Edge Assisted Misbehavior Detection for Platoons

Xinyue Kan[1], Akila Ganlath[1], Seyhan Ucar[2], Kyungtae Han[2], Prashant Tiwari[2], and Konstantinos Karydis[1]
Dept. of Electrical and Computer Engineering, University of California-Riverside, Riverside, CA, USA[1]
InfoTech Labs, Toyota Motor North America R&D, Mountain View, CA, USA[2]
{xkan001, aganl001, karydis}@ucr.edu, {seyhan.ucar, kyungtae.han, prashant.tiwari}@toyota.com

***Abstract*—Platoons are vehicle formations that allow better traffic management and safety. Although many efforts have been devoted to implement platooning, ensuring stability in the presence of misbehavior remains a challenge. The detection of the misbehavior is important, otherwise it may put the platoon as well as other vehicles in danger. In this paper, we propose an edge assisted misbehavior detection system for vehicular platoons where speed/acceleration information of platoon and observation from other connected entities are leveraged to detect the misbehaving vehicle(s). Extensive simulation in different settings shows that our proposed method can improve the misbehavior detection rate compared to existing baseline method.**

I. INTRODUCTION

Advances in the automotive industry are propelling the development of vehicles into higher levels of autonomy. This is evidenced by the availability of Level 2 Automation (Society of Automotive Engineers) technologies, such as Adaptive Cruise Control (ACC), in recent consumer vehicles. ACC increases fuel efficiency, driver comfort, and safety over human drivers by automatically adjusting the safe following distance and speed based on sensor feedback [1].

C-ACC is an extension of ACC which leverages vehicle-to-vehicle communication to make decisions *cooperatively*. This communication includes sharing the operating state of vehicles, such as speed, acceleration as well as distance from surrounding vehicles. C-ACC enables vehicles to be organized into close proximity formations called *vehicular platoons* [2]. Platooning allows for more efficient transportation as the cooperation among vehicles enhances their ability to plan ahead and drive closer than normal vehicles with small speed and distance variation [3].

One key objective of the vehicular platoon is *platoon stability*. A platoon is said to be stable if platoon followers follow the leader with minimal speed variation over time [4]. To ensure platoon stability, several methods [5]–[7] have been proposed with the assumption that trustworthy communication exists among vehicles. However, it has been demonstrated that malicious attacks on communication can severely degrade the performance of C-ACC and the stability of platoons [8], [9]. The Vehicular Ad Hoc Network (VANET) community have proposed security standards (i.e., IEEE 1609.2), which provide mechanisms against common external attacks and ensure message integrity and authenticity. In contrast, the case where the adversary is a *trusted insider*, such as compromised platoon members, are not addressed by the standards. Recently, a data-driven abnormal behaviour detection method is proposed for safeguarding the vehicular platoons in [10]. However, the proposed method suffers from artificial anomalies due to its uncooperative detection scheme where each vehicle runs the misbehavior detection algorithm independently.

On the other hand, edge computing is one of the techniques proposed to detect misbehaving vehicles [11], [12]. The edge layer performs trust model construction to detect the misbehaving vehicle as well as the trustworthy vehicles in [11]. [12] leverages traffic surveillance videos recorded by Drones to detect abnormal activities by identifying the vehicles and their trajectories. However, these approaches are not directly applicable to platoon and have the following drawbacks. First, trust value is computed based on predefined events in trust-based detection scheme and this scheme may suffer from event sparsity where it is misleading in trust value computation and trustworthy vehicle election. Second, extracting vehicle state information from transmitted video in order to detect anomalies may introduce large delays and overhead which is not tolerable in the time-critical vehicular platoon. Third, the method relies entirely on the availability of the edge layer.

In this paper, we investigate an edge assisted misbehavior detection scheme for vehicular platoon. The proposed method has the following properties: 1) it detects misbehaving vehicle(s) in a platoon in real-time. 2) it performs better at the edge when observations from connected entities (vehicles, RSUs, etc.) are available. 3) it requires neither large training data with prior knowledge, nor definition of normal behavior.

II. PRELIMINARIES

A. System Architecture

Fig. 1 illustrates a high-level overview of the proposed misbehaviour detection system. We distinguish different system components into three layers. At the vehicle layer, there exist both platooned and non-platooned connected vehicles. We assume that each vehicle in the platoon is capable of communicating with other connected entities through the vehicle-to-X (V2X) (e.g., IEEE 802.11p and/or LTE) communication. It can then process and store communicated data received from these entities.

