Reinforcement Learning Based Misbehavior Detection in Vehicular Networks

Roshan Sedar*, Charalampos Kalalas*, Francisco Vázquez-Gallego†, Jesus Alonso-Zarate†
*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain
†i2CAT Foundation, Barcelona, Spain
{roshan.sedar, ckalalas}@cttc.es, {francisco.vazquez, jesus.alonso}@i2cat.net

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838796

***Abstract*—Vehicle-to-everything (V2X) communication is contributing towards the realization of futuristic vehicular networks such as Internet-of-Vehicles (IoV). The IoV is expected to usher in a new direction of intelligence and networking to achieve the goal of intelligent transport systems, which rely on the secure exchange of messages between vehicles and infrastructure. However, the transmission of false/incorrect data by malicious vehicles may cause serious damages on road safety. Therefore, it is crucial to detect safety-threatening incorrect information and mitigate potentially detrimental effects on road users. In this paper, we propose a reinforcement learning (RL)-based misbehavior detection approach for V2X scenarios. In our method, the RL-based detection model processes V2X data broadcast by vehicles as time-series at the roadside units, and classifies incoming data as misbehaving or genuine. We evaluate the proposed RL-based approach for detection of various attack types using an open-source dataset, and compare its performance against recent work in misbehavior detection. Our scheme is able to detect all types of misbehavior with a superior recall of 0.9970 and an F1 score of 0.9845, yielding a significant improvement over the benchmarks. Our research outcomes further reveal that misbehaving vehicles can be detected with a great accuracy of 0.9882 by exploiting real-time V2X information.**

***Index Terms*—V2X, IoV, Misbehavior Detection, Reinforcement Learning.**

I. Introduction

Pervasive vehicle-to-everything (V2X) technology is paving the way for the Internet-of-Vehicles (IoV), a concept which has recently emerged from the Internet-of-Things. The IoV is essentially composed of vehicles, a communication network, and a service platform for supporting a variety of applications, including intelligent transport systems (ITS) applications and Internet services [1]. In such systems, vehicles send a large amount of data that include various parameters such as position coordinates, heading angle and speed of the vehicle, message timestamp, and the real/pseudo-identity of the vehicle, among others. This information is received at the infrastructure nodes, e.g., roadside units (RSUs) and edge/cloud servers, and can be utilized in driver assistance and safety applications for improving road safety and traffic efficiency as well as for cyber threat analysis.

Despite the multitude of benefits offered by V2X, the peculiar characteristics of V2X systems, in conjunction with the increased number of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication links due to the rapid growth of connected vehicles, introduce new security concerns and multifarious security attacks that have not been addressed in a similar context before [2]. Security threats and attacks in V2X systems can be originated from malicious outsiders and/or insiders. In contrast to an outsider, an insider possesses valid credentials to interact with other legitimate entities in the system. Malicious actions from rogue insiders, referred to as misbehaviors in V2X, are often difficult to detect and contain; they can inflict severe damage to road safety when transmitting false/incorrect data in safety-critical situations. Although several security schemes for V2X systems were proposed in recent literature [3]–[5], those cryptography-based techniques fall short in mitigating threats of rogue insiders, e.g., false data injection attacks.

In existing literature, several data-driven approaches have been proposed for misbehavior detection [6], and many of them rely on conventional machine learning (ML) methodologies. However, existing techniques are inadequate to adapt to evolving attack patterns in rapidly changing V2X environments. Also, the use of security thresholds in detectors (e.g., anomaly score-based methods) limits their applicability only to specific V2X scenarios. To this end, reinforcement learning (RL) can be identified as a highly effective approach that can consistently improve detection experience over time, while interacting with unknown environments without relying on security threshold values [7]. Although there have been several works related to ML-based misbehavior detection [8]–[11], to the best of our knowledge, RL-based misbehavior detection has not been studied before in this context.

