Deep Learning-based Intra-slice Attack Detection for 5G-V2X Sliced Networks

Abdelwahab Boualouache*, Taki Eddine Toufik Djaidja**, Sidi-Mohammed Senouci**,
Yacine Ghamri-Doudane[+], Bouziane Brik**, and Thomas Engel*
(*) FSTM, University of Luxembourg, Luxembourg, email: {firstname.lastname}@uni.lu
(**) DRIVE Laboratory, University of Bourgogne, France, email: {firstname.lastname}@u-bourgogne.fr
([+]) L3I Laboratory, University of La Rochelle, France, email: {firstname.lastname}@univ-lr.fr

***Abstract*—Connected and Automated Vehicles (CAVs) represent one of the main verticals of 5G to provide road safety, road traffic efficiency, and user convenience. As a key enabler of 5G, Network Slicing (NS) aims to create Vehicle-to-Everything (V2X) network slices with different network requirements on a shared and programmable physical infrastructure. However, NS has generated new network threats that might target CAVs leading to road hazards. More specifically, such attacks may target either the inner functioning of each V2X-NS (intra-slice) or break the NS isolation. In this paper, we aim to deal with the raised question of how to detect intra-slice V2X attacks. To do so, we leverage both Virtual Security as a Service (VSaS) concept and deep learning (DL) to deploy a set of DL-empowered security Virtual Network Functions (sVNFs) within V2X-NSs. These sVNFs are in charge of detecting such attacks, thanks to a DL model that we also build in this work. The proposed DL model is trained, validated, and tested using a publicly available dataset. The results show the efficiency and accuracy of our scheme to detect intra-slice V2X attacks.**

***Index Terms*—5G-V2X; Network Slicing; Security; Deep Learning; Intra-slice attack detection**

I. Introduction

Fifth-generation (5G) mobile systems came to address the new stringent requirements brought by the different verticals that aim to use these systems, such as transportation, manufacturing, energy, and e-Health, among others. As a key pillar of Cooperative Intelligent Transportation Systems (C-ITS), Connected and Autonomous Vehicles (CAVs) significantly benefit from 5G advances to enhance further road safety, traffic efficiency, and user convenience. As part of 3GPP Release 16 [1], 5G-Vehicle-to-Everything (5G-V2X) communication technology is specified to enable not only direct communication via the PC5 interface but also provides ultra-low and ultra-reliable communications. On top of this, CAVs are supported with a set of 5G enabling technologies, such as Software Defined Networking (SDN), Network Function Virtualization (NFV), Multi-access Edge Computing (MEC), and Network Slicing (NS). The latter allows the creation of independent virtual networks, targeting different requirements built on the top of the same physical infrastructure. The NS concept enables several V2X use-cases to operate together, such as automated lane merging/splitting and overtaking, real-time traffic flow regulation, and network-assisted vulnerable road user protection [2]. This will expand the exploitation degree from the 5G physical infrastructure while improving the performance of CAVs' applications and services. Besides, 5G-V2X communications are inherently vulnerable to various internal and external attacks such as denial of service, false information injection, and impersonation, leading to hazardous situations for CAVs and putting users' lives in danger. The situation gets worse with the introduction of network slicing. Indeed, combining 5G-V2X with NS will increase attack severity and open up a network slicing attack vector to CAVs [3]. We can classify these attacks into two main categories: (i) intra-slice attacks in which the attacker(s) and the target(s) belong to the same V2X Network Slice (V2X-NS), and (ii) inter-slice attacks in which the attacker(s) and/or the target(s) belong to different V2X-NSs. In this context, while cryptography solutions are efficient against external active V2X-NS attacks since attackers are not authenticated members of the 5G-V2X network, mitigating internal V2X-NS attacks is difficult using these solutions since the attackers are part of the network.

Intrusion detection systems are considered an efficient security mechanism to detect internal attacks [4], especially when taking advantage of the latest advances brought by machine/deep learning approaches. To this end, this paper focuses on leveraging Deep Learning (DL) approaches to detect intra-slice V2X attacks. More precisely, we design a new scheme that leverages both the concept of Virtual Security as a Service (VSaS) and DL for efficient intra-slice attack detection. VSaS has shown to be a promising approach to address 5G security concerns, thanks to the flexibility and elasticity support they provide [5]. VSaS consists in integrating built-in security Virtual Network Functions (sVNFs) in the slice life cycle. To ensure end-to-end security, we propose to deploy several sVNFs, within the V2X-NSs. Moreover, we build a new DL-based model that targets detecting intra-slice attacks. This DL model is executed on top of the deployed sVNFs. Our scheme is evaluated on attack detection capabilities and deployment performance.

