ISSN: 2093-5374 / E-ISSN: 2093-5382

Lightweight IDS Framework Using Word Embeddings for In-Vehicle Network Security

Harang Kim[1], and Hyun Min Song[2*]

[1]Department of Computer Science, Dankook University, Yongin, Republic of Korea.
harangkim@dankook.ac.kr, https://orcid.org/0009-0007-1455-6367

[2*]Department of Cybersecurity, Dankook University, Yongin, Republic of Korea.
hyunminsong@dankook.ac.kr, https://orcid.org/0000-0002-0048-4231

Received: January 03, 2024; Revised: February 14, 2024; Accepted: April 04, 2024; Published: June 29, 2024

Abstract

As modern vehicle systems evolve into advanced cyber-physical systems, vehicle vulnerability to cyber threats has significantly increased. This paper discusses the need for advanced security in the Controller Area Network (CAN), which currently lacks security features. We propose a novel Intrusion Detection System (IDS) utilizing word embedding techniques from Natural Language Processing (NLP) for effective sequential pattern representations to improve intrusion detection in CAN traffic. This method transforms CAN identifiers into multi-dimensional vectors, enabling the model to capture complex sequential patterns of CAN traffic behaviors. Our methodology focuses on a lightweight neural network adaptable for automotive systems with limited computational resources. At first, a Word2Vec model is trained to make the embedding matrix of CAN IDs. Then, using the pre-trained embedding layer extracted from the Word2Vec network, the classifier analyzes embeddings from CAN data to detect intrusions. This model is viable for resource-constrained environments due to its low computational expense and memory usage. Key contributions of this research are (1) the application of word embeddings for intrusion detection in CAN traffic, (2) a streamlined neural network that balances accuracy with efficiency, and (3) a comprehensive evaluation showing our model's competitive performance compared to relatively heavy deep learning models. Experimental results using the Car-Hacking dataset, widely used for automotive security research, demonstrate that our IDS effectively detects four different types of attacks on CAN. This work advances vehicle security technologies, contributing to safer transportation systems.

Keywords: Intrusion Detection System (IDS), Controller Area Network (CAN), Word Embeddings, Automotive Cybersecurity.

1 Introduction

The automotive industry is moving forward with interconnected technologies, which has resulted in the notable integration of sophisticated cyber-physical systems in modern vehicles. These vehicles are equipped with advanced electronics and networking capabilities, making them highly vulnerable to cyber threats (Lampe & Meng, 2023; Muralidharan., 2020l Stevovic et al., 2018). This increased connectivity of vehicles requires robust cybersecurity measures to protect vehicle systems to ensure operational safety and reliability. In modern vehicles, Controller Area Network (CAN) is a core

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 15, number: 2 (June), pp. 1-13. DOI: *10.58346/JOWUA.2024.I2.001*

*Corresponding author: Department of Cybersecurity, Dankook University, Yongin, Republic of Korea.

networking system, providing high-speed and reliable communication between Electronic Control Units (ECUs) in the vehicle systems. A CAN bus deliver messages transmitted by ECUs to control and maintain information consistency in various vehicle functionalities, including engine operations, safety features, and infotainment systems. However, CAN does not have built-in security features to protect itself, resulting in cyber-attack vulnerabilities. The safety and security of CAN traffic are critical since any unauthorized access could result in severe consequences and endanger passengers and other road users (Yu et al., 2022; Sreenivasu et al., 2022; Prasad Babu et al., 2023).

To solve these challenges, pioneer researchers proposed the various approaches for accurate detection of intrusions on CAN system using machine-learning based techniques (Ahmed et al., 2021; Gou et al., 2023). Thus, some studies proposed the deep learning model-based sophisticated Intrusion Detection System (IDS) leveraging sequential patterns in CAN traffic data (Javed et al., 2021; Song et al., 2020; Kutlu et al., 2021). Especially, RNN (Recurrent Neural Network)-based models are well-known for their capabilities to process time-series data. LSTM is the most widely used RNN architecture, and there are works to have tried to adopt them to CAN intrusion detection (Nayak et al., 2023; Hossain et al., 2020; Elshrkawey et al., 2021).

