5G V2X Misbehavior Detection as Edge Core Network Function based on AI/ML

Hadi Yakan, Ilhem Fajjari, Nadjib Aitsaadi, Cédric Adjih

5G V2X Misbehavior Detection as Edge Core Network Function based on AI/ML

Hadi Yakan*, Ilhem Fajjari†, Nadjib Aitsaadi*, Cedric Adjih‡
*Université Paris-Saclay, UVSQ, DAVID, F-78035, Versailles, France
†Orange Innovation, F-92320, Chatillon, France
‡Inria Saclay, F-91120, Palaiseau, France
firstname.lastname@ *uvsq.fr, †orange.com, ‡inria.fr

***Abstract*—As 5G Cellular Vehicle-to-Everything (C-V2X) technology takes the lead in V2X communication, it opens the possibility for telecommunication service providers to offer Vehicle-to-Network (V2N) services using their existing 5G network infrastructure. To enhance the security of 5G V2N services, in this paper we propose a novel collaborative V2X misbehavior detection system. This system would safeguard the V2X application servers (V2X ASs), deployed in the 5G edge network, from any malicious V2X position manipulation attacks. Our proposal includes two enhanced machine learning models. The first model utilizes historical data to conduct On-Road Plausibility Checks (ORPC), while the second model builds upon the first by enabling collaboration among edge detection nodes through the sharing of attack ratios for each vehicle. Our proposed models were tested using extensive 5G core-network emulations, yielding excellent results. The first model achieved a notable accuracy improvement from 73% to 91%, while the second model further enhanced the accuracy to an impressive 95%.**

***Keywords*—5G, V2X, C-ITS, Security, Misbehavior Detection, Machine Learning, MEC, Edge.**

I. Introduction

Every year, approximately 1.35 million people die due to road accidents worldwide, which translates to the loss of one life every 24 seconds. Despite legal and legislative efforts to improve road safety, the number of road fatalities remains unacceptably high. In response, Cooperative Intelligent Transport System (C-ITS) protocols and applications have been developed primarily to improve road safety and reduce accidents. These technologies offer a range of benefits, including traffic management, cooperative driving, and access to real-time information and entertainment.

C-ITS applications aim to enhance awareness and safety for all road users by enabling efficient communication between vehicles, infrastructure, pedestrians, and the Internet. This is made possible through the use of Vehicle-to-Everything (V2X) technology, which facilitates communication between these applications. By providing real-time information about traffic, weather conditions, and potential hazards, V2X helps to improve the comfort and safety of drivers, passengers, and pedestrians. V2X communications take many different forms, including: Vehicle-to-Vehicle (V2V), Vehicle-to-Pedestrian (V2P), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N). The latter form of communication enables vehicles to connect to V2X application servers and other network-hosted applications on the Cloud/Edge servers. V2N gives cars wide-area network capabilities and enables them to access various services, such as real-time traffic updates, weather information, and entertainment.

In the near future, it is expected that 5G V2X will become the standard communication method for vehicles, and that cloud-based C-ITS services will become widespread. As the V2X market grows, so will the number of C-ITS service providers and the services they offer. V2N communication will become increasingly important for these service providers, allowing for centralized road hazard warnings, traffic efficiency improvements, weather alerts and forecasts, pollution monitoring, and entertainment services to be provided to cars. However, securing these V2N-based services in this dynamic vehicular network environment remains a significant challenge that needs to be addressed.

Securing V2X communications against attacks is a critical challenge. Attackers can compromise the confidentiality, integrity, availability, and authenticity of V2X communications by employing tactics such as eavesdropping, location tracking, message manipulation, position falsification, blackhole and flooding attacks, impersonation, and sybil attacks. Addressing these challenges requires the development of advanced V2X-specific security measures to ensure the safety and privacy of V2X communications.

In this paper, we present a novel V2X edge misbehavior detection system that utilizes machine learning techniques and is compliant with 3GPP 5G-V2X architecture specifications. Our proposed system aims to detect and prevent position falsification attacks that may occur during V2N edge communications, thus ensuring the authenticity of data received by V2X application servers. The misbehavior detection application instances are implemented in the Edge, where they are interconnected and can collaborate to improve detection accuracy. We also provide an evaluation of the proposed system to demonstrate its effectiveness in detecting such attacks.

The paper is organized as follows. Section II presents the problem statement. In Section III, we review related work addressing V2X security and misbehavior detection. Section IV details the architecture of our proposed solution. Then, in Section V, we describe the two proposed machine learning models. Section VI provides the results of offline and online performance evaluations. Finally, in Section VII, we conclude the paper.