In this paper, we adapt an RL approach into time-series misbehavior detection in V2X scenarios, and propose a generic RL-based misbehavior detector. We demonstrate through extended experiments using an open-source dataset that our proposed detection scheme is highly effective in detecting various types of attacks, e.g., position falsification, sudden-stop, DoS, Sybil, etc., that can exist in V2X scenarios. The comparison of our experimental outcomes with recent prior work in misbehavior detection shows that the proposed RL-based detector yields superior performance against benchmark schemes, while accurately detecting all types of misbehavior attacks.

II. Background and Related Work

A vehicle is considered as misbehaving if it transmits incorrect/erroneous data when both hardware and software are operating as expected. Such abnormal behaviors may

occur due to either malfunctioning vehicle components or malicious actions by other users. Maliciously misbehaving vehicles tend to deviate from the expected behavior and transmit intentionally falsified information to mislead other V2X entities. Such misbehaviors often result in sophisticated attacks when attackers behave intelligently while conforming to normal system behavior.

The access to V2X datasets from real-world field studies can be difficult due to various policies and regulations, e.g., data protection laws and confidentiality agreements. Recently, a simulation-based V2X dataset was introduced in [10] as an evaluation baseline for the comparison of different misbehavior detection techniques. The dataset, called Vehicular Reference Misbehavior (VeReMi), is extensible and contains a number of misbehavior attacks. For attack detection, the authors of [10] have used a set of plausibility and consistency checks, e.g., confidence range on the parameters of exchanged messages, as well as ML-based techniques such as multi-layer perceptron and long short-term memory (LSTM) to classify genuine vehicles from misbehaving ones.

Several research works are available in the literature based on VeReMi dataset. The work in [8] integrates location and movement plausibility checks with ML algorithms, i.e., support vector machine (SVM) and K-nearest neighbor (KNN), to identify a set of position falsification attacks. Similarly, the authors of [11] apply supervised learning techniques (e.g., SVM, KNN, random forest) for detecting position falsification attacks, while KNN, decision trees and logistic regression are applied in [9] to quickly detect vehicles that transmit false alerts and falsified position information. In a similar line, the work in [12] applies conventional ML classification to detect falsified location information. However, the aforementioned approaches make use of traditional ML-based classification techniques which are restricted to a pre-trained set of attack labels, that limits their ability to adapt detection over time for possible new attack variants. Also, the analyses are mostly limited to a few types of misbehavior attacks.

A deep neural network (DNN) framework for anomaly detection has been recently proposed in [13]. By projecting high-dimensional V2X information into a lower-dimensional latent representation, the authors aim to differentiate data of genuine vehicles from misbehaving ones. The performance of six different DNN models is evaluated using VeReMi to demonstrate the effectiveness of their approach against several attacks. Nevertheless, the scheme relies on a threshold value to find the optimal accuracy for the classification.

III. V2X Network Model

In this section, we present the V2X scenario with the security attack model under consideration, by highlighting a set of attack types that can be highly effective against V2X networks.

A. Scenario

Fig. 1 illustrates a V2X network deployment where the involved vehicles are broadcasting Basic Safety Messages

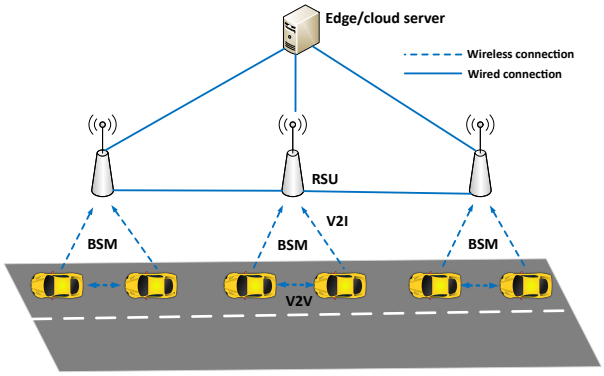

Fig. 1: Considered V2X network model

(BSMs). BSMs contain standard parameters such as the position, speed, acceleration and heading angle of the vehicle, and other relevant information. An RSU receives the messages sent from vehicles within its communication range, and the edge/cloud server aggregates information from RSUs deployed in a large geographical region.

