Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November 2022, Maldives

DAD: A Distributed Anomaly Detection framework for future In-vehicle network.

1st Elies Gherbi
Irt-SystemX
Palaiseau, France
elies.gherbi@irt-systemx.fr

2nd Blaise Hanczar
Univ Evry, IBISC
Evry, France
blaise.hanczar@univ-evry.fr

3rd Jean-Christophe Janodet
Univ Evry, IBISC
Evry, France
jeanchristophe.janodet@univ-evry.fr

4rd Witold Klaudel
Irt-SystemX
Palaiseau, France
witold.klaudel@ext.irt-systemx.fr

***Abstract*—Future in-vehicle (autonomous vehicles) network architectures will consider many aspects of modern network security by design. The general system contains many sub-systems related to different tasks with specific functional priorities and dedicated security mechanisms. In this work, we propose a Distributed Anomaly Detection (DAD) Intrusion Detection System (IDS) using a deep learning model that fits the in-vehicle network architecture. DAD aims to model the complex correlations among different views (sub-systems) by harnessing the joint distribution of the different sources of CAN (Controller Area Network) data. To this end, we propose DAD by jointly learning an anomaly detection model for critical applications such as security and maintenance while adopting the same isolation constraint on the sub-systems. On top of that, we introduce a new optimisation scheme that lowers both the computational inference time and the IDS's communication overhead.**

***Index Terms*—deep learning, intrusion detection system, in-vehicle communication, distributed network, anomaly detection.**

I. Introduction

The complexity of the multi-layered embedded infrastructures such as autonomous vehicle is constantly increasing. With the critical security concerns and for the safety of the passengers, future In-vehicle network architecture is composed of different subsystems embedded on electronic control units (ECUs). Each subsystem is responsible for specific services that ensure the autonomous functioning of the vehicle. For functional and security reasons, separate subsystems are isolated, forming a hierarchical architecture of the whole system. In that context, data is represented with different views (local data monitored on each ECU) and can include multiple modalities or various features. These views may be obtained from multiple sources or different feature subsets.

Building an in-vehicle Intrusion Detection System (IDS) needs to meet the hierarchical structure of the system. We set the in-vehicle architecture scheme as follows; each subsystem contains a probe (S) that monitors its local CAN network (Controller Area Network) environment and communicates only with the central probe that we call bastion (B) (see "Fig. 1").In that context, we define two strategies to build a Deep Anomaly Detection model for an in-vehicle IDS with its inner architectural characteristics (See "Fig. 2").

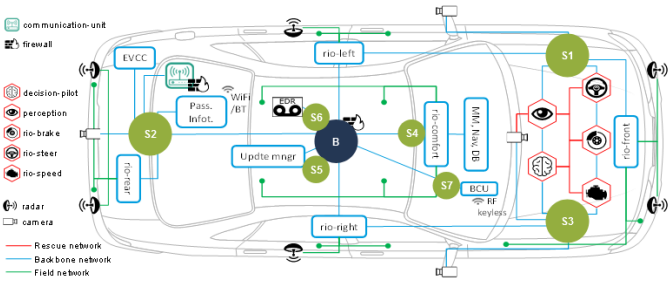
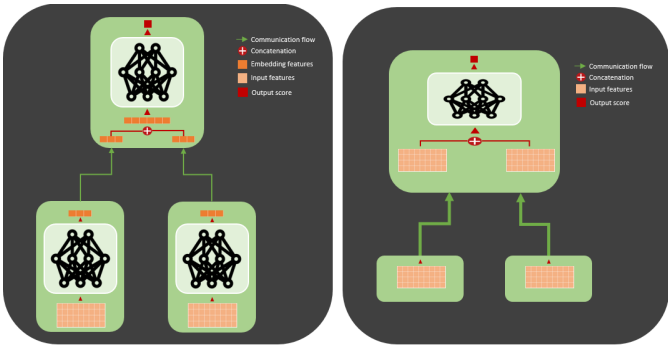


Fig. 1. Schema representing the different probes that monitor various subsystems of an autonomous vehicle architecture network.

