WIP: AMICA: Attention-based Multi-Identifier model for asynchronous intrusion detection on Controller Area networks

Natasha Alkhatib*, Lina Achaji*, Maria Mushtaq, Hadi Ghauch, Jean-Luc Danger
Télécom Paris, IP Paris, Palaiseau 91120, France
Université de Lorraine, CNRS, Inria, LORIA, Nancy 54000, France
Stellantis Group, Technical center of Velizy 78140, France
{natasha.alkhatib, maria.mushtaq, hadi.ghauch, jean-luc.danger}@telecom-paris.fr
lina.achaji@stellantis.com, lina.achaji@inria.fr

***Abstract*—The adoption of external connectivity on modern vehicles and the increasing integration of complex automotive software paved the way for novel attack scenarios exploiting the vulnerabilities of in-vehicle protocols. The Controller Area Network (CAN) bus, a widely used communication network in vehicles between electronic control units (ECUs), therefore requires urgent monitoring. Predicting sophisticated intrusions that affect interdependencies between several CAN signals transmitted by distinct IDs requires modeling two key dimensions: 1) *time dimension*, where we model the temporal relationships between signals carried by each ID separately 2) *interaction dimension* where we model the interaction between IDs, i.e., how the state of each CAN ID affects the others. In this work, we propose a novel deep learning-based multi-agent intrusion detection system, AMICA, that uses an attention-based self-supervised learning technique to detect stealthy in-vehicle intrusions, i.e., those that that not only disturb normal timing or ID distributions but also carried data values by multiple IDs, along with others. The proposed model is evaluated on the benchmark dataset *SynCAN*. Our source code is available at: https://github.com/linaashaji/AMICA**

***Index Terms*—CAN, intrusion detection, Controller Area Network, in-vehicle network, deep learning**

I. Introduction

With the increasing complexity and connectivity of modern vehicles, the expanding threat landscape of the in-vehicle network (IVN) which connects various electronic control units (ECUs) is raising concerns. A range of potential security risks can compromise the safety and functionality of a vehicle putting the life of drivers and passengers in danger. By connecting to the vehicles, hackers are in fact managing to discover and exploit the IVN systems' vulnerabilities and launch disruptive cyberattacks.

*These authors contributed equally to this work

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23033
www.ndss-symposium.org

Despite being widely deployed in cars today, the Controller Area Network (CAN) technology, responsible for delivering reliable communication between several ECUs lacks important security mechanisms mainly authentication and encryption. To mitigate the severe aftermath, numerous intrusion detection approaches have been proposed to detect attacks against IVN. In such settings, observations of CAN network packets are sent sequentially to a detector that is tasked with detecting threats, particularly those that may not be caught by other security measures, such as zero-day vulnerabilities or insider attacks.

Unfortunately, most available approaches for intrusion detection don't handle the challenging structure of the CAN protocol. A suitable IDS for CAN must take into consideration its challenging message transmission mechanism, i.e only a single message can be transmitted at any point in time. In fact, some intrusions can only be detected by monitoring the interdependencies of several transmitted signals asynchronously. To overcome the shortcomings of previous approaches in the context of CAN IDS, we present **AMICA**, an attention-based IDS, that monitors asynchronously transmitted signals carried by several CAN Identifiers (ID) in the CAN. Inspired by the success of BERT model in detecting in-vehicle intrusions and the outstanding performance of Transformers in addressing multi-agent problems, we propose a BERT-based IDS that predicts intrusions on CAN by modeling two key dimensions: 1) ***time dimension***, where we model the temporal relationships between signals carried by each ID separately 2) ***interaction dimension*** where we model the interaction between IDs, i.e., how the state of each CAN ID affects the others.

II. Related Work

Techniques based on unsupervised deep learning have been widely used for detecting known and unknown intrusions on the CAN bus [1], [2]. Despite their efficiency, most of these works consider detecting intrusions on signals transmitted by each ID separately without taking into consideration their interdependencies, thus incapable of detecting sophisticated attacks. Few works have addressed this challenging problem such as [5] who introduced separate LSTM modules for each

CAN ID followed by a shared autoencoder-based module "CANet" tailored to work on the signal space of CAN data. However, there are still some limitations to using LSTM for modeling sequential data. First, although LSTM can capture the sequential information by the recurrence formula, it cannot make each element in a sequence encoding the context information from both the left and right context. However, it is crucial to observe the complete context information instead of only the information from previous steps when detecting malicious attacks based on CAN messages.

