Anomaly Detection For In-Vehicle Communication Using Transformers

Victor Cobilean [1], Harindra S. Mavikumbure [1], Chathurika S. Wickramasinghe [1], Benny J. Varghese [2], Timothy Pennington [2] and Milos Manic [1], *Fellow IEEE*
[1]Department of Computer Science, Virginia Commonwealth University, Richmond, USA
(cobileanv, mavikumbureh, brahmanacsw)@vcu.edu, misko@ieee.org
[2] Energy and Environmental Science & Technology, Idaho National Laboratory, Idaho Falls, ID, USA
(benny.varghese, timothy.pennington)@inl.gov

IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society | 979-8-3503-3182-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/IECON51785.2023.10311788

***Abstract*—With the advancements of modern vehicle infrastructures, vehicles are increasingly relying on the signals received from a vast number of sensors and electronic components. Wireless technologies enable communication between vehicles and infrastructure, but it also increase the vulnerability surface. Malicious actors can remotely disrupt the vehicle's normal behavior, causing vehicle damage or worse, putting human lives in danger. To address these challenges, this paper proposes a transformer neural network-based intrusion detection system (CAN-Former IDS) that predicts anomalous behavior within the CAN protocol communication. Previous work typically addresses the prediction over the sequence of the CAN IDs. In this paper, we will simultaneously analyze both the sequence of IDs and the message payload values. The advantages of our approach are: 1) fully self-supervised training, which does not require labeled data, 2) self learning interactions between input tokens without relying on hand-crafted features. The transformer neural network is trained to predict the next communication sequence and anomalous communication is identified by comparing the real sequence to the predicted expected sequence. We evaluated our approach using a publicly available data set known as survival analysis data set, containing CAN communication from three different cars.**

***Index Terms*—Anomaly Detection, In-Vehicle Communications, Deep Learning, Transformer**

I. Introduction

Modern vehicles are equipped with an increased number of electronic parts and complex sensor systems for improving driver comfort by automating numerous tasks. These advancements have also allowed the vehicles to be incorporated into the concept of the "smart city", in which vehicles are interconnected and communicate with other vehicles and infrastructure components [1]. All of these technologies offer advantages to the end user in terms of improved traffic efficiency and safety, but they also present new opportunities for attackers to gain access to the internal communication bus of the vehicle and interfere with its normal operation [2].

The focus of this paper will be developing a deep learning-based Intrusion Detection System (IDS) for detecting abnormal behaviors in the Control Area Network (CAN) protocol, which is one of the most well-established and widely used protocols for in-vehicle communications. The CAN protocol uses a broadcasting architecture to allow multiple electronic control units (ECUs) to send messages across the bus. However, this architecture may expose vulnerabilities such as attackers gaining access to the communication bus to manipulate critical components including brakes and engine, posing a serious threat to the safety of the driver and other traffic participants [3].

Deep learning has demonstrated impressive results in detecting intrusions and is widely applied in securing critical infrastructure components. Neural networks are highly effective in learning from large amounts of data and extracting versatile features to describe the normal behavior of the system [4]. Furthermore, deep neural networks succeed at identifying interactions between components and finding meaningful representations of the inputs. Leveraging these advantages, we aim to apply neural networks to learn the sequential patterns of regular communication within the CAN bus.

In this paper, we will focus on the problem of intrusion detection in the CAN bus as a sequence-to-sequence (Seq2Seq) problem. Our goal is to use deep learning to predict the message sequence within the communication based on the provided communication sequence using a transformer neural network. The transformer architecture is the current state-of-the-art in sequence modeling, with remarkable results in domains such as natural language processing [5], speech recognition [6] etc. The transformer architecture uses the attention mechanism, allowing it to learn interdependencies across the entire input sequence. Advantages of this architecture over recurrent architectures is its ability to be efficiently parallelized, resulting in more effective training/inference and better performance in capturing long-term dependencies [5].

The main advantages of this approach are:

1) No need for any feature extraction methods, the model will learn the token representations by converting CAN communication of IDs and message payload into embedding.
2) The model doesn't require any labeled data for training. The IDS is fully trained through self-supervision.
3) High detection rate on survival analysis data set that contains data about three cars (Hyundai Sonata, Kia Soul and Chevrolet Spark).

The rest of the paper is structured as follows. The background and related work is presented in Section II. Section III describes the transformer architecture used for anomaly detection in the CAN bus. Section IV presents and discusses

Fig. 1. CAN frame

the experimental results. Finally, Section V concludes the paper.