1) Centralized anomaly detection modelling: In this case, the computational resources are located only on the bastion (See "Fig. 2(b)"), and the model is embedded on the bastion. Each probe sends its data information to the bastion. So, the bastion aggregates this information and predicts the system's state based on one model.
2) Distributed Anomaly Detection Modelling: (As described in "Fig. 2(a)"), a common strategy is to learn a joint representation coupled between multiple views at a higher level after learning several layers of view-specific features related to the specific probe. Each probe's particular view is represented with a lower dimension vector (feature vector). Those feature vectors jointly leverage the abundant and complementary information from multiple views. From a neural network modelling perspective, multi-view representation learning first learns the respective mid-level features for each view (probe) using a sub-neural network. The bastion then integrates the complementary knowledge of different views to comprehensively represent the initial data into a single and compact representation.

This work focuses on distributed anomaly detection using deep learning, represented in "Fig. 2(a)". We propose a Distributed Anomaly Detection framework (DAD) intrusion detection system using a multi-view deep learning architecture. The proposed deep learning architecture respects the in-vehicle



(a) Distributed Anomaly Detection Modelling (b) Centralized anomaly detection modelling

Fig. 2. (a) Represents the Distributed Anomaly Detection Modelling. (b) Represents the centralized model based on a single view.

network topology and constraints (probes communicate only with the bastion). It aims to classify the network's current state by learning about local patterns related to each view. Thus, the subspace learning-based approaches (multi-view representation learning) aim to obtain a joint feature vector subspace shared by multiple views (ECUs) representing the input data information. The received feature vector's dimensionality is lower than the input view, so subspace learning effectively reduces the "curse of dimensionality". Given this subspace, it is straightforward to conduct subsequent tasks, such as classification for anomaly detection. Furthermore, DAD reduces the communication overhead induced by integrating an intrusion detection system on top of the existing in-vehicle network functionalities. Indeed, in-vehicle systems generally have low computational power, and any additional computational resources (processing and communication overhead) can be translated into a high cost for automotive manufacturers. We conduct detailed experiments to assess the relationship between different sub-networks predictions and their relation in representation learning and accuracy of classification on different attack types using SynCAN dataset ([13]).Also, We added a communication overhead optimisation scheme in the inference phase. DAD IDS is the first practical in-vehicle intrusion detection that fits the distributed network architecture while introducing the notion of communication overhead optimisation to the best of our knowledge.

II. Related works

Multi-view and multi-modal Deep learning architectures have been considered for many anomaly detection problems, especially in network environments forming many swarms like VANet's (Vehicular Ad-Hoc Network), cloud computing, and edge computing areas [1]. In those cases, most data is collected from different sources, or different features extractors [2]–[5], [8]. In other words, data instances are usually depicted by heterogeneous feature spaces in the form of multiple views. Several multi-view learning approaches can be considered to develop an anomaly detection model-based IDS. Multi-view representation learning is used for multi-source data to facilitate the extraction of readily useful information when developing prediction models since it provides a holistic description of the system.

