AI-Powered Real-Time Channel Awareness and 5G NR Radio Access Network Scheduling Optimization

Ying Wang
Commonwealth Cyber Initiative
Virginia Tech
Arlington, VA, USA
ywang06@vt.edu

Adam Gorski
Virginia Tech Applied Research Corporation
Commonwealth Cyber Initiative
Virginia Tech
Arlington, VA, USA
gorski@vt.edu

Luiz A. DaSilva
Commonwealth Cyber Initiative
Virginia Tech
Arlington, VA, USA
ldasilva@vt.edu

***Abstract*—As with any other wireless technology, 5G is not immune to jamming. To achieve consistent performance, network resource scheduling must be optimized in a way that reacts to jamming in the NR channel environment. This paper presents a cognitive system for real-time Channel Awareness and Radio Access Network (RAN) Scheduling (CARS) optimization based on multi-dimensional temporal machine learning models. Our system automatically detects and classifies jamming in the channel environment and optimizes scheduling based on classification results and collected link parameters. Based on over-the-air (OTA) experiments, detection and classification time is less than 0.8 seconds, which enables real-time optimization. The system is evaluated and verified for OTA experimentation through integration to our end-to-end NR system. An Automated Jamming Module (AJM) is designed and implemented. Connecting the AJM to our NR system enables a comprehensive evaluation environment for our Jamming Detection and Classification Model (JDCM) and Modulation and Coding Scheme optimization model. The improvement in connection resiliency against Control Resource Set jamming is proof of the CARS concept for real-time channel awareness and scheduling optimization. Depending on channel conditions, CARS achieves a 30% or higher improvement in NR system throughput.**

***Index Terms*—5G, NR, testbed, MCS, CORESET, RAN, AI, ML, URLLC, jamming.**

I. Introduction

With the introduction of 5G New Radio (NR) there is a swell of new applications and services entering the cellular communications ecosystem. Both NR and earlier wireless cellular networks are vulnerable to jamming attacks, which create deliberate interference to hinder the communication of legitimate users [1]. Flexibility is key in NR [2], providing performance enhancements and allowing vertical customization; however, these advances also increase the complexity of security in NR. With new 5G technology comes further need to mitigate jamming aimed towards these often reliability- and latency-centric applications and services. Traditional case-by-case anti-jamming strategies and methodologies become limited and powerless as communication link configuration flexibility increases significantly. Thus, there is a strong need to understand to what extent NR systems are vulnerable to various types of jamming and consequently equip the networks with an intelligent solution that can autonomously detect jamming and continuously learn from the experience.

Jammers are malicious wireless nodes that cause intentional interference to wireless cellular networks. Types of physical and Radio Frequency (RF) layer jammers include: a regular jammer that continuously injects RF signals and tends not to follow any specific timing protocol [15]; a random jammer that conserves its energy by alternating between active and idle states; a responsive jammer that only injects RF when its monitoring receiver determines (sometimes using a deep learning model) that the victim transmitter is active [16]; a go-next jammer that hops between different frequencies; and a control-channel jammer that targets a control channel in order to block information exchange between the victim transmitter and receiver [15]. Control channel jamming can cause a denial of service (DoS) or denial of node access [1]. In 5G, the control channel most vulnerable to jamming is the Control Resource Set (CORESET). CORESET jamming is the focus of this paper.

In general, jamming effectiveness can be increased by frequently changing the pattern and frequency of jamming. Thus, a key to jamming mitigation is accelerating the system's detection of and reaction to jamming. Fast detection and differentiation of control and non-control channel jamming are two goals of our proposed jamming detection and classification model (JDCM) for 5G. With the proper training models, machine learning (ML) has proved to be an effective way to detect jamming. Syed et al. [24] proposed an NR intrusion detection system for jamming attacks based on Kullback Leibler Divergence and Hamming Distance models. Imen et al. [25] designed an intrusion detection mechanism to limit DoS attacks in wireless sensor networks. They also implemented five ML algorithms to detect and classify DoS attacks. Yi et al. [26] presented an ML method for launching jamming attacks in wireless communications and also introduced a defense strategy. Dimitrios et al. [27] presented a method for detecting and clustering RF jamming attacks based on the use of unsupervised ML.

This paper proposes an innovative cognitive system for real-time channel awareness and RAN scheduling (CARS) optimization based on multi-dimensional temporal-based learning models. The proposed method autonomously collects data from the NR gNodeB (gNB) and user equipment (UE),

detects and classifies jamming in the channel environment, optimizes RAN scheduling, and instantaneously implements the optimized schedule at the gNB. Our main contributions are summarized as follows:

- We implement a domain knowledge translation (DKT) and jamming scenario configuration (JSC) concept-based Automated Jamming Module (AJM). The AJM generates a jamming signal based on DKT and JSC input; it is integrated into our end-to-end NR system and enables a thorough assessment of NR jamming vulnerabilities.
- We design a real-time Data Collection Module that parses data from multiple observation points and connects to multiple ML models, enabling real-time decision support.
- We enable quantitative analysis of physical and media access control layer NR vulnerabilities.
- We develop an innovative AI-based modulation and coding scheme (MCS) optimization solution to counter NR CORESET jamming. This solution enhances NR system throughput by 30% in the presence of CORESET jamming.