In this work, we propose a novel method to improve in-vehicle network security using Natural Language Processing (NLP) techniques, specifically word embeddings (Rajesh et al., 2023; Nikitina et al., 2023). Word embedding is well-known for its ability to capture sequential patterns of words within sentences (Kostić et al., 2021; Bobir et al., 2024). It learns embedding matrix using a shallow neural network by training it to predict neighboring words of a given input word. Considering this, we utilize word embeddings to capture the sequential patterns of CAN messages sent over a CAN bus by making a Word2Vec model predict a given CAN ID's prior and following CAN ID. Through this, the Word2Vec model can learn an embedding matrix which is containing sequential patterns of CAN IDs in CAN traffic. We can reuse the embedding matrix to convert CAN IDs into multi-dimensional vectors in subsequent intrusion detection models. By converting CAN IDs into contextual embedding vectors, intrusion detection models can accurately identify signs of unauthorized behaviors within the CAN traffic. This study focuses on making lightweight intrusion detection systems for in-vehicle network systems that can quickly identify intrusions while using low computational resources, making them suitable for use in the resource-constrained environments typical in automotive systems. By leveraging a pretrained embedding matrix, we can simplify the architecture of intrusion detection neural network, and it can efficiently process and identifies patterns in CAN traffic with low computational resources. This allows us to make a lightweight IDS both effective and suitable for resource-constrained automotive environments.

In this paper, our contributions can be summarized as follows:

- The application of word embedding techniques to learn sequential patterns in CAN traffic, enhancing the ability to detect intrusion with relatively light neural network models.
- The development of a lightweight model that balances intrusion detection performance and computational efficiency, suitable for resource-constrained applications in vehicular contexts.
- A comprehensive evaluation framework that not only validates the model's effectiveness across various intrusion scenarios but also compares its performance with complex deep learning models, demonstrating superior adaptability and competitive detection accuracy.

The rest of this paper is organized as follows: Section 2 provides brief literature survey on prior studies about intrusion detection systems for automotive networks. Section 3 describes the methodology, detailing the innovative use of word embeddings and neural network design. Section 4 presents experimental results, demonstrating the effectiveness of our approach through comparative analysis. Section 5 concludes with a summary of findings and discusses potential directions for future research.

2 Related Works

The automotive network IDS has recently gained industrial and academic attention as autonomous vehicles have become popular. Researchers have devoted considerable attention to developing various methodologies such as machine learning, deep learning, and hybrid models to enhance intrusion detection capabilities. Lampe & Meng, (2023) conducted a comprehensive survey of automotive IDS approaches that use deep learning. They highlighted the advantages and limitations of various architectures when applied in the automotive industry. Ahmed et al., (2023) adopted a well-known deep learning architecture of VGG in computer vision tasks to detect malicious attacks on the Internet of Vehicles (IoV). Gou et al., (2023) proposed a tree-based ensemble network that classifies various attack types as multi-class classification problems. Javed et al., (2021) proposed a combined model consisting of convolutional neural networks (CNN) with attention-based Gated Recurrent Unit (GRU) networks named CANintelliIDS. Song, Woo & Kim, (2020) proposed a novel approach to learning CAN traffic sequential patterns using deep convolutional neural networks (DCNNs). They used a specialized preprocessing module named Frame Builder. It converts CAN IDs into binary representations. Then, the DCNN model learns the sequential characteristics within binary-represented CAN traffic and detects message injection attacks on the CAN bus (Srivastava et al., 2013; Yang et al., 2022).

Long Short-Term Memory (LSTM) network applications have proven particularly effective in processing time-series data. There are few studies that utilize LSTM-based neural network models to detect cyber-attacks on CAN bus systems. Yu et al., (2022) developed an LSTM-based IDS for VANETs that focuses on time series classification to identify false messages within network traffic. Similarly, Nayak et al., (2023) implemented an intelligent intrusion detection system using an LSTM. It showed the robustness of LSTM in handling network-specific challenges. Hossain et al., (2020) further validated the efficacy of LSTM models for in-vehicle network security. Longari et al., (2020) and Tanksale, (2020) both explored the utility of LSTM autoencoders in detecting anomalies within CAN traffic, underscoring LSTM's capability to learn from sequential and time-dependent data to enhance network security.