II. Problem statement

The current V2X security solutions rely heavily on cryptography, the majority of these solutions implement a Public Key Infrastructure (PKI) system to distribute and validate vehicular certificates. Vehicular certificates are typically smaller

in size than regular certificates and are used to authenticate vehicles. However, there are some limitations to the use of PKI in V2X, such as the high computational overhead required for signing and verifying certificates and the potential for a single point of failure in the PKI system. While Public Key Infrastructure (PKI) systems are crucial for protecting against external threats, additional measures are necessary to mitigate attacks that could be launched by insiders who have already authenticated and are part of the network. To address this issue, a Misbehavior Detection System (MDS) can be established, which acts like an Intrusion Detection/Prevention System (IDS/IPS) in traditional IT networks.

While numerous studies have investigated V2V misbehavior detection, it is possible for a vehicle to behave appropriately on V2V while acting maliciously on V2N. In such cases, traditional V2V misbehavior detection techniques will fail to detect and mitigate the attack. In this research paper, we focus on five types of position manipulation attacks that were identified in the VeReMi dataset [1]. These include: Attack type 1, where the attacker sends the same position information despite changes in its movement; Attack type 2, in which the attacker consistently adds a fixed value to its actual location, resulting in a parallel path; Attack type 4, where the attacker produces a new random location for each message it delivers; Attack type 8, where the attacker increments its actual location by a random value every time a message is sent; and Attack type 16, in which the attacker initially sends its precise location and then, after a variable period, starts sending a spoofed fixed position. These forms of position manipulation attacks can have serious consequences, particularly for C-ITS services that rely heavily on accurate location data.
As 5G V2X networks enable the deployment of low-latency V2N services, it is important to ensure that misbehaving vehicles cannot compromise these services. While V2V misbehavior detection is useful in some cases, it may not be enough to detect misbehaving vehicles on V2N. Therefore, we propose a V2X misbehavior detection system for the 5G Edge network that utilizes two advanced machine learning models to improve detection accuracy.

III. Related work

Ghosal et al. [2] provided a comprehensive overview of the V2X architecture and applications, along with an analysis of security challenges and requirements for both IEEE 802.11p and cellular-based V2X. The authors also proposed a classification scheme for different types of V2X attacks. To mitigate these attacks, the research community has proposed solutions in three major domains: i) symmetric key cryptography, ii) message authentication, and iii) privacy preservation using PKI-based, identity-based, or group-based solutions. The paper serves as an important reference for understanding the security issues and solutions in the V2X domain.

Lu et al. provide a comprehensive overview of the challenges and strategies to secure V2X services in a 5G network environment in [3]. The challenges are categorized into three groups: trust, security, and privacy. The authors also provide a summary of the key research papers addressing these challenges. The proposed solutions are classified based on their location in the 5G architecture: Data Network (DN)/Internet, 5G Core Networks, Network Edge, or V2X Communications Layer. This classification helps readers better understand the applicability and effectiveness of the proposed solutions in different parts of the 5G network.

Van der Heijden et al. developed VeReMi, a publicly available dataset of V2X attacks in [1], which was created using the SUMO [4] and Veins vehicular network simulators [5], based on the LuST scenario [6]. The dataset includes traces of both normally behaving and misbehaving vehicles, where the misbehaving vehicles perform five different position falsification attacks in V2V. The authors used the dataset to compare different plausibility checks, such as Acceptance Range Threshold (ART) [7], Simple Speed Check (SSC) [8], Sudden Appearance Warning (SAW), and Distance Moved Verifier (DMV) [9]. This dataset has been useful for developing and evaluating V2X misbehavior detection systems.

In [10], Bißmeyer et al. proposed a centralized misbehavior detection system for VANETs. This system receives and analyzes misbehavior reports sent by network nodes upon detection of an incident. Based on the plausibility of the data received in these reports, the centralized system makes the final decision about whether the reported vehicle is behaving correctly or not. The proposed system employs a Bayesian network to calculate the probability of misbehavior based on the reported data, and it is shown to be effective in detecting various types of misbehavior attacks in simulations.