The attack generation, vulnerability detection, scheduling optimization, and real-time feedback cycle forms a self-learning platform to address NR vulnerabilities. Continuously learning from different types of attacks improves NR system robustness. Domain knowledge is integrated into a data-driven approach for global optimization and fast convergence. The remainder of the paper is organized as follows: we first present our experiment-based NR CORESET jamming vulnerability analysis, followed by a description of our proposed CARS optimization. Then, we present the results of testbed experiments addressing CORESET jamming, and discuss conclusions and future work. Given additional data, our proposed system can be applied to other types of jamming and interference.

II. NR CORESET JAMMING VULNERABILITY

Jamming attacks pose serious risks to public communications systems, and particularly NR, which is expected to provide connectivity for self-driving cars, smart cities, public safety, and first responders, among others. Therefore, it is essential to assess the risks of jamming attacks on NR under various jamming conditions, including the jamming signal power, the duty cycle, the frequency range, etc. Lichtman et al. [6] note several reference signals in 5G NR that are vulnerable to jamming. Uplink (UL) jamming on the Physical Uplink Control Channel (PUCCH) and Physical Uplink Shared Channel (PUSCH) are both possible; however, the PUSCH represents the bulk of the frequency frame, making it inefficient to disrupt, and any control information on the PUCCH can be replicated on the PUSCH, making the PUCCH difficult to interfere with. Compared to the UL channels, downlink (DL) control channels such as the Physical Downlink Control Channel (PDCCH) and Primary Synchronization Signal/Secondary Synchronization Signal (PSS/SSS) prove more vulnerable to jamming. The NR CORESET carries Downlink Control Information (DCI), key to link quality management [7].

The CORESET is a combination of PSS/SSS and PDCCH channels; interfering with the CORESET is resource-efficient due to its mapping onto 127 sub-carriers within the same orthogonal frequency-division multiplexed (OFDM) symbol. If an attacker knows the carrier frequency, sub-carrier spacing, and physical resource block offset it is not difficult to find the center frequency of the CORESET. Because of the control channel information carried on the CORESET, the ease of CORESET jamming is a vulnerability that can threaten the link performance on the entire NR DL bandwidth.

Modulation and Control Scheme (MCS) selection is adopted to maximize NR scheduling efficiency . To make a reasonable MCS selection the NR system must have instantaneous and historical channel quality information. Standard NR MCS selection works well when no jamming or interference is present; however, our experimentation reveals that it is vulnerable to CORESET jamming. CORESET jamming can result in an inaccurate channel quality indicator (CQI) estimation, which in turn leads to an MCS selection that is sub optimal for current channel conditions. Fig. 1 shows a comparison between CORESET jamming (control channel jamming) and data channel jamming using the data collected on the Commonwealth Cyber Initiative (CCI) end-to-end 5G testbed. The CORESET is more vulnerable to jamming than the data channel, with a decreased throughput average and an increased throughput standard deviation as jamming power increases. With CORESET jamming present, MCS selection is significantly affected and becomes unstable at a jamming power higher than -12dB; this instability is reflected in a severely decreased throughput and increased re-transmission rate.

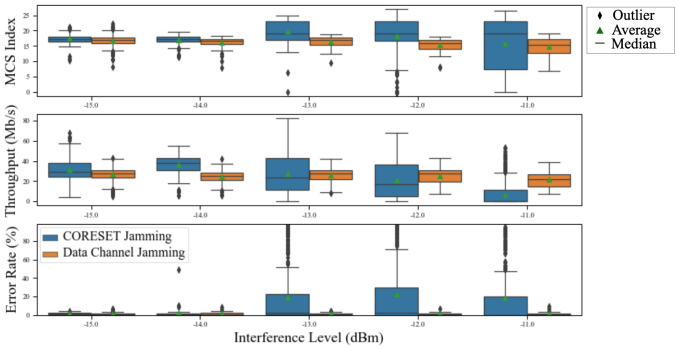

Fig. 1. MCS stability at various jamming signal power levels.

MCS selection in NR is an implementation-specific procedure. CQI is reported by the UE and can be used for MCS selection at the gNB. The MCS is periodically initialized according to the CQI and adjusted according to the packet error rate (PER). NR defines three tables of 4-bit CQIs (see Tables 5.2.2.1-1 to 5.2.2.1-3 in [19]); each CQI table is associated with one MCS table.