The remainder of this paper is organized as follows. Section II describes the main related work on attack detection in 5G-V2X networks. Our system model and targeted intra-slice V2X attacks are presented in Section III. The design of the DL model for intra-slice V2X attack detection is presented in Section IV. Section V depicts the performance evaluation results. Finally, Section VI concludes the paper.

II. Related Work

Various Machine Learning (ML)-based schemes have been proposed to detect attacks on 5G-V2X communications. Alheet *et al.* [6] proposed a centralized system based on supervised learning to detect network-level attacks. This system trains a multi-class classifier, using neural networks to deal with different attack types. Alternatively, Ashraf *et al.* [7] proposed a centralized attack detection system based on unsupervised learning. Thus, an anomaly detection model was trained based on Long Short-Term Memory (LSTM) auto-encoder architecture to distinguish the suspicious traffic from the normal one. Bangui *et al.* [8] proposed a hybrid attack detection system. A multi-classifier model was trained using Random Forest (RF) to detect known attacks, while an anomaly detection model was built using a variation of K-means to detect unknown attacks. Ghaleb *et al.* [9] proposed a collaborative attack detection system consisting of four main phases: (i) Individual ML model construction, (ii) ML models exchanging, (iii) ML model evaluation, and (iv) Collaborative system construction. The binary classification was used to build the model using random forest, XGBoost, and Support Vector Machine (SVM) algorithms. Shu *et al.* [10] proposed also a collaborative attack detection system based on SDN. This system enables multiple distributed SDN controllers to train a binary classifier based on DL combined with a generative adversarial network. Yang *et al.* [11] proposed a multi-tiered hybrid intrusion detection system. This uses multi-class classification models to detect known attacks and anomaly detection models to detect unknown attacks.

Although different attack detection systems with different architectures (centralized, distributed) and different learning methods (supervised, unsupervised, and hybrid) have been proposed, intra-slice V2X attack detection is not addressed yet. Thantharate *et al.* [12] and Kuadey *et al.* [13] proposed a DL-based attack detection system for Distributed Denial of Service (DDoS) attacks for 5G networks. However, these systems are (i) only designed to detect DDoS against the core network and (ii) do not consider attacks on the network access part, including V2X and the MEC.

III. Intra-slice V2X Attack Detection Scheme

Figure 1 shows 5G-V2X NS architecture consisting of (i) 5G New Radio (NR) including CAVs and gNodeBs, (ii) MEC nodes, and (iii) 5G Core networks. 5G-V2X NSs are created and managed by the Network Slice Manager (NSM). During the creation of the NSs, the NSM allocates the necessary storage and processing resource to satisfy the requirements defined in the Service-Level Agreement (SLA). The NSM also implements all required VNF service chaining while ensuring the isolation level defined in the SLA. The isolation level allows specifying VNFs dedicated to a V2X-NS and VNFs shared between V2X-NSs. Figure 1 presents two V2X-NSs with different network requirements. These NSs share VNFs at the core level and have dedicated VNFs at the MEC and NR levels. Each CAV is equipped with a 5G network card to communicate with other CAVs via the PC5 interface and C-V2X applications via the Uu interface. To ensure end-to-end

