Network Slicing for Vehicle Communications in Open Radio Access Networks

Saurabh Parkar
ECE Department
Stevens Institute of Technology
Hoboken, USA
sparkar@stevens.edu

Xiaochan Xue
ECE Department
Stevens Institute of Technology
Hoboken, USA
xxue2@stevens.edu

Dr. Shucheng Yu
ECE Department
Stevens Institute of Technology
Hoboken, USA
syu19@stevens.edu

Abstract*—The rapid advancement of connected and autonomous vehicles has significantly increased the demand for efficient and reliable Vehicle-to-Everything (V2X) communication systems, which are essential for ensuring real-time data exchange between vehicles, infrastructure, and other entities. Network slicing is a technique that creates multiple virtual networks optimized for specific services, and is important in addressing the diverse Quality of Service (QoS) requirements of V2X communication, such as ultra-low latency, high bandwidth, and reliability. However, traditional cellular networks often struggle to meet these demands due to their static and inflexible architectures. The introduction of the Open Radio Access Network (O-RAN) architecture addresses these challenges by incorporating intelligent controllers, specifically the Near-Real-Time RAN Intelligent Controller (Near-RT RIC), which enhances network slicing through dynamic and adaptive management of network resources. These controllers leverage real-time data and machine learning algorithms to optimize resource allocation, ensuring that each network slice meets its specific QoS requirements even in highly dynamic environments. In this paper, we present a method of network slicing through a deep learning approach within an O-RAN-based xApp to predict and manage network slices in a V2X environment. Our method is able to achieve a 92% accuracy in slice type prediction, demonstrating significant improvements in network performance and resource allocation. This work showcases the potential of combining O-RAN's intelligent control capabilities with advanced machine learning techniques to meet the stringent demands of dynamic vehicular networks.

Keywords*—*V2X, Network Slicing, O-RAN.

I. Introduction

The increasing advancements in vehicle technologies and the rise of connected and autonomous vehicles (CAVs) demand a need for a more efficient and intelligent transportation system. As vehicles become more interconnected and intelligent, the need for a reliable, low latency communications structure for them is also increased. Vehicle-to-everything (V2X) communications encompasses these various types of interactions, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) communication as shown in Fig.1. These interactions are fundamental to the operation of smart transportation systems, where vehicles share real-time information about their environment, enabling safer and more efficient travel. However, the dynamic and unpredictable nature of vehicular environments poses significant challenges to maintaining the high levels of reliability and low latency required by V2X applications. Traditional Radio Access Networks (RANs) often lack the flexibility and intelligence needed to meet these stringent requirements, especially in highly dynamic vehicular environments. This limitation of a Traditional RAN structure can be overcome by integrating the V2X communications with the Open Radio Access Networks (O-RAN) Architecture.

