Intelligent O-RAN-Based Proactive Handover in Vehicular Networks

1st Eshita Rastogi
Computer Science & Engineering
Sungkyunkwan University
Suwon, Republic of Korea
eshita@skku.edu

2nd Mukesh Kumar Maheshwari
Department of Electrical Engineering
Bahria University
Karachi, Pakistan
mukeshkumar.bukc@bahria.edu.pk

3rd Jaehoon (Paul) Jeong
Computer Science & Engineering
Sungkyunkwan University
Suwon, Republic of Korea
pauljeong@skku.edu

***Abstract*—The ongoing efforts of wireless service providers to build a Radio Access Network (RAN) architecture have been noted in recent years. The primary goal was to design an operator-defined RAN architecture capable of providing intelligent radio control for Fifth-Generation (5G) wireless networks as well as Beyond 5G (B5G). The Open Radio Access Network (O-RAN) Alliance was formed to transform the telecommunications ecosystem. In this paper, we propose an intelligent O-RAN framework for vehicle communication. We propose a Machine Learning (ML)-based model to predict the compatibility time during which a vehicle remains within communication range with another vehicle to establish a connection. We compare the performance of Gaussian Naive Bayes (GNB), K Nearest Neighbor (KNN), and Neural Networks (NN) in terms of training and testing accuracy. We believe compatibility time estimation helps to implement proactive forwarding for optimal network performance. Finally, we conclude our work by providing directions for future research.**

***Index Terms*—O-RAN, Vehicular Networks, Machine Learning, Cellular Networks, Handover, Mobility Prediction**

I. INTRODUCTION

In the current scenario of Fifth-Generation (5G) and Beyond 5G (B5G) communications networks, efficient messaging in dynamic and and mobile Vehicle-to-Everything (V2X) networks is very important [1]. With the continuous improvement of communication technology, connected vehicles will enable applications such as effective emergency response, intelligent traffic congestion control, cooperative driving, and safety notifications [2]. However, the speed and congestion of automobile traffic can reduce the performance of communication technology, which in turn adversely affects system performance. The growing interest in Artificial Intelligence (AI) and Machine Learning (ML) applications for quick and efficient decision-making has resulted in many desirable performance solutions. ML algorithms can be used in V2X communication, such as resource allocation prediction, road safety measures, power allocation, security challenges and routing [2]–[5].

To effectively support an unprecedented increase of the use-cases with varying QoS demands, the evolution towards 5G and B5G is the need of time. Open-Radio Access Network (O-RAN) is an architectural transformation using the concept of virtualization, flexibility, and intelligence [6], [7]. O-RAN can enable RAN with openness and required intelligence, which is also a primary motivation pursued by O-RAN alliance [6]. Openness can be understood as the removal of vendor constraints and proprietary hardware and software implementations by establishing open standard interfaces. This will help reduce operating costs. Intelligence is vital for deploying, optimizing, and operating wireless networks [8].

The internal structure of O-RAN consists of three main components: Open-Central Unit (O-CU), Open-Distributed Unit (O-DU), and Open-Radio Unit (O-RU). The O-CU and O-DU components are connected using the F1 interface, and the O-DU is connected to the O-RU using the O1 interface as shown in Fig. 2. 1 [9]. The O-RU consists of a User Control Synchronization (CUS) plane and an Open Management plane. Radio Intelligent Controller (RICs) in the O-RAN architecture is divided into (i) near-real-time (near RT-RIC) RICs and (ii) non-real-time (non-RT-RICs) RICs in O-RAN architecture. RIC aims to improve conventional network functions with embedded intelligence. Non-RT-RIC is implemented in Service Management Orchestration (SMO), which serves as a software platform for rApp designed to optimize RAN. Non-RT RICs in the O-RAN architecture operate for periods greater than 1s. This is where policies are set and RAN analysis is collected. Near-RT-RIC is placed in the peripheral network, allowing control and optimization of RAN elements. O-RAN's quasi-RT-RIC controller handles real-time inference problems with time scales from 10 ms to 1 s [8]. The highly modular and programmable structure of future split RANs is suitable for developing advanced AI-based modules to realize network optimization through powerful communication schemes.