Recent advancements include integrating machine learning techniques to optimize intrusion detection in smart vehicular networks. Alsarhan et al., (2020) introduced a novel machine learning-driven approach that combines rule-based filters with Bayesian learning to accurately detect and classify network behaviors in vehicular ad-hoc networks (VANETs), thus enhancing the reliability of security systems against sophisticated cyber threats. Islam et al., (2020) proposed a novel methodology for detecting anomalies using the graph structures of CAN traffic. Their system effectively identifies unusual patterns that imply malicious behaviors. This approach is an advanced application of graph theory in cybersecurity within intelligent transportation systems.

Yu et al., (2022) developed the TCE-IDS, a Time Interval Conditional Entropy-based Intrusion Detection System. It leverages the entropy measures of time intervals between messages to detect abnormalities in network traffic. This study demonstrates how entropy-based metrics can improve intrusion detection system accuracy. Kongjian et al., (2020) proposed a neural network-based intrusion detection system that integrates with domain expertise to generate unique features from automotive network data. Their approach emphasizes the importance of domain-specific adaptations to enhance effectiveness in intelligent vehicular environments. Kalkan & Sahingoz's, (2020) research highlights the utility of traditional machine learning techniques of decision tree-based algorithms in developing effective IDS solutions that can adapt to the dynamic nature of in-vehicle networks.

3 Methodology

This section introduces step-by-step the proposed lightweight automotive intrusion detection framework, which consists of a word embedding-based CAN traffic sequential pattern modeling and intrusion detection using a shallow neural network.

Data Collection

Collecting in-vehicle network data typically entails monitoring and recording the traffic delivered within a vehicle's network. Since CAN is the fundamental communication channel between ECUs, we can collect which messages are transmitted by the ECUs to control various functionalities of automotive systems and furthermore compromise ECUs to malfunction by injecting manipulated CAN messages.

Data collection in CAN networks generally employs a device connected directly to the CAN bus via OBD-II (On-Board Diagnostics) port, which is used to tap into the network without disrupting the ongoing communication. Figure 1 shows the OBD-II port configuration. Pins 6 and 14 are connected to CAN High and Low, respectively, to allow the CAN bus from an external device.

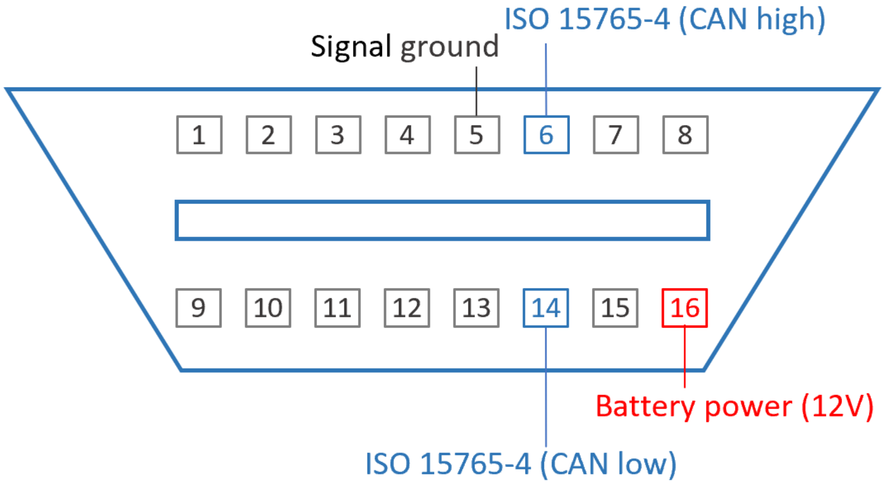
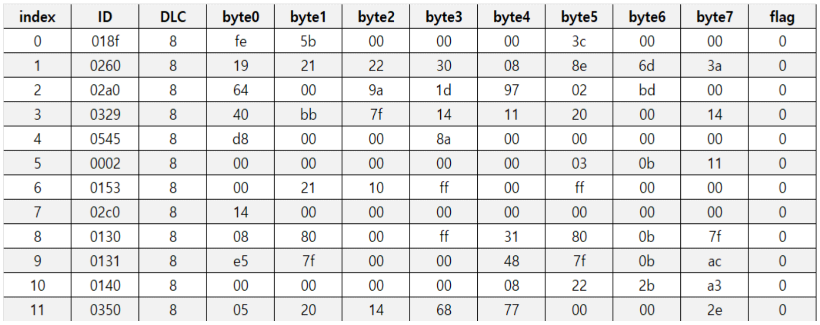