In [11], So et al. proposed a V2V misbehavior detection technique based on machine learning. They first created a machine learning version of the VeReMi dataset [1], and then proposed a model that uses six input features: two plausibility checks and four quantitative data about the vehicle's movement. The authors evaluated two machine learning algorithms, k-nearest neighbors (KNN) and support vector machine (SVM), and showed that both outperformed plausibility checks alone. In another study [12], Sharma et al. compared multiple machine learning algorithms, including Naïve Bayes (NB), Random Forest (RF), and Ensemble Boosting and Voting, and found that the RF and Ensemble algorithms performed better than the others.

In [13], Gyawali et al. presented a collaborative misbehavior detection system for V2V that leverages the feedback from vehicles to vehicular edge computing servers to predict the number of true feedbacks. They also used deep reinforcement learning to identify the optimal strategy for updating the reputation policy, which encourages vehicles to provide authentic feedback. In contrast, our proposed solution in this paper focuses on safeguarding V2N communications and emphasizes the collaborative aspect between misbehavior detection instances that run in the edge network of adjacent 5G service areas. While Gyawali et al. focused on V2V misbehavior detection, our work specifically addresses the need for V2X misbehavior detection to enhance the security of V2N services, which require low-latency and high-reliability communication.

While many misbehavior detection systems for V2X focus on V2V communications, protecting V2N communications and C-ITS service providers is equally important. To address this, we propose a collaborative security solution that operates within the 5G network edge and is compliant with 3GPP specifications. Our system is coupled with the 5G Core network

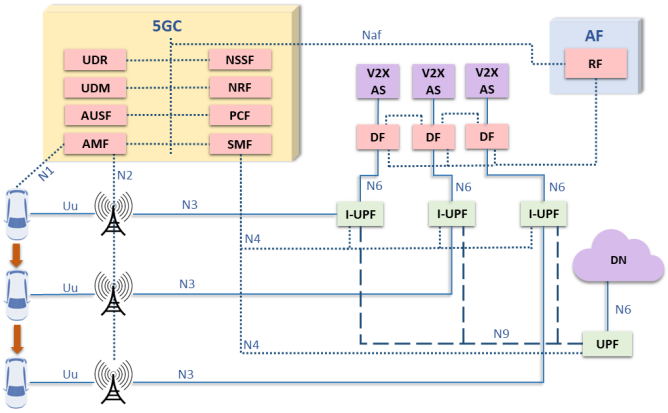

Fig. 1. 5G Edge Misbehavior Detection Architecture Proposal

(5GC) and is designed to protect V2X application servers from position falsification attacks, as listed in the VeReMi dataset. By sharing previous detection information between nodes, our system is able to analyze new V2X data in real-time and enhance the detection of attacks. When an attack is detected, the reporting entity can initiate countermeasures against the attacker by interacting with the 5G control-plane to prevent further attacks. Our approach provides a comprehensive and efficient solution for protecting V2N communications and improving the overall security of V2X systems.

IV. 5G Edge Misbehavior Detection Architecture

V2X Application Servers can gather information from vehicles and also provide them with a variety of C-ITS services. V2X ASs are hosted in the Edge, in the Cloud, or on the Internet. Regardless of their deployment location, V2X ASs need to be protected from position manipulation attacks and misbehaviors. To achieve this, a misbehavior detection system needs to be in place. It can be deployed either centralized or distributed with many instances. The deployment option of the misbehavior detection system depends on the architecture of the C-ITS application it is protecting. If the C-ITS application requires low latency, then it might be deployed in the edge network, where instances of the application are created on LADN (Local Access Data Network) next to local I-UPFs (Intermediate-UPFs). For each C-ITS application instance (V2X AS), a misbehavior Detection Function (DF) instance is created to protect it, as shown in Fig. 1.

Our proposed misbehavior detection system consists of two main components: one or many instances of i) Detection Functions (DFs): responsible for real-time analysis and monitoring of V2X traffic packets on the UP; and a single instance of ii) Reporting Function (RF): integrated within the 5G core network CP to allow telecommunications operators or legal authorities to revoke the access of reported malicious vehicles and stop the attack. It acts as an Application Function (AF), which controls the application's traffic flows by interacting with the 5GC network through 3GPP standard APIs.

Each DF instance will analyze the V2X traffic flowing through the path between vehicles and local V2X AS. DF instances can communicate and exchange detection information related to vehicles moving between their respective coverage areas.