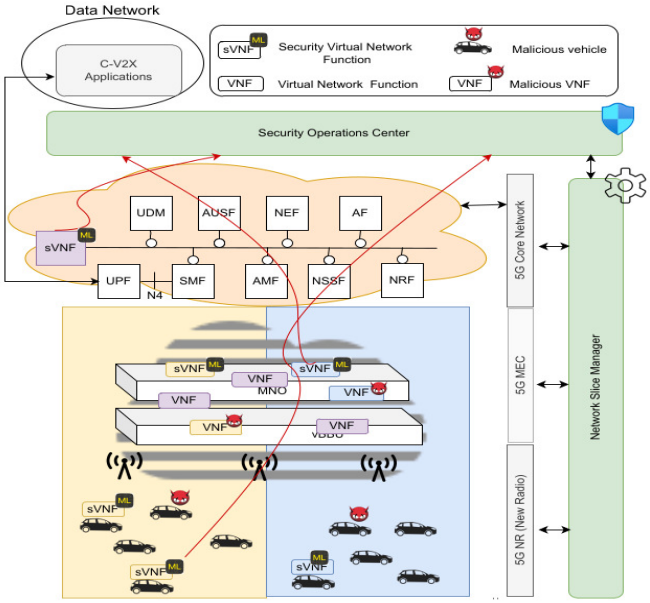

Fig. 1: DL-based Intra-slice attack detection scheme in 5G-V2X NS

security of V2X NSs, sVNF are also deployed within V2X-NSs during their creation. Our DL-based model deployed on top of sVNFs enables us to detect intra-slice V2X attacks and report them to the Security Operations Center (SOC). sVNFs are deployed at different levels of V2X-NSs. At the 5G NR level, sVNFs are deployed at some selected CAVs, according to predefined policies. For example, the NSM can obtain the list of most trusted vehicles from SOC. Thereby, it can deploy sVNFs on the most trusted CAVs. From the technical perspective, we expect that CAVs are equipped with a hypervisor that can support one or more sVNFs. At the MEC, sVNFs are deployed on the MEC nodes according to the number of CAV subscribed to a V2X-NS using that MEC node, and sVNFs can also be migrated from a MEC node to another node according to the mobility of the vehicles. sVNFs are also deployed at the core part of the network, according to the isolation level defined by the SLA. In a complete V2X-NS isolation scenario, each sVNF is dedicated to only one V2X-NS. However, in our example, since V2X-NSs share the core part, they also share the sVNF. As previously mentioned, detected attacks are reported to SOC, which is in charge of reacting to the detected attack and updating the list of trusted vehicles. Besides, and as aforementioned, we focus on detecting intra-slice V2X attacks. More precisely, two classes of attacks are considered in this paper.

- **Class 1 (Denial of service (DoS)):** The attackers of this class try to prevent V2X-NS members from having ordinal access to V2X-NS services. DoS can target different parts of V2X-NSs, ranging from the network access to the 5G core. Moreover, attackers can exploit many network protocols at different levels of the protocol stack. UDP flooding, HTTP DoS, and ARP flooding are examples of DoS attacks that might be used. DDoS is a variant of the DoS attack that involves multiple attackers belonging to

the same NS, which can collaborate and synchronize to perform the attack.

- **Class 2 (NSM Impersonation):** The NSM is the core component of the 5G-network architecture, which is in charge of the life cycle management of V2X-NSs. The NSM ensures the end-to-end monitoring of V2X-NSs, by continuous and dynamic interactions with the access and core network elements. This causes an increase in the risk of impersonation attacks. For example, an attacker can pretend to be an NSM to monopolize the NS resources for its own benefit. Impersonation is a multi-stage attack that starts by infiltrating the networks by exploiting protocol flaws.

IV. Deep Learning Model for Intra-slice Attack Detection

This section describes the process used to build and deploy our DL model on sVNFs. Figure 2 illustrates this process consisting of four main steps. We selected the CSE-CIC-IDS-2018 dataset [14] to train our DL model since it adequately covers different types of intra-slice V2X attacks addressed in this paper. During the dataset processing, we prepared the dataset for the next steps. Thus, we performed dataset cleaning, transformed features to a suitable format, performed data scaling, and split the dataset into three sub-datasets: training, validation, and test. The two following steps are aimed at training and validating the DL model. These steps take as an input previously processed sub-datasets. Moreover, they are iteratively run until finding the best hyperparameter configuration, which ensures the high accuracy of the classification model. The hyperparameters consist of the DL architecture, including the number of layers of DL and the number of nodes on each hidden layer, the optimizer, the learning rate, the batch size, and the number of epochs. After building the best DL model that ensures the highest detection of intra-slice V2X attacks. The DL model will be ready to be deployed on sVNFs. The last step is thus to deploy the DL model on top of sVNFs, which are distributed along the V2X-NS using virtualization/containerization technologies, such as Docker. This section focuses only on the dataset processing step. The rest of the steps are detailed in the next section.

CSE-CIC-IDS-2018 dataset is a collaborative project between the Communications Security Establishment and the Canadian Institute for Cybersecurity. This dataset was generated during 10 days of network traffic analysis. It includes 14 attack types divided into seven attack scenarios: (i) DoS (3 attacks), (ii) brute force (2 attacks), (iii) Heartbleed, (iv) web attacks (2 attacks), (v) infiltration, (vi) botnet and (vii) Low Orbit Ion Canon (LOIC). The dataset includes 80 features on network flows. It also includes a timestamp, which is divided into six features representing time in terms of the year, the month, the day, the hour, minutes, and seconds respectively. Thus, after converting the timestamp feature, the total number of features becomes 85. To train our DL model, we have a dataset containing a total of $9,232,943$ instances (rows) with 85 features (columns). Table I shows the dataset distribution per each attack type.