To address the existing limitations of LSTM-based models, researchers have started to leverage the Bidirectional Encoder Representations from Transformers (BERT) [4] for anomaly detection. Recently, Alkhatib et al. [10] proposed CAN-BERT, an IDS based on the BERT model tasked with monitoring the sequence of transmitted IDs through time and detecting intrusions on CAN bus by explicitly encoding the common patterns shared by all CAN ID normal sequences. However, their proposed model only regards the timing of each ID and/or the sequential nature of IDs and cannot detect attacks that do not disrupt normal timing or ID distributions.

Devising a multi-agent intrusion detection system is challenging since the interrelations between asynchronous CAN signals transmitted by distinct agents (IDs) are complex. To overcome the challenges of building such systems, Transformers have been widely adopted. In fact, Arnab et al. [6] and Gedas et al. [9] proposed a spatio-temporal model for video classification tasks. Achaji et al. [7] and Ye et al. [8] proposed a transformer-based multi-agent model that takes into account the interactions between several agents with respect to time and space for multi-agent trajectory prediction.

III. Proposed framework: AMICA

In this section, we describe our proposed model, AMICA, which tackles the asynchronous nature of the CAN protocol, in which messages are sent by various identifiers at different times. The different elements of the model are described as follows:

A. Input Formulation

Let $I = \{A_1, A_2, \ldots, A_N\}$ be the set of all possible CAN IDs. The input to the model, denoted by X, is composed of all the messages sent by different IDs in a temporal window of horizon T. We define the input as $X = \{X_1, X_2, \ldots, X_N\}$, where $X_i = \{x_{i,t_1}, \ldots, x_{i,t_j}, \ldots, x_{i,t_M}\}$ represents A_i's ordered set of message payloads, i.e. signals, transmitted during the horizon T, $t_j < T$ is the message global timestamp with respect to T, and N is the total number of IDs.

B. Temporal Module

1) Temporal Embedding: Message $x_{i,t} \in R^{d_i}$ transmitted at time t and carried by ID A_i is fed into its corresponding Temporal-based module TM_i. Each module TM_i has a different set of weights that will be optimized separately. Since different IDs can transmit various amounts of signals, all messages will be projected into a $d-$dimensional space via a single linear layer where $\bar{x}_{i,t} = W_i x_{i,t} + b_i$. $\bar{x}_{i,t}$ represents the projected signal, $W_i \in R^{d \times d_i}$ are the weights, and $b_i \in R^d$ represents the bias. Transformer models have no notion of time when computing attention for each of the input's elements. Usually, Transformer embedding layers are coupled with a positional encoding layer that will inject the timestamp of each input message $x_{i,t}$ when fed to its associated module TM_i. However, since messages are sent through asynchronous transmission, the input X composed of ordered X_i sets is an unordered set of $x_{i,t}$ signals. Thus, we employ a global positioning encoding layer that will encode the relative position of a message $x_{i,t}$ with respect to the input set X. The global positional encoding will follow the same sinusoidal formulation as in [3] and will be a function of the global timestamp t, $\tau(t)$. As opposed to the local positional encoding, a global positional encoding layer can be helpful to make the model learn the relative temporal dependencies between signals transmitted by different CAN IDs. The temporal embedding $\bar{x}_{i,t}$ will be aggregated by $\tau(t)$, $\bar{x}_{i,t} = \bar{x}_{i,t} + \tau(t)$.