Platoon stability is achieved by the periodical exchange of beacon messages. The beacon message may contain a platoon identifier, lane identifier, sequence number, acceleration, speed, position, and sender address of the transmitter. Upon reception of the beacon message, a platoon follower adjusts

*This work was done when Xinyue Kan and Akila Ganlath were with InfoTech Labs, Toyota Motor North America R&D, USA.

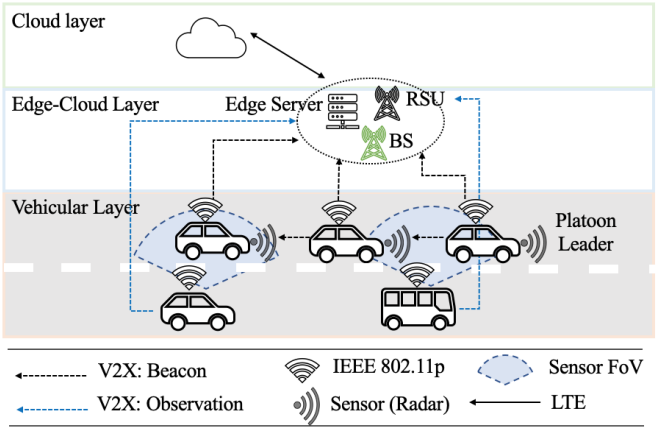

Fig. 1: Edge Assisted Misbehaviour Detection System

its own speed and distance to the preceding vehicle based on the speed and acceleration information of the vehicle itself and its preceding vehicle.

In the edge computing network, Roadside Units (RSUs) and cellular Base Stations (BS) are connected to an edge server over broadband connections, which forms an edge-cloud layer [13], [14]. We assume that the edge layer collects information about the platoon through beacon messages and observations from other connected entities. Beacon messages are broadcasted by platoon members and received by RSUs or BS via V2X. Observations, on the other hand, are generated by other connected entities (non-platooned connected vehicles, cameras and other sensors) and are composed of estimates of speed, acceleration, platoon length, etc.

B. Misbehaving Vehicles in a Platoon

We assume that the misbehaving vehicle (adversary) is a trusted insider, which is either a platoon follower or a leader that affects platoon stability by modifying the beacon messages: upon receiving a beacon message, the misbehaving vehicle alters the content (the acceleration and speed values are modified) and re-broadcasts. The re-broadcasted beacon message contains false information which degrades the platoon stability and jeopardizes the safety of other vehicles as well as the efficiency of the transportation system. Consider a scenario where the adversary modifies the acceleration of platoon from slowing down to speeding up, which may induce a chain of accidents. Though we label the misbehaving vehicle as an adversary, it is important to note that such behavior may arise due to non-malicious phenomenon, such as malfunction or sensor failure.

III. Misbehaviour Detection Method

In this section, we first present our time-series-analysis-based misbehavior detection method which can be applied on either the vehicle individually. Then, we further discuss the combined misbehavior detection with edge layer through the *observations* that are collected via the vehicular edge-computing network.

A. Time Series Analysis based Misbehaviour Detection

Consider V_s and V_r to be two vehicles in a platoon such that V_r trusts V_s and reacts based on information received from V_s. $\mathcal{X}_s$ and $\mathcal{X}_r$ represent time series formed by information related to V_s and V_r, respectively. Algorithm 1 runs with the input of vehicle's speed or acceleration values to detect the misbehavior(s).