Data from different views usually contains complementary information, and multi-view representation learning exploits this point to learn more comprehensive representations than single-view learning methods [7], [8]. There has been increasing research applied to multi-view learning using Deep Learning, deep architecture-based methods, including multi-modal deep Boltzmann machines, [9], multi-modal deep autoencoders [10], [11], and multi-modal recurrent neural networks [12]. In the existing In-vehicle IDS literature, most studies neglect the inherent multi-view property of data due to the lack of IDS problem modelisation concerning future in-vehicle architectures. Furthermore, the lack of related datasets is mainly the reason. Thus, building an in-vehicle IDS for the autonomous car is challenging for the scientific community. Few works tackle building an in-vehicle IDS that monitors the CAN bus network by proposing a distributed anomaly detection system structure. In [6], the author proposes a distributed IDS based on hierarchical temporal memory (HTM). The input at each detector in the sequence is the bits from the packet's data field related to each ID. Then the model using the HTM algorithm learns to predict the following data field of each ID. An overall score within a time window for the entire input sequence groups the different IDs scores. In [13] the authors also tackled the problem in the same manner in terms of distributed modelling, where the data input modalities are sequences related to a specific ID. The authors proposed multi-view architecture based on independent recurrent neural networks (LSTM for each ID that gets a sequence associated with this ID as inputs). The joint latent vector is fed into a subnetwork of consecutive linear layers in an autoencoder setting. At each time step, this subnetwork's task is to reconstruct the signal values of each possible input message solely based on the current joint latent vector. The drawback in the IDs-based distribution is that both propose several independent sub-network equal to the number of IDs. In other words, those architectures and algorithms do not consider the restrained resources in the in-vehicle systems; modern vehicles are composed of 70 to 100 ECU connecting to the in-vehicle network. The communication overhead is augmented with a number of signals and processing equal to the number of IDs available on CAN data.

In this work, we formulate the problem related to this new type of in-vehicle architecture and propose a general framework that can be extended to fit autonomous vehicle network systems.

III. Problem modelling

A. Problem Statement Centralized model

We consider a distributed architecture for network intrusion detection system. Firstly, we assume that we have p probes $(S_1, S_2, ..., S_p)$ that monitors the vehicle's different subnetworks (CAN bus communication between different ECUs).

The bastion B hosts a local anomaly detector D_B. The input of detector D_B is a joint vector $X = \bigcup_{i=1}^{p} T_i$ where T_i is the multivariate time series sent by S_i. The aim of D_B is to distinguish between the normal pattern and the attack pattern based on the join vector X obtained from the different probes S_i. Formally $D_B(X) \approx \mathbb{P}(y = 1|X)$ where $y \in [0, 1]$ (with 1 for normal example and 0 for anomalous examples).

1) Communication cost: To define the communication cost of the centralized model, we start by giving its functional process. The model D_B is embedded in the bastion, and it receives its input from the different probes S_i. The Input example $X = \bigcup_{i=1}^{p} T_i \in \mathbb{R}^{W*M}$, where W is the length of the sequence, and M the number of variable.

Since t_0 (initialisation of the detection model), each probes buffers the different information messages till it construct the first sequence $T_i^{t_0}$ with W as the length of the sequence. After sending their first sequence $T_i^{t_0}$, The different probes keeps buffering the information messages till the sequence reaches a length $step$, where step is the decay of the window. The $W - step$ represent the number of multivariate series from t_{i-1}. So, $T_i^{t_l} = U_{ibuff} \cup U_{ihist}$, where $U_{buff} \in \mathbb{R}^{step*M}$ and $U_{hist} \in \mathbb{R}^{(W-step)*M}$, t_l is the timing of detection after the initialisation with $l \in [1, \infty]$.

we give below the formula of the communication cost of a probe in a centralized model for a given test set containing N sequence examples.

$$\zeta_{centr}(test) = k|T_i^{t_0}| + H + (N - 1)(k|U_{ibuff}| + H) \quad (1)$$

where H is the incompressible cost of initiating a message sending and k is a parameter that allows adjusting the importance of both criteria.

B. Problem Statement Distributed

In the distributed setting, each probe S_i hosts a local anomaly detector D_i. The input of detector D_i is a multivariate time series T_i. The aim of D_i is to distinguish between the normal pattern and the attack pattern based on the features extracted from T_i only. Formally $D_i(T_i) \approx \mathbb{P}(y_i = 1|T_i)$ where $y_i \in [0, 1]$ (with 1 for normal example and 0 for anomalous examples).