II. Background and Related Work

A. CAN protocol

In this section we will present the technical part of the CAN communication protocol.

Controller Area Network (CAN) is a bus protocol that is widely used in in-vehicle communication, because of the low complexity, deterministic behavior, robustness against the electromagnetic interference and effective scalability [7]. CAN bus communications is based on two wire bus topology and is multi master broadcasting communication where every node can initiate communication and receive message in the network. The collision of the data is avoided by a bit-wise arbitration on the ID of the sender ECU. While the CAN bus can handle a large number of messages, it lacks mechanisms for authentication, encryption, and network segmentation. As a result, these features directly expose vulnerabilities in the CAN protocol, that make it easy for messages to be spoofed and sniffed [8].

The structure of CAN frame is presented in fig.1 and it is contains following components [7]:

1) **SOF (Start of Frame)** - signals the beginning of the frame.
2) **Arbitration Field** - contains the message ID and RTR (Remote Transmission Request) bit. RTR state is responsible for identifying the type of the frame: data or remote frame.
3) **Control Field** - represents the length of the data.
4) **Data Field** - contains the actual payload that the node wants to send (0-64 bits).
5) **CRC Field** - is used for error detection and for checking the integrity of the sent message
6) **Ack Field** - is used to acknowledge that a message was successfully received.
7) **EOF (End of Frame)** - signals the ending of the CAN frame.

B. Anomaly detection in CAN protocol

Because of the vulnerabilities described in previous section, detecting intrusions in the CAN bus is an important topic in the research community. In this section we will present several solutions that have been proposed in the literature. The authors of [9] are describing five feature categories that can be used for determining attacks on CAN bus communication: frequency-based, payload-based, signal-based, physical side-channel and others.

The authors of [10] used ResNet Autoencoder for unsupervised anomaly detection. The reconstruction error of the Autoencoder was used for determining if the the window is anomalous or not. Long Short Term Memory neural network was used in [11] for predicting next word for for each sender in the bus network, the words that are substantially different from the predicted words are flagged as anomaly. A GAN based anomaly detection system is proposed in [12], the anomaly detection is performed on binary images of the encoded sequence ID.

We identified several works that used transformer and attention based model for anomaly detection. In [13], a Bi-Directional GPT was used to for determining anomalies within the sequence of the IDs. During the inference, if the negative log-likelihood is higher than a pre-specified threshold then an anomaly is detected. In [14], a BERT architecture was used for predicting the masked IDs in the sequence, if the model is able to predict the masked ID than the sequence is normal. The authors of [15] presented two transformer architectures for classifying anomalies using a single message and a sequence of IDs. In this paper, we will extend the capabilities of the transformer architecture to perform the analysis on the sequence of IDs and payloads concomitantly.

III. Transformer for CAN Communication (CAN-former IDS)

In this section, we will present the transformer model for anomaly detection in CAN bus communication. We will use the decoder transformer for predicting the next token $[T_1, T_2, T_3 \cdots T_N]$ given the input context $[T_0, T_1, T_2 \cdots T_{N-1}]$. An example of transformation of a single CAN message into tokens is presented in Fig.2. An anomaly will be detected in case the differences between the predicted sequence and the actual received sequence will reach a certain threshold.

The input sequence length will be determined by the number of messages that will represent the context of the prediction. One message consists of 9 tokens (ID token + 8 bytes payload tokens), so if the maximum prediction length is 10 message, the length of the tokens sequence will be 90.

For creating the input sequence each CAN ID and its afferent sequence of eight bytes will be treated as separate token. The first layer is an embedding layer, which represents a look up table that will transform the input token to the embedding size ($Embd\_size$) defined as a hyper-parameter of the architecture. The size of the look table will be $L \times Embd\_size$ where:

$$L = 256 + UniqueIDs + None\_token \tag{1}$$

where: 256 represents the numbers of all possible transmitted bytes + the number of unique IDs in the training data + token that will be used as none token. Because we are not giving the model the payload size (Control Field value), we are inferring this property by filling the rest of the sequence in the message with the special token, so every message is composed of the same number of tokens.