Algorithm 1: Misbehaviour Detection

1 **Initialize:** $k \leftarrow 0$, $k^* \leftarrow 0$, empty time series $\mathcal{X}_s$, $\mathcal{X}_r$;
2 Start to collect time series data $\mathcal{X}_s$, $\mathcal{X}_r$ at time τ_0 ;
3 **while** $|\mathcal{X}_s| < \tau$ *or* $|\mathcal{X}_r| < \tau$ **do**
4 Add collected data to $\mathcal{X}_s$, $\mathcal{X}_r$;
5 **repeat**
6 **if** $\forall\ k' \in [0, min(|\mathcal{X}_s|, |\mathcal{X}_r|)/2] \cap [k-\epsilon, k+\epsilon] \iff L(\mathcal{X}_s, \mathcal{X}_r, k) \leq L(\mathcal{X}_s, \mathcal{X}_r, k')$ **then** $k^* \leftarrow k$;
7 **else** $k \leftarrow k+1$;
8 **until** $k \geq min(|\mathcal{X}_s|, |\mathcal{X}_r|)/2$ *or* $k^* > 0$;
9 $\mathcal{X} \leftarrow \mathcal{X}_s(k^*, \tau - k^*) - \mathcal{X}_r(\tau_0, \tau - k^*)$;
10 Apply moving median filter to $\mathcal{X}$, and get $\bar{\mathcal{X}}$;
11 Compute residual $\mathcal{R}_x = \mathcal{X} - \bar{\mathcal{X}}$;
12 $\mathcal{X}_{out} \leftarrow ESD(\mathcal{R}_x, N_m)$;
13 Return $\mathcal{X}_{out}$;

Let $\mathcal{A}_s(\tau_0, \tau)$, $\mathcal{A}_r(\tau_0, \tau)$ be the ordered list of acceleration collected in a window of length τ for V_s and V_r, respectively. $\mathcal{A}_r[t]$ and $\mathcal{A}_s[t]$ are the acceleration of the V_s and V_r at the time t, where $t \in [\tau_0, \tau_0 + \tau]$. Algorithm 1 runs on $\mathcal{A}_s$, $\mathcal{A}_r$ as input $\mathcal{X}_s$, $\mathcal{X}_r$ to detect the misbehaviour(s). The algorithm starts by collecting time series data (Line 1-4). In normal cases, the shape of $\mathcal{A}_r$ follows $\mathcal{A}_s$ smoothly with a time delay. With the occurrence of misbehavior(s), on the other hand, the relationship between the two time series data may become abnormal. The basic idea of our method is to identify the misbehavior by exploring the difference between $\mathcal{A}_s$ and $\mathcal{A}_r$. However, $\mathcal{A}_s$ and $\mathcal{A}_r$ may contain different number of elements due to 1) various data generation frequency among vehicles, and 2) packet loss caused by communication. These causes of anomaly fall out of the scope of this paper. On the other hand, $\mathcal{A}_s[t]$ and $\mathcal{A}_r[t]$ can be significantly different because of reaction and communication delays. Therefore, we pre-process the data to eliminate this difference (Line 5-10).

Let k represent the shifting length, such that $\mathcal{X}_s(k, \tau - k)$ and $\mathcal{X}_r(\tau_0, \tau - k)$ are sub-sequences of $\mathcal{X}_s$ and $\mathcal{X}_r$. Function $\mathcal{L}$ returns the *distance* of two sub-sequence as follows,

$$\mathcal{L}(\mathcal{X}_s, \mathcal{X}_r, k) = \mathcal{D}(\mathcal{X}_s(k, \tau - k), \mathcal{X}_r(\tau_0, \tau - k)) \ ,$$

$$\mathcal{D}(\mathcal{X}_s, \mathcal{X}_r) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(\mathcal{X}_s[i] - \mathcal{X}_r[i])^2} \ ,$$

where $|\mathcal{X}_s(k, \tau - k)| = |\mathcal{X}_r(\tau_0, \tau - k)| = n$. $\mathcal{D}(\cdot, \cdot)$ is the normalized Euclidean distance between two time series. We want to select a $k^* \in [0, \min(|\mathcal{X}_s|, |\mathcal{X}_r|)/2]$ such that when comparing the differences between two time series, the first local minima is reached at shifting size k^*.

With the selected k^*, two sub-sequences are generated through shifting from the original times series data. Then, we can obtain the new time series $\mathcal{X}$, which is the point-to-point difference between the two sub-sequences (Line 10). We compute the trend component, $\bar{\mathcal{X}}$, by applying a moving