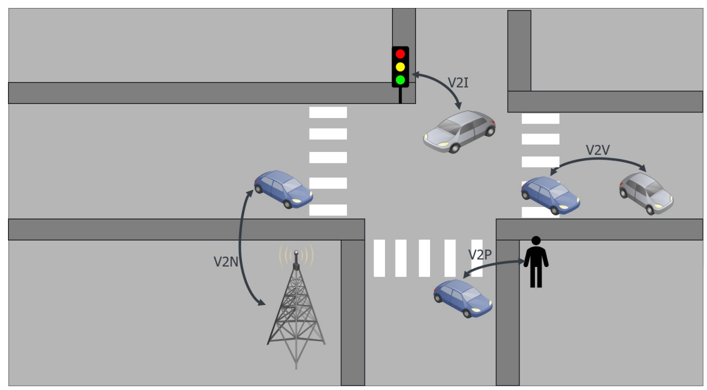

Fig. 1: Overview of current V2X Architecture

O-RAN offers a transformative approach to cellular architecture by promoting a virtualized, open, and intelligent architecture. O-RAN allows for disaggregated components to be connected through open-interfaces and optimized by Intelligent Controllers as shown in fig 2. This disaggregated approach to cellular networks allows for multi-vendor interoperability over the same infrastructure thus reducing operational costs and enabling network optimization through data-driven closed-loop control [1]. The O-RAN architecture is significantly shaped by the O-RAN Alliance promoting standardized and open-interfaces for network components and enabling Artificial Intelligence (AI) and Machine Learning (ML) based intelligent policy control for efficient network optimization and management. O-RAN integrates AI and ML into traditional RAN through RAN Intelligent Controllers (RICs): the Near-Real-Time RIC (Near-RT RIC) and the Non-Real-Time RIC (Non-RT RIC). The Near-RT RIC handles time-sensitive operations (10ms to 1 second) for real-time control, such as resource management, ensuring optimal network performance. The Non-RT RIC manages non-time-sensitive tasks (1 second or greater), focusing on long-term optimization, policy management, and network analytics. These controllers enhance network efficiency by enabling intelligent, adaptive control and orchestration, making O-RAN crucial for the evolution of next-generation cellular networks[2].

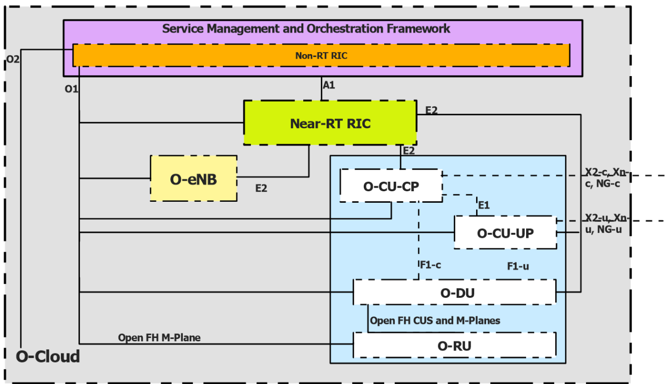

Fig. 2: O-RAN Architecture

connected and autonomous vehicles.[3]

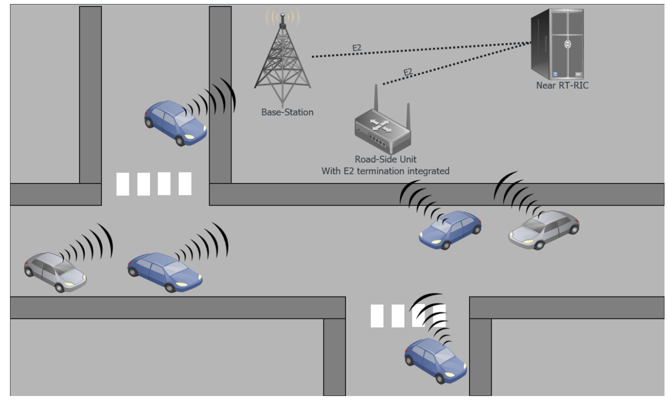

Fig. 3: Integrating RSUs into ORAN Architecture with E2 termination

Network slicing is the method of creating multiple virtual networks, each optimized for specific services ensuring efficient resource allocation for meeting the required Service Level Agreements (SLAs). Network slicing enables the allocation of dedicated resources to various services, such as safety-critical applications that require ultra-low latency and high reliability, or infotainment systems that demand high data rates. By segmenting the network into slices, service providers can ensure that each V2X service operates under optimal conditions, improving overall performance and efficiency. This approach allows for the dynamic and flexible management of network resources, crucial for addressing the diverse and stringent requirements of V2X communications in real-time. However, this process in V2X requires a definition of custom service architectures which may not always be compatible with the traditional cellular architectures due to less standardization of components. However, in O-RAN, network slicing is further enhanced by the RAN Intelligent Controllers, which can be used to dynamically manage these slices through specialized xApps, ensuring real-time optimization based on network conditions. These RICs also allow us to implement deep learning techniques as well for processing the network parameters and dynamically adjusting resource allocations, improving network and performance efficiency whilst providing a more standardized and flexible architecture as compared to V2X.