2) Temporal Encoder: The Temporal encoder is a bidirectional model that uses the scaled dot-product self-attention layer proposed by [1]. Unlike, LSTM-based solutions [5] that leverage left-to-right temporal conditioning, bidirectional models are strictly more powerful [4]. The self-attention layer is an operation over the queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$) vectors:

$$\begin{aligned} Attention(Q, K, V) &= softmax(QK^T/\sqrt{D} + M)V \\ &= AV \end{aligned} \quad (1)$$

$\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the parametric linear projections of the input embedding vector $\overline{X}_i$. In the Temporal Encoder case, the attention weights A denote the relative score given to each time step compared to the other time steps for each ID A_i. Since the number of transmitted messages carried by each ID is dynamic over time, we padded the input sequence to a fixed number of messages and then applied a corresponding padding mask M to the $Softmax$ function presented in (1). In addition to the attention layer, the temporal encoder is also composed of a point-wise feed-forward (Pff) and normalization layers (LN) presented by the following equations:

$$\overline{X}_i = LN(\overline{X}_i + Attention(\overline{X}_i, \overline{X}_i, \overline{X}_i)) \quad (2)$$

$$\overline{X}_i = LN(Pff(\overline{X}_i) + \overline{X}_i) \quad (3)$$

C. Interaction Module

The interaction module is responsible for encoding the interdependencies of signals carried by different IDs. In contrast to the spatio-temporal and multi-agent systems [6], [7], the signals are transmitted asynchronously in the CAN bus, i.e., it is not possible to catch the correlations between signals with different IDs at the same time. Thus, we build an encoder that takes as input an ordered set of signals, denoted as X_Δ, and which correspond to a set of encoded signals with different IDs in a range of horizon ΔT, where ΔT is a hyperparameter

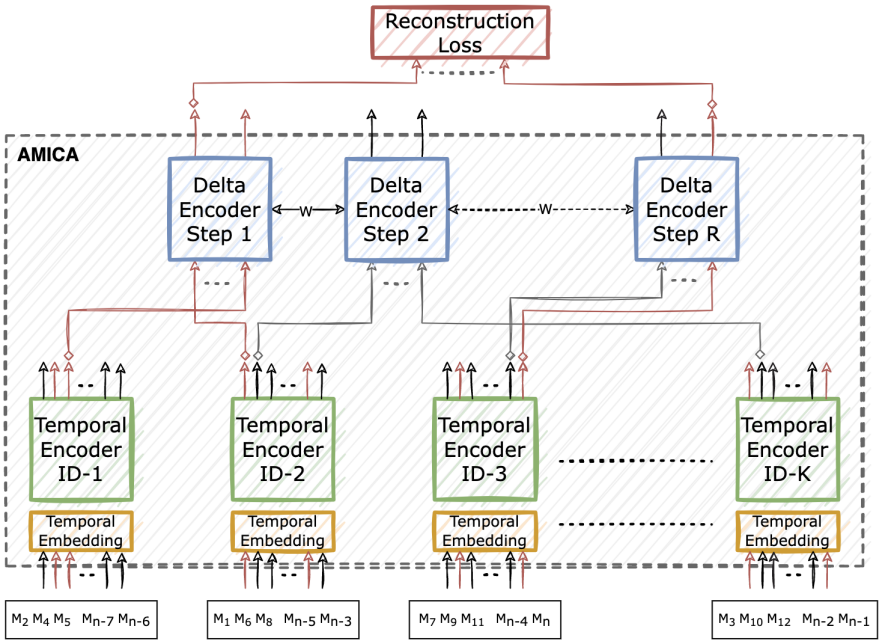

Fig. 1. Overview of the AMICA model architecture. A sequence of ordered CAN messages (signals) M_t are fed to their corresponding temporal ID encoder. In each sequence, a pre-defined ratio of the input messages is masked (colored in red). After being processed by their corresponding temporal ID encoder, the Delta encoder receives them and outputs the encodings of the previously masked messages in a fixed time window.

$\in [1, T]$. The ratio $R_\Delta = T/\Delta T$ represents a trade-off between the model interaction capabilities and the model complexity. For instance, $R_\Delta = 1$ means that the model can calculate a correlation between all signals transmitted during **T** timestamps. However, since the complexity of attention-based modules is quadratic in terms of sequence length, increasing R_Δ can decrease the model complexity from $c = O(T^2)$ in the ultimate case to $c = O(R_\Delta \cdot \Delta T^2) = O(T \cdot \Delta T) \leq O(T^2)$ in general cases.