The capabilities introduced by RICs, i.e. open interfaces and AI/ML workflow, help O-RAN support a wide range of advanced use cases. These include handover management for V2X communications, Quality of Service (QoS) optimization, dynamic route-based drones, and radio resource allocation and spectrum sharing. These applications demonstrate the practical utility of the O-RAN architecture [8], [10].

The flexibility offered by O-RAN motivates us to propose an intelligent O-RAN framework for vehicular communication. In this paper, we propose an intelligent model by investigating a reliable communication link between the two vehicles. The vehicle chooses such a path so that it may remain connected to a neighboring vehicle for a longer time and ensures an uninterrupted message delivery and reduced handover delay.

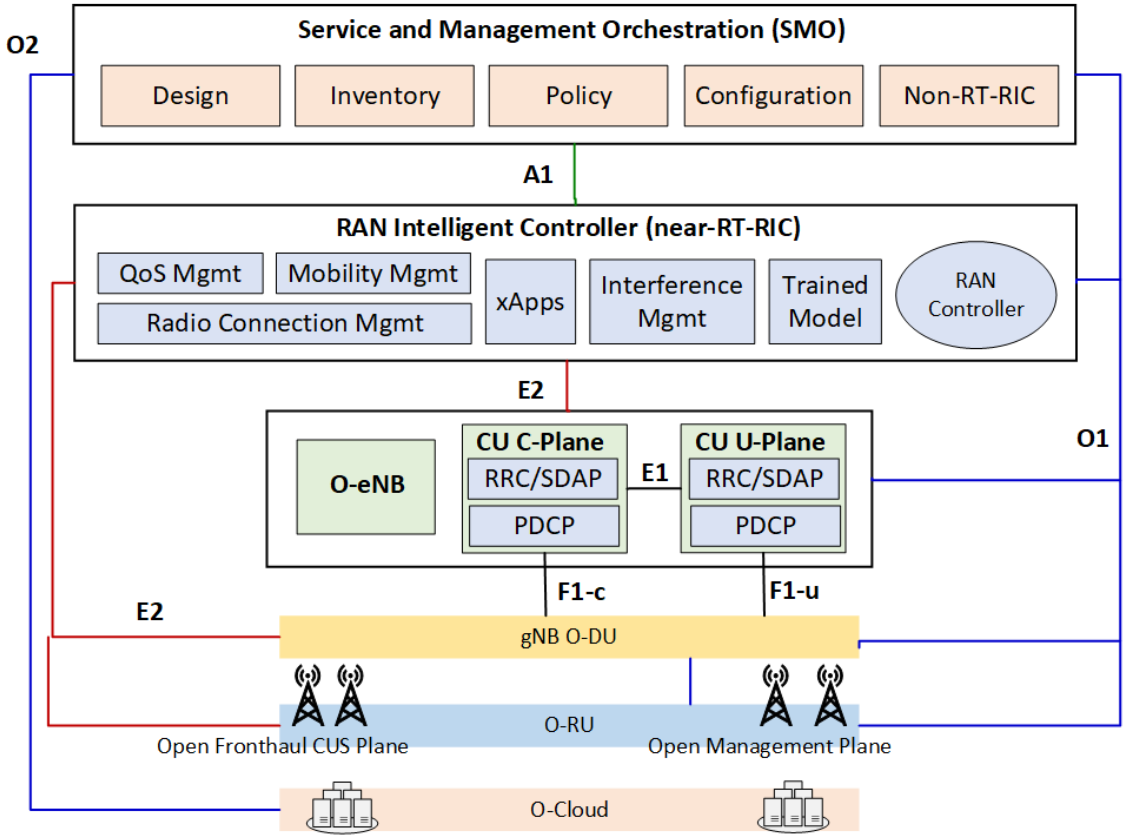

Fig. 1. Illustration of Open-RAN Architecture

In our scheme, the information of the vehicle is fed to a non-RT-RIC controller. The controller is trained to learn the mobility pattern of the vehicles offline. Then the trained model is placed on a near-RT-RIC controller which is capable of performing proactive handover in real-time which minimizes the communication overhead. This information can help in a proactive handover algorithm that reduces data interruptions and decreases delay. We evaluate widely used supervised ML algorithms and Neural Networks (NN) to predict a compatibility link between vehicles using the dataset described in [11]. We further analyze the training and testing accuracy of the proposed algorithms. The major contribution of this work includes the introduction of O-RAN framework for proactive handover to obtain seamless system performance in terms of service interruption delay and improved system performance.