In order to add information about the position of the token within the sequence, a positional embedding vector will be

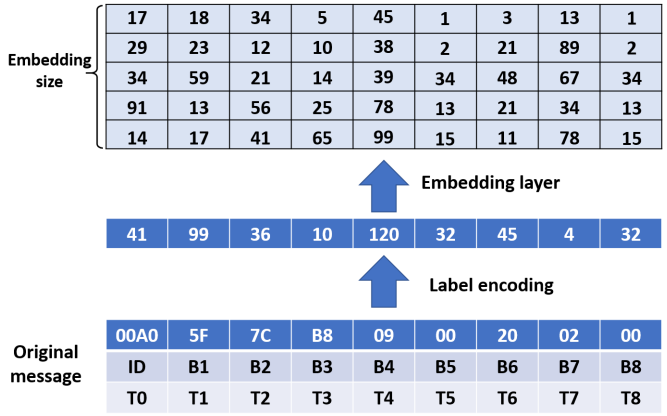

Fig. 2. CAN message transformation

added to each token. The positional embedding will be learned during the training process.

The architecture consists of repeating transformer blocks that consist of the following elements:

1) **Masked multi-head attention layer** - this layer is responsible for learning the interdependencies between the elements of the input sequence [5]. It implies linearly transforming the sequence into a query(Q), key(K), and value(V). The attention scores are obtained using the dot product between Q and K. A mask is applied to the obtained scores, so we allow tokens to have access to the information from the previous tokens, but not the future ones that need to be predicted. The normalized weights obtained using the softmax function will be used for computing the weighted sum over the V [5].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

For capturing multiple dependencies at the same time, multi-head attention is used, each head has a separate set of learnable parameters, and the result of each attention head is concatenated. A linear transformation is applied to the concatenated vector for obtaining the output value of the attention layer.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (3)$$

where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

2) **Add and Normalization layer** - adding residual connections [16] and layer normalization [17] for facilitating neural network optimization.
3) **Feed Forward layer** - represents a linear transformation followed by a non-linear activation function σ:

$$f(x) = \sigma(Wx + b) \quad (5)$$

We use the ReLU activation function.

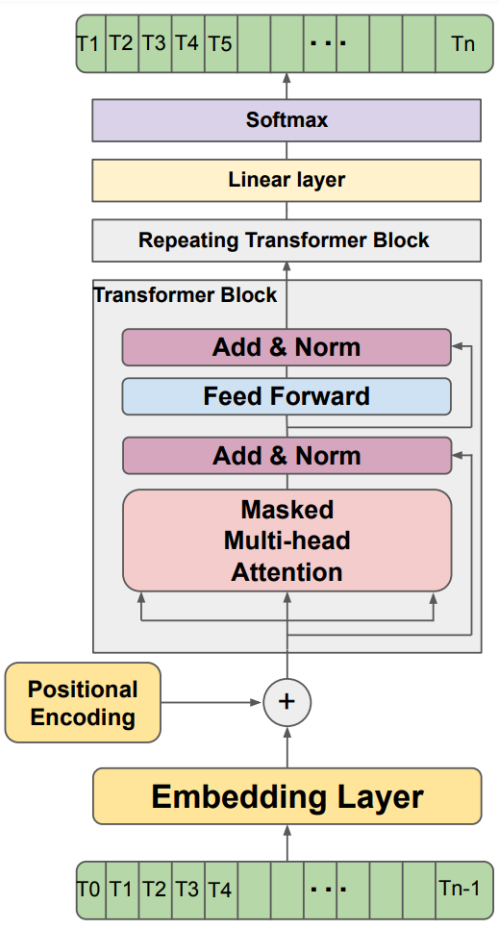

Fig. 3. Transformer architecture

After the last transformer layer is a linear transformation layer that has an output size equal to the number of elements in the look-up table. The last layer represents a softmax layer for getting the probability distribution over all possible tokens. The neural network is trained using the cross entropy loss function for the prediction of the next token in the sequence:

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \quad (6)$$

During inference time, given an input sequence, we try to predict the payload of the last message, generating the last eight tokens of the sequence. The input sequence is $[T_0, T_1, T_2 \cdots T_{N-8}]$ and we are generating the probabilities over all possible tokens for T_{N-7}. If the actual token T_{N-7} is not in the top k predictions, then the anomaly score is increased by 1. If the anomaly score is more than the set up threshold then the sequence is identified as anomalous. To determine the threshold, we can use a holdout subset from the training to perform a search for the best threshold that reduces the misclassification of the holdout subset.