Integration of O-RAN with V2X enables the deployment of more complex control mechanisms that can dynamically allocate resources and manage the network based on real-time data collected from vehicles and roadside units (RSUs). This integration of V2X with O-RAN includes extending the architecture of V2X components such as RSUs to include the E2 terminations as shown in Fig. 3. This allows for a more granular level of control, where the Near-RT RIC can directly manage and optimize V2X communications by adjusting parameters based on the specific needs of each vehicle or cluster of vehicles. This not only improves the efficiency and reliability of the V2X system but also allows for predictive maintenance and optimization, ensuring that the network can respond to potential issues before they impact service delivery. This integration provides a standardized, resilient, and scalable infrastructure capable of supporting the future demands of connected and autonomous vehicles.

This paper provides a network slicing solution based on the Deep Learning technique for V2X communications by integrating them within the latest O-RAN architecture through a xApp. The deep learning model is trained on the Berlin V2X Dataset. Training results show an accuracy of 92%. The base stations (eNBs) communicate with this model deployed on our xApp on the Near RT-RIC for predicting the type of slice based on the features obtained from User Equipment (UEs), the predictions are then sent back to eNBs for allocating appropriate resources based on slice type.

The rest of the paper is organized as follows. Section II discusses related work. Our design is elaborated in Section III. The experimental evaluations are illustrated in Section IV. Section V shows the future research directions. Section VI concludes this paper.

II. Related Work

Network slicing in traditional cellular networks is primarily based on a combination of rule-based and policy-driven approaches using predefined configurations and conditions to allocate network resources dynamically. But with current increasing trends in AI/ML show a significant rise in Network slicing based on Deep Learning Techniques.

With the rise of autonomous driving and connected vehicles, V2X services are becoming increasingly critical, demanding diverse Quality of Service (QoS) requirements, including ultra-low latency, high reliability, and high data rates. The work by ref. [4] provides the network slicing solution for V2X communications using Deep Learning technique through Deep Reinforcement Learning (DRL) approach. The paper proposes a cloud-based framework that integrates Software-Defined Networking (SDN) with Network Functions Virtualization (NFV) to create a centralized control environment for network slicing. This framework is divided into two domains: edge cloud and remote cloud. The edge cloud, positioned closer to the vehicular user equipment (VUE), ensures low latency by hosting service entities in proximity, while the remote cloud provides additional resources when the edge cloud's capacity is insufficient. Along with this they also define an intelligent network slicing architecture consisting of four layers namely, the Network Infrastructure Virtualization Layer virtualizes and abstracts resources into a multi-dimensional resource pool for allocation to network slices; the Intelligent Control Layer manages and deploys network slices using ML algorithms; Network Slice Layer defines network slices based on the QoS needs of V2X services, such as traffic safety, autonomous driving, and infotainment; Service Customized Layer: Aligns network slices with the specific Service Level Agreements (SLAs) of V2X service providers. It also defines the different slices important for V2X communications. We used these definitions to create our own slices in the dataset. Although their network slicing approach is able to efficiently optimize the network, the traditional V2X architecture lacks the flexibility needed to dynamically adapt to the highly variable conditions of vehicular environments. Also, the high mobility of vehicles makes it difficult to allocate resources efficiently. Here using O-RANs Near RT-RIC and dynamic nature we are able to efficiently process and predict network traffic and allocate resources faster.

Network slicing techniques in O-RAN are greatly enhanced by their Radio Intelligent Controllers, due to their ability to sense network conditions from the RAN elements and allocate resources accordingly. The work by ref. [5] provides a complete implementation of policy-driven network slicing within the ORAN architecture. The NexRAN framework is built upon the srsRAN open-source mobility stack with integration of O-RAN's E2 agent, enabling it to function in the O-RAN environment. This integration allows for facilitating network slicing, different network slices can share the same frequency band, and UEs can be explicitly associated with slices. The NexRAN xApp utilizes the standard O-RAN key performance measurements (KPM) service model to monitor the state of the RAN and control slice behavior based on policy-driven decisions. xApp reads the current state of RAN elements and adjusts the slicing configuration in real-time to meet predefined policies. This dynamic adjustment ensures optimal resource allocation across different network slices. Along with the network slicing application, the paper also highlights the effectiveness of POWDER testbed for conducting wireless experiments. The NexRAN framework provides a good reference for network slicing, however, their rule-based, policy-driven slicing may not be as efficient as employing DL methods for predicting network patterns leading to better adaptability under dynamic conditions of a highly mobile environment of V2X communications.