The remaining part of the paper is organized as follows: Section II describes the work related to the research carried out in this article. Section III discusses the proposed intelligent O-RAN framework for vehicle communication. The performance comparison of the proposed ML algorithms is presented in Section IV. Finally, the conclusion of the work with a future research direction is presented in Section V.

II. Related Work

The mobility mechanisms in wireless networks enable the users to move within the coverage area anywhere and still to be serviced. This makes handover management an important topic of discussion in the wireless community. A good amount of research is conducted in this direction. Interested readers can refer to the work found in [15].

Recently, the O-RAN network model is an interesting topic that has attracted the research community's attention. The flexibility offered by RIC, open interfaces, and AI/ML-based models can support new use cases, like in-car communication and more. In recent years, we have seen an increase in O-RAN-focused research on applications and use cases. The O-RAN Alliance has compiled a complete list of 19 use cases for O-RAN implementations described in [16]. The O-RAN network can manage the mobility or performance of mobile users by managing handover parameters, load balancing, multiple connections, and beamforming in the RAN. This can be achieved in a closed loop using state information from multiple base stations and predicting user mobility based on RAN information [16].

Authors in [17] exploit O-RAN architecture and propose ML-based proactive network optimization techniques to improve handovers. Authors in [18] present the handover problem using graph NN and reinforcement learning to achieve proactive and intelligent connection management and evaluate the performance in terms of throughput and network coverage. The O-RAN specification in [16] also includes context-based handover for vehicular scenarios, where xApps leverages non-

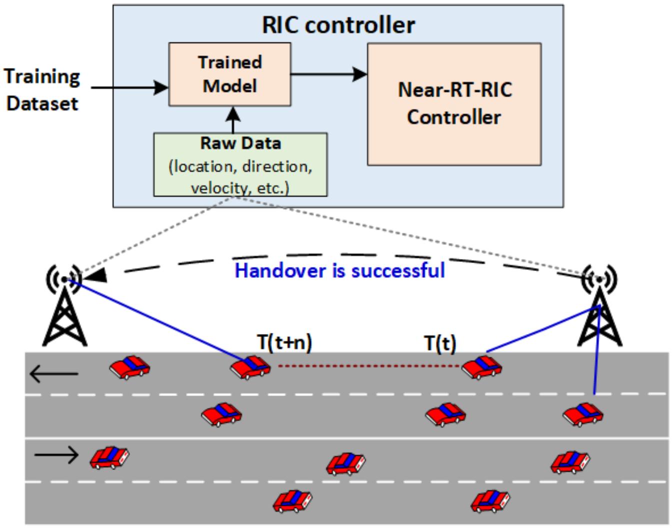

Fig. 2. Proposed O-RAN based Intelligent Handover Model

RT RIC information and inferences about AI/ML RAN data to manage the transmission. Unlike the other work, we exploit the importance of compatibility time between the vehicles as an important parameter for proactive handover.

In the preliminary work, we train a ML model using the dataset in the non-RT-RIC (rAPP). The trained model is then placed on the near-RT-RIC (xAPP). The accuracy of the ML models is analyzed in this work. The evaluation of the trained models is out of the scope of this paper and is considered as a part of our future work.

III. Proposed Intelligent Compatibility Time Model

In this section, we describe the proposed model followed by the description of the dataset used in this work. We also briefly describe the ML models used in the work.

A. A Proposed Model

In this work, a V2X scenario is considered where all vehicles are within gNB coverage. The gNB can serve applications that require high throughput and low latency. Each vehicle supports dual radios for cellular networks and Dedicated Short-Range Communications (DSRC) devices. In V2X communications, vehicles can share safety messages using short-range technology to ensure connectivity between vehicles. These safety messages contain information, such as position (x_i, y_i), velocity (v_i), direction (ϑ_i) for each vehicle i, and vehicle type. The time the vehicle j stays in the communication range (R_{ij}) of vehicle i is an integral part of the connection establishment process, which can be defined as compatibility time (γ_{ij}) [11], [19]. Using [11] and [19] the compatibility time can be calculated by the following equation:

$$\gamma_{ij} = \frac{R_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sqrt{|v_i + v_j||v_i - v_j|} + 1}. \quad (1)$$