TABLE I: Dataset distribution per each attack type

Attack type	Support
Benign	6484708
DDOS attack-HOIC	686012
DDoS attacks-LOIC-HTTP	576191
DoS attacks-Hulk	461912
Bot	286191
FTP-BruteForce	193360
SSH-Bruteforce	187589
Infilteration	161934
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS attack-LOIC-UDP	1730
Brute Force -Web	611
Brute Force -XSS	230
SQL Injection	87

Thus, this dataset contains several features with different scales, which slows down the training process. To address this issue, we also rescaled the dataset using MinMaxScaler, which normalizes dataset features to values in the range of [0,1]. Equation 1 gives the normalization done by MinMaxScaler.

$$x_{scaled} = \frac{x^i - x_{min}}{x_{max} - x_{min}} \tag{1}$$

MinMaxScaler deducts the minimum value of the feature from the original value and then divides the result by the range. The range is the difference between the maximum and minimum values of that feature. Before passing the dataset to the next step, it is split into training, validation, and test sub-datasets. Since this dataset contains more than 9 million rows, which is in the order of big data, we thus apply recommendations given in [15] by choosing 1% of the whole dataset as a test dataset and 1% of the training dataset as a validation dataset.

V. Performance Evaluation

In this section, we evaluate the performance of our DL model, which will be embedded in sVNFs. Our model consists of (i) an inputs layer with 84 neuron nodes and (ii) two hidden layers with 128 and 84 nodes, respectively. An output layer with 15 nodes based on one hot encoding to detect and identify intra-slice V2X attacks. The ReLU activation function is used for the hidden nodes, while the softmax function is used for the output layer. The model was trained on the Google Colab platform, using Compute Engine backend (TPU). Table II lists the hyperparameters of the model. We have used the ADAM optimizer with 0.01, as a learning rate for the gradient descent algorithm. During the training, we consider mini-batches of size 256, while the number of epochs is set to 50.

We considered a set of metrics to evaluate our DL model. These metrics include optimization metrics, such as the accuracy and F1-score, and satisfactory metrics, such as inference time and memory. Table III gives equations related to the evaluation metrics, where TP, FP, TN, and FN are the True Positive, the False Positive, the True Negative, and the False Negative, respectively. Figure 3 (a) and Figure 3 (b) show the loss and accuracy of the training and validation dataset,

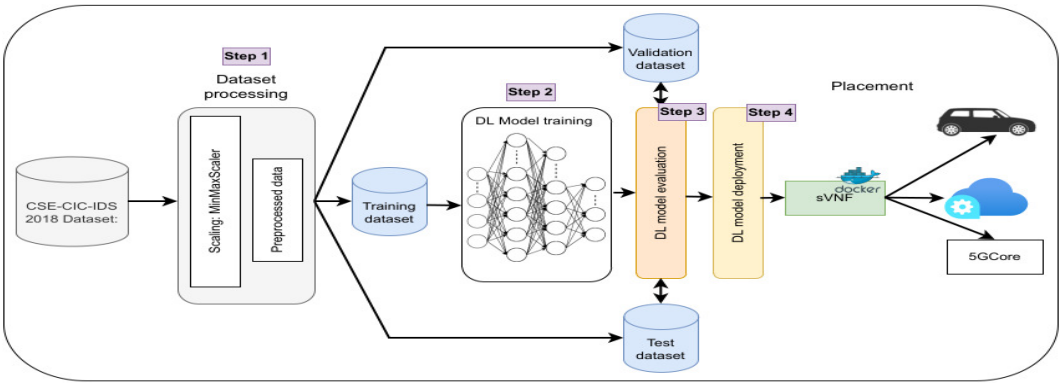

Fig. 2: The process to build and deploy the DL model for intra-slice V2X attack detection

