CANBERT: A Language-based Intrusion Detection Model for In-vehicle Networks

Ebelechukwu Nwafor
Department of Computing Sciences
Villanova University
Villanova, PA, USA
enwafor@villanova.edu

Habeeb Olufowobi
Department Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, USA
habeeb.olufowobi@uta.edu

***Abstract*—Controller Area networks (CAN) provide a standard means of communicating across vehicular electronic units without a centralized computing unit or complex dedicated wiring. Despite the benefits offered by in-vehicle networks, CAN networks have been susceptible to network attacks such as replay, fuzzing, and denial of service attacks. In addition, the proliferation of internet-connected vehicles motivates the need to build a robust and secure vehicular network system. Deep learning-based language models such as Bidirectional Encoder Representations from Transformers (BERT) models have proven to produce remarkable results for natural language tasks. BERT models provide a deep understanding of the underlying semantics in textual data. In this paper, we propose CANBERT, a language-based intrusion detection model for CAN bus. We leverage the power of transformer models to provide the detection of malicious attacks on the CAN network. We provide a thorough analysis of our approach using a CAN dataset produced in a realistic driving scenario which consists of a combination of normal data and malicious data from various types of attack scenarios such as Denial of Service (DoS), fuzzy, and impersonation attacks. Our approach is able to detect all of the attacks with high precision and accuracy. In addition, we compare our method with other baseline models and state-of-the-art deep learning intrusion detection approach for in-vehicle networks.**

***Index Terms*—Transformer-based Model, Controller Area Networks, Intrusion Detection Systems**

I. Introduction

In-vehicle networks such as Controller Area Networks (CAN) have witnessed widespread adoption as a means of communication across most vehicles and other embedded systems like medical equipment and aircraft applications. CAN is a vehicle-bus protocol that allows vehicular components known as electronic control units (ECUs) to communicate effectively on the network without a central computing system. ECUs are the core of the vehicular systems which control electrical components such as antilock braking, electric power steering, power windows, and heating and ventilation systems. Today's vehicle contains a total of over 100 ECUs. Messages across the vehicle are relayed on the CAN bus, which is disseminated to all ECU modules to support real-time execution of events.

In spite of the benefits that CAN protocol adoption offers, its has been met with issues one of which is data insecurity [1]. CAN was originally designed with security incorporated as an afterthought. This leaves the network susceptible to a myriad of potential zero-day cyber-attacks such as Denial of Service [2], [3], Fuzzing [4], and message injection attacks [5]. It is essential that more security tools are developed which provides real-time detection of intrusions on in-vehicle networks. The lack of security measures can lead to devastating long term effects in the safety of ECU systems across the network. Deep learning-based language models have proven to produce remarkable results for natural language tasks such as a machine translation [6], question answering [7], and sentiment analysis [8]. Bidirectional Encoder Representations from Transformers (BERT) [9] is a widely used transformer-based language model which pre-trains deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all neural network layers. The result of the pre-trained model can be fine-tuned on the last output layer providing state-of-the art results for most natural language understanding tasks while doing this at a faster rate.

In this paper, we explore the use of transformer-based language models to provide a means for intrusion detection on the CAN bus. To the best of our knowledge, this is the first approach in which language-based transformer models have been applied for intrusion detection on in-vehicle networks. We propose CANBERT, a BERT model trained to understand CAN messages. Furthermore, we fine-tune our CANBERT model on downstream task such as classifying CAN messages that originate on a CAN bus as either malicious or normal. This is achieved by training our dataset[1] containing already classified CAN data attacks on our model. The dataset was constructed by logging CAN traffic via the OBD-II port from a real vehicle while message injection attacks were performed. By doing so, this approach is able to detect unseen malicious CAN data that might originate in the CAN bus. We evaluate the feasibility of our approach by comparing the precision, recall and F1 scores of our approach to baseline models in addition to state of the art deep-learning based IDS model.

II. Background

In this section, we outline background information on the two major components in which our approach is built on: CAN bus and transformer-based BERT Model.

[1]https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

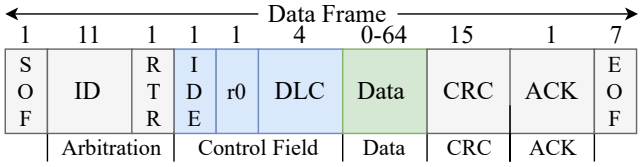

Fig. 1. CAN data and remote frame format with field lengths in bits.

