Location Anomalies Detection for Connected and Autonomous Vehicles

Xiaoyang Wang*, Ioannis Mavromatis*, Andrea Tassi*, Raúl Santos-Rodríguez† and Robert J. Piechocki*‡

* Department of Electrical and Electronic Engineering, University of Bristol, UK

† Department of Engineering Mathematics, University of Bristol, UK

‡The Alan Turing Institute, London, NW1 2DB, UK

Email: {xiaoyang.wang, ioan.mavromatis, a.tassi, enrsr, r.j.piechocki}@bristol.ac.uk

Abstract—Future Connected and Automated Vehicles (CAV), and more generally ITS, will form a highly interconnected system. Such a paradigm is referred to as the Internet of Vehicles (herein Internet of CAVs) and is a prerequisite to orchestrate traffic flows in cities. For optimal decision making and supervision, traffic centres will have access to suitably anonymized CAV mobility information. Safe and secure operations will then be contingent on early detection of anomalies. In this paper, a novel unsupervised learning model based on deep autoencoder is proposed to detect the self-reported location anomaly in CAVs, using vehicle locations and the Received Signal Strength Indicator (RSSI) as features. Quantitative experiments on simulation datasets show that the proposed approach is effective and robust in detecting self-reported location anomalies.

Keywords—anomaly detection, deep neural network, autoencoder, connected and autonomous vehicles (CAV), Intelligent Transportation Systems (ITS)

I. Introduction

The past decades have witnessed rapid developments in Intelligent Transportation Systems (ITS). Connected and Autonomous Vehicles (CAV) are an integral part of ITS and will redefine mobility, change the existing vehicle usage and pave the way for future transportation services. Vehicles collect the necessary information via a number of onboard sensors. This information is later disseminated to the surrounding environment in a Vehicle-to-Everything (V2X) fashion, negotiating manoeuvres and building a more agile, safe and efficient traffic network [1].

Without constant human supervision, the safety of CAVs heavily relies on the knowledge acquired from the connected surrounding environment [2]. However, this dependence of exchanged data brings into the surface several security risks and potential malicious cyberattack. The connected nature of the vehicles increases the risk of compromised vehicles on the road, and thus the demand for more sophisticated anomaly detection and cybersecurity protection techniques. In this work, we will focus on CAVs that maliciously exchange their falsified self-reported locations to the surrounding vehicles, presenting a novel way of detecting and counteracting on the abnormalities.

Anomaly detection is the identification of abnormal observations that do not conform to the expected behaviour. Ultimately, the goal is to present a quick and reliable alert when an anomaly occurs, helping the system to respond accordingly. Particularly for CAVs, it plays a crucial role in system malfunction detection, intelligent operations and cybersecurity protection. Growing efforts have been put into this area during the past years. For example, [3], [4] introduced intrusion detection methods for CAVs using deep learning and discriminant analysis. Additionally, [5] proposed a CAV misbehaviour detection method for service management.

In this paper, we develop a deep autoencoder approach for anomaly detection in CAVs. As an unsupervised method, autoencoders are capable of finding the latent patterns of data. It is generally accepted that most of the research activities in this area are carried out using synthetic datasets, generated by means of simulation frameworks [6]. Similarly, in our work, we will train and validate our deep autoencoder model using anomalous-free data, generated using OMNeT++ network simulator [7]. Later on, we introduce different abnormalities on our test dataset in order to evaluate the validity of our model.

The rest of this paper is organized as follows. Section II presents our problem description and our deep autoencoder approach to detect anomalies. Section III gives more insights about the generation of our synthetic dataset, the tools used for that and the details about our scenario. Section IV presents the results and our analysis. Finally, Section V concludes the paper summarizing our findings.

II. Methodology

A. Problem Statement

We consider a system where CAVs exchange beacons on a periodic basis. Also, we assume that anomalies are presented within the information encapsulated in the beacon frames, and more specifically at the self-reported CAV locations. The falsified information is then received by other surrounding CAVs or infrastructure network. Fig. 1 gives a brief summary of the anomaly scenario considered in this paper. The fake position reported from the "ghost" transmitter T' could be due to sensor malfunction in the actual transmitter T, or a possible information hacking during the transmission. The detection of self-reported location anomaly could further help with the root cause analysis and decision making in CAVs.