On the UE end, the measured CQI value becomes inaccurate in the presence of a CORESET jamming signal; in this case, the channel condition is not accurately reflected by the CQI. On the gNB end, the MCS based on reported CQI and PER

varies dramatically. Inaccurate CQI and unstable MCS leads to a lower average throughput and a higher throughput standard deviation, as well as a higher PER. The causalities among these parameters can be summarized as a chain relationship: CORESET jamming leads to poor CQI estimation, which leads to MCS changes, which in turn leads to a degradation in system performance. CORESET jamming has an indirect effect on the system performance through the intervening variable MCS. Thus, by optimizing the MCS, it is possible to eliminate performance degradation caused by CORESET jamming.

In the NR system, physical layer resources are more abundant but also more complex when compared to the 4G Long Term Evolution (LTE) standard. Thus, flexible scheduling algorithms and more accurate MCS values are of great significance in improving NR system performance. We propose a temporal-based ML model to dynamically configure the gNB and set the optimal MCS value when CORESET jamming is detected. Like the JDCM, the MCS optimization model requires a short input signal duration that allows for quick response time when jamming is detected. The MCS optimization model is trained on a non-real time basis and is stored in the SQL database to be called in real-time.

III. SYSTEM DESCRIPTION

In this paper we propose a novel RAN scheduling model (RSM) to improve overall performance of DL resource scheduling in the NR system when CORESET jamming is detected and channel quality estimation fails. The RSM is illustrated in Fig. 2. The model includes three components: jamming detection, MCS optimization, and scheduler implementation.

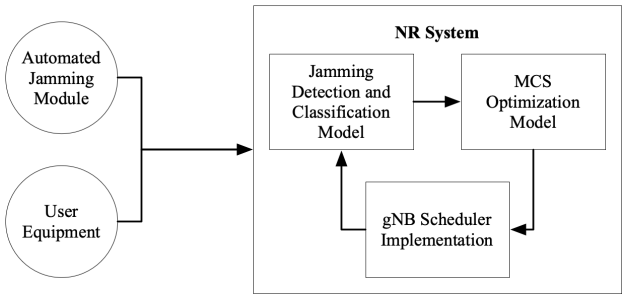

Fig. 2. RAN Scheduling Model.

As with most data-driven models, it is challenging to collect data and resource-intensive to label training data sets with comprehensive information. This process directly determines the accuracy and effectiveness of the RSM. The Automated Jamming Module (AJM) configures and generates various jamming signals. The NR signal and jamming signal data collected by the NR system are used for Jamming Detection and Classification Model (JDCM) training. The output of the JDCM triggers and is an independent feature of the MCS optimization model. Details of the AJM and JDCM are described in the next section. The gNB Scheduler implements the optimal scheduling based on the results of the MCS optimization model and is informed by collected NR system performance metrics. Optimized scheduling is implemented in real-time and serves as feedback to subsequent cycles of detection, analysis, optimization, and implementation.

A. Channel Awareness and RAN Scheduling System

The architecture of the CARS system detailed in Fig. 3. The first component of the CARS system is the AJM, which contains two input interfaces and one output interface. The first input interface, Domain Knowledge Translation (DKT), translates 3GPP standards information and integrates it into the jamming configuration and labeling process. For instance, if the targeted jamming channel is the CORESET, DKT will provide the domain knowledge corresponding to CORESET sub-carrier and frame location. The second input interface, Jamming Scenario Configuration (JSC), reflects user requirements or a test scenario description. Set preferences include frequency band, channel bandwidth, jamming power, and dynamic pattern of jamming in time and frequency domains. These preferences can be saved and reused as jamming scenarios. One jamming scenario example is an urban NR cell with a coexisting LTE signal and a high noise floor. The output interface is used for hardware configuration. Configuration information from the JSC is used in Device Interface and Configuration (DIC) to configure and subsequently run the signal generator device. The DIC process allows for abstraction of signal generator configuration away from the AJM. This abstraction removes the limitation of the system needing a specific signal generator, enabling replacement, upgrade, or coexistence of multiple signal generators within the system.

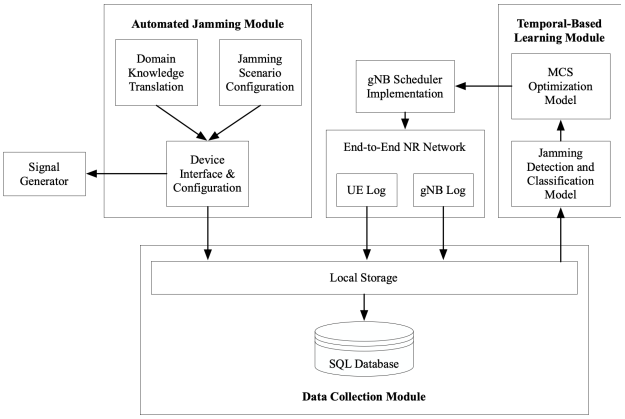

Fig. 3. NR Channel Awareness and RAN Scheduling System.