Figure 1: OBD-II Port Configuration. Pins 6 and 14 are Connected to CAN High/Low Respectively to Allow Access to the CAN Bus

Once a hardware device is attached to the CAN bus via an OBD-II port, software tools like CANopen are used to record the traffic. These tools read the binary data frames relayed on the CAN bus, and output interpreted CAN traffic in a format that typically consists of a timestamp, CAN ID, DLC, and the data payload. The timestamp indicates when a message was transmitted, the CAN ID identifies the message's purpose or originating ECU, and the data payload contains the actual information sent in the message, as shown in Figure 2. It is essential to prepare a comprehensive CAN traffic dataset in automotive cybersecurity applications. Each CAN message in collected data should be labeled as benign or abnormal behaviors to evaluate developed intrusion detection systems properly. In this study, we utilize the Car-Hacking Dataset provided by the Hacking and Countermeasure Research Laboratory (HCRL) (Song et al., 2020), which is widely used for automotive security research.



index	ID	DLC	byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7	flag
0	018f	8	fe	5b	00	00	00	3c	00	00	0
1	0260	8	19	21	22	30	08	8e	6d	3a	0
2	02a0	8	64	00	9a	1d	97	02	bd	00	0
3	0329	8	40	bb	7f	14	11	20	00	14	0
4	0545	8	d8	00	00	8a	00	00	00	00	0
5	0002	8	00	00	00	00	00	03	0b	11	0
6	0153	8	00	21	10	ff	00	ff	00	00	0
7	02c0	8	14	00	00	00	00	00	00	00	0
8	0130	8	08	80	00	ff	31	80	0b	7f	0
9	0131	8	e5	7f	00	00	48	7f	0b	ac	0
10	0140	8	00	00	00	00	08	22	2b	a3	0
11	0350	8	05	20	14	68	77	00	00	2e	0

Figure 2: Exemplary CAN Traffic Sample. There are CAN Identifier, DLC (Data Length Code), Payload. The Flag is added while Data Collection Process. Timestamps have been Omitted for Brevity

We decided to leverage this pre-collected, well-curated, and publicly open dataset instead of collecting data directly from a vehicle's CAN network. It requires an extensive setup and potentially raises privacy or legality issues. It also provides message-level labels to support the development of automotive intrusion detection systems. It includes four types of attack scenarios, such as Denial of Service (DoS), fuzzing, and spoofing attacks targeting drive gear and RPM values, along with a clean driving scenario.

Each subset is designed to simulate distinct operational scenarios and attack strategies within a CAN bus and is described as follows:

- **Normal Traffic** This subset captures the routine communication within the CAN network, reflecting typical vehicle operations and providing a critical comparative backdrop for identifying anomalies and malicious behaviors in the network.
- **DoS Attack Data** This subset is the scenario where the network is intentionally overloaded with messages to disrupt the normal functions of the vehicle. In the CAN bus, messages are rapidly sent with the intent to prevent legitimate communications. The injected messages often contain the most dominant CAN ID, the lowest value, ensuring they take priority in the network and effectively prevent transmitting normal messages.
- **Fuzzing Attack Data** This subset includes instances where random or malformed messages are deliberately injected into the network. Since the CAN protocol does not have a way to identify the sender node, any node connected to the CAN bus can imitate other ECUs. The purpose of fuzzing attacks is to explore vulnerabilities of the target system, such as triggering system errors or causing the automotive system to behave unpredictably.
- **Spoofing Attacks** These subsets are characterized by targeted manipulations of specific message ID and payloads, such as drive gear and RPM values on dashboards. If spoofing attacks successfully executed, it could make attacker to take control of vehicle operations or mislead the driver, thereby introducing serious safety risks.

The dataset provides a controlled test environment to evaluate intrusion detection models. It helps us to develop robust detection models to effectively classify normal behaviors and attacks in the CAN network. Thus, it ensures that the proposed method covers a comprehensive range of attack vectors, enhancing the robustness and relevance of our findings in the context of automotive cybersecurity.

Data Preprocessing

The data preprocessing step converts raw CAN traffic data into a format suitable for analysis with word embedding techniques. It focuses on the sequential patterns in CAN traffic and simplifies the dataset's essential features for lightweight intrusion detection models.