We aim to design a self-learning process for detecting the self-reported location anomalies. To do so, the Received
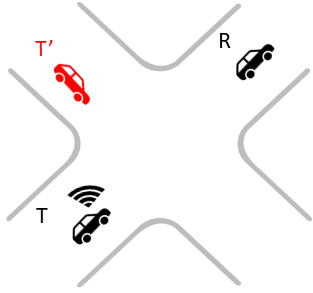
Fig. 1: Self-reported location anomaly in CAVs. Here T is the transmitter in its real location, while T' is "ghost" transmitter. R, being the receiver, receives the beacon from T with the faulty location of T'.

Fig. 2: The seven-layer, fully-connected deep autoencoder structure designed in this paper.

Signal Strength Indicator (RSSI) is chosen as a proxy for the distance separation between two CAVs. The RSSI represents the beacon's signal strength and it is dependent on the maximum broadcasting power, the antenna gains, and the attenuation from the channel and the distance. RSSI has been widely used for indoor localization [8]–[10], human activity recognition [11] and movement tracking [12] in wireless networks. For CAVs, when a packet is received, the RSSI along with the transmitter-reported locations and the receiver self-location could form a strong state description for the self-reported location anomaly detection.

For our system, we will consider the RSSI between the different pairs of CAVs exchanging beacons in a Vehicle-to-Vehicle (V2V) fashion. As shown in Fig. 1, T and R are the real transmitter-receiver pair, while T' is a "ghost" transmitter. Locations of T, R, and T' are represented as l_T, l_R and $l_{T'}$, respectively. The distance between each pair of T, R and T' are $D_{T,R}$, $D_{T,T'}$ and $D_{T',R}$, respectively.

B. The Deep Autoencoder Approach

We design a Deep Autoencoder (DAE) for CAV self-reported location anomaly detection in an unsupervised manner. Autoencoders are a special type of artificial network, encoding high-dimensional data into a latent space by replicating the input in the output [13]. The idea for training a DAE for anomaly detection is to feed anomaly-free data into the network so it can learn the anomaly-free manifold and the corresponding latent space. Once the model has learned the anomaly-free manifold for a specific task, the error between the DAE input and output would be a strong indicator for recognizing anomalous samples.

Here we use a seven-layer, fully-connected autoencoder structure, as shown in Fig. 2. Note that the first hidden layer H_1 has more neurons than the input layer L_1 in our model. L_1 can be seen as a data interpolation layer, which helps the model to learn the proper latent space. Symmetrically, the output layer L_2 transfers the interpolated data into its original dimension. Hidden layers H_1 to H_5 build the standard DAE structure, with a latent space $L \in \mathbb{R}^{1 \times 20}$. The activation function is discarded at the bottleneck H_3 and the output layer
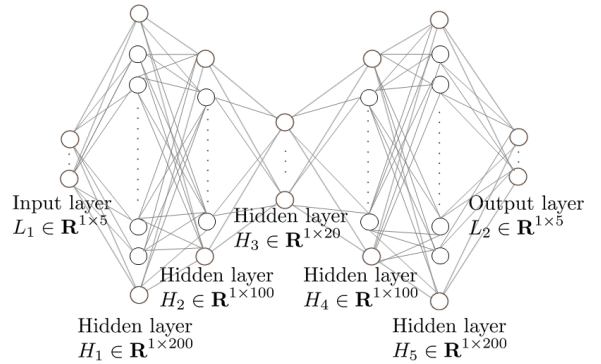
L_2. Dropout is not used in our model.

