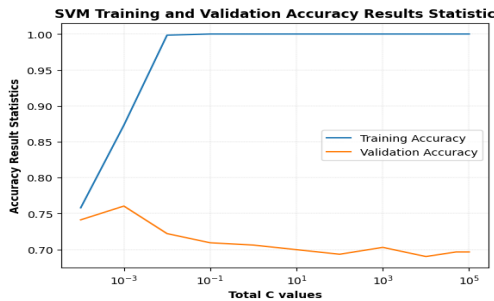


Fig. 6: Graphical Depiction of Training and Validation accuracy of SVM

2) Statistical Analysis of DNN:

In this section, we examine the criteria, how the DNN model's is evaluated for performance metrics and other influencing factors. During implementation, we used TensorFlow's

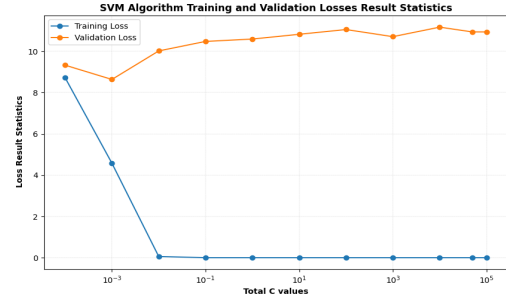


Fig. 7: Graphical Depiction of Training and Validation Losses Through SVM

Keras API for emergency and traditional vehicle classification. The network accepts "256 × 256 × 3256 × 256 × (3)" input images with 3 RGB channels. The model is composed of four convolutional blocks such as 1. 2D convolutional layer (Conv2D) with the activation function set as ReLU. 2. Batch normalization is used to stabilize the activations and expedite training. 3. Max-pooling layer of a pool size 2×2×2 is used to downsample the spatial dimensions by half after every block. 4. The number of filters in the convolutional layers starts at 32 and doubles in each subsequent block until it reaches 256 in the fourth block. After the convolutional blocks, the spatial feature maps are flattened into a 1D vector such as a dense layer of 512 nodes with ReLU. dropout regularization rate of 0.1 to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training time, and a final dense layer with a single neuron is used with a sigmoid activation function to make binary decisions for classification. The model employs an exponential decay learning rate scheduler, which is initially managed with a learning rate of 0.001, and it decays at a rate of 0.7 every 1000 steps. Likewise, the Adam optimizer is utilized with the previously defined learning rate schedule to adjust weights during training. Finally, the model is compiled using the binary cross-entropy loss function with the objective to make it suitable for binary classification tasks. Additionally, the accuracy metric is tracked during training. For performance evaluation, we check the training and validation accuracy followed by training and validation losses, as shown in Figure 8.

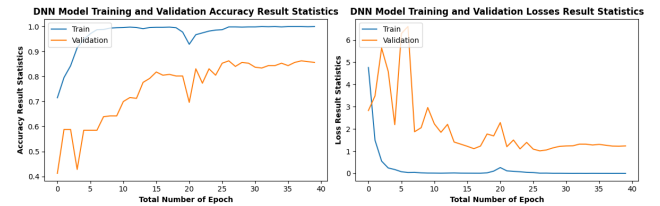


Fig. 8: Graphical Depiction of Training and Validation accuracy and Losses of DNN

V. EXPERIMENT SET-UP AND RESULTS DISCUSSION

In this section, we explore the operation of the DNN model in a network and its associated results. For our implementation, we utilized an Intel(R) Core(TM) i5-1035G1 CPU running

at 1.00GHz (1.19 GHz with 32.0 GB of RAM, 31.8 GB usable) in conjunction with Google Colab Pro. Following the training phase, our primary objective was to apply the model's output to the vehicles connected in the network and evaluate the resulting statistics for various communication metrics. Initially, we constructed the network topology, which included vehicles, micro base stations, and base stations. We established connections among all participating entities using the Dijkstra algorithm [34]. We ensured that regular vehicle traffic adhered to a hop-count communication framework, whereas emergency vehicle traffic was routed through micro base stations to base stations for further processing.