CAN ID Extraction Each CAN message consists of multiple fields, as shown in Figure 2, but we only use the CAN ID field to capture sequential patterns of CAN traffic. The CAN ID is a unique identifier representing what kinds of information the message carries. The model simplifies the input data by focusing solely on CAN IDs to reduce the computational cost. It is aligned to create a lightweight intrusion detection system. In this context, other fields, such as Timestamps and payload, were excluded during the preprocessing pipeline. This decision is based on the nature of typical CAN bus attacks, where adversaries perform message injection, which alters the sequence of CAN IDs observed during legitimate vehicle operation. The model targets the intrusions by concentrating on CAN IDs, which cause deviations from typical CAN ID sequences.

Sequence Creation To capture the dynamics of CAN traffic, the preprocessed CAN IDs were reassembled into sequences. Each sequence is a continuous partial segment of traffic, preserving the order and occurrence of CAN IDs over a fixed time window. We constructed sequences by time window 10 for the Word2Vec model so it can learn the contextual relationships, which are consecutive CAN IDs conditional probabilities. Contextual size is set to 2, meaning preceding or following two CAN IDs are considered contextual CAN IDs of a given central CAN ID.

As a result, the preprocessing step extracts CAN IDs from raw CAN traffic and make them into chunk sequences of CAN IDs, setting the foundation for the subsequent application of advanced machine-learning techniques to learn sequential patterns of CAN IDs in vehicular networks. This approach simplifies the raw CAN data and help the model to learn sequential patterns of CAN traffic effectively.

Model Architecture and Design

Our model for detecting intrusions in CAN traffic has two distinct stages. Figure 3 illustrates the proposed CAN sequential pattern learning using word embedding techniques and an intrusion detection system. Initially, we focus on developing a comprehensive representation of CAN IDs by leveraging advanced word embedding techniques. Specifically, we utilize the Word2Vec algorithm (Mikolov et al., 2013), adopting the skip-gram model complemented by negative sampling to learn meaningful embeddings efficiently and effectively. Following this, the second stage of our approach integrates these pre-trained embeddings into a shallow neural network for intrusion detection. This two-tiered strategy ensures a robust foundation for understanding and analyzing sequential patterns in CAN traffic, which is critical for accurately identifying anomalous behaviors.

CAN ID Embedding The initial stage focuses on capturing the semantic similarities between different CAN IDs based on their contextual usage within the traffic sequences. To accomplish this, we employ the Word2Vec algorithm in its skip-gram configuration, which is particularly effective in preserving the local context of words (or CAN IDs, in our case). Negative sampling is used to efficiently handle the large number of CAN IDs by improving the quality of the embeddings and speeding up the training process. The output from this stage is an embedding table where each CAN ID is represented as a dense vector in a N-dimensional space. The embedding layer is trained using large volumes of CAN traffic data under normal operating conditions to ensure that it accurately captures the typical patterns of network communication. Once trained, this embedding layer acts as a feature extractor in the intrusion detection model.

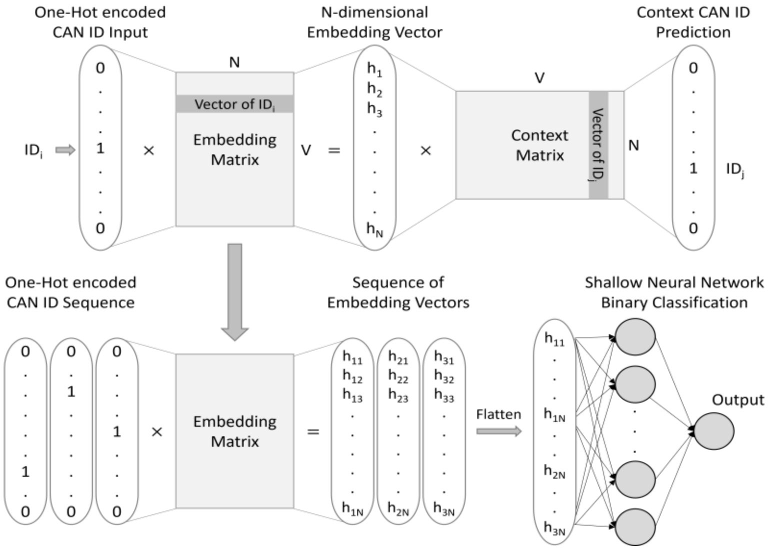

One-Hot encoded CAN ID Input
N-dimensional Embedding Vector
Context CAN ID Prediction
ID_i
N
Vector of ID_i
Embedding Matrix
V
×
=
V
Context Matrix
Vector of ID_j
N
ID_j
One-Hot encoded CAN ID Sequence
Sequence of Embedding Vectors
Shallow Neural Network Binary Classification
Embedding Matrix
Flatten
Output