1) Training: As mentioned in Sec. II-A, the location of the receiver $l_R \in \mathbb{R}^{1 \times 2}$, the self-reported transmitter location $l_T \in \mathbb{R}^{1 \times 2}$ (when there is no anomaly during transmission) or $l_{T'} \in \mathbb{R}^{1 \times 2}$ (when anomaly happens), along with the RSSI value V_{RSSI} form the feature set for self-reported location anomaly detection in CAVs. During training, anomaly-free samples $X = [l_R, V_{\text{RSSI}}, l_T] \in \mathbb{R}^{1 \times 5}$ are fed to the DAE. The model is trained by minimizing the loss function L_t:

$$\min_{\Theta} L_t(\Theta) = ||X - X'||_2 \qquad (1)$$

Here X' is the output of the DAE, and Θ represents the parameters. The goal is to train the DAE generating X' close to X. Gradient descent optimization is used for training this model, with the learning rate of 0.00095.

2) Validation: To validate the performance of the trained DAE, the anomaly-free data is split into the training set and validation set with the proportion of 0.8 and 0.2, following the same data distribution. During validation, samples from the validation set are fed into the trained model. The validation loss is calculated as:

$$L_v = ||X_v - X'_v||_2 \qquad (2)$$

Since DAE is an unsupervised method, here we introduce the adjusted mutual information (AMI) score [14] to evaluate the relation between L_t and L_v. It gives an unbiased evaluation of our trained model. The mean value and variance of L_t and L_v are also calculated. Results are shown in Table I. Though the mean values and variances of two loss distributions are slightly different, they can be considered as nearly identical in the context of AMI.

TABLE I: Model evaluation by testing

	AMI	Mean value	Variance
Training loss	1.0	4.11×10^{-5}	8.13×10^{-9}
Testing loss		5.37×10^{-5}	5.36×10^{-9}

3) Anomaly Detection: After the DAE model is well-tuned and validated, it can be applied for anomaly detection. Potential anomalous samples Y go through the DAE, with an

output Y'. The difference between Y and Y' can then be used for anomaly detection, while L_t and L_v serve as references.

III. DATA GENERATION

For this particular work, we assume that each vehicle generates one beacon per second. Each beacon is encapsulated in a UDP packet with a total length 140 B. Each UPD packet is broadcast in the network to the surrounding vehicles. We choose a $2\,\text{km} \times 2\,\text{km}$ area in central Bristol, UK as our simulation scenario. The number of vehicles within our system is constant (always 150 vehicles). To simulate our scenario, we used OMNeT++ [7] and our modified INET framework [15]. The vehicles mobility traces were generated using SUMO traffic generator [16] and parsed within our framework. Our INET framework was further modified with a logging interface that logs all the packets generated, transmitted and received, in a space-separated file format. These traces will be later used for our anomaly detection algorithm.

In particular, at the transmitter (TX) side, we find at first the wireless interface ID (e.g., *ScenarioWorking.node[1].wlan[0].radio*), followed by the node ID (e.g., *"1"* for the TX example). For this work, we assumed that all the vehicles are equipped with one IEEE 802.11p transceiver, operating at the frequency band of $5.9\,\text{GHz}$. The next entry is the packet ID, i.e. *"UDPData-50 1027"*, used to reconcile the transmitted with the received packets. *UDPData-50* is the data structure called *Signal* within INET, that represents the physical phenomena of transmitting a packet. The number following the signal is the sequence number of the event generated in INET. *Start* fields represent the timestamp that the UPD packet started being transmitted (in seconds), followed by the position of the vehicle in space (given in meters). SUMO, when parses a real-world map, converts all the geolocation coordinates into a Cartesian plane with the southern-west map corner being the origin of the Euclidean space. Similarly, *End* shows the timestamp that the transmission was over, followed by the position of the vehicle on that particular time.