The work by ref. [6] highlights the effectiveness of using deep learning to enhance network slicing within O-RAN. Their Network Slice Radio Resource Management (NSRRM) xApp monitors the RAN conditions and adjusts RAN configurations dynamically to ensure that the SLAs of different network slices are met. Their xApp uses the BigDL Choronos tool for time-series predictions, to develop flexible training pipelines that can handle various data sampling granularities and loss function designs. Based on the traffic predictions, the NSRRM xApp calculates the appropriate radio resources to be reserved or prioritized for each network slice. This ensures that the network slices can meet their SLA requirements even under fluctuating network conditions.

III. Our Solution

Network slicing in our O-RAN environment is implemented through a xAPP on the Near RT-RIC, this xAPP holds our Deep Learning Model and predicts the slice type based on the transmissions feature vector obtained from the UEs, and the predictions are sent back to the eNBs to allocate the appropriate resources as per the slice type as shown in Fig. 4. The Near RT-RIC's lower response times help in fast allocations of these resources as per the slice type, thus optimizing the network efficiently in real-time.

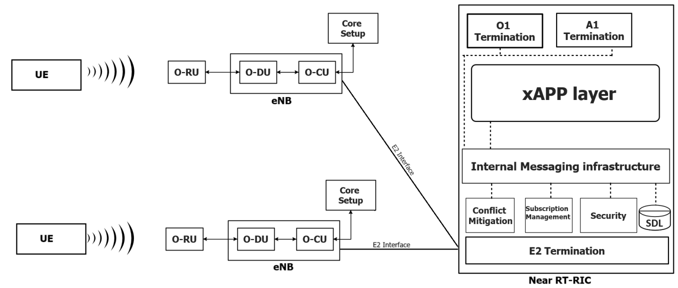

Fig. 4: O-RAN Setup Design

A. Description of Dataset

We used the Berlin V2X dataset ref. [7] for training our deep learning model for network slicing. This dataset provides a comprehensive number of different features associated in V2X communications collected from 4 different vehicles over a course of 3 days under various environments such as Avenue, Highway, Residential, Tunnel, and Park settings. The features of the dataset are listed in Fig. 5.

Data category	Source	Tool	Sampling interval	Features
Cellular	DME	Mobile Insight	10 ms	PHY: SNR, RSRP, RSRQ, RSSI
			20 ms	PDSCH/PUSCH: RBs, TB Size, DL MCS, UL Tx Power
			Event-based	RRC: Cell Identity, DL/UL frequency, DL/UL bandwidth
		ping	1 s	Delay
		iperf	1 s	DL Datarate, Jitter
	Server	iperf	1 s	UL Datarate, Jitter
Sidelink	SDR UE	tcpdump	Event-based	SNR, RSRP, RSRQ, RSSI, Noise Power, Rx Power, Rx Gain
Position	GPS		1 s	Latitude, Longitude, Altitude, Velocity, Heading
Side information	Internet database	HERE API	5 min	Traffic Jam Factor, Traffic Street Name, Traffic Distance
		DarkSky	1 hour	Cloud cover, Humidity, Precipitation Intensity & Probability, Temperature, Pressure, Wind Speed

Fig. 5: Berlin V2X Features

The dataset itself does not consists of the slice type feature so we created our own slice type feature by setting up our arbitrary thresholds for three slices: Low Latency, High Bandwidth, and General. The Features selected for these slices are as follows:

- Low Latency:

- **ping_ms** (< 2500): Measures the round-trip time it takes for a packet to travel to a server and back, directly reflecting network latency. A threshold of 2500 ms is used to categorize connections with relatively low latency. While very low latencies (e.g., < 100 ms) are ideal for real-time applications, a more inclusive threshold helps capture a broader range of low-latency scenarios suitable for applications that require responsive communication but are not strictly real-time.