The second component of the CARS system is the CCI 5G testbed, a stable end-to-end NR network that supports real-time configuration and remote data access.

The third component of the CARS system is the Temporal-Based Learning Module, which uses various ML algorithms as well as domain and policy knowledge to analyze collected system data and execute a gNB scheduler response that optimizes NR system throughput. Both control data absent

interference and data exposed to various types of jamming are collected, parsed, and stored in an SQL-based database.

The fourth component of the CARS system is the Data Collection Module. This module includes local storage as well as a local and cloud-based SQL database that stores parsed data samples. The data collection process is detailed in the next section.

B. Data Collection Process

Fig. 4 shows the data collection process. The gNB captures control messages and user plane statistical data associated with an E-UTRAN NR Dual Connectivity (ENDC) session, storing it in JavaScript notation (json) format. The raw .json files captured by the gNB detailing the NR connection are sent to local storage where the current instance of collected data is then labeled according to jamming scenario settings. This instance of labeled data is copied to a cloud server where all historical records reside. Back in local storage, the labeled data instance is parsed, extracting key performance indicators (KPI) relevant to experimentation. Important KPI include DL channel throughput, PER, MCS, signal-to-noise ratio (SNR), CQI, and power headroom report. Using these KPI we can measure a baseline for the NR system and compare it with KPI behavior in the presence of a jamming signal. The pattern of signal degradation informs the effectiveness of jamming on the DL channel throughput distribution, which can then be used to train the JDCM. During experimentation, parsed data can be visualized in real-time. The parsed data is stored in an SQL database in the cloud server. The SQL database allows the components of CARS to be distributed in different computers or Local Area Networks (LANs). The SQL database is used in the JDCM, whereas the historical labeled raw data is used as a backup. The structure of one SQL database is shown in Fig. 5.

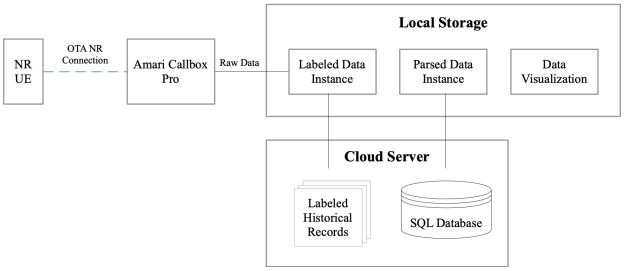

Fig. 4. Data collection process.

C. ML-Based Jamming Detection and Classification

The labeled data flowing into the JDCM acts as the foundation and accelerator for ML-based jamming classification. In current research, ML in NR systems presents some limitations due to the high volume of data (equivalent to real-world commercial network loading) needed to build effective models for optimal over-the-air (OTA) performance. The CARS system and intelligent SQL database meet these demands and continuously boost the accuracy and effectiveness of the

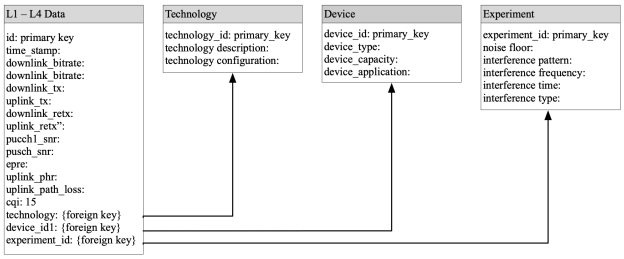

Fig. 5. SQL database structure.

JDCM. In a jammed environment, channel variations can be unpredictable. A model that is able to quickly assess the channel environment and immediately take action can significantly improve network performance. For this purpose we propose a temporal-based learning model. Similar to sequential learning algorithms like Long Short Term Memory (LSTM), the temporal-based learning model can not only process single data points, but also data sequences. Moreover, compared to LSTM, a temporal-based learning model requires less data and shorter data sequences to reach the same level of accuracy.

Data is taken from a period of time nT where n is an integer and T is the sampling time period. This data is fed into a temporal-based ML model. The labeled data where a targeted type of jamming is detected is marked as positive and the labeled data where it is not detected is marked as negative. Throughout the detection process we experimented with several ML techniques, including logistic learning, random forest, decision tree, adaptive boosting, support vector machine, and expectation-maximization. A logistic learning model was selected based on performance accuracy and the amount of data required. The model involves less computational complexity and is more effective compared to LSTM and others. The experimentation section provides a comparison of the performance of each ML technique in the context of jamming detection and classification.