The logging of each packet at the RX side follows a similar structure. Again we find the wireless interface ID followed by the node ID, the packet ID and the starting and ending timestamp of the reception as well as the positions of the RX vehicle. On top of that, our logging interface saves the RSSI of all the received packets (as it is being calculated within the INET framework). In order to calculate the RSSI for each packet, we take into account the building layout and the position of the vehicles. A scalar radio medium was chosen for our configuration, meaning that the analog signal power is represented with a scalar value over frequency and time. As a path loss model, we chose Rician Fading with a path loss exponent $\alpha = 2.4$ and a Rician K-factor equal to $k = 8\,\text{dB}$. Finally, the obstacle loss model chosen was the *DielectricObstacleLoss*. This model calculates the dielectric and reflection loss along the straight path considering the shape, the position, the orientation, and the material of obstructing physical objects. The rest of our simulation parameters can be

TABLE II: Simulation Parameters.

Parameter	Value	
Simulation Time	1800	s
Carrier Frequency	5.9	GHz
Bandwidth	10	MHz
Path Loss Model	Rician Fading	
Path Loss Exponent α	2.4	
Rician K-factor k	8	dBm
UDP Packet Length	140	B
UDP TX Interval	1	s
TX Power	27	dBm
TX/RX Antenna Gain	9	dBi
TX/RX Modulation	QPSK	
RX Sensitivity S_{th}	−88	dBm
SNIR Threshold $SNIR_{\text{th}}$	10	dB
Background Noise	$\mathcal{N}(-110, 3)$	dBm
Connector and Cable Losses	3	dBm

found in Table II. Also, the reasoning for choosing this setup was derived from the performance investigation in [17].

A UDP packet is considered as deliverable under the following conditions. At first, the RSSI is compared with the sensitivity threshold S_{th} for the chosen Modulation and Coding Scheme (MCS). When the RSSI is lower than S_{th}, it is considered as non-deliverable. If the RSSI is above the S_{th}, then it is compared with the Signal-to-Noise-plus-Interference (SNIR) threshold $SNIR_{\text{th}}$. When below that, the packet is always considered as non-decodable due to errors introduced from the channel. The last case is when the RSSI is greater than the $SNIR_{\text{th}}$. For that, a Packet Error Rate (PER) value is calculated based on the current SNIR. This value is later compared with a random number chosen from a uniform random distribution and if it is greater, the packet is considered as delivered. All the successfully received packets are logged in the dataset. These RX entries can then be reconciled with the transmitted ones using the combination of the values found at the packet ID (signal and sequence number), as mentioned above.

IV. RESULTS AND DISCUSSIONS

To evaluate the proposed algorithm, an information hacking process has been imitated. More specifically, we consider a number of CAVs as hacked and force them to report random wrong locations. In this process, the inaccessible areas for CAVs, like buildings, grass-areas and rivers are avoided to increase the reasonableness of samples. Anomalous samples have the format of $[l_R, V_{RSSI}, l_{T'}]$, while normal samples are $[l_R, V_{RSSI}, l_T]$ without hacking. Here $l_{T'}$ is CAV-accessible locations in our simulation area. Anomaly detection is carried out for each received packet.

To conduct quantitative evaluation, anomalous samples are classified into 8 datasets by $D_{T,T'}$. Table III shows the details of each anomaly dataset, with the corresponding $D_{T,T'}$ range and mean values. Besides, the one-class Support Vector Machine (SVM) method has been introduced for comparison [18]. One-class SVM is an unsupervised learning approach for novelty and outlier detection. By estimating the support of a high-dimensional distribution, the unknown data can be

classified as inliers or outliers. Here the one-class SVM can be seen as a benchmark for the location anomaly detection problem.

TABLE III: Anomaly datasets (AD) (distance unit:m)

	Number of samples	$D_{T,T'}$	$\overline{D_{T,T'}}$
AD1	1000	[0,10)	7.16
AD2	1000	[10,20)	15.77
AD3	1000	[20,30)	27.30
AD4	1000	[30,40)	35.70
AD5	1000	[40,50)	46.43
AD6	1000	[50,100)	78.94
AD7	1000	[100,500)	347.42
AD8	1000	≥ 500	711.12

A. Results

The performance of the anomaly detection model proposed in this paper is depicted with the Receiver Operating Characteristic (ROC) curve, along with the area under curve (AUC) value. ROC curve plots the correspondence between the False Positive Rate (FPR) and the True Positive Rate (TPR) over all possible classification thresholds. The data generated in Sec. III serves as an anomaly-free baseline.