Let's suppose, we have a network with a total of V_N vehicles. Among these vehicles, one specific vehicle, denoted as $V_{i(s)}$ from the set V_N , is engaged in regular traffic and communicates in the network using channel Y_2 . Vehicle $V_{i(s)}$ has multiple paths available for transmitting its data to a remote destination (R_d). These paths involve a certain number of intermediate vehicles $V_j \in V_N$ for data transmission. The network's connectivity and capacity are represented by matrices $G_c [V_i, j]$ and $C_c [V_i, j]$, respectively. Moreover, the communication cost of each hop count/next hop count in the network is symbolized as NH_c , and the total communication cost for all hops involved is referred to as $Comm_{Total_c}$. Moreover, there is a specific communication channel, assisted by μBS , dedicated to transmitting priority-class traffic, which helps improve the communication statistics for emergency vehicles. During our simulations, we collected data on various metrics to demonstrate the effectiveness of this approach when compared to rival protocols.

A. Communication Cost

In this part, we evaluate the communication cost of the proposed model with existing protocols. But before that, we check the communication cost for priority traffic and ordinary traffic (Emergency vehicles and Traditional vehicles) by considering the fixed length message scenario.

Furthermore, we used the mandatory messages (starting from initiation and ending at communication termination) to compute the communication cost in the context of energy consumption for the proposed paradigm in the presence of traditional protocols such as Ad hoc On-Demand Distance Vector (AODV) [35], Destination-Sequenced Distance Vector (DSDV) [36], and Low-Energy Adaptive Clustering Hierarchy (LEACH) [37]. For our calculations, we took into account a message length of 512 bytes and a Transmission Energy Consumption Rate (TSCR) of 0.002 watts. This simplifies to: Transmission energy (T_x) = Message length \times TSCR $\Rightarrow 0.002 \times 512 = 1.024$ mW. During the reception of the message at the receiver side (R_x), the same energy is consumed (1.024 mW). Given that each vehicle exchanges at least three messages with relying (next hop) vehicles before transmitting data, the total energy consumption during the connection establishment process = $3 \times (T_x + R_x) = 3 \times (1.024 + 1.024) = 6.144$ mW. Thus, the total energy consumption = $6.144 + 2.048$ mW = 8.192 mW.

Following this, if the vehicle uses a micro base station (μBS),

then its hop count will be two, and energy consumption with the same scenario (connection establishment) would be: $8.192 \text{ mW} + 2 \times 2.048 \text{ mW} = 12.288 \text{ mW}$. Conversely, if it uses traditional Dijkstra with five hop counts, then it would be: 18.51 mW. Similarly, we considered a scenario to calculate the total communication cost in the context of exchanging the total number of bits for different numbers of messages for our prototype and comparative protocols. The result statistics are shown in Table I.

1) *Comparative Protocols Communication Cost* : In this subsection, we analyze the latency results of comparative protocols as discussed in references [35][36] and [37]. Our observation from the operational scenario of these protocols indicates that they tend to consume more energy in the context of communication costs when transmitting data from the same vehicles to the destination location. Furthermore, we have noted that some of these protocols update their routing table through broadcast messages, while others adopt a hop-to-hop or hop-to-cluster update mechanism, resulting in additional energy consumption. For instance, in the case of hop count communication, the connected entities continuously broadcast messages in the network to update their routing table, whereas, in the case of cluster head communication, they select the cluster head based on defined parameters. But this process, consumes extra energy, and increases the communication cost. In this context, we would like to acknowledge that the LEACH protocol enables the source vehicle to transmit data within a single hop count. However, it's important to note that the other 10 candidate vehicles also send their candidacy requests to the source vehicle, which consequently consumes energy (9×1.024 mW). During simulations of these protocols, we have obtained the undermentioned results (Table II).

B. Energy Consumption Results

In this section, we delve into the energy consumption statistics of the proposed approach when compared to other protocols. From the statistical analysis conducted in the previous section, it becomes evident that the proposed approach exhibits the lowest communication cost when compared to all traditional protocols, based on energy consumption during the communication process. Given the operational requirements of cooperative Internet of Vehicle networks, particularly with the embedded of resource-limited embedded sensors. It is imperative to showcase the collective energy consumption statistics of the proposed model in the presence of comparative protocols. Th results obtained for energy consumption are shown in Figure 9

C. Latency Result Statistics

In this section, we discuss the latency result statistics of the proposed framework with comparative protocols, because the utmost objective of this work is to ensure delay-sensitive communication in Cooperative Internet of Vehicles. Given that, we focused on the vehicle categorized as "emergency vehicles" to observe their communication statistics in terms of latency. Furthermore, to assess the proposed paradigm with traditional protocols such as AODV, DSDV, and LEACH, we