The probes are not allowed to exchange information between them. They exchange information only with the bastion B. We want to limit this exchange of information as much as possible to reduce the communication overhead. To achieve this goal, we use two levers. On one hand, the detectors do not send information at each step of time: it assesses the probability of an attack $D_i(T_i)$ and raises an intrusion alarm $e_i \in \{0, 1\}$ if this value is larger than a given threshold τ_i. Formally,

$$e_i = \begin{cases} 0, & \text{if } D_i(x_i) \geq \tau_i \\ 1 \end{cases} \quad (2)$$

If no probe raises the alarm, then no information is sent to the bastion by any probe. On the other hand, if only one of the probes raises the alarm, the bastion asks all the probes to transmit information about their status and local data. This is the second lever where we act: as shown below, the detectors are based on deep learning and designed to provide a condensed (summarized) representation of the data using a function G_i (the layer befor the logit of D_i). Therefore, the detector D_i does not transmit raw input data T_i but a condensed representation denoted $v_i = G_i(T_i)$. The bastion hosts an anomaly detector D_B that groups the different representation v_i to decide on the global system behaviour $D_B(V) \approx \mathbb{P}(y_B = 1|V)$.

This allows reducing the communication overhead. To quantify this, let us introduce the following,

communication cost function ζ: Given any vector W , let $|W|$ denote the size of the vector; then

$$\zeta(W) = k|W| + H \quad (3)$$

where H is the incompressible cost of initiating a message sending and k is a parameter that allows adjusting the importance of both criteria.

Now, with respect to an input series of raw data T_i, let N be the number of data points in a given dataset; N is the number of messages that would transit through the network to the bastion B in the absence of local detector D_i (Centralized model see "Fig. 2(b)"). In the presence of detector D_i, the number of such messages is reduced, so let N_e be the number of messages sent from D_i to the bastion B. Moreover, the transmitted information is v_i, which is much lighter than T_i, so the communication gain of our techniques is the following:

$$gain = \sum_{1}^{N} \zeta(T_i) - \sum_{1}^{N_e} \zeta(v_i) \quad (4)$$

IV. Proposed model

The overall goal is to build a model (DAD) that detects attacks in a car when they happen. The model must respect and fit the in-vehicle network's distributed architecture and, more precisely, reduce the communication overhead brought by the detection process. We propose a hierarchical Distributed intrusion detection system based on multi-view deep learning architecture (See "Fig. 3") to respond to the abovementioned problem (Distributed Anomaly Detection Modelling)). The model has multiple input sequences T_i related to each probe S_i that will be fed to a Temporal Convolutional Network (TCN) [14], [16]. The model returns multiple outputs $\{\hat{y}_i\}_{i=1}^{P}$ and $\hat{y}_B$. $\hat{y}_i = D_i(T_i)$ represents the local normality probability for each probe, and $\hat{y}_B = D_B(V)$ represents the bastion probability of the normality of the whole system. We denote y_d the final decision based on the combination of $(\{\hat{y}_i\}_{i=1}^{P}, \hat{y}_B, e_i)$ ($e_i \in [0, 1]$ represent if the local probe sends or not the vector). We want to optimise the overhead communication induced by embedding our IDS in the in-vehicle network system. Thus, using a distributed network reduces the size of the information