D. MCS Optimization Model

Total NR system throughput can be calculated as the throughput sum of all connected User Equipments (UEs). The throughput of each UE is dependent on the MCS and PER as well as packet size and distribution, control overhead, and upper layer configuration. The gNB-controlled MCS index is a value between 0 and 28 that determines modulation order, target code rate, and spectral efficiency. An overestimation of the MCS for the current channel condition can cause a dramatic increase in PER. Bonafé et al. [14] calculated expected PER at different MCS levels; results show that at any given channel condition, an increase in MCS results in an increase in PER. For example, for a channel with a SNR of 10dB and MCS 6, PER is 0.1%. However, if we choose MCS 10 the PER increases by a factor of 10, and if we choose MCS 16 the PER increases by a factor of 100.

Collection of training data for the MCS optimization model requires sweeping through relevant jamming frequency, jamming power, jamming bandwidth, and MCS index values.

Various NR system KPI are logged and labeled automatically. All collected data is injected into a temporal-based learning model for MCS optimization. KPI used in the JDCM are the same as those used in the MCS optimization model. We are able to accomplish three things with the collected data: first, we get a clear picture of the effect that the MCS index has on NR system throughput under various jamming conditions. Second, we can determine the ideal MCS index for different jamming levels and categories. Finally, we can compare the performance of the model's ideal MCS index with the performance of the system's auto-set MCS index; in this way we are able to improve upon the MCS index setting in the presence of jamming.

The MCS optimization ML model is shown in Fig. 6. The value of n is empirically determined and must be large enough to detect the patterns needed to differentiate between different MCS settings. Observation time is represented as nT.

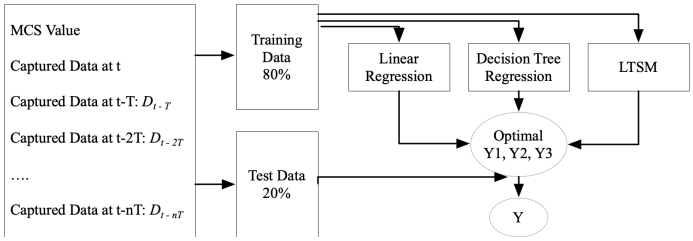

Fig. 6. Temporal-based learning module.

The description of the model is as follows. Given a series of observations $X(1)$, $X(2)$, $X(3)$, ..., $X(t)$, ..., $X(\Delta)$, we train regression models to predict the value labels y_1 and y_2. Here, t indexes sequence steps and Δ represents the length of the sequence.

- Independent variables (observations), $X(t)$ include features D_t, D_{t-T}, ..., D_{t-nT} and an MCS value. Each set of D_t includes the parameters in (1). The value of n is empirically determined as the value needed for differentiating patterns of signals in physical layer.
- The dependent variables y_1, y_2, are respectively the average and standard deviation (SD) of throughput for the given MCS value in $X(t)$. Two separate regression models are trained for predicting the value of y_1 and y_2.
- MCS indexes 0 to 28, as well as jamming signal frequency and power, are traversed to train the models.
- The MCS index predicted to yield the maximum throughput with minimal standard deviation is selected as the optimal MCS at time t.
- Gradient descent is used as the change in each weight to ensure that the weights quickly converge to a result without oscillations.
- Multiple models are compared to select the one that fits the data most closely.

At time t, the parameters of the NR communication link feeding into features D_{t-nT} include throughput th_t, signal-to-noise ratio snr_t, CQI cqi_t, PER per_t, power headroom phr_t, energy per resource element $epre_t$, and UL path loss pl_t.

$$D_t = [th_t, snr_t, cqi_t, per_t, phr_t, epre_t, pl_t] \tag{1}$$

Dependent variables (labels) of the model are shown in (2) and (3). We separately train two regression models to predict the average of the throughput y_1 and standard deviation of the throughput y_2 at a given MCS index k. t_0 and $t_0 + \delta T$ are the start and end times of data collection, respectively. th_t is the measured throughput value at time t when MCS is set to k.

$$y_1 = \frac{1}{\delta} \sum_{t=t_0}^{t_0+\delta T} th_t \tag{2}$$

$$y_2 = \sqrt{\frac{\sum_{t_0}^{t_0+\delta T} (th_t - y_1)^2}{\delta - 1}} \tag{3}$$

When CORESET jamming is detected we can predict the expected throughput average and standard deviation at different MCS levels. Prioritization of maximizing average throughput or minimizing throughput standard deviation is dependent on use-case scenario and can factor into optimal MCS selection. An example is: $max(p_1 y_1 - (1 - p_1) y_2)$ in which case if maximizing the value of average throughput is preferred, then p_1 is set closer to 1; if minimizing the value of throughput standard deviation is preferred, then, p_1 is set closer to 0.