B. Attack Model

In what follows, we briefly discuss the misbehavior attacks, originally defined in [10], which are relevant to our scenario. In the considered V2X scenario, an attacker is a misbehaving vehicle that transmits falsified information embedded in the BSMs. It is worth noting that a combination of multiple attacks may occur at once:

Position falsification attack: A vehicle transmits falsified position coordinates. This could lead to four different attack variants: i) *constant position*, the attacker transmits fixed position coordinates; ii) *constant position offset*, the attacker transmits the real position coordinates with a fixed offset; iii) *random position*, the attacker transmits newly generated random position coordinates; and iv) *random position offset*, the attacker transmits the real position coordinates with a random offset.

Speed falsification attack: A vehicle transmits falsified speed values in its BSM, following a similar approach as in position falsification attack. This results in *constant speed, constant speed offset, random speed* and *random speed offset* attack variants.

Sudden-stop attack: A vehicle may behave normally for a certain time-period and then start sending falsified information, i.e., fixed-position and zero-speed values, in subsequent time-steps.

Data replay attack: A vehicle re-transmits or replays valid BSMs previously received from other vehicles. In this case, the vehicle uses its own identity while replaying the data, and tries to exploit the conditions that existed at the time of the original BSM transmission. The attack could also be carried out in *Sybil* mode by changing the attacker's identity to avoid detection. The attacker uses multiple valid pseudonym certificates of compromised vehicles to realize an attack in the *Sybil* mode.

Delayed messages attack: A vehicle may transmit BSMs containing all relevant fields with correct data, but with a delay shift from real-time.

Denial-of-service (DoS) attack: A vehicle transmits BSMs at a frequency higher than the limit set by the standard. Such high volume of data transmission would result in extensive periods of network congestion and unavailability to serve other legitimate vehicles. DoS attacks may also be launched by setting all BSM fields to random values, i.e., *DoS random* attacks. A DoS random attack can be carried out in *Sybil* mode by changing pseudonym on every BSM.

Traffic congestion Sybil attack: In this attack, a fake road traffic congestion is generated with a grid of ghost vehicles in a selected geographical region. Valid pseudonymous identities and BSM frequency may be used for every ghost vehicle.

Disruptive attack: The behavior of this attack is similar to a data replay attack, where a vehicle re-transmits previously transmitted messages by other vehicles. BSMs are selected at random and flood the network with stale data to disrupt genuine information from being propagated. This attack may also be carried out in *DoS* and *Sybil* modes.

IV. Misbehavior Detection Approach

In this section, we present the proposed RL-based approach for misbehavior detection in the considered V2X scenario.

A. Reinforcement Learning for Misbehavior Detection

Time-series anomaly detection involves sequential decision-making and can be modelled as a Markov decision process (MDP). The action of anomaly detection will change the environment based on the decision of either normal or anomalous behavior at time-step t; subsequently, the next decision at time-step $t+1$ will be influenced by the changing environment at the previous time-step t. Thus, the application of an RL model becomes a natural fit for time-series anomaly detection [14].

In the context of V2X, the aggregated information at each RSU (Fig. 1) constitutes a time-series repository of received BSMs with intrinsic temporal and spatial interdependencies. The information contained in each BSM is constantly evolving over time along the vehicle trajectory, while BSMs from neighboring vehicles exhibit high spatial dependency. Hence, misbehaving vehicles can be potentially detected by sequentially analyzing their mobility patterns using an RL model.