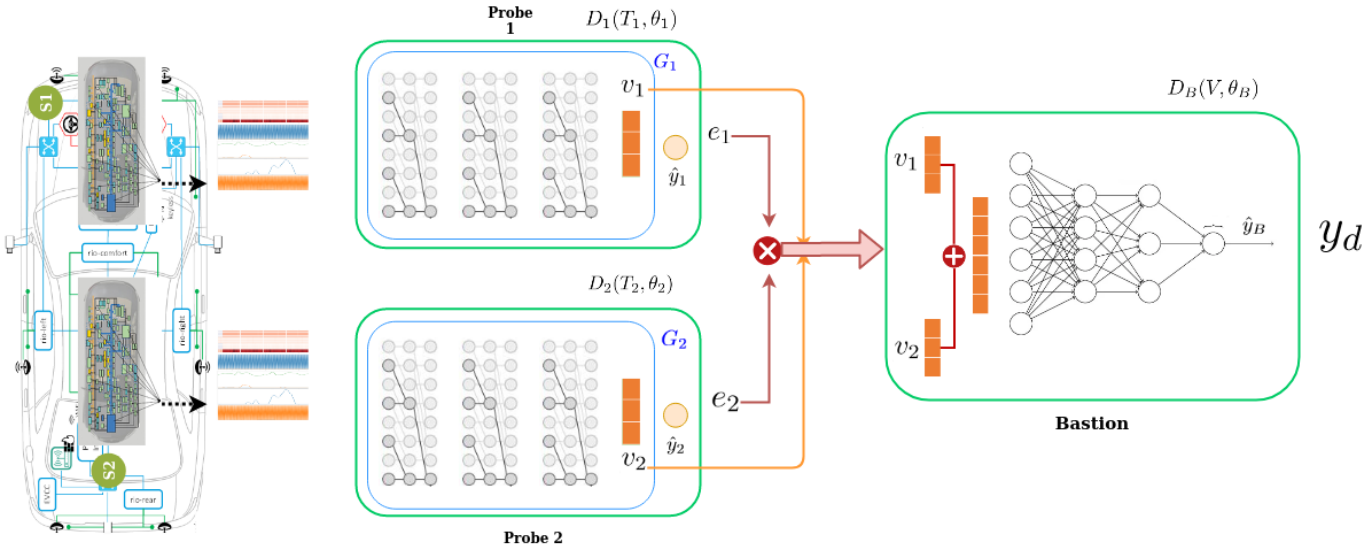


Fig. 3. The framework of the proposed model. We consider two-model classification, but the proposed framework is not limited to two-model. The raw feature consists of the matrix occurrence of the encoded multivariate binary sequence. Both of them are fed to the models D_1f and D_2 for sequence feature extraction and local anomaly prediction. The D_B model receive both v_1 and v_2 for a global anomaly detection prediction y_d .

sent from the probes to the bastion by sending the feature vector v_i representation alternatively of data input sequence T_i. The second point is to reduce the communication rate between the probes and bastion. The idea is to prevent sending the features vector representation when the probes are certain of their local pattern's normality. To this end, we introduce a new target e_i for each probe to decide whether to send the feature vector v_i to the bastion based on threshold optimisation that we will explain below.

The DAD framework has two stages: Model learning and threshold optimisation.

A. Model learning

Since dependencies among local labels and models play an important role in extracting hierarchical dependencies, we privilege the local classification and global classification, meaning that each sub-network model D_i learns a local feature representation relative to each probe. Those representations will then be jointly fed to the upper sub-network classifier D_B to learn about a global pattern based on the join feature representation vector. Hence, giving a second assessment of the network's general state with the overall network information's abstraction. we define the objective function as follow:

$$L_{DAD} = \sum_{i}^{p} L_{S_i} + L_{S_B} \tag{5}$$

The first terms is the loss functions of model D_i. The next term is the global classifier's D_B loss function, which analyses dependencies among the feature vectors v_i to learn the network system's global pattern. Both terms represented with the *binary cross entropy* loss function "(6)" "(7)". We note that Y represent the true label for a given training sample (T, Y).

$$L_{S_i}(T_i, Y) = -Ylog(D_i(T_i))) - (1-Y)log(1-D_i(T_i))) \tag{6}$$

$$L_{S_B}(V, Y) = -Ylog(D_B(V))) - (1-Y)log(1-D_B(V))) \tag{7}$$

The proposed framework leverages local label dependencies and models relations through feature representation to facilitate learning. It constructs a deep network representation and classifier for each view and can make a single view prediction for a specific local view. We propose combining the sub-networks D_i and the global classifiers D_B for the model learning. The backpropagation is used to train the sub-networks and classifiers jointly.