Fig. 3 shows the detection results for AD1 to AD8, with AUC values in the right bottom of each subfigure. The diagonal black dash lines show where FPR = TPR. If the ROC curve falls along the diagonal line, the result is evaluated as no better than the random classification, where the AUC would be 0.5. From Fig. 3(a) we see the scenario where the faulty location $l_{T'}$ is close to the real location l_T, i.e., $D_{T,T'} < 10\,\mathrm{m}$. For that case, the chance to detect an anomaly almost equals to a random chance for both SVM and DAE. Under this situation, RSSI is not enough for self-reported location anomaly detection. When $D_{T,T'}$ reaches $20\,\mathrm{m}$, the detection rate is clearly boosted for DAE. When $D_{T,T'}$ is larger than $30\,\mathrm{m}$, there is a very good chance to detect anomalies with low FPR using DAE method. When the ghost vehicle T' is $100\,\mathrm{m}$ away from T, the detection rate for DAE is nearly 100%, clearly better than SVM.

To further investigate how the anomaly detection performance changes with $D_{T,T'}$, we calculated the detection rate when FPR = 0.2. Fig. 4 shows the detection rate when T' is located in the different radius ranges of T. We can also see that when $D_{T,T'} < 30\,\mathrm{m}$, it is difficult to detect self-reported location anomalies with the current model and features.

B. Transmission Direction Anomaly Analysis

According to the results in Sec. IV-A, when $D_{T,T'} > 30\,\mathrm{m}$, there is a very good chance to successfully detect abnormalities. In this section, we investigate another factor that could influence the detection performance. Consider a situation where $D_{T,T'} > k$, here k is the detectable threshold. In the meantime, $|D_{T',R} - D_{T,R}| < \epsilon$, where ϵ is a small distance. Under such situation, the distance between the 'ghost' transmitter T' and the real transmitter T is noticeable, while their distances to the receiver R are very close. It can be seen as a location anomaly, which mainly changes the data transmission direction

with the same transmission distance. We would like to test the performance of our model in such a situation.

We generated two additional anomaly datasets, AD9 and AD10, to study the transmission direction anomaly. In AD9, $D_{T,T'} > 30\,\mathrm{m}$ where $|D_{T,R} - D_{T',R}| < 1\,\mathrm{m}$. The wrong locations of T' in AD9 imitate CAVs reporting wrong transmission directions. In AD10, we still have $D_{T,T'} > 30\,\mathrm{m}$ while $|D_{T,R} - D_{T',R}| \in [10, 20]\mathrm{m}$, for the sake of comparison.

Detection results on AD9 and AD10 are shown in Fig. 5. In Fig. 5(a), even though $D_{T,T'}$ has reached 30m, the detection performance is not satisfying comparing to Fig. 3(d) to Fig. 3(h). The detection rate of DAE is overall lower than the results in AD4 to AD8, with lower AUC value. The performance of SVM in AD9 is close to a random classification. In Fig. 5(b), the detection performance improved as $|D_{T,R} - D_{T',R}|$ increases. It can be seen as evidence that the trained DAE model mainly learns the correspondence between the transmitter/receiver distance $D_{T,R}$ and the RSSI. Directional information is not learned because of the lack of relevant features. Data from multiple sources/sensors could be further considered for different types of anomalies.

V. CONCLUSION

In this study, we investigate the self-reported location anomaly detection problem in CAVs. To train a self-learning approach for anomaly detection, a deep autoencoder is designed. The proposed model is trained and tested on simulation data generated by OMNeT++, our modified INET framework and SUMO traffic generator, in an area of Bristol, UK.

Illustrative results about the self-reported location anomaly detection performance have been presented by only using the transmitter/receiver location pairs and RSSI values as features. We show that the DAE approach proposed in this paper is capable of detecting CAV location anomaly in a complicated scenario. We also present an insightful analysis of the detectable range of the proposed model, as well as the detection performance on transmission direction anomaly.