B. Reinforcement Learning Model

We consider an RL-based misbehavior detector deployed in an RSU, where it acts as an agent that interacts with the V2X environment to learn the optimal detection policy. Based on the current state s_t at time-step t, the agent takes an action a_t to maximize its reward r_t. The reward is offered to the agent by the environment, and the environment subsequently moves to a new state s_{t+1} following the MDP. This is done repeatedly until the optimal detection policy π is learned. In this work, model-free and value-based Q-learning method is adopted to train the RL model for estimating the action-value function $Q(s,a)$ [15]. The optimal detection policy can thus be obtained using the Q-value. Moreover, the ϵ-greedy technique is utilized in Q-learning to balance the strategy between exploration and exploitation.

We consider the case where the training of the RL model is performed on the edge/cloud server, assuming it has superior computational resources over RSUs, whereas detection is performed at an RSU level. The BSMs, upon received at the RSU, are processed as time-series data and are fed to the pre-trained RL model for classification. The detection is performed at the RSU since the vehicle may not have the complete information in its communication range during a short period of time. It is also assumed that RSUs are considered trusted infrastructure nodes.

C. Reinforcement Learning Model Parameters

In what follows, we describe the parameters pertaining to the RL model used in our methodology.

The **agent** takes the V2X time-series data and prior related decisions as inputs (i.e., state s_t), and generates the new decision made (i.e., action a_t) as output. The agent's actions at each time-step t are selected by the detection policy π. Thus, the agent's experience at each time-step, i.e., $e_t =< s_t, a_t, r_t, s_{t+1} >$, stores all the behaviors of the misbehavior detector. By learning from experience, the misbehavior detector is consistently improved to obtain a better estimation of the $Q(s,a)$ function. This process is referred to as experience replay memory, through which the model training is performed. The goal of the agent is to maximize the expected sum of future discounted rewards by learning the optimal detection policy. The discounted reward return is expressed as $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, where γ denotes the discount factor that specifies the importance of long-term rewards and T is the terminal step. The agent updates its model in order to improve the accuracy in decision-making. The Q-value in the model can be updated iteratively with learning rate α and discount factor γ as expressed by the following formula

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)). \quad (1)$$

Fig. 2 illustrates the Q-network for learning the $Q(s,a)$ function. The agent's Q-network is composed of an LSTM layer, which is a type of RNN model that can effectively capture long-term temporal dependencies in the input data. The LSTM layer is used to extract the sequential information from the input time-series, i.e., state s_t. The output of LSTM is then fed into a fully-connected neural network, which generates two Q-values, i.e., $Q(s_t, a_t = 0)$ and $Q(s_t, a_t = 1)$, as choices for the action a_t. Of these choices, the final output of the agent's action is then decided for the current state s_t, i.e., $a_t = 0$ or $a_t = 1$.

The **environment** of the RL model controls the training of the agent. It takes the action a_t performed by the agent as its input, and consequently generates a reward r_t and the next environment state s_{t+1} for the agent. In this setting,

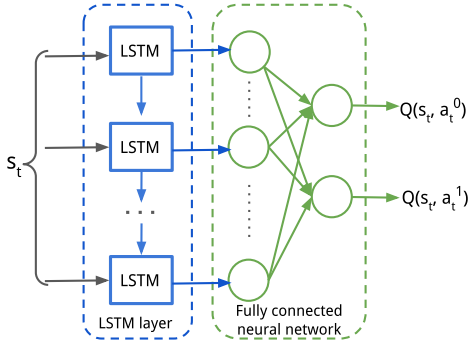

Fig. 2: Q-network of the agent

the environment contains a large population of BSMs with misbehavior attacks labels.

The **state** contains the sequence of previous actions denoted by $s_{action} =< a_{t-1}, a_t, ..., a_{t+n-1} >$, and the current BSM information denoted by $s_{time} =< X_t, X_{t+1}, ..., X_{t+n} >$. $X_t \in R^d$ is a d-dimensional feature vector at time-step t, including information on d different features. According to the state design, the next action taken by the agent depends on the previous actions and the current V2X information.