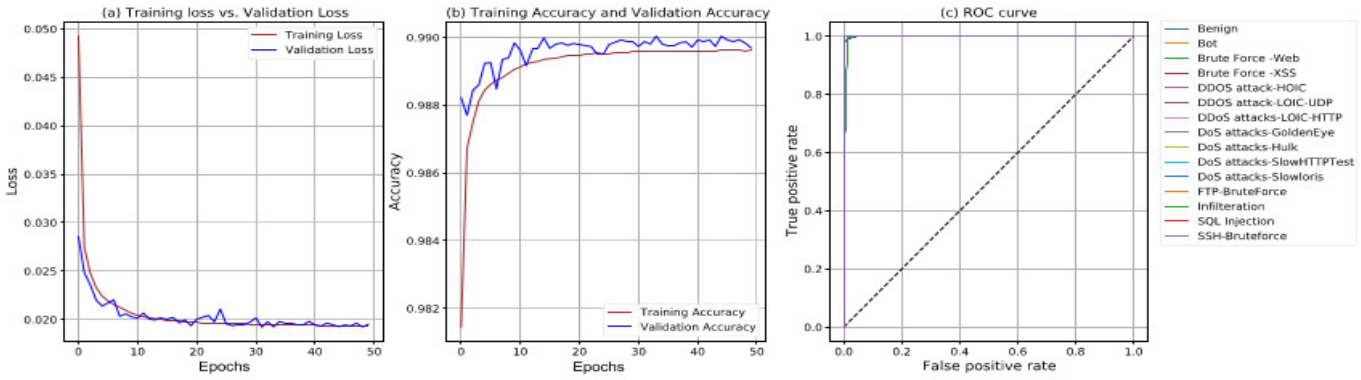

Fig. 3: Training process

TABLE II: Training parameters of the DL model

Parameter	Value
Optimizer	ADAM
Learning rate	0.01
Number of epochs	50
Batch size	256
Ratio of validation dataset	1%
Ratio of test dataset	1%

TABLE III: Metrics

Metric	Formula/Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1- score	$2X\frac{Precision*Recall}{Precision+Recall}$
ROC	the trade-off between the true positive rate and the false positive
AUC	the ability of a classifier to distinguish between classes.

respectively. We can see that loss significantly decreases since the first steps of training. We can also see that our model is doing well on both the training and the validation datasets. Indeed, the accuracy surpasses 98% both the training and validation data set. Table IV shows the performance of our model on the test dataset. As we can observe, our DL performs better on the test dataset. We have obtained 99% for both accuracy and F1-score. We also obtained 99% for The Area Under the Curve (AUC). These results demonstrate the efficiency of our model in detecting not previously seen attack instances and distinguishing between attack classes.

TABLE IV: Attack detection results

Accuracy	Precision	Recall	F1-score	AUC
0.99	0.98	0.95	0.99	0.99

Table V shows the detection results of our DL model per each type of attack. These results demonstrate the high capability of our DL model, to distinguish the benign traffic network from the malicious one. Indeed, our DL model with 99% of F1-score regarding identifying benign network traffic. In addition, as shown in the Receiver Operator Characteristic (ROC) curve, illustrated in inf Figure 3 (c), attack curves are closer to the top-left corner, which further proves high performance in classifying intra-slice V2X attacks. Note the backline illustrates the performance of a basic classifier. Moreover, Table V shows high accuracy (more than 97% of F1-score) to detect most of the attacks (12/14). However, Bute Force-Web and Infiltration attacks have obtained less interesting results. This is due to the number of instances used in the test dataset.

Table VI gives the evaluation of the DL model in terms of satisfactory metrics. Our model was trained for 94.8 seconds,

TABLE V: Multi-class results of the DL model

Traffic type	Precision	Recall	F1-score	Support
Benign	0.99	1.00	0.99	64848
Bot	1.00	1.00	1.00	2862
Brute Force-Web	1.00	0.67	0.80	6
Brute Force-XSS	1.00	1.00	1.00	2
DDOS attack-HOIC	1.00	1.00	1.00	6860
DDOS attack-LOIC-UDP	1.00	0.94	0.97	17
DDoS attacks-LOIC-HTTP	1.00	1.00	1.00	5762
DoS attacks-GoldenEye	1.00	1.00	1.00	415
DoS attacks-Hulk	1.00	1.00	1.00	4619
DoS attacks-SlowHTTPTest	1.00	1.00	1.00	1399
DoS attacks-Slowloris	1.00	1.00	1.00	110
FTP-BruteForce	1.00	1.00	1.00	1934
Infilteration	0.77	0.58	0.66	1619
SQL Injection	1.00	1.00	1.00	1
SSH-Bruteforce	1.00	1.00	1.00	1876