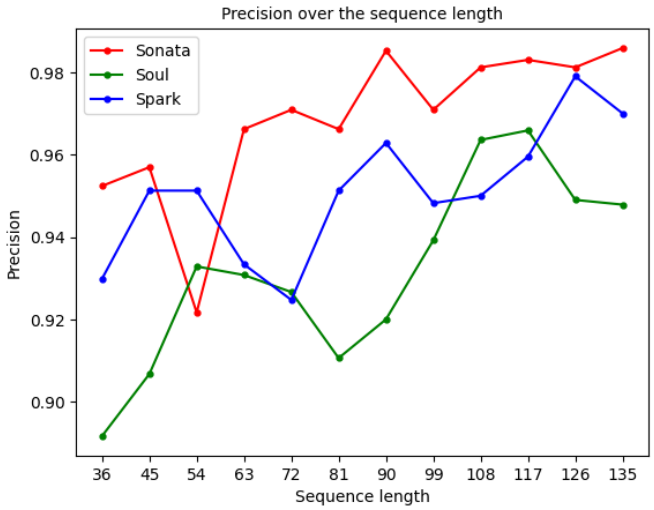

Fig. 4. Precision over the sequence length

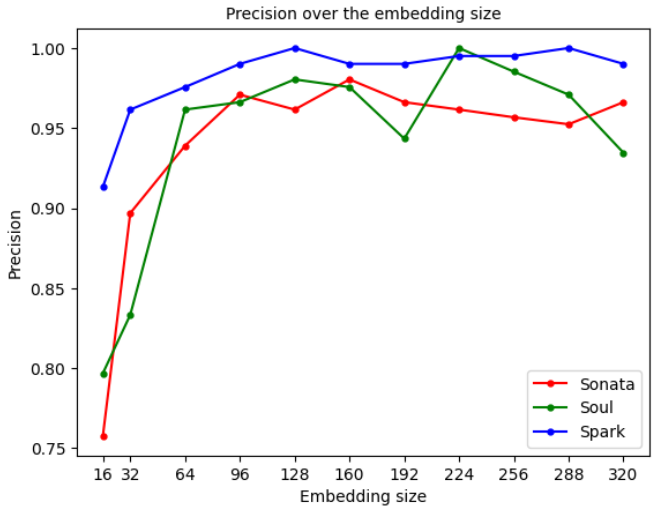

Fig. 5. Precision over the size of the embedding

IV. Experiments

In this section, we will present the experiments for testing our approach and will provide a comparison to other intrusion detection algorithms for CAN bus communication.

We will test our approach on survival analysis data set [18]. This data set contains data for three car models: Hyundai Sonata, Kia Soul, and Chevrolet Spark. The authors of the data set performed 3 types of attacks for each car, but we will focus on investigating the detection of the malfunction attack, because in this attack all the injected messages have IDs that were present in the training data. For other attacks an error will be raised since there is no embedding representation in the look up table for some random generated IDs during the attack.

In the Fig. 4 is presented the precision of the models for a specific length of the sequence for an embedding size equal to 32. We can observe that with a shorter context, the model is not able to predict with confidence the next tokens in the sequence, which results in higher false positives. The optimal length of the sequence should be around 11-15 messages (99-135 tokens).

In the Fig. 5 we have plotted the precision of the proposed architecture based on the embedding size. All the results are presented for a length of the sequence of 90 tokens. From the initial embedding size of 16 we notice a significant increase of the precision of the model up to an embedding size of 128. Afterwards the precision of the model doesn't change much. Furthermore, for larger embedding sizes we can notice a drop in the prediction performance more accentuated for the Sonata and Soul models. The performance drop is related to the over fitting of the model, so the optimal embedding is around 128. Higher embedding size is increasing the size of the model without bringing substantial performance improvements.

In the table I we are presenting the results of our model in comparison to other unsupervised algorithms. The best results were obtained for an embedding size of 128, and a sequence length of 126 tokens (14 messages). The number of heads in the attention layer is equal to 4 and the number of transformer blocks is equal to 10. We are using a dropout probability of 0.25 for avoiding overfitting. During the inference time, we are generating the payload of the last message (8 tokens) and we are checking for the best 15 predictions and the anomaly threshold is 3.

We are comparing our model to two other deep learning architectures: the ResNet Autoencoder and the Long-Short Term Memory (LSTM) Autoencoder. We are also comparing our approach to a graph-based anomaly detection approach [19]. The ResNet and LSTM Autencoder are trained to reconstruct the input window of the extracted features [10]. The G-IDCS [19] is using features extracted from the constructed graph of the CAN communication.