1) Delta Embedding: Similar to the temporal embedding in Section (III-B1), we have applied temporal encoding with the purpose of injecting a notion of time to the later attention modules. However, the temporal encoding in this module is a local temporal encoding $\tau(p)$ representing the relative position p of a signal w.r.t the local X_Δ input. Additionally, a second encoding that reflects the ID encoding is implemented in this module. It injects an encoding $\tau(i)$ based on the message carried by the A_i for each message in X_Δ. The output embedding will be formulated as $\bar{x}_{\Delta_{i,p}} = W x_{\Delta_{i,p}} + \tau(i) + \tau(p)$, where W are the weights of the linear projection layer.

2) Delta Encoder: The delta encoder has the same architecture and associated equations of the temporal encoder presented in section (III-B2) following equations (2), (3). Contrary to the Temporal Module that has N separate Temporal Encoders, the Delta encoders will all share the same trainable weights independently of the temporal window to which X_Δ belongs.

IV. Training Procedure

A. Forward Pass

The detection of stealthy attacks against CAN necessitates monitoring large window sizes of CAN messages (signals). However, large input sequences have always been a major problem for various deep learning models resulting in vanishing gradient for LSTMs and quadratic time increase for Transformers. To mitigate this issue, for each time window w, we first feed the signals that are handled by their respective ID to their corresponding temporal encoder, i.e., although if a huge time lapse occurs between same-ID signals, the generated temporal encodings contain the signals' contextual representations. Once all encodings are obtained, they are fed to the delta encoder which iteratively processes and outputs the masked inputs for each interval $\Delta_T << w$.

B. Backward Pass

Although the iterative process handled by the delta encoder is time-consuming when training, it is advantageous for model optimization during backpropagation. In fact, after each Δ_T, the model's loss is backpropagated and the model's learnable parameters are updated. For instance, if $\Delta_T = 250$ and $w = 5000$, we will perform 20 backpropagations per batch.

To train the AMICA model using self-supervised learning techniques, we leverage the masked language modeling (MLM) objective function, proposed originally for training BERT [4]. The MLM objective consists in masking a percentage of the input sequence at random, and then predicting the masked signals using the output representations. The masked signals will be forecasted by leveraging contextual

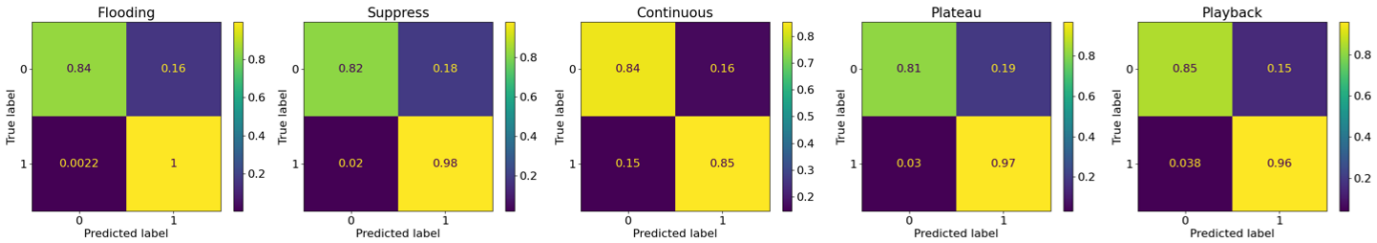

Fig. 2. Normalized confusion matrices representing the performance of AMICA when detecting diverse attacks on CAN bus for a given threshold ϕ.

representation from non-masked input signals, thus the bi-directional capabilities of the model. For the AMICA model, we did not embed the input payload values into a discrete vocabulary. Therefore, the masked signals are replaced by predefined mask-value that is chosen empirically. In addition, we consider the mean squared error loss function (depicted in Equation 4) as the regression-based function for training:

$$L = \sum_{i}^{M} MSE(\hat{y}_{[Mask_i]}, x_{[Mask_i]}) \tag{4}$$

where M is total number of masked signals, $\hat{y}$ is the prediction of the $i-th$ masked signal, and x is the original value of the $i-th$ masked signal.