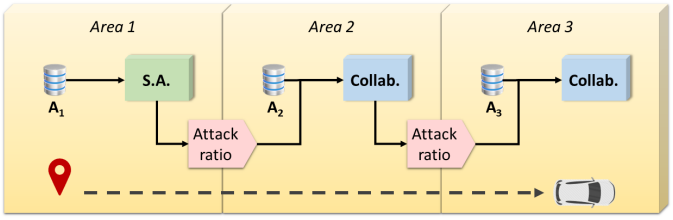

Fig. 2. Machine Learning Models

V. ORPC Proposal: AI-based Detection

We propose two machine learning models: Standalone (S.A.) and Collaborative (Collab.). Each proposed model provides an improvement over [11], where the authors proposed using machine learning to enhance misbehavior detection results. Their proposal utilizes six different input features to predict if the vehicle's behavior is benign or not. Four out of the six features use quantitative values, and they are: i) Difference between the calculated average velocity based on displacement and time and the predicted average velocity based on reported velocity and time in the X direction, ii) Same as feature i) but in the Y direction, iii) The magnitude of features i) and ii), iv) Difference between displacement based on calculated distance and predicted displacement based on average velocity. The two remaining features are plausibility checks' results, namely: v) Location Plausibility Check (LPC) result and vi) Movement Plausibility Check (MPC) result.
We propose two additional features, which are detailed later in this section: vii) On-Road Plausibility Check (ORPC), used in both S.A. and Collab.; and viii) Attack Ratio from the previous area, used in Collab. only.

Location Plausibility Check uses the current vehicle's speed and location, average acceleration, and Gaussian distribution to provide Confidence Intervals (CIs) for predicting the plausible range of future vehicle position in both the X and Y directions. If the new location reported by the vehicle falls inside the 95% Confidence Interval (CI) in both directions, the new position's score is set to 0, indicating that the position is probable. However, the score is incremented by 1 per direction if the new location is considered to be outside 95% CI but inside 99% CI. Lastly, the score is incremented by 2 per direction if the new location falls outside the 99% CI. The score range for the location plausibility check is thus between 0 and 4, where a score of 4 indicates that the new location received from the vehicle is most probably not realistic.

Movement Plausibility Check (MPC) is a comparison between displacement and velocity. If the vehicle is reporting no displacement but the average velocity is not zero, the score is set to 1, if not, the score is 0.
S.A. utilizes seven features i to vii, while Collab. has eight i to viii. For simplicity, the first seven features are represented in Fig. 2 as "A". Collab. uses all the input features used in S.A., however, it has an eighth feature that leverages the resulting attack ratio calculated by a previous model.

Our proposal, On-Road Plausibility Check (ORPC), evaluates whether the newly received position is on the road or not. To do this, ORPC utilizes previous anonymous location data collected from cars that have already traversed the covered

region. Latitudes and longitudes are recorded without any vehicle identification or labeling. They denote locations where vehicles are often anticipated to pass. When a new location is received, we can thus determine how close it is to the nearest recorded position. If it is relatively close, the new location is plausible, but if it is distant, it is considered improbable.

As the historical data is unlabeled, supervised machine learning cannot be applied. Instead, we must use unsupervised machine learning, like anomaly detection. It is the process of identifying data points that deviate from the anticipated pattern of a particular group. One of the strategies for detecting anomalies uses KNN functions to accomplish the assessment.

The KNN functions are used in a non-traditional manner in order to discover anomalies using the following method: i) Fitting a KNN model to the historical location data. ii) Estimating an average distance between adjacent recorded locations. iii) Estimating the optimal cut-off distance value; if it is exceeded, the new location is deemed abnormal. iv) Computing the distance between the current location of the vehicle and the closest recorded historical location (nearest neighbor). v) Comparing the determined distance to the cut-off value. If the estimated distance is less than the cut-off value, the new location is considered to be on-road and the ORPC result is set to 0. In contrast, if the distance exceeds the threshold, the location is deemed off-road and consequently improbable, and the ORPC result is set to 1.

The eighth feature, "Attack Ratio" of a vehicle is the ratio of the number of attack messages predicted by the previous model to the total number of messages sent by the vehicle while passing through the previous area. It is only shared upon the vehicle's transition from one area to another. It acts as a reputation for the car that is exchanged between areas, which is similar to the machine learning concept proposed in [14].

For instance, in Fig. 2, when a new vehicle enters Area 1 and it's not known by any other neighboring area, S.A. is chosen to analyze this vehicle's traffic because the previous attack ratio cannot be determined. Once this vehicle moves to Area 2, the attack ratio calculated in Area 1 will be transmitted and used as an input to Collab. instance of Area 2. When this car moves from Area 2 to Area 3, the new attack ratio predicted by the Collab. instance in Area 2 will transition as an input feature to the next Collab. instance in Area 3, and so on.