The **action** space is defined as $\mathcal{A} = \{0,1\}$, where 1 indicates the detection of an attack and 0 represents the genuine behavior. The deterministic detection policy π can be expressed as a mapping, i.e., $\pi : \mathcal{S} \longmapsto \mathcal{A}$, from states to actions, where $\pi(s)$ denotes the action that the agent takes at state s. In a given state s_t, the agent selects the action based on the optimal detection policy given by

$$\pi^* = \arg\max_{a \in \mathcal{A}} Q(s, a). \tag{2}$$

The **reward** r_t helps the agent to learn an effective detection policy, and it is offered as feedback (i.e., positive/negative) for an action a_t taken in state s_t. The reward r_t for an action a_t under state s_t is computed based on the ground truth values of BSMs. Concretely, a positive reward is given to the agent for correctly detecting an attack, i.e., true positive (TP), or a normal state, i.e., true negative (TN); otherwise, a negative reward is given to the agent for incorrect identification of a normal state as an attack, i.e., false positive (FP), or an attack as a normal state, i.e., false negative (FN). In safety-critical V2X scenarios, the correct identification of misbehavior is vital in order to mitigate potential hazardous situations. Therefore, the agent is penalized more for FN actions than for FPs. The reward function can be expressed by

$$r(s_t, a_t) = \begin{cases} A & \text{if the action is a TP,} \\ B & \text{if the action is a TN,} \\ -C & \text{if the action is an FP,} \\ -D & \text{if the action is an FN,} \end{cases} \tag{3}$$

where $A, B, C, D > 0$, with $A > B$ and $D > C$.

V. Experiments and Performance Evaluation

In this section, we evaluate the effectiveness of the RL-based approach applied for V2X misbehavior detection by performing experiments using the VeReMi dataset.

A. Dataset Description and Pre-processing

The VeReMi dataset [10] includes 19 misbehavior attack types as described in section III-B. The dataset models two road traffic densities under each attack scenario: high-density (37.03 vehicles/km^2) and low-density (16.36 vehicles/km^2). For each attack scenario, a log file per vehicle is generated, which contains BSM data transmitted by neighboring vehicles over its entire trajectory. Each scenario contains a ground truth file to record the observed behavior of all participating vehicles. BSMs constitute a three-dimensional vector[1] for position, speed, acceleration and heading angle features. The proportion between misbehaving and genuine vehicles is set to 30% to 70%, respectively, for all the simulations in [10].

Fig. 3 depicts a raw sample of BSM data which contains six selected fields, i.e., timestamp, pseudo-identity, position, speed, acceleration and heading angle. Based on feature analysis and data pre-processing, we selected these six fields as the most relevant feature set related to attack detection. In particular, we use the Euclidean norm of the position, speed, acceleration and heading vectors. Furthermore, the ground truth data do not contain labels; therefore, a label for each data point was generated by comparing the ground truth value against the actual transmitted value recorded in the log file of each vehicle. An attack label 1 was added if the transmitted data diverges from the ground truth; otherwise, 0 was added for the genuine data.

sendTime	PseudoID	pos	spd	acl	hed
2413.315025	102072	[861.5896564521556, 723.8492038508903, 0.0]	[11.016630549605496, -9.454566846922763, 0.0]	[0.190557869967025, -0.16227766035749402, 0.0]	[0.74415546691166, -0.6680064678320771, 0.0]
2413.506941	102132	[153.28767426827244, 900.7165052363796, 0.0]	[6.831779729000001e-06, 6.831779729000001e-06, 0.0]	[2.582766878e-06, 2.582766878e-06, 0.0]	[-0.067946182302661, -0.9976889877664751, 0.0]
2413.756941	102132	[153.28767426827244, 900.7165052363796, 0.0]	[6.831779729000001e-06, 6.831779729000001e-06, 0.0]	[2.582766878e-06, 2.582766878e-06, 0.0]	[-0.067946182302661, -0.9976889877664751, 0.0]
2414.006941	102132	[153.33397441474833, 900.7130309086808, 0.0]	[6.831774301e-06, 6.831774301e-06, 0.0]	[2.582766878e-06, 2.582766878e-06, 0.0]	[-0.067946182302661, -0.9976889877664751, 0.0]
2414.256941	102132	[153.33397441474833, 900.7130309086808, 0.0]	[6.831774301e-06, 6.831774301e-06, 0.0]	[2.582766878e-06, 2.582766878e-06, 0.0]	[-0.067946182302661, -0.9976889877664751, 0.0]
2414.315025	102072	[872.0232881512283, 713.7651577970362, 0.0]	[10.416278774860185, -10.085278581022612, 0.0]	[-0.48815459315556003, 0.47401496049310604, 0.0]	[0.702749880995823, -0.7114369998533641, 0.0]
2414.506941	102132	[153.33397441474833, 900.7130309086808, 0.0]	[6.831774301e-06, 6.831774301e-06, 0.0]	[2.582766878e-06, 2.582766878e-06, 0.0]	[-0.067946182302661, -0.9976889877664751, 0.0]