V. Anomaly Score

The quadratic error between the overall masked inputs and their reconstruction are used to predict whether or not the sequence of CAN IDs with their respective signals is anomalous. The sequence is considered anomalous if the total sum of the masked signals' reconstruction error is above a fixed threshold. It's noteworthy to mention that the threshold is determined based on normal data as we are considering a self-supervised learning problem. We thus consider the following formulated labeling criteria:

$$Y = \begin{cases} 1 \text{ (abnormal)} & \text{if } \sum_{\mathbf{t=1}}^{\mathbf{w}} \mathbf{E}(\mathbf{s_t}) \geq \phi \\ 0 \text{ (normal)} & \text{otherwise} \end{cases}$$

where Y is the CAN sequence's label, $\mathbf{E}$ is the reconstruction error function, $\mathbf{s}_t$ is a corresponding masked signal, w is the considered sequence length, and ϕ is the determined threshold.

VI. Experiment & Results

To evaluate our proposed method, we used the benchmark synthetic CAN dataset ***SynCAN*** [11]. The dataset is composed of 10 different message IDs, each with different amounts of signals per ID and different noisy time frequencies. The data contains signals that are dependent on one or multiple other signals. The test data is composed of 6 subsets of equal time length: plateau, continuous change, playback, flooding, suppress and normal. We refer readers to [11] for further information. For training and evaluation, we use the pyTorch framework. All computations are performed on a Tesla V100 with 32 GB of installed physical memory (RAM). We train the network for 80 iterations with batch size 16. Every element in a batch is a series of 5000 consecutive messages. During a single iteration, a back-propagation is performed every 250 time steps ($\Delta_T = 250$) in order to update the network weights. We use the normal validation set to compute the anomaly detection threshold ϕ. We set ϕ to $\mu + 1.5\sigma = 0.001$ where μ is the mean reconstruction loss on normal data and σ is the standard deviation.

TABLE I
Performance of AMICA for different attack types

Attack	Recall	Precision	F1-score	AUC
Plateau	0.97	0.81	0.88	0.89
Continuous	0.74	0.85	0.79	0.85
Playback	0.96	0.78	0.86	0.91
Suppress	0.98	0.82	0.89	0.90
Flooding	0.99	0.87	0.93	0.92

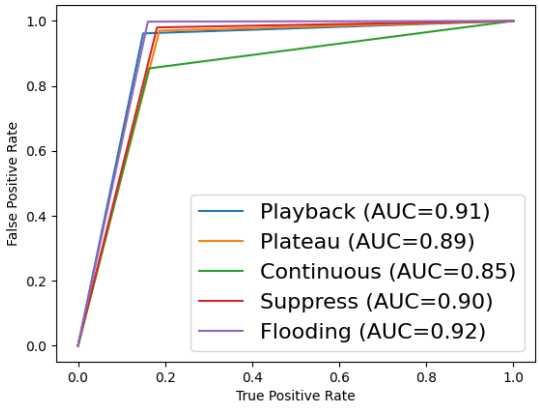

Fig. 3. ROC curves for diverse types of attack on CAN bus. As the corresponding AUC values approach 1, the AMICA model is thus assumed have a good measure of separability between the normal class and the different attack types.

We present our results in Table I, Fig. 2, and Fig. 3. We conclude several performance metrics from the visualized confusion matrices (seen in Fig. 2) to assess the performance of AMICA given the threshold ϕ including *precision*, *recall*, *F1-score* (depicted in Table I), in addition to the *ROC* curve along with *AUC* value (shown in Fig. 3). Our proposed approach is most prominent when detecting *Timing Transparent*