TABLE I: Communication cost statistical results analysis of μ BS and Hop count Dijkstra

Vehicle Type	No of Hops	μ BS	BS	No of Messages	Total Consumed Energy	Total Bits Utilized at first reach
Emergency Vehicles	1	1	1	5	12.288 mW	2,560
Traditional Vehicles	5	0	1	8	18.51 mW	4,096

TABLE II: Communication cost result statistics of comparative protocols

Protocol Name	Vehicle Type	No of Hops	BS	No of Messages	Total Consumed Energy	Total Bits Utilized at first reach
AODV [35]	Traditional Vehicle	17	1	20	75.776 mW	10,240
DSDV [36]	Traditional Vehicle	6	1	11	30.720 mW	5,632
LEACH [37]	Traditional Vehicle	1	1	$4 + 9(R_x)$	17.408 mW	6,656

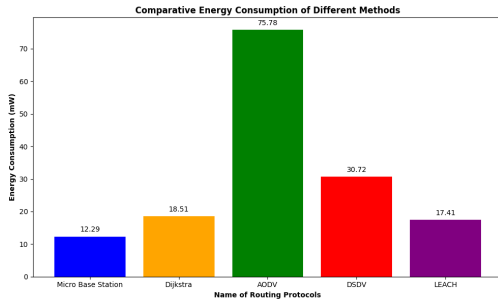


Fig. 9: Micro-base station and hop count communication latency visual representation

check it in the simulation environment by considering the source (vehicle) and destination point by exchanging messages among them. Although our proposed model categorizes the traffic into two different classes, but still, it is important to note how effectively it ensures delay-sensitive information in the network, when needed. In contrast, the traditional protocols follow the same tactics, never categorize the traffic, and even some modified versions do categorize the traffic, but still, they do not achieve the results we are getting here. Moreover, the results obtained during simulations are shown in Figure 10.

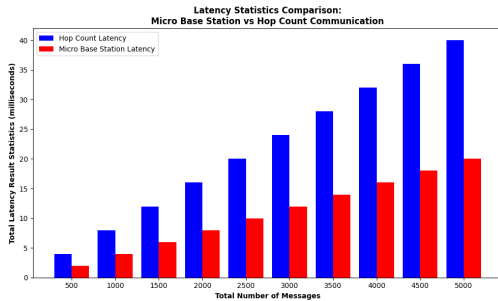


Fig. 10: Micro-base station and hop count communication latency visual representation

D. Throughput Result Statistical Analysis

In this section, we continue to check how well our new approach performs compared to traditional methods by taking into account the throughput statistics. For this, we took into account the throughput statistics. Given that, we have noted

in the model evaluation phase that the proposed framework classifies the traffic into two classes such as emergency vehicle traffic and traditional traffic. Emergency vehicles can communicate directly with remote destinations through micro-base stations, while regular vehicles follow a hop-count communication method called the "Dijkstra algorithm" to transmit data. By doing this, we've noticed that emergency vehicles using μ BS create a congestion-free environment for regular vehicles using the Dijkstra algorithm, making better use of the network's bandwidth. To test this idea, we ran simulations by passing the model output to connected vehicles, while guiding the traffic into the considered scenario. During an evaluation with comparative protocols, we noted great improvement in the results statistics of throughput. This improvement occurred because traditional protocols like AODV and DSDV continuously broadcast routes among interconnected vehicles, which creates extra overhead to the wireless communication channel. As a result of this extra overhead both the throughput and overall network performance are affected. Similarly, in the LEACH protocol, several vehicles (nodes in the context of the network) advertise their candidacy for cluster head positions which also causes communication overhead as nearby nodes respond to these candidacy requests. Which as a result degrade the network performance in the context of throughput. The comparative results obtained during simulations are shown in Figure 11.