IV. Results

The results in this section are based on the data collected in the CCI 5G testbed integrated with the CARS system. The CCI 5G testbed is equipped with a commercially available Amari Callbox Pro which provides NR core network capability and software-configurable eNB and gNB RAN capabilities. We connect a 5G-capable Android UE Samsung S20 to this RAN. Considering current commercially available UE frequency band support, we selected the E-UTRAN New Radio – Dual Connectivity (ENDC) Non-Standalone mode setting of LTE b2 (1960 MHz DL) and NR n71 (634.5 MHz DL) for all experiments reported in this paper.

A. Jamming Detection and Classification Modeling

We test the efficacy of several ML models in jamming detection and classification. These models include: Logistic Regression, Ada Boost, Quadratic Discriminant Analysis, Random Forest, and Decision Tree. Fig. 7 plots the Receiver Operating Characteristic (ROC) curve for each model and a comparison of Area Under Curve (AUC) values. Table I shows a direct comparison of the resulting precision, recall, and F1 parameters. A deep learning model is not selected due to the volume of data it requires and its computation complexity. With the amount of data collected for this experiment, the deep learning model exhibits instability and performs less accurately compared to Ada Boost and Logistic Regression. Both Ada Boost and Logistic Regression show high AUC and model robustness. When comparing optimal recall and precision the Logistic Regression model demonstrates the highest performance. Thus, the Logistic Regression model is selected and used in the JDCM.

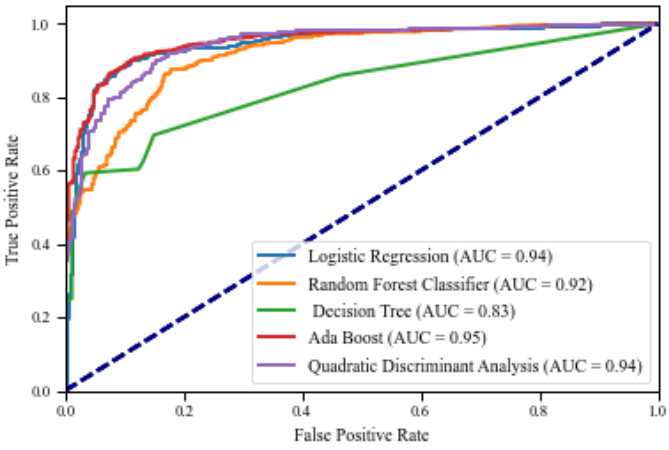

Fig. 7. Jamming Detection and Classification Model AUC/ROC comparison.

TABLE I
JAMMING CLASSIFICATION MODEL ACCURACY ANALYSIS

Classification Model	*Precision*	*Recall*	*F1*
Logistic Regression	0.87	0.99	0.93
Random Forest	0.83	1.00	0.91
Decision Tree	0.86	0.97	0.91
Ada Boost	0.86	0.97	0.91
Quadratic Discriminant Analysis	0.94	0.50	0.65

B. MCS Optimization Modeling

The first step of MCS optimization is to traverse all MCS values in the range $[0, 28]$, then predict the throughput average and standard deviation at each MCS. Different from the classification model used for the JDCM, MCS prediction uses a regression model. Regression results are generated based on a decision tree regression algorithm. Based on the predicted value of throughput average and standard deviation, the second step of MCS optimization is to apply the predicted values to a specific use case to select the optimal MCS. This section contains two sets of results: the regression model and the accuracy of MCS selection.

Decision tree regression is selected for optimal MCS determination. Linear regression and LSTM were evaluated as well; however, both yield lower performance. Decision tree regression builds the model in the form of a tree structure. It breaks the dataset down into smaller subsets while simultaneously and incrementally developing an associated decision tree. The result is a decision tree with decision nodes and leaf nodes. This model performs the best due to its recognition of the jamming pattern and relationship between selected MCS and predicted throughput.

Fig. 8 shows average and standard deviation of throughput when a randomly changing jamming frequency and a randomly traversed MCS input are used. The blue line indicates the predicted throughput value, and the orange line shows the actual throughput value. Overall, the R^2 value of the average throughput regression model is 0.94 and the R^2 value of the standard deviation of throughput regression model is 0.86.

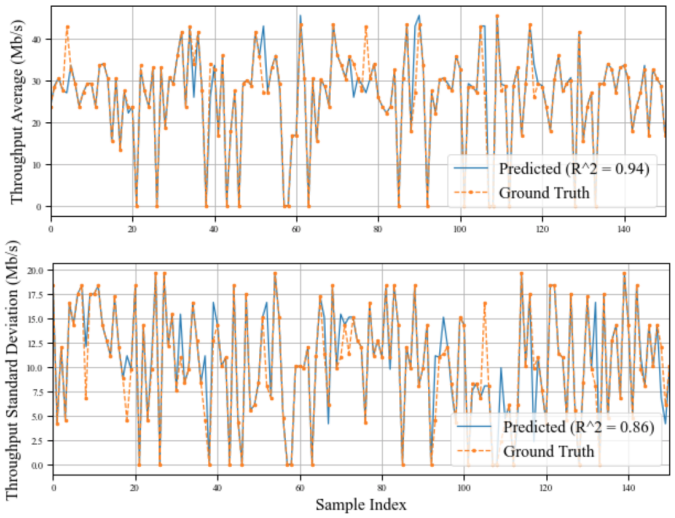

Fig. 8. Average and standard deviation of throughput regression at mixed MCS levels.