Fig. 3: A raw sample of a log file with BSM data

In our experiments, the high-density dataset was used to train the RL model under each attack scenario; thus, the RL model is allowed to detect and learn attack patterns more frequently. On the other hand, the low-density dataset was used to test the ability of the RL model in detecting attacks when attack patterns are less frequent in the dataset.

B. Baseline Schemes and Performance Metrics

Although several research works [8], [9], [11], [12] are available in recent literature, most of the proposed schemes have been evaluated only with a few attack types, e.g., position falsification. We therefore compare our RL-based detection results with two recent works presented in [10] and [13],

[1]It is noted that z-dimension entries are zero-valued for all features.

wherein they extend the detection to all attack types in VeReMi dataset. In particular, the work presented in [13] demonstrates overall superior performance against the proposed schemes in the original VeReMi paper [10]. We deem that the detection of all attack types is important in the case of assessing a practical application of misbehavior detection in V2X scenarios. Therefore, the comparison is performed for all attacks scenario considered in [10], [13]. To achieve this, a single dataset was created by merging all data from each attack type, resulting in 19 attacks with labelled data points.

The detection performance of the RL-based approach has been evaluated based on commonly used metrics, i.e., *Accuracy*, *Precision*, *Recall* and *F1 score*, which are defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{6}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{7}$$

The accuracy indicates the ratio of all correct predictions to the total number of considered input samples. Higher precision values indicate low FP rates, and higher recall values indicate low FN rates. F1 score is the harmonic mean between precision and recall metrics, and it is used when FPs and FNs are vital. Thus, a higher F1 score indicates a better detection performance in our experiments.

C. Single Attack Type Detection

In this section, we discuss the performance of our approach based on the obtained results for each attack type detection, highlighting the outcomes which give the best and moderate performances.

Table I depicts the results of the RL-based detection per attack type (in total, 19 attack types). Results show that for all types of attacks, RL-based detection is performed effectively with over 0.98 of recall, demonstrating the ability to detect misbehaving vehicles accurately. In particular, we can notice that the attack types 5, 7, 13 to 16, 18 and 19 yield the best detection performance with over 0.99 of F1 score, resulting in a high precision and high recall at the same time with over 0.99. This manifests that the RL-based approach is able to markedly distinguish misbehavior from the genuine behavior, with very low number of false alarms. Also, the attack types 16 and 18 are detected with 1.0 of F1 score, with zero false alarms. We further notice that the attack types 13 to 16, 18 and 19 generate an increased number of BSMs due to their behavior; thus, the high proportion of attack data availability at the training helps the RL model to learn the attack patterns more frequently. Additional features such as timestamp and pseudo-identity are used for the detection of such high-frequency and high-volume traffic attacks, contributing to a better detection performance.