B. Threshold optimisation

After the learning phase, the proposed approach has multiple output prediction values. For a given test samples the following probabilities are yield $preds = (\{\hat{y}_i\}_{i=1}^{P}, \hat{y_B})$. The intrusion detection final decision based on the DAD model is not directly based on the values returned in $preds$. Indeed, we add another step that we call the optimization step. As explained above, we aim to make DAD predictions more economical in terms of communication overhead. To this end, we deduce e_i from $\hat{y}_i$ according to two strategies: Naive test thresholding and Learned models.

1) Naive test thresholding: In this strategy, we deduce e_i using a test set to assess the probability of the lower bound of $\{\hat{y}_i\}$ in this given test set.

2) Communication label creation: The goal of this approach is to propose a communication label creation process based on the rendering of the learned DAD model. In other words, we want to learn Y_{ci}, the labels that will tells if for a given examples, we can trust the probes (local detectors D_i) to assess the state of the network, if so, no communication is needed with the bastion, thus reducing the communication rate between the probes and the bastion. On the other hand, if the labels tells that we need to send a message because we can not trust the rendering of the local detector, the feature vector v_i are sent to the bastion for a prediction using $D_b(V)$. Based on Y_{ci}, we will learn models and embed them on each probe beside D_i.

Y_c expresses for a given example if it's sufficient for local probes to assess the state of the network, or the detection need an upper level that had access to more information through the joint vector V.

We define the *probability error* P_{ei} for the probes and P_{eb} for the bastion. P_e is obtained as follow:

$$P_e = \begin{cases} \hat{y}, & \text{if } y = 0 \\ 1 - \hat{y}, & \text{if } y = 1 \end{cases} \tag{8}$$

Our aim using P_e is to assess how often the local detection is worth trust compared to the model D_B. To this end, we present the difference $P_{ei} - P_{eb}$ as a confidence threshold that need to be established to decide for a given example if the local decision is sufficient or not.

The finale decision $\hat{y}_d, \hat{y}_d : (\{\hat{y}_c i\}_{i=1}^{P}, y_B)$ is given below "(10)", y_c labeling construction using the equation "(9)" considers only normal examples to assess the difference $P_{ei} - P_{eb}$, so when we say that the local decision is sufficient it means that we trust the decision of the normality yield simultaneously at all probes. For anomalous examples, $y_c = 1$. In that case, we aim to reduce the communication rate between probes and bastion for normal examples.

$$y_{ci} = \begin{cases} 0, & \text{if } P_{ei} - P_{eb} < \lambda \quad and \quad y = 1 \\ 1, & \text{if } P_{ei} - P_{eb} \geqslant \lambda \quad or \quad y = 0 \end{cases} \tag{9}$$

$$y_d = \begin{cases} 1, & \text{if } y_{ci} = 0, \quad \forall y_{ci} \in Y_c \\ \hat{y}_b, & \text{if } y_{ci} = 1, \quad \exists y_{ci} \in Y_c \end{cases} \tag{10}$$

V. EXPERIMENTS AND RESULTS

This section describes the experimental design we used to build the DAD framework. We provide a detailed analysis of the experiments; we start with adapting the SynCAN data for a multi-view input. The second part is to train the sub-networks jointly. The third part is calibrating the different thresholds to obtain good accuracy detection while reducing the communication overhead. Finally, we discuss the communication cost reduction and compare the naive and the learned-based strategy and further contribution of this work.

A. Dataset

SynCAN dataset is synthetic data proposed in [13]; the data set consists of 10 different message IDs, each with varying amounts of signals per ID and different noisy time frequencies. The total number of signals is 20. The data is created in such a way that it is similar to real CAN traffic. It contains physical values (signals), timestamps and IDs. We use a training data set of about 16.5 hours and a test data set of about 7.5 hours of CAN traffic. We evaluate our model on the following attack type : Plateau attack, Continuous change attack, Playback attack, Suppress attack, Flooding attack. Most attack in the test data last 4s.