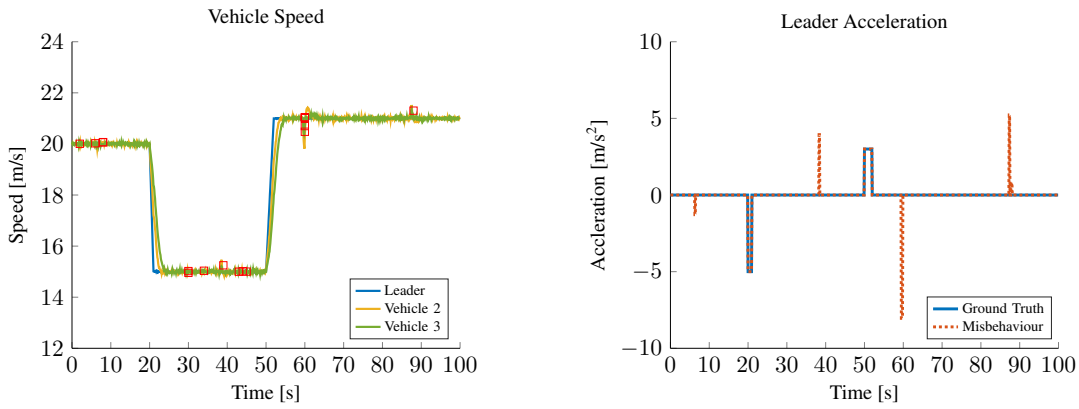

Fig. 2: Speed Profile of Platoon Followers Under Misbehaviour (left), Misbehaving Vehicle Characteristics (right)

median filter to $\mathcal{X}$, then subtract the trend component from $\mathcal{X}$ to get the residual $\mathcal{R}_x$ (Line 11-12). Following that, the Generalized Extreme Studentized Deviate (ESD) [15] is run on $\mathcal{R}_x$ to detect misbehaviors (Line 13). ESD takes the parameters the residual and the upper bound of on the number of potential misbehaviour, N_m. After ESD execution, misbehaviour(s) is detected and added to $\mathcal{X}_{out}$. $\mathcal{X}_{out}$ consists of the simulation time and the abnormal values from the misbehaviour.

B. Edge Assisted Misbehavior Detection

Algorithm 1 can be run by platoon members individually to detect the misbehaviour. However, it has been shown that running misbehavior detection individually can be insufficient. On the other hand, the observation of platoon members from other connected entities at the edge layer may improve the misbehavior detection.

On the edge server, *observations* related to platoon are collected through other connected entities. We assume that connected vehicles have on-board sensors (e.g., camera, radar, and sonar, etc.). On-board sensors can measure odometric parameters such as distance to the preceding vehicle (with an additive white Gaussian noise on the observed parameters) [16]. It is assumed that the edge server is capable of associating the beacon messages with speed observations. For vehicles V_s and V_r, observed speeds form two time series $\mathcal{O}_s(\tau_0, \tau)$ and $\mathcal{O}_r(\tau_0, \tau)$ for time window τ. Then, Algorithm 1 is run with inputs $\mathcal{O}_s(\tau_0, \tau)$ and $\mathcal{O}_r(\tau_0, \tau)$ on the edge server.

We balance the misbehaviour detection results from individual vehicles, $\hat{\mathcal{A}}(\tau_0, \tau)$, and from *observations*, $\hat{\mathcal{O}}(\tau_0, \tau)$, using a weighted voting strategy. If a misbehavior is detected by an individual vehicle at time $t \in [\tau_0, \tau_0 + \tau]$, $\hat{\mathcal{A}}[t] = 1$, otherwise, $\hat{\mathcal{A}}[t] = 0$. $\hat{\mathcal{O}}$ follows the same scheme. Consider $\hat{\mathcal{A}}[t]$ and $\hat{\mathcal{O}}[t]$ as two voters for $t \in [\tau_0, \tau_0 + \tau]$, the voting scheme can be described as $[\mathcal{Q} : w_{\hat{\mathcal{A}}}, w_{\hat{\mathcal{O}}}]$. $\mathcal{Q}$ is the quota of votes, which indicates the number of votes needed to determine the existence of the misbehaviour at t. $w_{\hat{\mathcal{A}}}$ and $w_{\hat{\mathcal{O}}}$ are weights for $\hat{\mathcal{A}}$ and $\hat{\mathcal{O}}$, which represents the number of votes, respectively. The value of $w_{\hat{\mathcal{A}}}$ and $w_{\hat{\mathcal{O}}}$ can be based on the current vehicular network condition, including sensor quality of service, network connection, etc. For instance, we can set $w_{\hat{\mathcal{A}}}$ and $w_{\hat{\mathcal{O}}}$ to 1 to equally weight both sides. On the other hand, $\hat{\mathcal{O}}$ may get higher votes if the vehicle in a platoon is observed by multiple connected vehicles as compared to observations by only one connected vehicle.