VI. CONCLUSION

In this paper, we proposed a hybrid technique to address quality of service (QoS) challenges within cooperative Internet of Vehicles (C-IoV) networks. Given that, we introduce a hybrid framework known as the Traffic Congestion and Priority Aware Algorithm (TCPAA). Our proposed technique uses computer vision with a Deep Neural Network to classify the vehicles into two classes "real-time" such as emergency vehicles and traditional vehicles, in the other words emergency traffic or traditional traffic. With this, we used the Dijkstra algorithm to facilitate traditional vehicle traffic, while a network special component "micro base stations" was adopted for the first time to facilitate the priority traffic. Considering the unique traffic facilitation and real-time vehicle categorization makes our work the first of its kind. Consequently, it is important to answer why we chose DNN instead of other comparative algorithms. For this, we evaluated different algorithms on

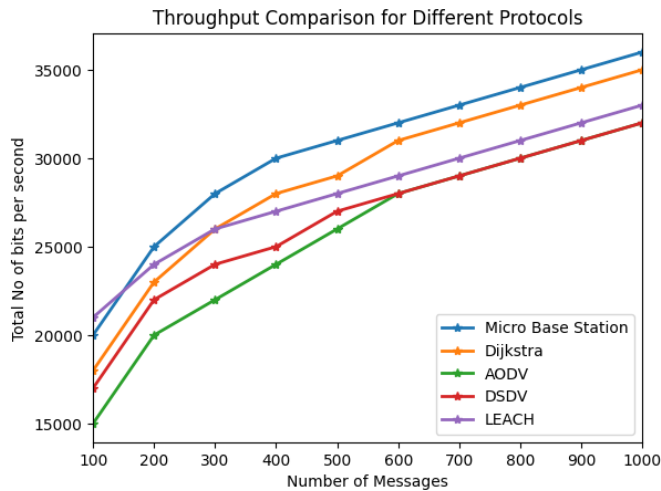


Fig. 11: Throughput comparative analysis of the proposed prototype with comparative protocols

our dataset and noted that DNN was among the best in terms of comparative metrics. Thereafter, we checked the proposed framework in the network topological infrastructure of C-IoV for communication metrics such as latency, communication cost, and throughput, etc. During analysis, we observed that our proposed framework outperforms the traditional protocols such as AODV DSDV, and LEACH in the aforesaid metrics. Finally, we ensured the reliability of the proposed model in all aspects starting from scratch to network deployment. Therefore, we are confident this model will help and improve the communication metrics of the future C-IoV networks.

ACKNOWLEDGEMENT

This work was supported in part by the Guangdong Provincial Key Laboratory of Human Digital Twin (2022B1212010004) and the Endowed Professorship from the Shenzhen Holdfound Foundation