- **jitter** (< 0.003): Represents the variability in packet arrival times. Lower jitter indicates more consistent network performance, which is crucial for real-time applications. A threshold of 0.003 is chosen to identify scenarios where network performance is stable and predictable. High jitter can disrupt real-time communications, so lower jitter is indicative of a network suitable for latency-sensitive applications.
- **PCell_RSRP_max**(> -100): Measures the power level of the signal received from the cell tower, reflecting the strength of the connection. A stronger signal (closer to 0 dBm) generally leads to more reliable communication with fewer errors, reducing the likelihood of re-transmissions and lowering latency. The threshold of -100 dBm is chosen to filter out weak signals that may cause delays.
- **PCell_SNR_1**(> 10): Indicates the quality of the signal relative to background noise. A higher SNR leads to clearer signals with fewer errors, which supports faster and more reliable data transmission, contributing to lower latency. A threshold of 10 is selected to ensure that only scenarios with relatively good signal quality are considered for low latency applications.
- **speed_kmh**(> 5 km/h): Faster movement may necessitate lower latency to maintain seamless connectivity, especially in mobile scenarios.Devices in motion, especially those moving faster than 5 km/h, often require more responsive network performance to handle handovers between cell towers and maintain a stable connection. This threshold ensures that the network slice is suitable for mobile use cases.

- High Bandwidth:

- **datarate**($> 30Mbps$): A threshold of 30 Mbps is chosen to identify scenarios where a high data rate is necessary. This is typical for applications that require fast download or upload speeds, such as video streaming, large file transfers, or high-resolution video conferencing.
- **PCell_Downlink_frequency** (> 1500 MHz): Higher frequency bands generally support higher data rates due to increased bandwidth availability. A threshold of 1500 MHz is chosen to distinguish scenarios where the network is operating on a frequency that can support higher bandwidth.
- **PCell_Downlink_bandwidth_MHz**(> 15 MHz): Wider bandwidths allow more data to be transmitted simultaneously, making them suitable for high-bandwidth applications. A threshold of 15 MHz is set to ensure the network slice is appropriate for scenarios that require substantial data throughput.
- **PCell_Downlink_Num_RBs**(> 50): Represents the number of resource blocks allocated for downlink transmission, a key determinant of network capacity. More resource blocks indicate a larger share of the channel's bandwidth, supporting higher data rates. A threshold of 50 ensures that only scenarios with sufficient network resources are considered for high bandwidth classification.
- **PCell_Downlink_TB_Size**(> 1000): The size of the transport block in the downlink, which indicates the amount of data being transmitted. Larger transport blocks correspond to higher data throughput, which is essential for high-bandwidth applications. This threshold ensures that the network slice can handle substantial data loads.
- **PCell_Downlink_Average_MCS**(> 10): Represents the efficiency of data transmission, with higher MCS values indicating better spectral efficiency. A threshold of 10 is chosen to identify conditions where the network is optimized for fast data transmission.

- General: The data-points that don't match either of these conditions where set to general slice.

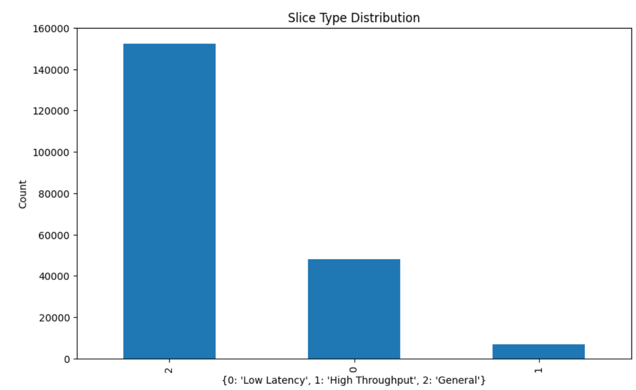

Fig. 6: Slice Distribution

Based on the above thresholds we got the following distribution of slice type, with the general type being most dominantly present with nearly 73% of data points, followed by Low latency with 23% and 4% of High Bandwidth as shown in Fig. 6. This distribution gives an accurate representation of a usual environment considering a majority of Vehicles/Devices would be in general with only a few being in Low Latency or Higher Bandwidth requirements.