We propose ML-based algorithms for the prediction of compatibility time (γ). Here ML algorithms run on the non-RT-RIC controller, which is trained using the mobility pattern of the vehicles offline, as shown in Fig. 2. The trained model is placed on the near-RT-RIC controller which is capable of performing proactive handover in real-time to minimize the communication overhead and improving network performance. This model can help in a proactive handover which may improve data interruptions and decreases delay. We evaluate widely used supervised ML algorithms and Neural Networks (NN) to predict the compatibility link in the vehicular environment using the dataset described in [11]. The ML models discussed in our work are described in Section III-B. We further analyze the training and testing accuracy of the proposed algorithms.

B. ML Models

In this subsection, we will describe the ML models and Neural Network architecture used in our analysis.

1) Gaussian Naive Bayes (GNB): Gaussian Naive Bayes is often used for estimating the probability of instances in each class. It can quickly create classification models with less computational effort. A gaussian distribution works well when the data is continuous, which means the continuous values associated with the class are distributed according to normal distribution [12]. The likelihood of the features is assumed using the following equation:

$$P(A|B=b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} exp\left(-\frac{(a-\mu_b)^2}{2\sigma_b^2}\right), \quad (2)$$

where μ_b and σ_b are the mean and standard deviation of continuous variable A computed for the class b of B.

2) K Nearest Neighbor (KNN): The KNN algorithm is a supervised learning classifier that is non-parametric and easy to implement. It uses proximity to classify or predict the clustering of an individual data point. KNN can be used for both regression and classification problems, but it works well for classification algorithms. The KNN algorithm assumes that similar points can be found next to each other [13]. The algorithm can easily adapt new training samples because all training data is stored in memory.

3) Neural Network (NN): NN is an iterative learning process consisting of input, hidden, and output layers that perform operations on the data. One of the applications of NN is the classification of labeled data sets. The NN algorithm extracts the features passed to the classifier for classification. The input layers correspond to the number of layers, and the output layers have a node for each layer. During the training phase, the network is trained by adjusting the weights to predict the correct class label of the input samples [14]. The main advantages of NN include high noise tolerance and the ability to classify patterns for which the network has not been trained.

IV. Results and Discussion

In this section, we describe the features of the dataset followed by the simulation setup. Next, we describe the simulation results.

A. Dataset Description

In this work, we make use of the information in the dataset described in [11] to estimate and predict the best route to provide vehicle communication. The dataset generated in [11] utilized the position (x_i, y_i), velocity (v_i), direction (ϑ_i) for each vehicle i, and velocity information of the vehicle. The data features are described and calculated as follows:

- **Euclidean distance** (α_{ij}) between the vehicle i and j, with 2D locations (x_i, y_i) and (x_j, y_j) can be calculated as follows:
$$\alpha_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{3}$$
- **Relative velocity** (ΔV_{ij}) between the two vehicle i and j can be defined as follows:
$$\Delta V_{ij} = \sqrt{|v_i + v_j||v_i - v_j|} + 1. \tag{4}$$
- **Direction difference** $(\Delta\vartheta_{ij})$ is calculated using the current direction of the vehicles (ϑ_i and ϑ_j) relative to each other and is derived as follows
$$\Delta\vartheta_{ij} = |((\vartheta_i - \vartheta_j + 180)\%360 - 180)|. \tag{5}$$
- **Direction difference label** $(\Delta\vartheta^l_{ij})$ represents the direction of vehicles i and j relative to each other, i.e. same, opposite direction, neither and is expressed as follows
$$\Delta\vartheta^l_{ij} = \begin{cases} 0 \text{ (same)} & \text{if } \Delta\vartheta^l_{ij} \leq 60, \\ 1 \text{ (opposite)} & \text{if } \Delta\vartheta^l_{ij} \geq 120, \\ 2 \text{ (neither)} & \text{otherwise.} \end{cases} \tag{6}$$
- **Tendency label** (T^l_{ij}) shows that vehicles (i, j) are moving in the same or opposite direction. The connectivity time can be calculated based on this characteristic. For instance, if vehicles are moving toward each other, connectivity time will be longer and vice versa. It can be defined as follows:
$$T^l_{ij} = \begin{cases} 0 & \text{if } \Delta\vartheta^l_{ij} == 2 \ \& \ \alpha_{ij}(t_2) - \alpha_{ij}(t_1) < 0, \\ 1 & \text{if } \Delta\vartheta^l_{ij} == 2 \ \& \ \alpha_{ij}(t_2) - \alpha_{ij}(t_1) > 0. \end{cases} \tag{7}$$
where $\alpha_{ij}(t_1)$ and $\alpha_{ij}(t_2)$ are vehicle inter-distance at time t_1 and t_2 respectively.
- **Compatibility time label** (γ^l_{ij}) shows the connectivity duration between two vehicles (γ^l_{ij}) and is defined as follows:
$$\gamma^l_{ij} = \begin{cases} C0 & \text{if } \gamma_{ij} == 0, \\ C1 & \text{if } \gamma_{ij} > 2 \ \& \ \gamma_{ij} \leq 5, \\ C2 & \text{if } \gamma_{ij} > 5 \ \& \ \gamma_{ij} \leq 10, \\ C3 & \text{if } \gamma_{ij} > 10 \ \& \ \gamma_{ij} \leq 15, \\ C4 & \text{if } \gamma_{ij} > 15. \end{cases} \tag{8}$$