REFERENCES

- [1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE transactions on industrial informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
- [2] M. Adil, J. Ali, M. Attique, M. M. Jadoon, S. Abbas, S. R. Alotaibi, V. G. Menon, and A. Farouk, "Three byte-based mutual authentication scheme for autonomous internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9358–9369, 2021.
- [3] Y. Miao, Q. Tong, K.-K. R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8681–8691, 2019.
- [4] M. Adil, H. Song, A. Farouk, and Z. Jin, "Auto-cioV: Autonomous connected internet of vehicles security requirements, open challenges with future research directions," *IEEE Network*, vol. 37, no. 4, pp. 238–246, 2023.
- [5] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive iot data and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1457–1477, 2017.
- [6] M.-A. Meyer, C. Granrath, G. Feyrerl, J. Richenhagen, J. Kath, and J. Andert, "Closed-loop platoon simulation with cooperative intelligent transportation systems based on vehicle-to-x communication," *Simulation Modelling Practice and Theory*, vol. 106, p. 102173, 2021.
- [7] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2020.
- [8] X. Cheng, C. Chen, W. Zhang, and Y. Yang, "5g-enabled cooperative intelligent vehicular (5genciv) framework: When benz meets marconi," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 53–59, 2017.
- [9] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019.
- [10] R. Mangiaracina, A. Perego, G. Salvadori, and A. Tumino, "A comprehensive view of intelligent transport systems for urban smart mobility," *International Journal of Logistics Research and Applications*, vol. 20, no. 1, pp. 39–52, 2017.
- [11] A. H. Sodhro, M. S. Obaidat, Q. H. Abbasi, P. Pace, S. Pirbhulal, G. Fortino, M. A. Imran, M. Qaraqe *et al.*, "Quality of service optimization in an iot-driven intelligent transportation system," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 10–17, 2019.
- [12] Z. Partovi, M. Zarei, and A. M. Rahmani, "Data-centric approaches in the internet of vehicles: a systematic review on techniques, open issues, and future directions," *International Journal of Communication Systems*, vol. 36, no. 3, p. e5383, 2023.
- [13] B. Hildebrand, M. Baza, T. Salman, S. Tabassum, B. Konatham, F. Amsaad, and A. Razaque, "A comprehensive review on blockchains for internet of vehicles: Challenges and directions," *Computer Science Review*, vol. 48, p. 100547, 2023.
- [14] L. Wen, S. Gao, Q. Luo, W. Hu, and Y. Yin, "Wideband dual circularly polarized antenna for intelligent transport systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5193–5202, 2020.
- [15] B. Alaya, R. Khan, T. Moulahi, and S. E. Khediri, "Study on qos management for video streaming in vehicular ad hoc network (vanet)," *Wireless Personal Communications*, vol. 118, no. 4, pp. 2175–2207, 2021.
- [16] N. Kalsoom, I. Ahmad, R. Alroobaea, M. A. Raza, S. Khalid, Z. Ahmed, and I. Ali, "Architecture for resource allocation in the internet of vehicles for cooperating driving system," *Journal of Advanced Transportation*, vol. 2021, pp. 1–11, 2021.
- [17] L. Miao and D. Leitner, "Adaptive traffic light control with quality-of-service provisioning for connected and automated vehicles at isolated intersections," *IEEE Access*, vol. 9, pp. 39 897–39 909, 2021.
- [18] A. S. A. Raj and Y. Palanichamy, "An aerial intelligent relay-road side unit (air-rsu) framework for modern intelligent transportation system," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 965–986, 2020.
- [19] J. Choi, V. Marojevic, C. B. Dietrich, J. H. Reed, and S. Ahn, "Survey of spectrum regulation for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 140 145–140 160, 2020.
- [20] S. Zeaddally, M. A. Javed, and E. B. Hamida, "Vehicular communications for its: Standardization and challenges," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 11–17, 2020.
- [21] K. Xiong, S. Leng, C. Huang, C. Yuen, and Y. L. Guan, "Intelligent task offloading for heterogeneous v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2226–2238, 2020.
- [22] N. Srinidhi, G. Sunitha, S. Raghavendra, S. D. Kumar, and V. Chang, "Hybrid energy-efficient and qos-aware algorithm for intelligent transportation system in iot," *International Journal of Grid and Utility Computing*, vol. 11, no. 6, pp. 815–826, 2020.
- [23] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: Toward iot-enabled smart urban traffic control and management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4063–4071, 2019.
- [24] H. Bi, E. Gelenbe, and Y. Chen, "Incentive mechanism for collective coordination in an urban intelligent transportation system using g-networks," in *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2020, pp. 1–8.
- [25] W. Ni, W. Wu, and K. Li, "A message efficient intersection control algorithm for intelligent transportation in smart cities," *Future Generation Computer Systems*, vol. 76, pp. 339–349, 2017.
- [26] A. Moubayed, A. Shami, P. Heidari, A. Larabi, and R. Brunner, "Edge-enabled v2x service placement for intelligent transportation systems," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1380–1392, 2020.
- [27] C. Roy and S. Misra, "Safe-passé: Dynamic handoff scheme for provisioning safety-as-a-service in 5g-enabled intelligent transportation sys-

tem,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5415–5425, 2021.

- [28] H. Hamidi and A. Kamankesh, “An approach to intelligent traffic management system using a multi-agent system,” *International Journal of Intelligent Transportation Systems Research*, vol. 16, pp. 112–124, 2018.
- [29] S. Kaffash, A. T. Nguyen, and J. Zhu, “Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis,” *International journal of production economics*, vol. 231, p. 107868, 2021.
- [30] A. Pradhan and G. Mahinthakumar, “Finding all-pairs shortest path for a large-scale transportation network using parallel floyd-warshall and parallel dijkstra algorithms,” *Journal of computing in civil engineering*, vol. 27, no. 3, pp. 263–273, 2013.
- [31] Z. Huang, “Extensions to the k-means algorithm for clustering large data sets with categorical values,” *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.
- [32] Ö. Akar and O. Güngör, “Classification of multispectral images using random forest algorithm,” *Journal of Geodesy and Geoinformation*, vol. 1, no. 2, pp. 105–112, 2012.
- [33] R. G. Brereton and G. R. Lloyd, “Support vector machines for classification and regression,” *Analyst*, vol. 135, no. 2, pp. 230–267, 2010.
- [34] M. Sheykhoumousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, “Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308–6325, 2020.
- [35] I. A. Aljabry and G. A. Al-Suhail, “A qos evaluation of aodv topology-based routing protocol in vanets,” in *2022 International Conference on Engineering & MIS (ICEMIS)*. IEEE, 2022, pp. 1–6.
- [36] P. Kanani, N. Patil, V. Shelke, K. Salot, A. Nanavati, N. Damodaran, and S. Desai, “Improving qos of dsdv protocol to deliver a successful collision avoidance message in case of an emergency in vanet,” *Soft Computing*, pp. 1–11, 2023.
- [37] A. Naeem, M. Rizwan, S. Alsubai, A. Almadhor, M. Akhtaruzzaman, S. Islam, and H. Rahman, “Enhanced clustering based routing protocol in vehicular ad-hoc networks,” *IET Electrical Systems in Transportation*, vol. 13, no. 1, p. e12069, 2023.