VI. Performance Evaluation

To evaluate the performance and efficiency of our proposed ORPC scheme, we utilize two evaluation approaches: i) offline: numerically with VeReMi-ML dataset using Python and Scikit-learn [15], and ii) online: using the free5GC [16] 5G network emulator and real-time attacks and predictions.

We utilize the VeReMi-ML dataset to generate traffic sent by vehicles to the V2X application servers. During the offline evaluation, we use the recorded traces of both normal and false vehicles' positions in the dataset. Since the dataset was initially designed for VANET, we had to make adjustments to accommodate our V2N scenario environment. The primary modifications are: i) Sort by sender: The original dataset is sorted according to the sender/receiver pair. Assuming complete 5G coverage and full message visibility, the first change we made was to consolidate all the dataset's messages, sort them by sender ID and sending time, and then filter and eliminate duplicate messages. By the end of this step, we will have a list of transmitted messages per sender ID. ii) Real-time assessment: While the original algorithm executes prediction after receiving the very last message from a vehicle being examined, we opt to use real-time assessment, which utilizes the two most recently received messages only. This modification allows triggering prediction immediately after the reception of every message, which helps make real-time evaluations while restricting the number of cached locations to only one, preserving the privacy of the path.

TABLE I
Offline Results: ORPC

ML Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
w/o ORPC	0.8926	0.9131	0.6921	0.7874
w/ ORPC	0.9504	0.9609	0.8626	0.9091

The performance of the models is determined by the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) they produce. *Accuracy* is the ratio of correctly classified instances to the total number of instances: $\frac{TP+TN}{TP+TN+FP+FN}$. *Precision* indicates the ratio of correctly predicted attacks to the overall detection predicted by the model: $\frac{TP}{TP+FP}$. *Recall*, also called detection rate, represents the ratio of correctly predicted attacks to the overall actual attacks: $\frac{TP}{TP+FN}$. High precision implies a small number of false positives. And high recall suggests fewer false negatives, which also translates to an increased detection rate. Consequently, a performing model is characterized by high precision and high recall. The *F1-score* is not a distinct metric; rather, it combines precision and recall into a single value. It is calculated using: $2 \times \frac{Precision \times Recall}{Precision + Recall}$.

During the initial offline training and validation phases, we compared several machine learning algorithms and noticed that Random Forest (RF) was among the best performers on our training dataset. A result that confirms the findings in [12]. Therefore, we utilize RF for all the evaluation models in this paper.

We evaluate the effectiveness of adding ORPC as a seventh input feature. And we compare the performance with the original model, which uses only six features (listed in Section V) and does not include ORPC. A summary of this offline evaluation is shown in TABLE I. The original model and the improved model scored, respectively, 0.8926 and 0.9504 on accuracy, 0.9131 and 0.9609 on precision, 0.6921 and 0.8626 on recall, 0.7874 and 0.9091 on F1-score. These results show significant improvements in accuracy, detection rate, and false positive rate when using ORPC.

After proving the effectiveness of ORPC in improving results, we utilized it in both S.A. and Collab. models' evaluation. We recall that the only difference between them is that the latter utilizes the vehicle's attack ratio it receives from the previous area as an additional input feature.

Before presenting the offline results of the comparison between Standalone and Collaborative models, we will explain the process of training and testing both models, which is depicted in Fig. 3. The offline evaluation dataset is divided into three main subsets, which are split based on vehicles' IDs. The first subset X is reserved for S.A. training, the second subset Y is dedicated to Collab. training, and the third subset