A. CAN Architecture Overview

CAN bus is a serial communication protocol comprising of a set of nodes called ECUs. The CAN bus was originally designed for the automotive industry but has seen wide adoption in other domains such as industrial automation and embedded controls. The CAN bus is a high-speed, cost-effective network technology with a reduced amount of wiring used by the ECUs to transmit messages related to their functions. CAN supports up to 8 bytes of data payload, as shown in Fig. 1 and implements the carrier sense multiple access protocols with collision detection and arbitration on message priority (CSMA/CD+AMP) use in automotive applications. Messages on the bus contain a unique ID that specifies their priorities and meaning, and the lower the message ID, the higher its priority. Higher priority messages are transmitted first on the bus due to message arbitration, ensuring the non-interrupted transmission of the highest priority message. The CAN bus is a robust protocol that supports static fixed priority non-preemptive scheduling of messages, error detection, signaling, and fault confinement. However, the CAN bus has been an attractive target for cyber attackers as it does not implement any security protocol (no encryption or authentication), making it susceptible to cyber and physical attacks. Attacks on the bus, such as denial of service (DoS), spoofing, and masquerade, have been demonstrated using simple tools by various researchers in the literature [10], [11].

B. BERT Model Overview

Transformer-based models [12] are widely used in natural language processing tasks and provide state-of-the-art results for most tasks with high accuracy. These models are a specialized form of deep learning architecture developed using a multi-head attention mechanism that involves a series of attention heads in the encoder and decoder of sequence to sequence tasks. One benefit of the transformer models over other types of deep learning architecture is their ability to process sequential data all at once, achieved using the self-attention and positional context of every input sequence. For more information on transformer-based models, we refer the reader to state-of-the-art literature by Vasawani et al. [12].

BERT, a form of an encoder-only transformer-based model, provides deep bidirectional training of unlabelled text by combining left and right contexts in both layers. This allows fine-tuning of the last layer of a pre-trained BERT model on various language tasks. In this approach, sentences are broken into tokens and are used as input into the model with special characters used to indicate the start of every sentence [CLS], masked words [MASK], and the end of every sentence [SEP]. BERT was trained using datasets from BookCorpus containing 800 million words and Wikipedia containing 2500 million words. BERT training involves two major steps as depicted in Fig. 2; pre-training and fine-tuning, which are discussed in greater detail below:

1) Pre-training: In order to train bidirectional deep representation of the tokens, BERT is trained on unlabelled data using two tasks; Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM, often referred to as the Cloze task in the literature [13], provides a masked representation of about 15% of the training data. The hidden tokens chosen at random are used as input vectors to an output softmax layer of the transformer model. The model then tries to predict the masked portion containing the hidden token. This way, the model can efficiently learn the various representation of the text such that it can be further fine-tuned on other downstream tasks such as classification.

In addition to MLM, BERT also performs an NSP task which deals with training a BERT model to understand sentence relationships such that given a sentence, the model is able to predict if two sentences accompany each other. The goal of this training process is to improve the performance of the model for downstream tasks such as natural language inference [14]. However, in subsequent iterations of BERT implementations such as RoBERTa [15], this approach has been shown not to improve the overall performance of the model, and is not required to provide bidirectional understanding. In addition, it increases the computation overhead of model pretraining. Hence, NSP task was not used in training our CANBERT model.

2) Fine-tuning: This process is relatively straightforward and comes after the pre-training phase where pre-trained model parameters are initialized, and all of the parameters are fine-tuned from the labeled data in a downstream task such as sequence to sequence task, question answering, and classification task. Dataset from the downstream tasks is trained on the output layer of a pre-trained model, making the training quicker than training from scratch. In a classification task, the resulting output layer is placed on a softmax function which generates a probability distribution based on the likelihood of the data is a member of every class.

III. Related Work

Several studies have developed intrusion detection systems for in-vehicle networks. Most of which involve the application of techniques such as specification-based approach [16]–[19] that utilizes a predefined set of specifications like message timing to classify messages on the CAN bus. Fingerprint-based approach [20], [21] classifies malicious messages based on a signature of network events derived from message features as they execute on the network bus. Recently, machine learning [22]–[25] approaches using recent advances in deep learning techniques such as recurrent neural networks (RNN) have been investigated to detect malicious messages on a CAN bus. We present some of the recent work on in-vehicle network intrusion detection below:

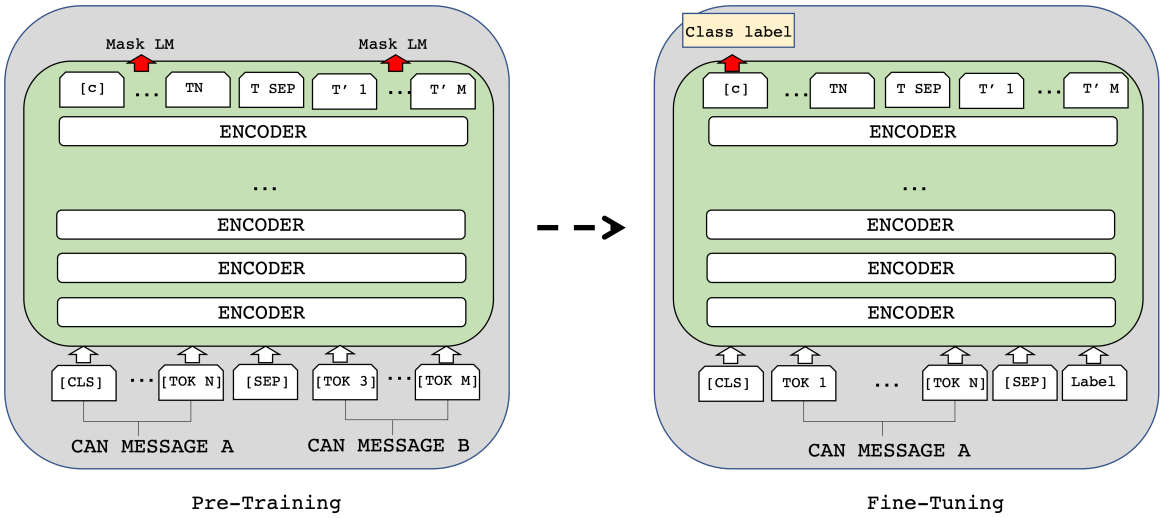

Fig. 2. CANBERT pre-training and fine-tuning system architecture.

Song et al. [26] proposed a method that consists of two deep-learning models, a generator, and a detector for in-vehicle network intrusion detection using noised pseudo normal data. The generator model creates noised pseudo-normal data using Long-Short Term Memory (LSTM), while the detector model uses the Reduced Inception-ResNet model for training sequences of CAN messages to detect anomalies. Kang and Kang [22] proposed an intrusion detecting system for in-vehicle networks using deep belief networks. Song et al. [27] proposed a detection system based on a deep convolutional neural network (DCNN) called inception-ResNet by creating a frame builder module that converts the CAN bus data into a grid-like structure that serves as input to the DCNN. This way, the CNN-based classifier learns the sequential temporal patterns in the CAN traffic data without additional feature engineering. Levi et al. [28] utilized the hidden Markov model to learn the behavior of CAN messages on the network bus. Seo et al. [29] present a detection system using Generative-Adverse Network (GANs) by transforming CAN data into a one-hot encoding vector and converted into an image, which is then used as input into the GAN deep learning model. The model then sets series of threshold values at various discriminator levels to detect incoming malicious messages. Anjum et al. [25] proposed an anomaly detection technique based on extreme gradient boosting machine (GBM) learning algorithm. The authors categorize unexpected occurrences in the CAN data payload and obtain optimum model performance by performing individual parameter tuning and cross-validation, followed by early stopping rounds.

With the recent advancements in deep-learning intrusion detection models, to the best of our knowledge, there is no prior work that deals with the use of language-based models such as bidirectional encoder transformer architecture for developing intrusion detection systems on in-vehicle networks.

IV. Threat Model

The main goal of an attacker is to compromise the ECU modules in the vehicle via the CAN bus. An attacker may gain access to the bus through an unpatched vulnerability. This vulnerability provides the attacker access to inject malicious messages that are directly relayed to other ECUs across the network. The threats that our intrusion detection system mitigates are in line with threats outlined in previous research on in-vehicle networks [16], [29]. In order to guarantee the applicability of the proposed method, we provide the following stipulations in which our approach can efficiently function in detecting network intrusions. We assume that attacks performed on the device are conducted on a remote or local network using measures such as fuzzing, DoS, and message injection attacks. Our approach, however, does not cover the security beyond the aforementioned network parameters (e.g., physical network security and side-channel attacks caused by power consumption or electromagnetic leaks are beyond the scope of this work). Next, we provide further details on the attack types that our approach detects.