The ML classification model needs training data in the form of a six column header i.e. α_{ij}, ΔV_{ij}, $\Delta\vartheta_{ij}$, $\Delta\vartheta^l_{ij}$ and T^l_{ij}. These five features and the target γ_{ij} are chosen by hit and trial methods for optimal performance.

TABLE I
TRAINING AND TESTING ACCURACY

Models	Training Accuracy	Testing Accuracy
Gaussian Naive Bayes	97.67	97.28
K Nearest Neighbor	99.04	99.53
Neural Network	98.75	98.72

B. Performance Evaluation

In this article, we intend to calculate the compatibility time between vehicles using the data set D obtained from work in [11]. The total number of samples in the data set is 20,000. We randomly partition the dataset into training (60%), testing (20%), and validation (20%) datasets. We train our model using the training dataset and the testing and validation dataset is used to evaluate the effectiveness of the trained model. The data is prepossessed to make it suitable for training the ML classifiers and Neural Networks. The dataset consists of 5 features $\{\alpha_{ij}, \Delta V_{ij}, \Delta\vartheta_{ij}, \Delta\vartheta^l_{ij}, \Delta\vartheta^l_{ij}\}$ and 5 output class labels $\{C0, C1, C2, C3, C4\}$ which correspond to the different connectivity duration between two vehicles i and j.

The two ML classifiers used in this work are GNB and KNN. We have also used NN classifier for multiclass classification. We consider a three-layer feed-forward NN. The batch size considered in our work is 16 and the learning rate is assumed to be 0.001. The ReLU activation function and the Adam optimizer are used. We use cross-entropy loss to minimize the loss in the model. All ML models are imported from Python Sci-kit-learn API while the NN model is imported from PyTorch in our Python simulation.

It is imperative to meticulously observe and analyze the training process of the ML and NN models to avoid overfitting and underfitting problems and evaluate the trained model on the dataset. Fig. 3 shows the accuracy curve of the GNB. The training accuracy is very high initially and decreases with increasing the number of samples. However, the testing accuracy is low at the beginning and increases gradually. The training and test accuracy becomes more realistic once the entire dataset is used for training. Another learning curve for the GNB classifier shows that the training accuracy remains constant regardless of the size of the training set. While the testing accuracy increases with the size of the training dataset.

Fig. 4 shows the accuracy curve of the KNN with respect to K nearest neighbors. It can be seen from Fig. 4 that when using higher values of K, the accuracy decreases. This is because the model becomes complicated and will use more data points that reduce the flexibility of the model. Fig. 5 shows the accuracy curve of NN with respect to the number of epochs. We can see from Fig. 5 that the accuracy increases with the increase in the number of epochs. During each epoch, the model is exposed to the same data multiple times which allows the model to learn from the data and improve its accuracy considerably. Table I describes the training and testing accuracies of the models proposed in the work.