IV. Performance Evaluation

We have conducted a simulation study to investigate the feasibility of proposed misbehaviour detection methods. The goal of the simulations is to compare the performance of proposed individual and edge assisted version of misbehavior detection schemes, denoted by *Individual* and *Edge-Assisted*, to the previously proposed baseline misbehaviour detection method, denoted by *ESD* [15].

A. Simulation Setup

The acceleration and speed observations are collected via simulation in VEhicular NeTwork Open Simulator (VENTOS) [17]. VENTOS is a simulator integrating realistic mobility generator, Simulation of Urban Mobility (SUMO) [18]; the packet level simulator, OMNET++ [19] and vehicular communication platform Vehicles in Network Simulation (Veins) [20]. Time series analysis and misbehavior detection schemes are implemented in Python and integrated with VENTOS. The simulated topology consists of 20 connected vehicles (3 of them are platoon enabled) and vehicles are injected into the road according to the Poisson process at a rate of 0.5 vehicles per second rate. A platoon consists of 3 vehicles and *Vehi* refers to *i-th* vehicle in the platoon, with *Veh1* as the *Leader*. The vehicles possess two communication interfaces: IEEE 802.11p and LTE. In the simulation, the leader, *Veh1* is misbehaving where it manipulates the acceleration fields of beacon message. Figure 2 demonstrates the speed profile of platoon followers under misbehaving vehicle and it shows the misbehavior characteristics.

B. Results and Analysis

We conduct 6 case studies by considering various window sizes ($\tau \in \{0.8, 1, 2, 5, 10, 50\}$ seconds). The system alerts the occurrence of a misbehaviour if $\hat{\mathcal{A}}[t] = 1$ and $\hat{\mathcal{O}}[t] = 1$ at any time.

The results of detection rate is averaged over 100 Monte Carlo trials, each of which contains different misbehaviour characteristics in terms of occurrence time and duration. Results are given in Table I where two cases $\tau = 1$ s and $\tau = 2$ s are shown in Fig. 3. The simulation results suggest

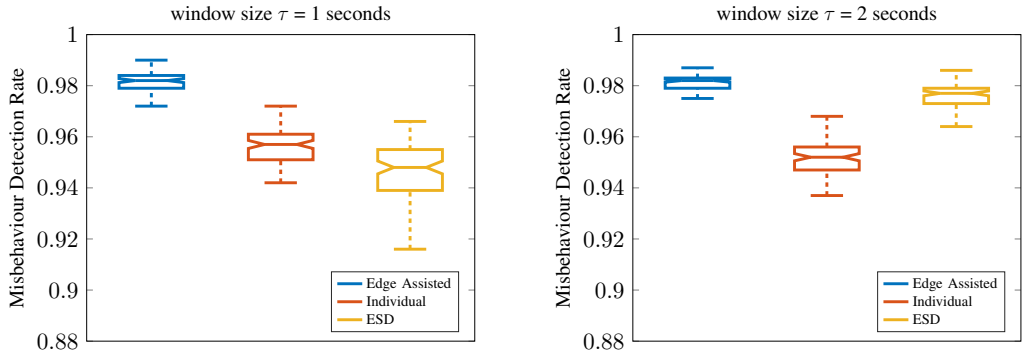

Fig. 3: Misbehaviour Detection Rate for $\tau = 1$ second and $\tau = 2$ seconds

that the proposed *Edge Assisted* method can detect the occurrence of altered beacon messages from the misbehaviour with a success rate near 98% and small standard deviation. For each selected window length, the *Edge Assisted* method outperforms the *Individual* and baseline approach *ESD*. This results reveal the benefit of cooperative misbehaviour detection at the edge layer. The *Edge Assisted* utilizes not only the values from platoon members but also the observation from other connected entities which improves the detection rate of altered beacon messages.