- **Denial of Service Attack:** This form of attack involves overloading the network with empty or malicious messages. This process can lead to over-utilization of resources across the network which might have adverse effects on the ECUs contained in the network.
- **Network Fuzzing:** Similar to brute force attack, this involves the injection of a combination of known malicious messages (fuzz vectors) in the hopes of exploiting a network vulnerability.
- **Message Injection Attack:** This type of attack involves the addition of malicious messages at various time interval across the network which can cause the ECU module to malfunction. An example of such an attack is injecting repeated malicious messages to activate the break module.
- **Spoofing Attack:** This type of attack involves mas-

TABLE I
DATASET SUMMARY

Dataset	No of Malicious Messages	No of Normal Messages
DoS Attack	587521	3047062
Fuzzy Attack	491,847	3259177
RPM Spoofing	654,897	3,925,329
Gear Spoofing	597,252	3,805,725
Attack free state	N/A	2,369,868

querading or impersonating a source by sending messages with false ID with the intent of learning network parameters in order to insert malicious messages on the network.

V. CANBERT APPROACH

In this section, we outline steps taken in CANBERT approach. We provide details on the data processing, and implementation techniques utilized to train our transformer-based BERT model. Fig. 3 displays a high-level workflow for the CANBERT approach. The main idea is training the BERT model to understand semantics in CAN messages. Once we achieve this, we are able to fine-tune more specific classification tasks on the top layer of the neural network using labelled CAN dataset which are derived from realistic driving scenarios. We outline each process in greater detail below.

A. Data Processing

To train our model, we utilize an open-source dataset developed by Lee et al. [30] which consists of attack free and malicious data generated by logging CAN traffic via the OBD-II port from a real vehicle while various message injection attacks were performed on the vehicle's CAN bus. The authors perform the following attacks: DoS attacks by injecting messages of '0x000' CAN ID in a short cycle, fuzzy attacks by injecting messages of spoofed random CAN ID and DATA values, and impersonation attacks by injecting messages of impersonating node with an arbitration ID of '0x164'. The dataset consists of information such as timestamps (in seconds), CAN ID, DLC, and 8 bytes of data. Table I summarizes the datasets used in the evaluation.

B. Data Pre-processing

For our data processing efforts in both the pre-training and fine-tuning steps, we utilize labeled datasets that comprise attack and normal data. Before training the BERT model, we performed initial preprocessing steps by placing our datasets into a common data format. In this format, we concatenate each data component contained in each CAN message using a space string as the delimiter. This information provides a more string-like representation for input into our BERT model, as transformer-based models work well with string. For the tokenization process, we utilize Byte-level Byte-Pair encoding tokenizers that split words into appropriate tokens to ensure that the most common words are represented as a single token while less common words are broken down into subtokens. The tokenization process converts CAN message into appropriate numerical representations, which are used as input to the encoder models.

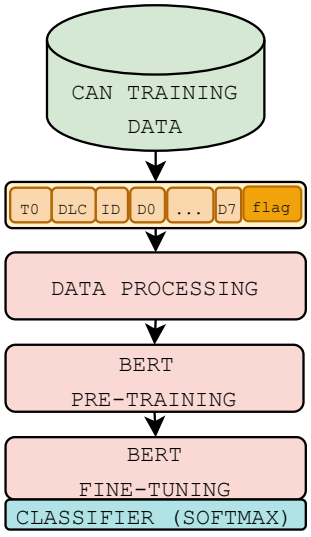

Fig. 3. CANBERT system workflow

C. Pre-training

Once the dataset has been pre-processed using the aforementioned steps, it is then used as an input into the transformer model. Our approach involves training our bidirectional encoder model from scratch in order to properly learn a detailed semantic representation of CAN messages. To achieve this step, we leverage prior work conducted by Liu et al. [31]. In this work, they designed RoBERTa, an optimized approach for the pre-training BERT model. This approach provides some improvements over the standard BERT approach such as the elimination of the NSP task and reducing the parameters that allow the model to be optimally trained at a faster time than the traditional BERT model. Our model is trained on the input datasets, which consist of normal driving conditions. These datasets are trained on the bidirectional encoder models using the Masked Language Model task where 15% of the data is masked for training to avoid allowing each word indirectly predict the target word in a multi-layered context. With this task, we are able to provide a deep bidirectional representation of each CAN message contained in the normal driving dataset. This way, the model properly learns about CAN syntax. Next, we discuss our fine-tuning efforts.