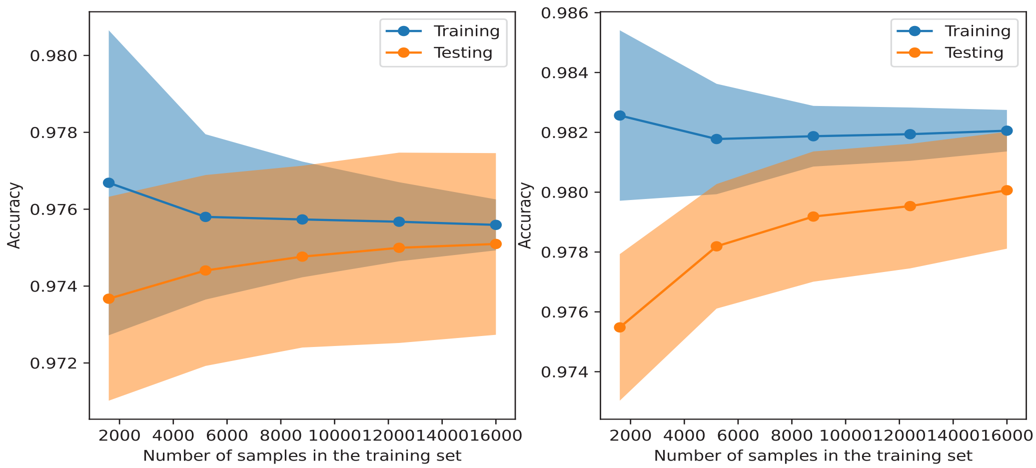

Fig. 3. Training and Testing Accuracy of GNB

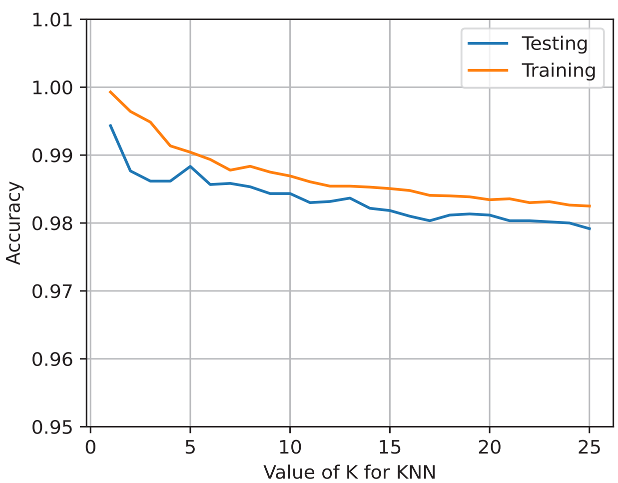

Fig. 4. Training and Testing Accuracy of KNN

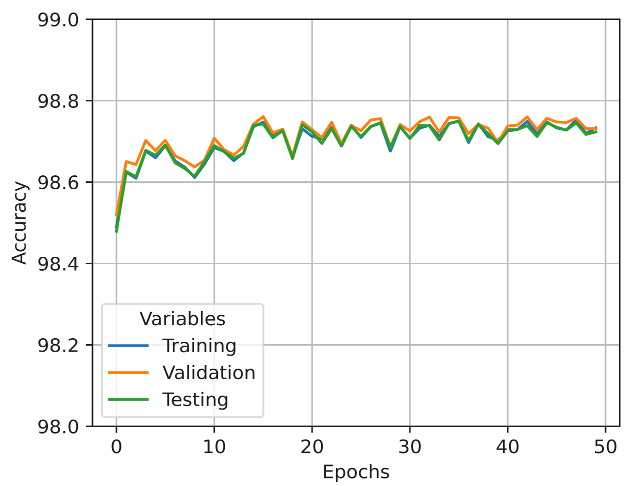

Fig. 5. Training, Testing, and Validation Accuracy of NN

V. Conclusion

In this work, we discuss that vehicles in highly mobile environments will be connected to each other for a longer time depending on the compatibility time. The capability time is calculated based on the communication range, Euclidean distance, and vehicle velocity. We propose an ML-based model to predict the compatibility time. We compare the performance of GNB, KNN, and NN in terms of training and testing accuracy. We endorse that KNN-based compatibility time prediction is the most suitable option for the ML model based on the training and testing accuracy.