Because of the large size of the training data set, the transformer architecture performs the best on the Kia Soul. The large amount of messages allows our model to perform better (recall 1.0 and F1 score of 0.9873) than other models.

In comparison to the graph-based approach, our model has comparable recall, identifying most of the anomalous sequences for Hyundai Sonata and Chevrolet Spark. However, our model has a lower precision. The difference are related to the inclusion of payload values in the analyzed sequence, which introduces more noise and, as a result, higher rates of false positives. At the same time, because our architecture considers both the IDs and payload of the messages, it would be able to detect more stealthy attacks that involves payload manipulations. Furthermore, our approach is more flexible because it does not rely on any feature extraction method and instead relies on the neural network to learn the useful interactions on its own (Fig.6).

For a deeper understanding of the model inner workings, in the Fig.6 we present the visualization of the token embedding of Hyundai Sonata the using the T-SNE algorithm. It is possible to observe that ID's that are more frequently in the

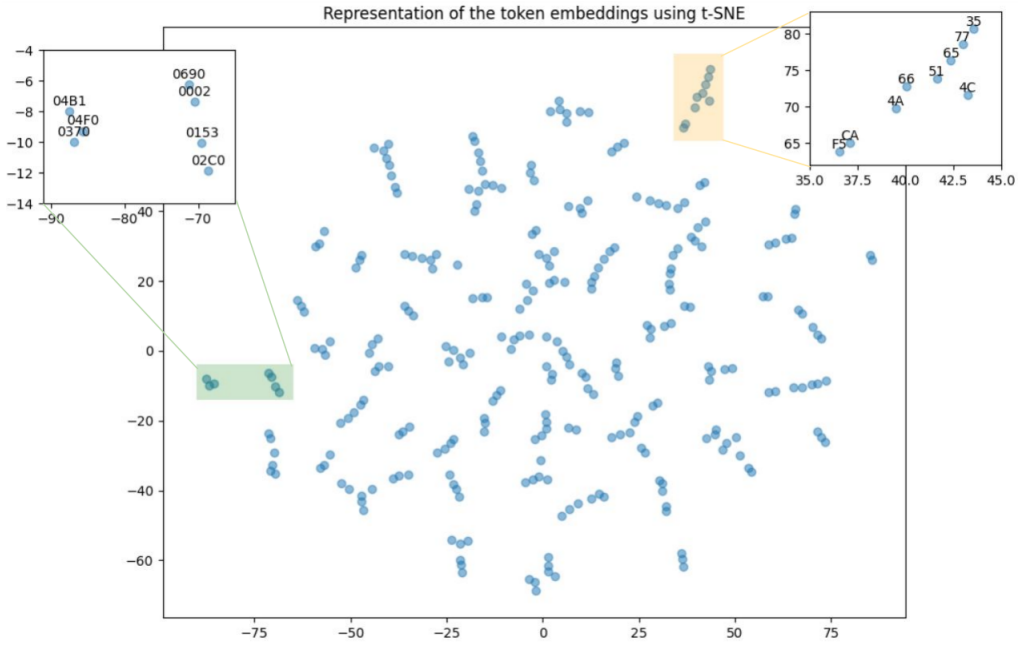

Fig. 6. Representation of tokens embedding using T-SNE

TABLE I
EXPERIMENTAL RESULTS

Car type	Model	Accuracy	Precision	Recall	F1
Kia Soul	**ResNet AE**	0.7032	0.6248	0.41	0.4951
	LSTM AE	0.9124	0.9614	0.7848	0.8641
	G-IDCS [19]	0.9815	0.9592	0.9903	0.9745
	CAN-Former	0.9975	0.9749	1.0	0.9873
Hyundai Sonata	**ResNet AE**	0.9697	0.9994	0.9273	0.9620
	LSTM AE	0.9974	0.9995	0.9942	0.9968
	G-IDCS [19]	0.9985	1.0	0.9964	0.9982
	CAN-Former	0.9835	0.9501	0.9911	0.9702
Chevrolet Spark	**ResNet AE**	0.5446	0.4888	0.8599	0.6233
	LSTM AE	0.5678	0.5042	0.8022	0.6192
	G-IDCS [19]	0.9975	0.9936	1.0	0.9968
	CAN-Former	0.9948	0.9812	0.9972	0.9892

same sequence are mapped closer in the embedding space. An example are IDs $(0 \times 002, 0 \times 153, 0 \times 2C0)$ that are situated in the green sub-region of the representation. At the same time the payload tokens are mapped together (yellow sub-region) if the tokens have the similar position in the message payload, for example tokens $(35, 77, 65)$ are more likely to be last bytes in the payload sequence of the message.