D. Fine-tuning

The fine-tuning process involves training the top layer of the bidirectional encoder model on classification tasks (Intrusion classification in our case). This way, we are able to provide a fine-tuned layer that classifies CAN messages as either malicious or normal based on the deep bidirectional representation of CAN messages previously learned in the pre-training phase. In order to achieve the task of fine-tuning, we utilize three datasets containing labeled information of a combination of normal and malicious CAN messages.

VI. EXPERIMENTAL EVALUATION

In this section, we discuss experimental evaluations and provide details on the experimental parameters used in our

TABLE II
RESULTS FROM THE BASELINE AND CANBERT MODELS.

	RPM Spoofing Attack				Gear Spoofing Attack				DoS Attack				Fuzzy Attack			
Model Name	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
XGBoost	1.00	0.98	0.98	0.98	1.00	0.98	0.98	0.97	0.98	0.95	0.95	0.95	1.00	1.00	1.00	1.00
KNN	1.00	0.97	0.96	0.96	1.00	0.97	0.97	0.96	0.99	0.92	0.93	0.92	1.00	0.96	0.96	0.96
Logistic Regression	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.97	0.95	0.94	0.94	0.99	0.98	0.98	0.98
RandomForest	1.00	0.97	0.97	0.97	1.00	0.97	0.97	0.97	0.99	0.94	0.94	0.94	1.00	1.00	1.00	1.00
MultinomialNB	0.86	0.95	0.95	0.95	0.87	0.95	0.95	0.95	0.84	0.90	0.91	0.90	0.87	0.95	0.95	0.94
SVM	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.91	0.91	0.91	0.91	0.98	0.98	0.98	0.98
CANBERT	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

approach.

A. Baseline Models

In order to evaluate the effectiveness of our approach, we developed and compared our approach to some baseline models. The baseline models consist of several linear machine learning models which have also been shown to produce high accuracy on most classification tasks and in detecting CAN anomalies. These models were implemented using the sckit-learn package, a python library for machine learning. We generate our baseline models using datasets that contain attacks and normal CAN messages. For each datasets, we performed an 80/20 split between training and test data respectively with stratification on class labels to account for class imbalance. Each model was trained with K-fold cross-validation where $k = 10$. Experiments for the baseline model generation were run on a 51GB RAM CPU using Google Colab[2].

B. CANBERT

The model pre-training and fine-tuning process were conducted on a TPUv2.8 with 8 cores and a total TPU memory of 64GB using the Tensorflow Cluster in Google Colab. CANBERT was pre-trained on a dataset that consists of only normal CAN messages and then fine-tuned on a dataset that consists of both normal and various malicious attacks as outlined in Table I. For the process of pre-training and fine-tuning, we utilized a batch size of 64.

We also utilized Adam as the optimizer of choice, with a learning rate of 1e-5. For fine-tuning, we utilized categorical cross entropy as the loss function and accuracy, precision, and recall as the validation metric. The data was divided into 64% training samples, 20% validation samples, and 16% test samples. The number of epochs was set to 3 for fine-tuning.

VII. RESULT

In this section, we provide results derived from evaluating our baseline and CANBERT models. We also compare CANBERT to other state of the art deep learning based in-vehicle intrusion detection approach.

A. Comparing our baseline models to CANBERT

Table II presents the results of the baseline model generation and the proposed CANBERT approach. From the presented results, we can see that most models except the naive Bayes model detect all types of attacks with over a 90% F1 score. The result is indicative that our baseline models can also detect network intrusions with high accuracy. Logistic regression and SVM performed the best in detecting spoofing and fuzzy attacks. XGBoost performs best in detecting DoS attacks with an F1 score of 95%. CANBERT outperforms all the baseline models in detecting all types of attacks with a slight increase in F1 scores across all attack sets.

Compared to the baseline results, the proposed approach performs the best with an accuracy and F1 score of 1.00 across all datasets. This shows that our approach is able to detect attacks at a near-perfect rate of 100%. Our approach was able to detect all of the malicious CAN messages contained in the dataset.