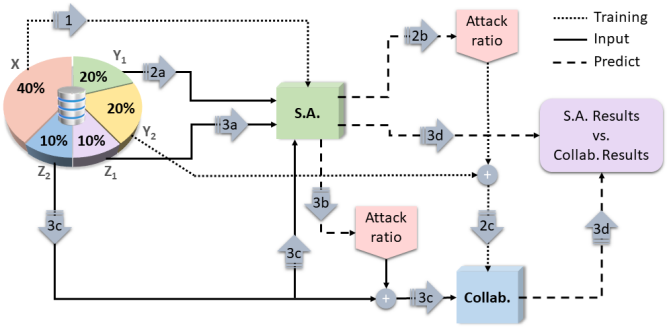

Fig. 3. Offline Training and Evaluation Process

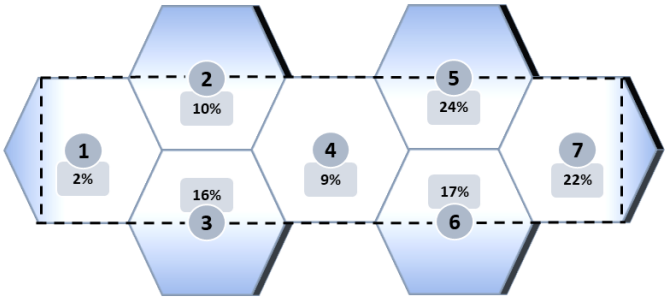

Fig. 4. Online Scenario: 5G service areas and V2X messages distribution

TABLE II
OFFLINE RESULTS: STANDALONE/COLLABORATIVE MODELS

ML Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
S.A.	0.9400	0.9335	0.8518	0.8908
Collab.	0.9721	0.9338	0.9716	0.9523

Z is used to test both models and compare their performances.

First, the data sent by vehicles in subset X is processed to produce the seven input features, which are then utilized to train the S.A. model **(Step 1)**. After S.A. model is trained, the second step is to train the Collaborative model. Subset Y is used for this purpose, and it's divided into two additional subsets: Y_1 and Y_2. Where each vehicle ID has its messages split equally between them. The first half of the sent messages of every vehicle Y_1 are assumed to be moving through Area 1 and are evaluated using S.A. **(Step 2a)**, and the second half of the messages Y_2 are under Area 2 coverage, where the Collab. model is used for predictions. After S.A. performs its evaluation on Y_1, the attack ratio for the vehicle is calculated based on the ratio of the number of predicted attack messages by S.A. to the total number of its predictions **(Step 2b)**. Next, the second half of the vehicles' messages, which are part of subset Y_2, are processed. The attack ratio, calculated in the previous step, is used as an eighth feature, and all eight features are utilized to train the Collab. model **(Step 2c)**.

After training the S.A. and Collab. models, the third step is to utilize the test subset Z to compare the performances of both models. Similar to subset Y, subset Z is divided into two halves. The first half Z_1 is evaluated using the S.A. model only **(Step 3a)**, which will produce the attack ratio of Area 1 **(Step 3b)**. For a fair evaluation, we limit the comparison of the models to subset Z_2 only, where the same data is evaluated using S.A. and Collab. **(Step 3c)**. And finally, the results of the two models are compared **(Step 3d)**.

The results of this process are shown in TABLE II. The Standalone and Collaborative models' scores are, respectively, 0.9400 and 0.9721 on accuracy, 0.9335 and 0.9338 on precision, 0.8518 and 0.9716 on recall, 0.8908 and 0.9523 on F1-score. The better performance of the Collaborative model during the offline evaluation on all metrics is an indicator of the favorable impact of adding the attack ratio feature.

To confirm these findings in a more realistic environment that contains many areas, we performed an online evaluation. We implement an emulation of the 5G core network using free5GC, including eight UPFs for the user-plane. Seven of them serve as I-UPFs, also called branching UPFs, providing

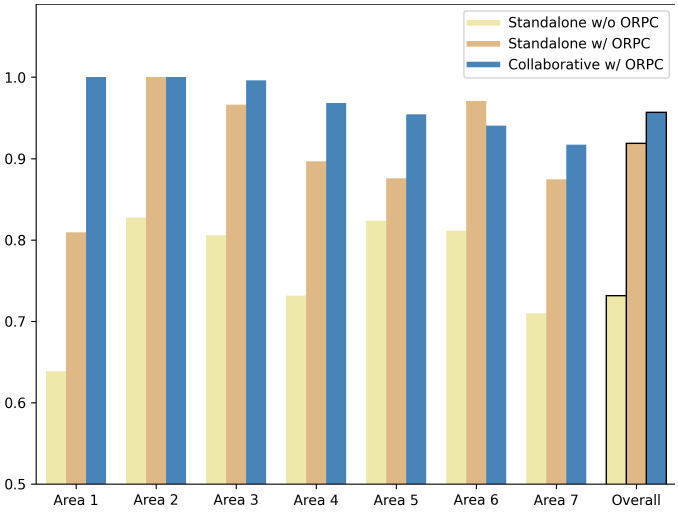

Fig. 5. Online Results: Accuracy of Standalone and Collaborative Models

access to local edge networks, or LADNs. And one PSA-UPF (PDU Session Anchor) providing access to DN. Next to each branching UPF, two application instances are created in the edge network: an instance of V2X AS and another instance for DF.