Knowing the vehicle's communication path and compatible timing is a step toward proactive vehicle delivery and networking. The flexibility brought by O-RAN prompts us to think of an intelligent O-RAN framework for vehicle communication. Non-RIC-based controllers can be trained offline, and the trained model is fed back to a nearby RT-RIC controller for compatible timing prediction. This will help reduce communication costs by optimizing transmissions and improving overall network performance. In future work, we will study the smart handover process in detail using the information discussed in the work. Additionally, we will consider more constraints in the system model which will involve more complexities, thereby impacting the performance of NN.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. 2023R1A2C2002990). This work was supported in part by the Institute of Information & Communications Technology

Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2022-0-01015, Development of Candidate Element Technology for Intelligent 6G Mobile Core Network). Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," Proc. IEEE, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[2] A. Awang, K. Husain, N. Kamel, and S. Aissa, "Routing in vehicular ad-hoc networks: A survey on single- and cross-layer design techniques, and perspectives," *IEEE Access*, vol. 5, pp. 9497–9517, 2017.

[3] M. A. Hossain, R. Md. Noor, K.-L. A. Yau, S. R. Azzuhri, M. R. Z'aba, and I. Ahmedy, "Comprehensive survey of machine learning approaches in cognitive radio-based vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 78054–78108, 2020.

[4] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, "Reinforcement learning based radio resource control in 5G vehicular network," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 611–614, May 2020.

[5] H. Zhang, S. Chong, X. Zhang, and N. Lin, "A deep reinforcement learning based D2D relay selection and power level allocation in mmWave vehicular networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 416–419, Mar. 2020.

[6] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, "Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 21–27, 2021.

[7] "O-RAN Alliance Introduces Minimum Viable Plan Towards Commercial O-RAN Solutions," 2020. [Online]. Available:https://www.o-ran.org/resources

[8] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," arXiv:2202.01032 [cs.NI], Jan 2023. [Online]. Available:https://arxiv.org/abs/2202.

[9] M. Mustafa, J.M. Batalla, E. Pallis, G. Mastorakis, E. K. Markakis, and C. X. Mavromoustakis. "On analyzing beamforming implementation in O-RAN 5G." *Electronics* vol. 1, no. 17, 2021.

[10] ORAN Alliance, "O-RAN use cases and deployment scenarios," Tech. Rep., Feb. 2019. [Online]. Available: https://www.o-ran.org/resources

[11] F. H. Kumbhar, and S. Y. Shin. "Novel vehicular compatibility-based Ad Hoc message routing scheme in the Internet of Vehicles using machine learning."*IEEE Internet of Things Journal* vol. 9, no. 4, pp. 2817–27, 2828, 2021.

[12] R. Islam, M. K. Devnath, M. D. Samad, Syed Md J. Al Kadry, "GGNB: Graph-based Gaussian naive Bayes intrusion detection system for CAN bus." *Vehicular Communications*, vol. 33, 2021.

[13] M. Bansal, A. Goyal, A. Choudhary, " A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning." *Decision Analytics Journal*, vol. 3, 2022.

[14] J. T. Hancock, T. M. Khoshgoftaar. "Survey on categorical data for neural networks." *Journal of Big Data*, vol. 7, pp. 1-41, 2020.

[15] M. Tayyab, X. Gelabert and R. Jäntti, "A survey on handover management: From LTE to NR," in IEEE Access, vol. 7, pp. 118907-118930.

[16] O-RAN Working Group 1, "O-RAN Use Cases Detailed Specification 6.0," O-RAN.WG1.Use-Cases-Detailed-Specification-v06.00 Technical Specification, July 2021.

[17] H. Kumar, V. Sapru, and S. K. Jaisawal, "O-RAN based proactive ANR optimization," in Proc. *IEEE Globecom Workshops* (GC Wkshps, Taipei, Taiwan, 2020, pp. 1–4.

[18] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. Talwar, "Connection management xAPP for O-RAN RIC: A graph neural network and reinforcement learning approach," in Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2021, pp. 936–941.

[19] M. H. Eiza, T. Owens, and Q. Ni, "Secure and robust multi-constrained QoS aware routing algorithm for VANETs," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 1, pp. 32–45, Jan./Feb. 2016.