As the SynCAN dataset's capture was mainly done by indexing only the different messages' IDs, we do not have the information of the ECUs related to each ID. Thus we cannot split into different views according to a semantic locality pattern as shown in the architecture presented in "Fig. 3". We split the IDs according to each ID's frequency, obtaining a balanced rate number of messages at each probe. We obtain two views referring to two probes S_1 and S_2, and each probe monitors five IDs. The following step is to create the sequence matrix for T_1 and T_2. We obtain two datasets with an input size is (100*12), (100*8), where 100 is the length of the obtained sequence and 8 and 12 are the numbers of signals monitored at each probe S_1 and S_2 respectively.

For the anomalous examples labelling we choose to set $y_1 = y_2 = y_b = 0$ if $y_1 = 0$ or $y_2 = 0$. This labelling aims to encourage the sub-network to learn more about the certainty of sending the message by backpropagating the error relative to the all system state while having only the information on the local sequence. We split the dataset into 80% training set and 20% for testing.

B. Training settings

We train our DAD model for 100 epochs and set the batch size to 100. We used the ADAM optimizer for the gradient descent, and we put the learning rate to 0.0001 (decayed by a factor of 0.1 after 50 epochs). As for the TCN parameters on each probe S_1, S_2, we use one dilated factor d=2 in the TCN. We use Keras and TensorFlow for the training under Nvidia Tesla M40 GPU. We set the size of the joint layer V to 100 as it shows the best trade-off between the number of parameters and the model's performance in the F1 measure.

C. Results

We start by showing and assessing the results concerning the accuracy of detecting anomalies at each probe. We evaluate DAD model performance on the different probes using F1-measure, Precision and Recall metrics.

From Table I we observe a clear gap between the results on the global classifier D_B than D_1 and D_2. First, the observation of the results on y_B concludes that hierarchical learning from the feature vector representation v_1, v_2 obtained from D_1 and D_d is effective. The sub-network sequence modelling task is well learned, and it preserves the information in the feature vector V with a reduced dimension where $|V| < |T_1| + |T_2|$,

TABLE I
F1 PERFORMANCE ON DAD MODEL HIGHLIGHTS THE DIFFERENCE BETWEEN THE PREDICTION ON THE LOCAL PROBES (S_1, S_2) AND THE GLOBAL PROBE BASTION (S_b)

attack	probe	precision	recall	F1
continues	B	0.995	0.998	**0.997**
continues	$S1$	0.974	0.894	0.932
continues	$S2$	0.953	0.825	0.884

so D_B Discriminator can differentiate between normal traffic and attacks. On the other hand, we can explain the under-performance of the sub-network results D_1 and D_2 compared to D_B with the lack of a general view of the system and more correlation dependencies are needed. Thus, local pattern modelling is not sufficient to get good results in detecting specific attacks.

TABLE II
COMMUNICATION GAIN RESULTS

	Naive threshold selection			Learned communication		
	F1	Ne	gain	F1	Ne	gain
Continues	0.997	0.898	0.102	0.997	0.668	**0.332**
Plateau	0.990	0.915	0.09	0.990	0.793	**0.207**
Suppress	0.998	0.972	0.028	0.998	0.752	**0.248**
Flooding	0.995	0.966	0.034	0.995	0.592	**0.408**
Playback	0.996	0.928	0.072	0.996	0.480	**0.520**

In table II, we observe more gain using the learned communication strategy compared to naive threshold selection. We keep the same performance as before introducing the optimisation flag e_i. The grid-search on threshold space aims to find the tuple (t_1, t_2) that satisfies the strict condition of $error = 0$ (error related to the optimisation process False-negative that could be detected if the v_i are sent to D_b). Indeed, the rate of communication optimisation (gain) can be more important if we neglect the error side effect, which directly impacts the DAD model's overall performance. In our case, the tradeoff between optimisation and the model's performance in detecting an attack remains clear. We cannot tolerate a reduction in the model performance for a computational reason.