We use UERANSIM [17] to simulate UEs and the RAN part of the system (gNBs). UERANSIM exchanges control messages with free5GC to authenticate and register 5G base stations and UEs/vehicles. When a vehicle is registered and authenticated, the required user-plane GTP (GPRS Tunneling Protocol) tunnels are created. The vehicle, depending on its actual location, can then communicate with its local V2X AS through the gNB/UPF pair serving that 5G service area.

In the dataset, the range of coordinates sent by vehicles forms a rectangular field. To provide cellular connectivity across the map and maximize the number of transiting vehicles between areas, we divide the map into seven coverage areas, as shown in Fig. 4. Each area is a small cell that consists of a gNB and is served by an I-UPF and local V2X AS and DF instances. Neighboring DF instances are interconnected and therefore can share the attack ratios of vehicles moving between them.

The results of the online testing phase are summarized in Fig. 5 and detailed in TABLE III. Both Standalone and

TABLE III
ONLINE RESULTS: STANDALONE AND COLLABORATIVE MODELS

		Attack 1		Attack 2		Attack 4		Attack 8		Attack 16		All Attacks	
Metric	**Area**	*S.A.*	*Collab.*	*S.A.*	*Collab.*	*S.A.*	*Collab.*	*S.A.*	*Collab.*	*S.A.*	*Collab.*	*S.A.*	*Collab.*
Accuracy	*Area 1*	0.7182	1.0000	0.9945	1.0000	1.0000	1.0000	1.0000	1.0000	0.7143	1.0000	0.8092	1.0000
	Area 2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 3	0.9729	0.9986	0.9533	1.0000	0.9986	1.0000	0.9953	1.0000	0.9729	0.9898	0.9661	0.9962
	Area 4	0.8555	1.0000	0.9782	1.0000	0.9966	1.0000	0.9897	1.0000	0.8555	0.9048	0.8964	0.9683
	Area 5	0.8252	0.9313	0.9869	1.0000	0.9995	1.0000	0.9896	1.0000	0.8256	0.9313	0.8756	0.9542
	Area 6	0.9797	0.9679	0.9545	0.9196	0.9938	0.9601	0.9857	0.9334	0.9739	0.9197	0.9707	0.9405
	Area 7	0.8209	0.8760	0.9924	1.0000	0.9995	1.0000	0.9886	1.0000	0.8203	0.8745	0.8746	0.9169
	Overall	**0.8912**	**0.9501**	**0.9779**	**0.9863**	**0.9982**	**0.9932**	**0.9908**	**0.9886**	**0.8909**	**0.9315**	**0.9187**	**0.9569**
Precision	*Area 1*	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 3	0.9972	1.0000	0.9970	1.0000	0.9973	1.0000	0.9973	1.0000	0.9972	1.0000	0.9994	1.0000
	Area 4	0.9905	1.0000	0.9929	1.0000	0.9932	1.0000	0.9931	1.0000	0.9905	1.0000	0.9984	1.0000
	Area 5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 6	0.9860	0.9365	0.9866	0.9364	0.9877	0.9415	0.9875	0.9383	0.9872	0.9366	0.9974	0.9869
	Area 7	0.9985	1.0000	0.9990	1.0000	0.9991	1.0000	0.9990	1.0000	0.9985	1.0000	0.9998	1.0000
	Overall	**0.9956**	**0.9885**	**0.9965**	**0.9895**	**0.9966**	**0.9896**	**0.9965**	**0.9895**	**0.9957**	**0.9882**	**0.9992**	**0.9978**
Recall	*Area 1*	0.4333	1.0000	0.9890	1.0000	1.0000	1.0000	1.0000	1.0000	0.4286	1.0000	0.7709	1.0000
	Area 2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 3	0.9485	0.9973	0.9093	1.0000	1.0000	1.0000	0.9932	1.0000	0.9485	0.9797	0.9599	0.9954
	Area 4	0.7179	1.0000	0.9633	1.0000	1.0000	1.0000	0.9862	1.0000	0.7179	0.8096	0.8771	0.9619
	Area 5	0.6504	0.8627	0.9738	1.0000	0.9991	1.0000	0.9792	1.0000	0.6510	0.8626	0.8507	0.9450
	Area 6	0.9710	1.0000	0.9214	0.9002	1.0000	0.9813	0.9838	0.9278	0.9602	0.9005	0.9672	0.9408
	Area 7	0.6422	0.7517	0.9858	1.0000	1.0000	1.0000	0.9782	1.0000	0.6410	0.7486	0.8497	0.9002
	Overall	**0.7839**	**0.9100**	**0.9593**	**0.9830**	**0.9998**	**0.9968**	**0.9851**	**0.9876**	**0.7850**	**0.8733**	**0.9031**	**0.9503**
F1-score	*Area 1*	0.6047	1.0000	0.9945	1.0000	1.0000	1.0000	1.0000	1.0000	0.6000	1.0000	0.8706	1.0000
	Area 2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Area 3	0.9722	0.9986	0.9512	1.0000	0.9986	1.0000	0.9952	1.0000	0.9722	0.9897	0.9793	0.9977
	Area 4	0.8324	1.0000	0.9779	1.0000	0.9966	1.0000	0.9896	1.0000	0.8324	0.8948	0.9338	0.9806
	Area 5	0.7882	0.9263	0.9867	1.0000	0.9995	1.0000	0.9895	1.0000	0.7886	0.9262	0.9193	0.9717
	Area 6	0.9784	0.9672	0.9529	0.9180	0.9938	0.9610	0.9857	0.9330	0.9735	0.9182	0.9821	0.9633
	Area 7	0.7817	0.8582	0.9924	1.0000	0.9995	1.0000	0.9885	1.0000	0.7807	0.8562	0.9187	0.9475
	Overall	**0.8772**	**0.9476**	**0.9775**	**0.9862**	**0.9982**	**0.9932**	**0.9908**	**0.9886**	**0.8779**	**0.9272**	**0.9487**	**0.9735**