(T.T.) [12] attacks such as *suppress* (F1-score ≈ 0.9, AUC ≈ 0.9) and *flooding* (F1-score ≈ 0.93, AUC ≈ 0.92), as these attacks can hypothetically be identified by monitoring the frequency of the CAN IDs, i.e., the appearance of new IDs or disappearance of usually present IDs, irrespective of the values of their carried signals. The *Timing Opaque (T.O.)* [12] attacks including *plateau* (F1-score ≈ 0.88, AUC ≈ 0.89), *continous* (F1-score ≈ 0.79, AUC ≈ 0.85), and *playback* (F1-score ≈ 0.86, AUC ≈ 0.91) attacks are slightly less detectable as they are considered to be stealthier, i.e., they do not disrupt normal timing or ID distributions, and thus would not be detected by simply monitoring the frequency of the CAN IDs. In fact, these attacks solely manipulate the signal values and affect their correlations by overwriting the transmitted signals with constant values (*plateau*), slowly drifted values from their true value (*continuous*), or by playback of recorded time series of values of that same signal over a period of time (*playback*). Interestingly, our proposed model AMICA excelled in detecting both *T.T.* and *T.O.* attacks, whereas the other methods in the literature only fit in frequency-based IDSs or payload-based IDSs.

VII. Conclusion & Future Work

In this paper, we presented a novel deep learning-based multi-agent system, AMICA, for detecting intrusions on the widely deployed in-vehicle communication network CAN bus. As most of the intrusions can only be detected by monitoring long sequences of ordered CAN messages, we develop a model that overcomes this challenge by 1) incorporating information from different relation types between asynchronous signals and IDs, respectively in each stage of the model due to the *attention mechanism* 2) devising a suitable training process. Additionally, unlike the best-known methods so far, our approach is designed to detect all types of attacks on the CAN bus, particularly those that affect the interrelations between asynchronous signals of distinct CAN IDs. To evaluate our method, we leverage the SynCAN dataset and obtain promising results. Our model excels in detecting *T.T.* attacks, but it is also able to detect stealthier attacks with slightly less performance. The obtained results open many avenues for future research. It would be interesting to compute an anomaly threshold for each signal separately and to study whether this can lead to better model performance, particularly when detecting *T.T.* attacks. Finally, it is important to explicitly perform an ablation study on the model architecture along with an exhaustive hyperparameter tuning to enhance its effectiveness and complexity.

Acknowledgement

This work was carried out in the framework of the OpenLab “Artificial Intelligence” in the context of a partnership between INRIA institute and Stellantis company. We would also like to express our deep gratitude to the Connected Cars and Cyber Security academic chair (C3S) at Telecom Paris University for their generous support in funding our research work. This academic chair’s commitment to advancing research in the field of connected cars and cyber security has been invaluable in enabling us to pursue our work and achieve our research goals.

References

[1] Narasimhan, Harini, R. Vinayakumar, and Nazeeruddin Mohammad. ”Unsupervised Deep Learning Approach for In-Vehicle Intrusion Detection System.” IEEE Consumer Electronics Magazine (2021).
[2] Longari, Stefano, et al. ”CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network.” IEEE Transactions on Network and Service Management 18.2 (2020): 1913-1924.
[3] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).
[4] Devlin, Jacob, et al. ”Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 (2018).
[5] Hanselmann, Markus, et al. ”CANet: An unsupervised intrusion detection system for high dimensional CAN bus data.” Ieee Access 8 (2020): 58194-58205.
[6] Arnab, Anurag, et al. ”Vivit: A video vision transformer.” Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[7] Achaji, Lina, et al., ”PreTR: Spatio-Temporal Non-Autoregressive Trajectory Prediction Transformer,” 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 2022, pp. 2457-2464, doi: 10.1109/ITSC55140.2022.9922451.
[8] Yuan, Ye, et al. ”Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting.” Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[9] Bertasius, Gedas, Heng Wang, and Lorenzo Torresani. ”Is space-time attention all you need for video understanding?.” ICML. Vol. 2. No. 3. 2021.
[10] Alkhatib, Natasha, et al. ”CAN-BERT do it? Controller Area Network Intrusion Detection System based on BERT Language Model.” arXiv preprint arXiv:2210.09439 (2022).
[11] SynCAN Dataset https://github.com/etas/SynCAN
[12] Verma, Miki E., et al. ”Addressing the Lack of Comparability & Testing in CAN Intrusion Detection Research: A Comprehensive Guide to CAN IDS Data & Introduction of the ROAD Dataset.” arXiv e-prints (2020): arXiv-2012.