B. Deep Learning Model

We built a simple Dense model for classification of the data points as shown in Fig. 7. The deep learning model takes in a feature vector of UEs transmissions consisting of the features discussed in Section III-A for predicting the different slices. Some Dropout layers were also added to the model to overcome the disparity of slice-type distribution and avoid overfitting on the general slice type due to its majority.

IV. Evaluation

A. Experiment Setup

We conducted our experiments using the POWDER Wireless testbed, where we simulated an Open Radio Access Network (O-RAN) environment. In this setup, X-310 USRPs were used to emulate as Base-Stations(eNBs) connected to a Cloud Node

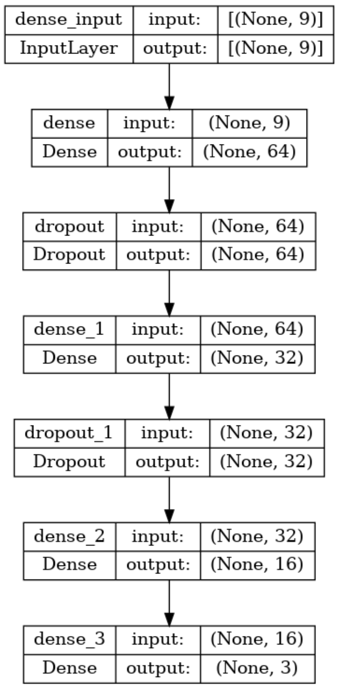

Fig. 7: Classification Model

that formed the Near RT-RIC, and the B210 USRPSs as the User Equipments (UEs) as shown in Fig. 8. The UEs here are supposed to act as vehicles in an O-RAN integrated V2X environment.

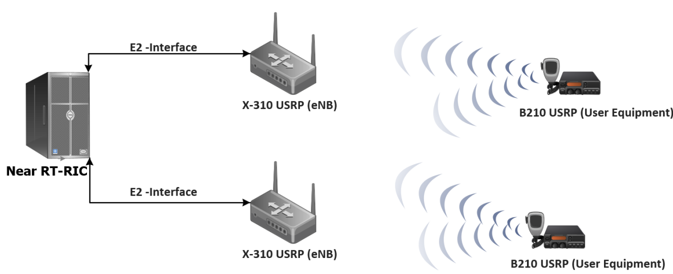

Fig. 8: O-RAN Setup on POWDER testbed

B. Network Slicing Results

The dataset is trained on the deep learning model and gives an accuracy of 92% on the training set and 93% on the validation set as shown in Fig 9.

Although these training metrics show good results with a smooth increasing accuracy and decreasing loss. The predictions however show a slight underperformance for the high-bandwidth slice as shown in the confusion matrix Fig. 10. With a majority of high-bandwidth points being classified into general. This performance can be attributed to the disparity in slice-distribution as shown in Fig. 6. However, the model is still able to predict Low Latency slices correctly with high accuracy. This suggests the need for either optimizing the thresholds or increase in data-points for high bandwidth under current threshold values.