TABLE III
COMPARING CANBERT TO GIDS

	GIDS	CANBERT
Attacks	**Accuracy**	**Accuracy**
RPM Spoofing	1.00	1.00
Gear Spoofing Attack	1.00	1.00
Fuzzy Attack	0.99	1.00
DoS Attack	1.00	1.00

B. Comparing the State of the Art to CANBERT

We compare the accuracy of our approach to a state-of-the-art deep learning approach proposed by Seo et al. [29] which utilizes GANs, a specialized form of deep learning model for detecting malicious CAN messages. Table III presents the accuracy and precision of the results derived from both CANBERT and GIDS. The results show that the proposed approach performs just as well as the GIDS in detecting the injected malicious CAN messages in all the categories of the attack datasets. It also performs slightly better with a slight increase in accuracy in detecting fuzzy attacks. This result is indicative of the robustness of our model in detecting a series of malicious attacks that can be found in CAN bus systems.

VIII. LIMITATIONS

This approach provides a great start to the adoption of transformer-based models for detecting malicious intrusions on in-vehicle networks. However, it does have some limitations. One of the main limitations of this approach is the high computational requirement and processing time involved in pre-training. Due to the resource constraint that most in-vehicle networks impose, real-time training with such an approach might be challenging. One way of mitigating such restriction is by pre-training the model offline and also employing the

[2]https://colab.research.google.com/

use of efficient transformer models [32] that are optimized for training smaller datasets in memory-constrained environments.

IX. Conclusion and Future Work

In this paper, we propose a language-based model for detecting malicious messages on in-vehicle networks. This model is built with the power of transformer architecture which provides a means of learning deep semantic bidirectional relationships that exist in CAN messages. We compare our approach with baseline and state-of-the-art models on in-vehicle intrusion detection. The proposed approach is able to detect a wide range of CAN bus intrusions with high accuracy. In the future, we hope to explore the use of a wide array of datasets that contains a more comprehensive driving scenario. In addition, we hope to explore transformer-based models that are optimized for memory-constrained environments.

References

[1] B. Groza and P.-S. Murvay, "Security solutions for the controller area network: Bringing authentication to in-vehicle networks," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 40–47, 2018.

[2] M. Bozdal, M. Randa, M. Samie, and I. Jennions, "Hardware trojan enabled denial of service attack on can bus," *Procedia Manufacturing*, vol. 16, pp. 47–52, 2018, proceedings of the 7th International Conference on Through-life Engineering Services. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918312794

[3] W. Si, D. Starobinski, and M. Laifenfeld, "Protocol-compliant dos attacks on can: Demonstration and mitigation," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–7.

[4] H. Lee, K. Choi, K. Chung, J. Kim, and K. Yim, "Fuzzing can packets into automobiles," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 817–821.

[5] Y. Zhang, B. Ge, X. Li, B. Shi, and B. Li, "Controlling a car through obd injection," in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, 2016, pp. 26–29.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[7] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 633–644. [Online]. Available: https://aclanthology.org/D14-1070

[8] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 959–962.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[10] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *black hat USA*, vol. 2014, p. 94, 2014.

[11] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *20th USENIX Security Symposium (USENIX Security 11)*, 2011.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[13] W. L. Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism & Mass Communication Quarterly*, vol. 30, pp. 415 – 433, 1953.

[14] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," 2015. [Online]. Available: https://arxiv.org/abs/1508.05326

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[16] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "Saiducant: Specification-based automotive intrusion detection using controller area network (can) timing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1484–1494, 2020.

[17] S. Halder, M. Conti, and S. K. Das, "Coids: A clock offset based intrusion detection system for controller area networks," in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, ser. ICDCN 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3369740.3369787

[18] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno, "Automotive intrusion detection based on constant can message frequencies across vehicle driving modes," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, 2019, pp. 9–14.

[19] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 220–225.

[20] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*, 2015, pp. 45–49.

[21] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 911–927. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[22] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[23] S. Tariq, S. Lee, and S. S. Woo, "Cantransfer: Transfer learning based intrusion detection on a controller area network using convolutional lstm network," in *Proceedings of the 35th annual ACM symposium on applied computing*, 2020, pp. 1048–1055.

[24] V. S. Barletta, D. Caivano, A. Nannavecchia, and M. Scalera, "Intrusion detection for in-vehicle communication networks: An unsupervised kohonen som approach," *Future Internet*, vol. 12, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1999-5903/12/7/119

[25] A. Anjum, P. Agbaje, S. Hounsinou, and H. Olufowobi, "In-vehicle network anomaly detection using extreme gradient boosting machine," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2022, pp. 1–6.

[26] H. M. Song and H. K. Kim, "Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1098–1108, 2021.

[27] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209619302451

[28] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected car cybersecurity," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–7.

[29] 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)*, 2018, pp. 1–6.

[30] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, 2017, pp. 57–5709.

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[32] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1909.10351