Muhammad Adil is currently a Ph.D. student in the Department of Computer Science and Engineering at the University at Buffalo, The State University of New York, USA. He received the Chair’s Fellowship from the department in 2022. Mr. Adil obtained BS and MS degrees in Computer Science from the Virtual University of Lahore, Pakistan, in 2017 and 2020 respectively. He has CCNA and CCNP certifications. Adil’s research interest includes Networking, Cybersecurity, Cyber-Physical Systems (CPS), Unmanned Aerial Vehicles (UAVs),

Internet-of-Things (IoT), and Wireless Sensor Networks (WSN). He has many publications in prestigious journals such as IEEE Internet of Things, IEEE Transactions of Intelligent Transportation Systems, IEEE Transactions on Industrial Informatics, IEEE Transactions on Network Science and Engineering, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Vehicles, IEEE Transactions on Green Communication Networking, IEEE Sensor Journal, IEEE Network Magazine, IEEE SMS Magazine, IEEE Micro Magazine, ACM Computing Surveys, ACM Transactions on Sensor Networks, Computer Networks Elsevier, Sustainable Cities and Societies, and many more. In addition, he is a member of the IEEE computer society, IEEE Industrial Electronics, IEEE Cybersecurity, IEEE Young professionals, IEEE Industrial Electronic Ambassador (Region-1) and London Journal Press ClubUK, as an Honory member. He is reviewing for prestigious journals, such as IEEE IoTJ, IEEE Sensors, IEEE Systems, IEEE TII, IEEE TCCN, IEEE TITS, IEEE TIV, IEEE TAI, IEEE TSMS, IEEE TGCN, IEEE WCL, IEEE Communication Magazine, IEEE Network Magazine, IET Communication, Elsevier Journals such as Computer Networks, JNCA, FGCS, Computers & Security, etc.



Houbing Song (M’12–SM’14–F’23) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, in August 2012. He is currently a Professor, the Founding Director of the NSF Center for Aviation Big Data Analytics (Planning), the Associate Director for Leadership of the DOT Transportation Cybersecurity Center for Advanced Research and Education (Tier 1 Center), and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us), University of Maryland, Baltimore County (UMBC), Baltimore, MD. He serves as an Associate Editor for IEEE Transactions on Artificial Intelligence (TAI) (2023-present), IEEE Internet of Things Journal (2020-present), IEEE Transactions on Intelligent Transportation Systems (2021-present), and IEEE Journal on Miniaturization for Air and Space Systems (J-MASS) (2020-present). He was an Associate Technical Editor for IEEE Communications Magazine (2017–2020). He is the editor of more than 10 books, the author of more than 100 articles and the inventor of 2 patents. His research interests include AI/machine learning/big data analytics, cyber-physical systems/internet of things, and cybersecurity and privacy. His research has been sponsored by federal agencies and industry. His research has been featured on popular news media outlets, including IEEE Spectrum, IEEE GlobalSpec&39;s Engineering360, IEEE Transmitter, insideBIGDATA, Association for Uncrewed Vehicle Systems International (AUVSI), Security Magazine, CXOTech Magazine, Fox News, U.S. News & amp; World Report, The Washington Times, and New Atlas.



Neeraj Kumar (SMIEEE) (2019, 2020, 2021 highly-cited researcher from WoS) is working as a Full Professor in the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala (Pb.), India. He has published more than 600 technical research papers (in top-cited journals and conferences which are cited more than 48,770 times from well-known researchers across the globe with a current h-index of 114). He has guided many research scholars leading to Ph.D. and M.E./M.Tech.