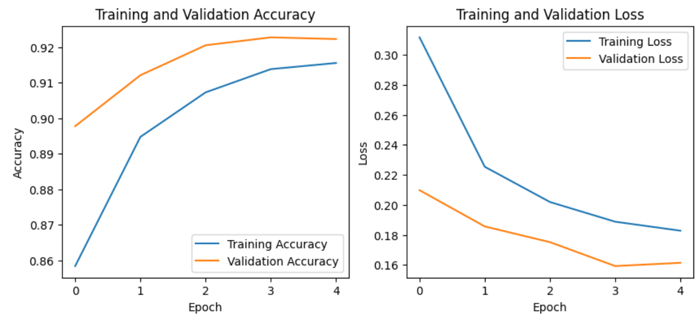

Fig. 9: Deep Learning Model Training History

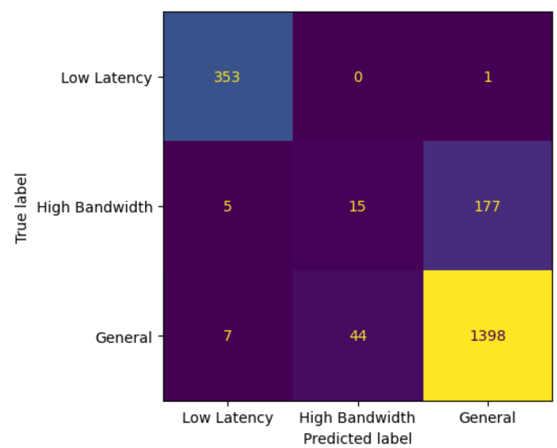

Fig. 10: Confusion Matrix for Predictions on Validation set

V. Future Directions

The current disparity in slice-type distribution for our data shows that the data points of high-bandwidth are over-fitting towards the general type, however, the results for low-latency slice clearly show the validity of our model. This disparity can also be attributed to a large amount of empty values for features at several time-stamps due to different data collection intervals as shown in Fig. 6. These empty values were filled in with the columns' mean values for the current implementation. Future work will focus on optimizing how the slice-type labels are defined as well as collecting our own dataset over the mobile vehicle endpoints available on the POWDER Platform.

VI. Conclusion

This paper has demonstrated the successful implementation of network slicing based on Deep Learning technique over the O-RAN architecture for V2X communications. The deployment of a deep learning technique in predicting network patterns for network slicing and resource allocations also further helps to optimize the O-RAN networks efficiency in real-time as compared to standard rule-based slicing. It also highlights the advantages of integrating V2X communication into the O-RAN architecture thus standardizing the V2X architecture and providing it with benefits of the O-RANs intelligent control structure thus further optimizing the V2X network efficiency and optimizations.

REFERENCES

[1] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *arXiv preprint arXiv:2202.01032*, 2022.

[2] B. Brik, H. Chergui, L. Zanzi, F. Devoti, A. Ksentini, M. S. Siddiqui, X. Costa-Pérez, and C. Verikoukis, "A survey on explainable ai for sixth generation (6g) open radio access network (o-ran): Architecture, use cases, challenges and research directions," *arXiv preprint arXiv:2307.00319*, 2023.

[3] F. Linsalata, E. Moro, M. Magarini, U. Spagnolini, and A. Capone, "Open ran-empowered v2x architecture: Challenges, opportunities, and research directions," in *2024 IEEE Vehicular Networking Conference (VNC)*, 2024, pp. 113–116.

[4] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for v2x services toward 5g," *IEEE Network*, vol. 33, no. 6, pp. 196–204, 2019.

[5] D. Johnson, D. Maas, and J. Van Der Merwe, "Nexran: Closed-loop ran slicing in powder -a top-to-bottom open-source open-ran use case," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, ser. WiNTECH '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 17–23. [Online]. Available: https://doi.org/10.1145/3477086.3480842

[6] S.-P. Yeh, S. Bhattacharya, R. Sharma, and H. Moustafa, "Deep learning for intelligent and automated network slicing in 5g open ran (oran) deployment," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 64–70, 2024.

[7] R. Hernangomez, P. Geuer, A. Palaios, D. Schäufele, C. Watermann, K. Taleb-Bouhemadi, M. Parvini, A. Krause, S. Partani, C. Vielhaus, M. Kasparick, D. F. Külzer, F. Burmeister, F. H. P. Fitzek, H. D. Schotten, G. Fettweis, and S. Stanczak, "Berlin v2x," 2022. [Online]. Available: https://dx.doi.org/10.21227/8cj7-q373