TABLE VI: DL model deployment performance

Training time (s)	**Size (KB)**	**Inference time (s)**
94.8	309.808	0.058

which is an acceptable time to train the model. Our model also has a small storage size (less than 1 MB), making them lightly deployable in sVNFs, at different levels from CAVs to the 5G core. Moreover, the Inference Time (IT) is short. It takes less than 58 ms to decide if an event is an attack or not, which demonstrates the fast detection of our scheme, leading to an immediate reaction after detecting an attack. Figure 4 shows the IT required by the DL model for verifying 1000 events. The x-axis is the number of events in each Inference Operation (IO). As depicted, IT decreases with the increase of the number of events in each IO. If events are sequentially verified, then 1000 IOs are performed; thereby, IT equals 62.29s. However, if 100 events are verified in each IO, then 10 IOs are performed; thus, IT equals 1.72s. We can explain these results by the role of the vectorization technique in making the DL model runs faster.

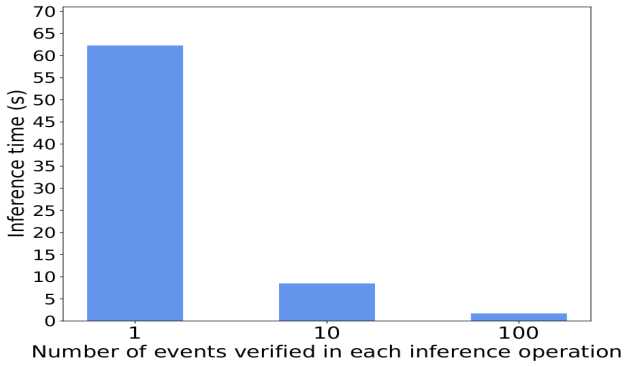

Fig. 4: The inference time vs. the number of events verified in each inference operation (the total number of events equals 1000)

VI. Conclusion

The failure to detect intra-slice 5G-V2X attacks could jeopardize the safety of users. This paper has designed a novel scheme for detecting intra-slice V2X attacks. Our scheme combines the flexibility of virtual security as a service and the power of deep learning to efficiently detect the attack while taking deployment indicators into account. We plan to perform feature engineering to optimize the size model further while enhancing the performance in future work.

Acknowledgment

This work was supported by the 5G-INSIGHT bilateral project, (ANR-20-CE25-0015-16), funded by the Luxembourg National Research Fund (FNR), and by the French National Research Agency (ANR).

References

[1] 3GPP TS 23.287, "Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services," Jul 2020.

[2] C. Campolo, A. Molinaro, and V. Sciancalepore, "5G Network Slicing for V2X Communications: Technologies and Enablers," *Radio Access Network Slicing and Virtualization for 5G Vertical Industries*, pp. 239–257, 2021.

[3] H. Mun, M. Seo, and D. H. Lee, "Secure privacy-preserving v2v communication in 5g-v2x supporting network slicing," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[4] 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 (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[5] Y. Khettab, M. Bagaa, D. L. C. Dutra, T. Taleb, and N. Toumi, "Virtual security as a service for 5G verticals," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018, pp. 1–6.

[6] K. M. Ali Alheeti and K. McDonald-Maier, "Intelligent intrusion detection in external communication systems for autonomous vehicles," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 48–56, 2018.

[7] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled lstm autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[8] H. Bangui, M. Ge, and B. Buhnova, "A Hybrid Data-driven Model for Intrusion Detection in VANET," *Procedia Computer Science*, vol. 184, pp. 516–523, 2021.

[9] F. A Ghaleb, F. Saeed, M. Al-Sarem, B. Ali Saleh Al-rimy, W. Boulila, A. Eljialy, K. Aloufi, and M. Alazab, "Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for vanet," *Electronics*, vol. 9, no. 9, p. 1411, 2020.

[10] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[11] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multi-Tiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet of Things Journal*, 2021.

[12] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 0762–0767.

[13] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "DeepSecure: Detection of Distributed Denial of Service Attacks on 5G Network Slicing-Deep Learning Approach," *IEEE Wireless Communications Letters*, 2021.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.

[15] DeepLearning.AI, "Setting up your ML application: Train/dev/test sets," *Coursera. Available online: https://cs230.stanford.edu/files/C2M1.pdf (accessed on 2 March 2022)*, 2022.