His research is supported by funding from various competitive agencies across the globe. His broad research areas are Green computing and Network management, IoT, Big Data Analytics, Deep learning and cyber-security. He has also edited/authored 10 books with International/National Publishers like IET, Springer, Elsevier, CRC. He is serving as an editor of the ACM Computing Survey, IEEE Transactions on Sustainable Computing, IEEE TNSM, Elsevier Computer Communication, Wiley International Journal of Communication Systems. He has won the best papers award from IEEE Systems Journal in 2018, in 2020, and IEEE ICC 2018, Kansas City in 2018. He has also won best paper award from Elsevier JNCA in 2021 and IEEE Comsoc IWCNC 2021. He has won the Outstanding Leadership Award from IEEE Trsutcom in 2021.



Mian Ahmad Jan (Senior Member, IEEE) received the master’s degree in mobile computing from the University of Bradford, U.K., and the Ph.D. degree from the University of Technology Sydney (UTS), Australia. He is currently an Assistant Professor with the Department of Computer Science, School of Computing and Informatics, University of Sharjah, United Arab Emirates. Prior to it, he was affiliated with Abdul Wali Khan University Mardan, Pakistan. His research interests include cyber security, energyefficient routing, and secured communication for wireless sensor networks and the Internet of Things. He has published his research in prestigious IEEE TRANSACTIONS, Elsevier journals, and Tier-A conferences. He has been an editor of various journals and he has been the general chair and the session chair for IEEE and Springer conferences. He has been among the World’s Top 2% scientists for three years in a row: 2021, 2022, and 2023, respectively



Amiya Nayak (Senior Member, IEEE) received the B.Math. degree in computer science and combinatorics and optimization from the University of Waterloo, Waterloo, ON, Canada, in 1981, and the Ph.D. degree in systems and computer engineering from Carleton University, Ottawa, ON, Canada, in 1991. He has more than 17 years of industrial experience in software engineering, avionics and navigation systems, and simulation and system level performance analysis. He is currently a Full Professor with the School of Electrical Engineering and Computer

Science, University of Ottawa, Ottawa, ON, Canada. He has authored or coauthored more than 300 research articles in international journals and conferences. His research interests include mobile computing, wireless sensor networks, and Internet of Things.



AHMED FAROUK (Member, IEEE) is currently an Assistant Professor with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, South Valley University. Before that he was a Post-Doctoral Research Fellow with Wilfrid Laurier University, Waterloo, ON, Canada, and Ryerson University, Toronto, ON, Canada. He is one of the Top 20 technical co-founders of the Quantum Machine Learning Program by Creative Destruction Laboratory, University of Toronto, Toronto. He is also selected as Top 25 of Innovate to 150 Canada to

showcase the best of Toronto's next generation of change-makers, innovators, and entrepreneurs. He was selected by the scientific review panel of the Council for the Lindau Nobel Laureate Meetings to participate in the 70th Lindau Nobel Laureate Meeting. His volunteering work is apparent since he was appointed as the Chair of the IEEE Computer Chapter for the Waterloo-Kitchener area. He is also selected for IEEE and IET Young Professional Ambassador.



ZHANPENG JIN (S'07-M'10-SM'15) is currently a Xinshi Endowed Professor in the School of Future Technology at the South China University of Technology, China, and also an Associate Professor in the Department of Computer Science and Engineering at the University at Buffalo, USA. He was an Associate Professor in Electrical and Computer Engineering, and Biomedical Engineering, at Binghamton University, as well as a Postdoctoral Research Associate at the University of Illinois at Urbana-Champaign (UIUC). He received his Ph.D. degree in Electrical

Engineering from the University of Pittsburgh. His research interests include ubiquitous computing, human-computer interaction, and AI-powered smart health and smart home. He is a Senior Member of ACM and IEEE, and serves as an Associate Editor for the following journals: ACM Computing Surveys, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), and Elsevier Computers in Biology and Medicine. He has published over 100 papers in international journals and conferences. He received the IEEE Region 1 Technological Innovation Award. His pioneer work in brain biometrics was selected and named as one of the "Future Technology: 22 Ideas About to Change Our World" by BBC Science Focus.