V. CONCLUSION AND FUTURE WORK

In this paper we presented a transformer neural network architecture for anomaly detection in CAN bus communication. The transformer is trained using self-supervised learning to learn the normal sequence of CAN communication by predicting the next tokens given the input sequence of the tokens. The detection of anomaly is done by comparing the predicting sequence and the received sequence and if substantial differences are detected the sequence is flagged as anomalous. The main advantages of the model are that it does not require labeled attack data for learning and that it considers both ID and payload values at the same time. We tested the approach on the survival data set that has data for three different types of cars. Our model performed best on the Kia Soul, which has the largest data set. For other types of vehicles, the model detects the majority of anomalous sequences but has a lower precision due to the more noisy signal that are coming from the payloads.

Further research directions are testing the model on other attacks and testing the transfer learning capabilities of the architecture. Furthermore, we would like to test different types of encoding and tokenizations of the input sequence, so the model will be capable to encode unseen IDs based on IDs present in the training data. Developing an alternative embedding pipeline for adding more information about different tokens (ID or payload token type) and relative positional information within the sequence.

ACKNOWLEDGEMENTS

The Department of Energy partly supported this work through the U.S. DOE Idaho Operations Office under Contract DE-AC07-05ID14517, and partly by the Commonwealth Cyber Initiative, an Investment in the Advancement of Cyber Research and Development, Innovation and Workforce Development (cyberinitiative.org).

References

[1] D. Swessi and H. Idoudi, "A comparative review of security threats datasets for vehicular networks," in *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2021, pp. 746–751.

[2] G. Karopoulos, G. Kambourakis, E. Chatzoglou, J. L. Hernández-Ramos, and V. Kouliaridis, "Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy," *Electronics*, vol. 11, no. 7, p. 1072, 2022.

[3] A. Martínez-Cruz, K. A. Ramírez-Gutiérrez, C. Feregrino-Uribe, and A. Morales-Reyes, "Security on in-vehicle communication protocols: Issues, challenges, and future research directions," *Computer Communications*, vol. 180, pp. 1–20, 2021, ISSN: 0140-3664.

[4] H. S. Mavikumbure, C. S. Wickramasinghe, D. L. Marino, V. Cobilean, and M. Manic, "Anomaly detection in critical-infrastructures using autoencoders: A survey," in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2022, pp. 1–7.

[5] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[7] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of Automotive Controller Area Network Intrusion Detection Systems," *IEEE Design & Test*, vol. 36, no. 6, pp. 48–55, Dec. 2019, Conference Name: IEEE Design & Test, ISSN: 2168-2364.

[8] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, "Ai-based intrusion detection systems for in-vehicle networks: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–40, 2023.

[9] M. E. Verma *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:2012.14600 [cs], Jan. 2022.

[10] C. S. Wickramasinghe *et al.*, "Rx-ads: Interpretable anomaly detection using adversarial ml for electric vehicle can data," *arXiv preprint arXiv:2209.02052*, 2022.

[11] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 130–139.

[12] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, IEEE, 2018, pp. 1–6.

[13] M. Nam, S. Park, and D. S. Kim, "Intrusion Detection Method Using Bi-Directional GPT for in-Vehicle Controller Area Networks," *IEEE Access*, vol. 9, pp. 124 931–124 944, 2021, Conference Name: IEEE Access, ISSN: 2169-3536.

[14] N. Alkhatib, M. Mushtaq, H. Ghauch, and J.-L. Danger, "CAN-BERT do it? Controller Area Network Intrusion Detection System based on BERT Language Model," in *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, ISSN: 2161-5330, Dec. 2022, pp. 1–8.

[15] T. P. Nguyen, H. Nam, and D. Kim, "Transformer-Based Attention Network for In-Vehicle Intrusion Detection," *IEEE Access*, vol. 11, pp. 55 389–55 403, 2023, Conference Name: IEEE Access, ISSN: 2169-3536.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[18] M. L. Han, B. I. Kwak, and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," *Vehicular communications*, vol. 14, pp. 52–63, 2018.

[19] S. B. Park, H. J. Jo, and D. H. Lee, "G-idcs: Graph-based intrusion detection and classification system for can protocol," *IEEE Access*, vol. 11, pp. 39 213–39 227, 2023.