Fig. 9 illustrates the impact of the weight of the throughput average on accuracy of MCS optimization in the presence of CORESET jamming. A correct prediction is defined as the selected MCS resulting in a throughput greater than 90% of the optimal throughput performance. A throughput average weight approaching 1 signifies a greater importance in maximizing average throughput. A throughput average weight approaching 0 signifies a greater importance of minimizing throughput standard deviation. As illustrated in Fig. 8, the MCS prediction for throughput standard deviation is less accurate than the MCS prediction for throughput average, leading to the decline in the prediction result when the throughput average weight is less than the standard deviation weight. The solid blue line represents the estimated accuracy of the MCS optimization process with no CORESET jamming. The dashed orange line represents the estimated accuracy of MCS optimization with CORESET jamming given the assumption that CORESET jamming is correctly detected. Throughput performance may vary as the criteria of correct prediction of MCS optimization changes or throughput average weight changes.

C. System Performance

Fig. 10 compares the overall NR system throughput improvement with CARS, in which the JDCM and MCS optimization model are integrated. When the MCS is automatically set by the NR system, the data connection is vulnerable to a significant decrease in throughput in the presence of CORESET jamming. When the data connection drops due to jamming, throughput approaches 0. The application of CARS leads to increased throughput and increased resilience to a CORESET jamming signal.

V. CONCLUSION

In this paper we have proposed a cognitive system for real-time channel awareness and RAN scheduling optimization based on multi-dimensional temporal-based learning models.

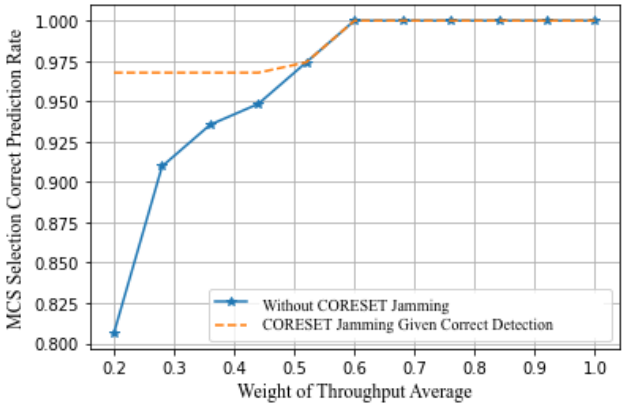

Fig. 9. Decision accuracy for MCS optimization.

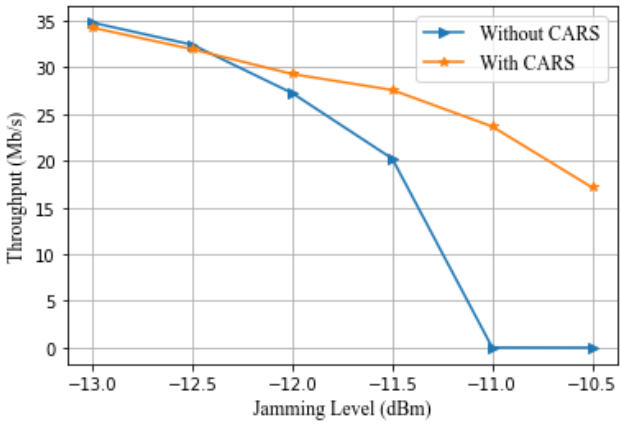

Fig. 10. System performance comparison with and without CARS.

Jamming generation, jamming detection, system optimization, and real-time implementation form a closed-loop self-learning platform that addresses NR jamming vulnerabilities and improves NR robustness. Further configuration and data collection with the CARS system can enable classification of a larger set of malicious and non-malicious interference signals. Future work includes outdoor experimentation with traffic from many users in distributed geographic areas which can improve the robustness and applicability of the JDCM and MCS optimization model through collection of higher-variability data. The future work includes deploying the CARS in outdoor environments with a large-scale number of devices and users.

VI. ACKNOWLEDGMENT

This work was funded by the Commonwealth Cyber Initiative (CCI), a research, innovation, and workforce initiative of the Commonwealth of Virginia.