Results further reveal that for attack types 9 and 12, RL-based detection performs moderately with 0.9274 and 0.9012 of precision values, respectively. Attack type 9 is the sudden-stop attack, where the attacker behaves normally for a certain time-period, and then stops based on a pre-defined probability. However, there is no certainty that the attacking vehicle will stop. In addition, the erratic behavior over time between attack and genuine misleads the RL model to incorrectly identify genuine state as attack, resulting in increased FP rates. Attack type 12 is the delayed message attack, where the messages contain genuine vehicles' information, but are transmitted with a delay shift from real-time. Due to attack data being identical to genuine data, the RL model is tricked by delayed message attack, resulting in increased FPs.

TABLE I: Detection performance per attack type over the test dataset

Type	Attack	Accuracy	Precision	Recall	F1
1	Constant Position	0.9868	0.9588	0.9984	0.9782
2	Constant Position Offset	0.9981	0.9629	0.9982	0.9803
3	Random Position	0.9886	0.9642	0.9985	0.9810
4	Random Position Offset	0.9886	0.9632	1.0000	0.9812
5	Constant Speed	0.9988	0.9968	0.9995	0.9982
6	Constant Speed Offset	0.9923	0.9766	0.9978	0.9871
7	Random Speed	0.9985	0.9987	0.9963	0.9975
8	Random Speed Offset	0.9915	0.9774	0.9945	0.9858
9	Sudden Stop	0.9811	0.9274	1.0000	0.9623
10	Disruptive	0.9896	0.9664	1.0000	0.9829
11	Data Replay	0.9894	0.9656	0.9999	0.9825
12	Delayed Messages	0.9666	0.9012	0.9976	0.9470
13	DoS	0.9999	0.9999	1.0000	0.9999
14	DoS Random	0.9997	0.9996	1.0000	0.9998
15	DoS Disruptive	0.9991	0.9984	1.0000	0.9992
16	Traffic Congestion Sybil	1.0000	1.0000	1.0000	1.0000
17	Data Replay Sybil	0.9938	0.9981	0.9809	0.9894
18	DoS Random Sybil	1.0000	1.0000	1.0000	1.0000
19	DoS Disruptive Sybil	0.9972	0.9998	0.9940	0.9970

D. All Attacks Detection

Table II shows the overall detection performance of the RL-based approach for all attacks scenario in comparison to VeReMi [10] and DeepADV [13]. The best achieved performance values are marked in bold. Results show that for overall detection of all attacks, the RL-based approach achieves superior performance with respect to accuracy, recall and F1 metrics. In particular, recall of 0.9970 demonstrates that the RL model is able to accurately detect all misbehavior types present in the dataset, which in fact is essential for safety-critical V2X scenarios. The higher F1 score of 0.9845 further indicates the better detection performance of the RL-based approach. However, we can notice the lower precision value of 0.9724 compared to [10], [13]. As mentioned before, the characteristics of some attacks, e.g., type 9 and 12, tend to resemble the genuine behavior; thus, this misleads the RL model to trigger FPs over some genuine states. Also, the RL model is penalized less for FPs than for FNs, which allows

tolerating FPs as long as they are not excessive. Overall, the RL-based approach is able to detect all types of misbehavior with a greater accuracy of 0.9882.

TABLE II: Performance comparison for all attacks scenario

Detection technique	Accuracy	Precision	Recall	F1
RL-based approach	**0.9882**	0.9724	**0.9970**	**0.9845**
DeepADV [13]	0.9800	**0.9960**	0.9560	0.9760
VeReMi [10]	0.9293	0.9912	0.8228	0.8992

E. Analysing Detection Model Performance

Fig. 4 demonstrates the performance of the RL model using the cumulative sum of all rewards obtained over the number of training episodes under all attacks scenario. An episode represents the sequence of agent-environment interactions between initial and terminal states. At each interaction, the agent is offered a reward for the action taken. For rewards, the values of $5, 1, 1, 5$ are used for $A, B, -C, -D$, respectively. We can notice that the RL model is consistently improved while detecting and learning.