TABLE I: Misbehaviour Detection in Different τ

	Edge Assisted		Individual		ESD	
τ(s)	mean	std	mean	std	mean	std
0.8	0.980	0.003	0.980	0.004	0.967	0.005
1	0.981	0.004	0.957	0.005	0.947	0.011
2	0.981	0.003	0.952	0.007	0.977	0.004
5	0.977	0.004	0.931	0.012	0.975	0.004
10	0.974	0.006	0.932	0.012	0.961	0.004
50	0.970	0.005	0.966	0.007	0.961	0.004

Moreover, *Edge Assisted* is more robust than *ESD* to changes of window length where its accuracy range is between $[0.970, 0.981]$ for $\tau \in [0.8, 50]$ seconds, whereas the accuracy range is $[0.947, 0.977]$ for *ESD*. The performance with small window length enables fast response to misbehaviour especially in time-critical vehicular platoon.

V. Conclusion and Future Work

We investigated an edge assisted misbehavior detection method to identify the misbehaving vehicle(s) and its altered beacon messages in a platoon. The simulation results for different settings show that the proposed method outperforms the existing methods, and achieves the detection rate near 98%. Besides, with a window length of 0.8 second, our method achieves near-real-time detection of vehicle misbehavior without large training dataset and prior knowledge. The fast detection is beneficial in terms of revocation mechanism where the rogue vehicles are removed from platoon membership.

Future work would concentrate on testing the proposed method in an experimental setting to study the impact of sensors noise and network speed. We aim to focus on designing an edge assisted platoon revocation mechanism which removes the misbehavior from platoon membership. Moreover, we plan to evaluate the performance of misbehavior detection in platoon maneuvers, such as entrance and exit, in which platoon is more vulnerable than usual.

References

[1] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Trans. on Intell. Transp. Syst.*, vol. 4, no. 3, pp. 143–153, 2003.
[2] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful Degradation of Cooperative Adaptive Cruise Control," *IEEE Trans. on Intel. Transp. Syst.*, Feb 2015.
[3] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *ITS World Congress*, 2012.
[4] Y. Zheng, S. Eben Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Trans. on Intell. Transp. Syst.*, vol. 17, no. 1, pp. 14–26, Jan 2016.
[5] S. Santini, A. Salvi, A. S. Valente, A. Pescap, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with inter-vehicular communications," in *IEEE Conf. on Comput. Comm.*, 2015.
[6] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is Your Commute Driving You Crazy?: A Study of Misbehavior in Vehicular Platoons," in *ACM Conf. on Security & Privacy in Wireless and Mobile Netw.*, ser. WiSec '15, 2015.
[7] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Veh. Comm.*, 2015.
[8] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control," in *IEEE Vehicular Networking Conference*, 2017.
[9] S. Ucar, S. C. Ergen, and O. Ozkasap, "IEEE 802.11p and Visible Light Hybrid Communication Based Secure Autonomous Platoon," *IEEE Transactions on Vehicular Technology*, Sep. 2018.
[10] S. Ucar, S. Ergen, and O. Ozkasap, "Data-driven Abnormal Behavior Detection for Autonomous Platoon," in *IEEE Vehicular Networking Conference*, 2017.
[11] S. Yang, Z. Liu, J. Li, S. Wang, and F. Yang, "Anomaly detection for internet of vehicles: A trust management scheme with affinity propagation," *Mobile Information Syst.*, vol. 2016, pp. 1–10, 03 2016.
[12] N. Chen, Z. Yang, Y. Chen, and A. Polunchenko, "Online anomalous vehicle detection at the edge using multidimensional SSA," in *IEEE Conf. on Comput. Comm. Workshops*, May 2017, pp. 851–856.
[13] S. Raza, S. Wang, M. Ahmed, and M. Anwar, "A Survey on Vehicular Edge Computing: Architecture, Applications, Technical Issues, and Future Directions," *Wireless Communication and Mobile Computing*, 2019.
[14] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular Ad-hoc Network with Fog Computing," in *IFIP/IEEE Int. Symp. on Integrated Netw. Management*, 2015.
[15] B. Rosner, "Percentage points for a generalized ESD many-outlier procedure," *Technometrics*, 1983.
[16] R. Guo, B. Ayinde, H. Sun, H. Muralidharan, and K. Oguchi, "Monocular Depth Estimation Using Synthetic Images With Shadow Removal," in *Intell. Transport. Syst. Conf.* IEEE, 2019.
[17] "VENTOS," http://goo.gl/OueFkO.
[18] "SUMO," http://sumo.sourceforge.net/.
[19] "OMNET++ Networ Simulator," https://omnetpp.org/.
[20] "Veins," http://veins.car2x.org/.