REFERENCES

[1] Arjoune, Youness, and Saleh Faruque. “Smart Jamming Attacks in 5G New Radio: A Review.” IEEE, 2020. 1010–1015. Web.
[2] Tripathi, Nishith, Reed, Jeffrey ”5G Cellular Communications: Journey Destination”, www.thewirelessuniversity.com, 2019
[3] O. Osanaiye, A. Alfa, and G. Hancke, “A statistical approach to detect jamming attacks in wireless sensor networks,” Sensors, vol. 18, no. 6, p. 1691, 2018.
[4] A.Marttinen, A.M.Wyglinski, and R.Jantti,“Statistics-based jamming detection algorithm for jamming attacks against tactical manets,” in 2014 IEEE Military Communications Conference. IEEE, 2014, pp. 501–506.
[5] Ali-Tolppa, Janne et al. “SELF-HEALING AND RESILIENCE IN FUTURE 5G COGNITIVE AUTONOMOUS NETWORKS.” ITU, 2018. 1–8. Web.
[6] Lichtman, Marc et al. “5G NR Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation.” IEEE, 2018. 1–6. Web.
[7] Takeda, Kazuki et al. “Understanding the Heart of the 5G Air Interface: An Overview of Physical Downlink Control Channel for 5G New Radio (NR).” (2019): n. pag. Print.
[8] Braun, Volker, Karol Schober, and Esa Tiirola. “5G NR Physical Downlink Control Channel: Design, Performance and Enhancements.” IEEE, 2019. 1–6. Web.
[9] Girke, Felix et al. “Towards Resilient 5G: Lessons Learned from Experimental Evaluations of LTE Uplink Jamming.” (2019): n. pag. Print.
[10] Rupprecht, David et al. “Breaking LTE on Layer Two.” IEEE, 2019. 1121–1136. Web.
[11] Liu, Jin et al. “Initial Access, Mobility, and User-Centric Multi-Beam Operation in 5G New Radio.” IEEE communications magazine56.3 (2018): 35–41. Web.
[12] Lagen, Sandra et al. “New Radio Physical Layer Abstraction for System-Level Simulations of 5G Networks.” (2020): n. pag. Print.
[13] Yin, Hao et al. “Predicting Channel Quality Indicators for 5G Downlink Scheduling in a Deep Learning Approach.” (2020): 1-6. N. pag. Print.
[14] Olmos Bonafé, Juan José, Albert Serra, and Sílvia Ruiz Boqué. On the Definition of Reference Scenarios for LTE-A Link Level Simulations Within COST IC1004. N.p., 2013. Print.
[15] K.Grover,A.Lim,and Q.Yang,“Jamming and anti-jamming techniques in wireless networks: a survey,” International Journal of Ad Hoc and Ubiquitous Computing, vol. 17, no. 4, pp. 197–215, 2014.
[16] T. Erpek, Y. E. Sagduyu, and Y. Shi, “Deep learning for launching and mitigating wireless jamming attacks,” IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 1, pp. 2–14, 2018.
[17] M. Mezzavilla et al., “A lightweight and accurate link abstraction model for the system-level simulation of LTE networks in ns-3,” Proceedings of the 15th ACM Int. Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Oct. 2012.
[18] R. Patidar et al., “Link-to-System Mapping for ns-3 Wi-Fi OFDM Error Models,” Workshop on ns-3, June 2017.
[19] 3GPP TS 38.214, TSG RAN; NR; Physical layer procedures for data, Release 15, v15.5.0, Mar. 2019.
[20] Yang, Shun-Ren et al. “Multi-Access Edge Computing Enhanced Video Streaming: Proof-of-Concept Implementation and Prediction/QoE Models.” IEEE transactions on vehicular technology 68.2 (2019): 1888–1902. Web.
[21] Ma, Bo, Weisi Guo, and Jie Zhang. “A Survey of Online Data-Driven Proactive 5G Network Optimisation Using Machine Learning.” IEEE access 8 (2020): 35606–35637.
[22] Wang, Wang. “Physical-Layer Security of 5G Wireless Networks for IoT: Challenges and Opportunities.” IEEE internet of things journal 6.5 (2019): 8169–8181. Web.
[23] Hachimi, Kaddoum. “Multi-Stage Jamming Attacks Detection Using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks.” (2020): n. pag. Print.
[24] S. M. Danish, A. Nasir, H. K. Qureshi, A. B. Ashfaq, S. Mumtaz and J. Rodriguez, ”Network Intrusion Detection System for Jamming Attack in LoRaWAN Join Procedure,” 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018.
[25] Iman Almomani, Bassam Al-Kasasbeh, and Mousa AL-Akhras, WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks, Journal of Sensors, 16 pages, 2016.
[26] Y. Shi, Y. E. Sagduyu, and J. H. Li, ”Adversarial Deep Learning for Cognitive Radio Security: Jamming Attack and Defense Strategies,” 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, 2018, pp. 1-6.
[27] D. Karagiannis, A. Argyriou, ”Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning,” Vehicular Communications, Volume 13, 2018, Pages 56-63.