Collaborative models analyzed in real-time more than 28,000 messages sent by vehicles transiting between two or more neighboring areas.

VII. CONCLUSION

In this paper, we propose a collaborative V2X misbehavior detection system to protect V2X application servers in the 5G edge network. To detect position manipulation attacks, we propose two improved machine learning models. Our work presents a step towards exploring the benefits of using collaboration between edge network nodes to enhance detection results. More studies are needed to explore different methods of collaboration and address more sophisticated V2X attacks.

REFERENCES

[1] K. F. van der Heijden R.W., Lukaseder T., "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," *Chang B., Li Y., Zhu S. (eds) Security and Privacy in Communication Networks. SecureComm 2018*, vol. 254, 2018.

[2] A. Ghosal and M. Conti, "Security issues and challenges in V2X: A Survey," *Computer Networks*, vol. 169, p. 107093, Mar. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1389128619305857

[3] R. Lu, L. Zhang, J. Ni, and Y. Fang, "5G Vehicle-to-Everything Services: Gearing Up for Security and Privacy," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 373–389, Feb. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8897696/

[4] D. Krajzewicz and J. E. M. Behrisch, L. Bieker, "Sumo homepage," http://sumo.sourceforge.net/.

[5] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, January 2011.

[6] L. Codeca, R. Frank, and T. Engel, "LuST: a 24-hour Scenario of Luxembourg City for SUMO," p. 10.

[7] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, "Decentralized position verification in geographic ad hoc routing," vol. 3, no. 4, pp. 289–302.

[8] H. Stübing, J. Firl, and S. A. Huss, "A two-stage verification process for car-to-x mobility data based on path prediction and probabilistic maneuver recognition," in *2011 IEEE Vehicular Networking Conference (VNC)*, 2011, pp. 17–24.

[9] R. K. Schmidt, T. Leinmüller, E. Schoch, A. Held, and G. Schäfer, "Vehicle behavior analysis to enhance security in vanets," 2008.

[10] N. Bißmeyer, J. Njeukam, J. Petit, and K. M. Bayarou, "Central misbehavior evaluation for vanets based on mobility data plausibility," ser. VANET '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 73–82. [Online]. Available: https://doi.org/10.1145/2307888.2307902

[11] S. So, P. Sharma, and J. Petit, "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, FL: IEEE, Dec. 2018, pp. 564–571. [Online]. Available: https://ieeexplore.ieee.org/document/8614116/

[12] P. Sharma and H. Liu, "A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9245568/

[13] S. Gyawali, Y. Qian, and R. Q. Hu, "Deep reinforcement learning based dynamic reputation policy in 5g based vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6136–6146, 2021.

[14] R. Akbani, T. Korkmaz, and G. V. S. Raju, "A machine learning based reputation system for defending against malicious node behavior," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[16] free5GC Project, https://www.free5gc.org/.

[17] UERANSIM, https://github.com/aligungr/UERANSIM.