VI. CONCLUSION

In-vehicle systems are getting more and more evolvement. This opens a new kind of architecture into in-vehicle systems. Building an intrusion detection system must adapt to those architectures and constraints. In this work, we formulated the integration of an IDS into future in-vehicle distributed architectures. We propose a framework DAD that fits the in-vehicle distributed architecture. DAD can capture local view patterns using the Temporal Convolutional Network (TCN)and send a reduced size feature vector to the system upper layer (bastion). Our framework also reduces the communication overhead brought by the intrusion detection system. Also, in this work, we set the e_i by manually searching threshold analysis through the t_i space. We introduced multiple-stage learning, where the communication overhead optimisation will be the second learning stage of the model. Nonetheless, future work will integrate recent works on the estimation of prediction uncertainty [15] to design robust and trustworthy machine learning models with epistemic and aleatoric uncertainties.

REFERENCES

[1] Karopoulos, G., Kampourakis, G., Chatzoglou, E., Ramos, H. & V Kouliaridis Demystifying in-vehicle Intrusion Detection Systems: A survey of surveys and a meta-taxonomy. ELECTRONICS. 11, 1072 (2022)

[2] Peng, Z., Luo, M., Li, J., Liu, H. & Zheng, Q. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. Proceedings Of The Twenty-Seventh International Joint Conference On Artificial Intelligence, IJCAI-18. pp. 3513-3519 (2018,7)

[3] Ding, K., Li, J., Bhanushali, R. & Liu, H. Deep Anomaly Detection on Attributed Networks. Proceedings Of The 2019 SIAM International Conference On Data Mining (SDM). pp. 594-602 (2019)

[4] Marcos Alvarez, A., Yamada, M., Kimura, A. & Iwata, T. Clustering Based Anomaly Detection in Multi-View Data. Proceedings Of The 22nd ACM International Conference On Information & Knowledge Management. pp. 1545-1548 (2013)

[5] Ji, Y., Huang, L., He, H., Wang, C., Xie, G., Shi, W. & Lin, K. Multi-view Outlier Detection in Deep Intact Space. 2019 IEEE International Conference On Data Mining (ICDM). pp. 1132-1137 (2019)

[6] Wang, C., Zhao, Z., Gong, L., Zhu, L., Liu, Z. & Cheng, X. A Distributed Anomaly Detection System for In-Vehicle Network using HTM. IEEE Access.PP pp. 1-1 (2018,1)

[7] Li, Y., Yang, M. & Zhang, Z. A Survey of Multi-View Representation Learning. IEEE Transactions On Knowledge And Data Engineering. 31, 1863-1883 (2019)

[8] Xu, C., Tao, D. & Xu, C. A Survey on Multi-view Learning. ArXiv. abs/1304.5634 (2013)

[9] Srivastava, N. & Salakhutdinov, R. Multimodal Learning with Deep Boltzmann Machines. J. Mach. Learn. Res.. 15, 2949-2980 (2014,1)

[10] Feng, F., Wang, X. & Li, R. Cross-Modal Retrieval with Correspondence Autoencoder. (Association for Computing Machinery,2014)

[11] Wang, W., Arora, R., Livescu, K. & Bilmes, J. On Deep Multi-View Representation Learning. Proceedings Of The 32nd International Conference On International Conference On Machine Learning - Volume 37. pp. 1083-1092 (2015)

[12] Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K. & Darrell, T. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions On Pattern Analysis And Machine Intelligence. 39, 677-691 (2017)

[13] Hanselmann, M., Strauss, T., Dormann, K. & Ulmer, H. CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data. IEEE Access. 8 pp. 58194-58205 (2020)

[14] Bai, S., Kolter, J. & Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. CoRR. abs/1803.01271 (2018)

[15] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U., Makarenkov, V. & Nahavandi, S. A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. Inf. Fusion. 76, 243-297 (2021,12)

[16] Gherbi, E., Hanczar, B., Janodet, J. & Klaudel, W. Deep Learning for In-Vehicle Intrusion Detection System. Neural Information Processing - 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18-22, 2020, Proceedings, Part IV. 1332 pp. 50-58 (2020)