As can be observed from Fig. 4, the agent sacrifices its rewards at the beginning. The accumulation of negative rewards in early episodes is due to the false alarms when the agent takes random actions. However, from episode 273 onwards, the agent improves its rewards by learning the optimal detection policy, resulting in higher and stable accumulated rewards. The fluctuations of accumulated rewards after 273 episodes indicate that the agent learns various attack patterns whilst repeatedly recouping its cost of learning. The RL environment consists of BSM time-series with varying lengths of different attack types, which contribute to oscillations in the trends of accumulated rewards. Thus, such iterative process of detecting and learning allows the RL-based detector to adapt to rapidly changing V2X environments, yielding superior performance in the presence of multifarious attacks.

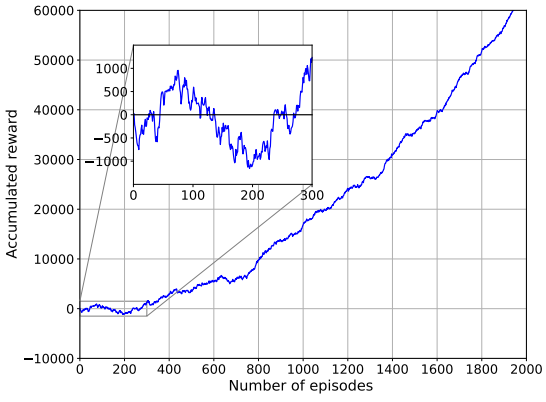

Fig. 4: Cumulative sum of all rewards over training episodes

VI. Conclusion

With the advent of V2X technology, there is a sharp increase in road traffic data from the rapid growth of connected vehicles in intelligent transport systems. This gives rise to an increased number of V2X links, resulting in entirely new and multifarious security attacks in vehicular networks. In this paper, we have proposed a novel RL-based misbehavior detection approach and evaluated its effectiveness based on a widely used open-source dataset. Our evaluation yields superior performance for accuracy, recall and F1 metrics compared to benchmark schemes. We demonstrate that the RL-based approach is capable of improving its detection experience over time, and adapt to changing V2X environments. In the path forward, we aim to incorporate RL-based trust of infrastructure nodes into a collaborative misbehavior detection framework.

Acknowledgment

This work is partly supported by the H2020-INSPIRE-5Gplus project (under grant agreement No. 871808), by the Spanish MINECO project SPOT5G (TEC2017-87456-P), and by the Generalitat de Catalunya under Grant 2017 SGR 891.

References

[1] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C. Lin, and X. Liu, "Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[2] A. Ghosal and M. Conti, "Security issues and challenges in V2X: A Survey," *Computer Networks*, vol. 169, p. 107093, 2020.

[3] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A Security Credential Management System for V2X Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[4] J. Cui, X. Zhang, H. Zhong, Z. Ying, and L. Liu, "RSMA: Reputation System-Based Lightweight Message Authentication Framework and Protocol for 5G-Enabled Vehicular Networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6417–6428, 2019.

[5] B. Ying and A. Nayak, "Anonymous and Lightweight Authentication for Secure Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 626–10 636, 2017.

[6] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 779–811, 2019.

[7] R. Sedar, C. Kalalas, F. Vázquez-Gallego, and J. Alonso-Zarate, "Reinforcement Learning-based Misbehaviour Detection in V2X Scenarios," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 109–111.

[8] 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*, 2018, pp. 564–571.

[9] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *2019 IEEE International Conference on Communications*, 2019, pp. 1–6.

[10] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets," in *2020 IEEE International Conference on Communications*, 2020, pp. 1–6.

[11] 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, 2021.

[12] F. Hawlader, A. Boualouache, S. Faye, and T. Engel, "Intelligent misbehavior detection system for detecting false position attacks in vehicular networks," in *2021 IEEE International Conference on Communications Workshops*, 2021, pp. 1–6.

[13] T. Alladi, B. Gera, A. Agrawal, V. Chamola, and R. Yu, "DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021.

[14] C. Huang, Y. Wu, Y. Zuo, K. Pei, and G. Min, "Towards experienced anomaly detector through reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018.

[15] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.