The proposed model connects unsupervised learning and anomaly detection in CAVs. It indicates that the powerful properties of unsupervised learning can bring benefits to future ITSs. The DAE approach proposed in this paper can be applied as a pre-processing step to the control centre of CAVs, verifying the clearness and reliability of the transmitted data before taking it to the CAV decision-making process.

REFERENCES

[1] S. E. Shladover, "Connected and automated vehicle systems: Introduction and overview," *Journal of Intelligent Transportation Systems*, vol. 22, no. 3, pp. 190–200, Jun. 2018.
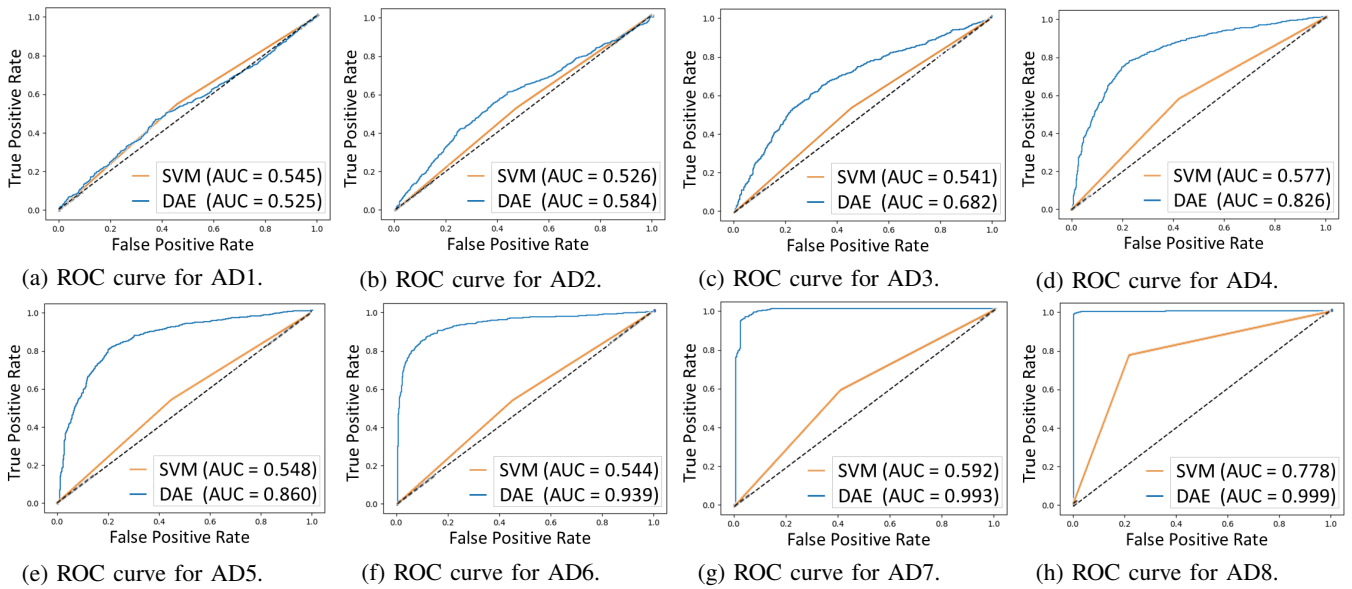
(a) ROC curve for AD1. (b) ROC curve for AD2. (c) ROC curve for AD3. (d) ROC curve for AD4.

(e) ROC curve for AD5. (f) ROC curve for AD6. (g) ROC curve for AD7. (h) ROC curve for AD8.

Fig. 3: Anomaly detection results in AD1 to AD8.
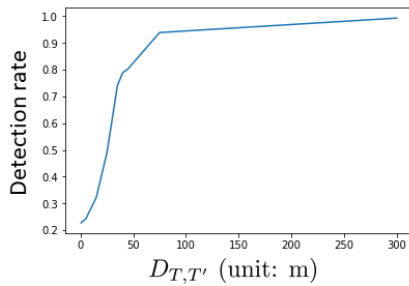
Fig. 4: Anomaly detection rate under different distances $D_{T,T'}$, with FPR $= 0.2$.

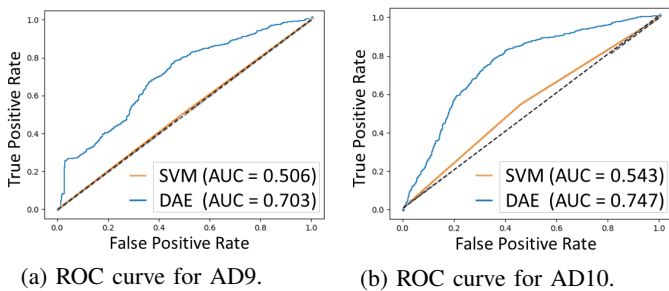(a) ROC curve for AD9. (b) ROC curve for AD10.

Fig. 5: Anomaly detection results in AD9 and AD10.

[2] K. Kockelman, P. Avery, P. Bansal, S. D. Boyles, P. Bujanovic, T. Choudhary, L. Clements, G. Domnenko, D. Fagnant, J. Helsel *et al.*, "Implications of Connected and Automated Vehicles on the Safety and Operations of Roadway Networks: A final Report," Tech. Rep., Aug. 2016.

[3] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using Discriminant Analysis to Detect Intrusions in External Communication for Self-driving Vehicles," *Digital Communications and Networks*, vol. 3, no. 3, pp. 180–187, Feb. 2017.

[4] K. M. Ali Alheeti and K. McDonald-Maier, "Intelligent Intrusion Detection in External Communication Systems for Autonomous Vehicles,"

Systems Science & Control Eng., vol. 6, no. 1, pp. 48–56, Feb. 2018.

[5] O. Berlin, A. Held, M. Matousek, and F. Kargl, "POSTER: Anomaly-based Misbehaviour Detection in Connected car Backends," in *Proc. of IEEE VNC 2016*, Dec. 2016, pp. 1–2.

[6] G. K. Rajbahadur, A. J. Malton, A. Walenstein, and A. E. Hassan, "A Survey of Anomaly Detection for Connected Vehicle Cybersecurity and Safety," in *Proc. of IEEE IV Symposium 2018*, Jun. 2018, pp. 421–426.

[7] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *Proc. of IEEE ICST 2008*, Mar. 2008.

[8] R. McConville, D. Byrne, I. Craddock, R. Piechocki, J. Pope, and R. Santos-Rodríguez, "Understanding the quality of calibrations for indoor localisation," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE, 2018, pp. 676–681.

[9] D. Byrne, M. Kozlowski, R. Santos-Rodriguez, R. Piechocki, and I. Craddock, "Residential wearable rssi and accelerometer measurements with detailed location annotations," *Scientific data*, vol. 5, p. 180168, 2018.

[10] R. McConville, D. Byrne, I. Craddock, R. Piechocki, J. Pope, and R. Santos-Rodriguez, "A dataset for room level indoor localization using a smart home in a box," *Data in brief*, vol. 22, pp. 1044–1051, 2019.

[11] M. Mukherjee and A. B. Bhattacharya, "RSSI Based Indoor human activity Recognition System," *Techno International Journal of Health, Engineering, Management and Science (TIJHEMS)*, Dec. 2018.

[12] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor Positioning Algorithm Based on the Improved RSSI Distance Model," *Sensors*, vol. 18, no. 9, p. 2820, Aug. 2018.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, May 2014.

[14] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clustering Comparison: Variants, Properties, Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, Oct. 2010.

[15] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Poster: Parallel Implementation of the OMNeT++ INET Framework for V2X Communications," in *Proc. of IEEE VNC 2018*, Dec. 2018, pp. 1–2.

[16] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, Dec. 2012.

[17] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Agile Calibration Process of Full-Stack Simulation Frameworks for V2X Communications," in *Proc. of IEEE VNC 2017*, Nov. 2017, pp. 89–96.

[18] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.