Figure 3: Schematic Illustration of the Intrusion Detection Model Incorporating Word2Vec Embeddings

The diagram describes the transformation of a one-hot encoded CAN ID vector into a multidimensional embedding vector. Subsequently, it shows how this embedding matrix is used to convert a given sequence of CAN IDs in the intrusion detection model.

This reuse of the pre-trained embeddings allows for transfer learning, where knowledge from the general task of understanding CAN traffic is applied to the specific task of intrusion detection.

Intrusion Detection Model

The intrusion detection model employed in this study is predicated upon the utilization of previously trained word embeddings, specifically designed to interpret CAN traffic data. At the core of our model lies an elegantly architected shallow neural network optimized for the intrusion detection within vehicular networks.

The network architecture features several strategically integrated components:

- **Embedding Layer** This layer is a dense layer with the embedding matrix as the parameter matrix extracted from the pre-trained Word2Vec. It effectively converts each CAN ID into a unique multi-dimensional vector. These embeddings contain the semantic meanings of the CAN IDs. It helps the model to recognize the sequential patterns of CAN traffic more efficiently.
- **Concatenation and Flattening** After converting CAN IDs of a given sequence into embedding vectors, we need to reduce the embedding vectors to a single vector. In this work, we concatenated these vectors to form a composite feature set flattened into a unified feature vector. Additive reduction of vectors also could be considered in this step, but we do not deal with the comparison between vector reduction techniques in this work. A context vector combining information from multiple CAN messages improves neural network's interpretative capability.
- **Hidden Layer** A single dense layer with Rectified Linear Unit (ReLU) activation receives the combined context vector. This dense layer analyzes the context vector and produce more informative features. The ReLU activation function introduces non-linearity to the model, allowing the model to capture complex relationships within the feature set and improve its predictive performance.

- **Output Layer** The final layer is a dense layer with a single neuron used for binary classification. This layer does not use an activation function, outputting logits directly, which are then interpreted using a threshold (typically 0.5 when used with a sigmoid function during inference).

Our methodology aims to develop an efficient and effective intrusion detection system for vehicular networks. Basically, we use a Word2Vec algorithm to create the embedding matrix which contains sequential pattens of CAN ID sequences and then reuse the embedding matrix in front of a shallow neural network-based binary classification model for intrusion detection. By leveraging the pre-trained embedding matrix, we could make classification model simple that requires low computational resources. This approach ensures our model's competitive performance on CAN intrusion detection and efficient operation in resource-constrained environment, making it well-suited for real-world vehicular environments.

4 Experimental Result

This section presents the results of the experiments conducted on applying word embedding techniques to enhance the performance of the CAN intrusion detection system. First, we present the dataset overview and clarify evaluation metrics to evaluate the proposed intrusion detection model compared to other deep learning-based models.

Dataset Description

The study utilizes the HCRL Car-Hacking Dataset, a publicly available dataset designed to enhance intrusion detection in in-vehicle networks. This dataset has been constructed by collecting CAN traffic data from an actual vehicle via the OBD-II port while performing message injection attacks. The dataset comprises four types of message injection attack subsets and a clean subset. This makes it an invaluable resource for developing and evaluating intrusion detection systems.

Each attack session in the attack subsets lasts 3 to 5 seconds, and the entire traffic takes about 30 to 40 minutes. Table 1 provides an overview of the HCRL Car-Hacking Dataset. There are over 15 million CAN messages, about 85% of normal and 15% of injected attack messages. The composition of this dataset allows for strict evaluation of intrusion detection models by providing a realistic simulation of potential cybersecurity threats within vehicular networks.

Table 1: Overview of the HCRL Car-Hacking Dataset

Attack Type	Number of Messages	Number of Normal Messages	Number of Injected Messages
DoS Attack	3,665,771	3,078,250 (83.97%)	587,521 (16.03%)
Fuzzing Attack	3,838,860	3,347,013 (87.19%)	491,847 (12.81%)
Spoofing Drive Gear	4,443,142	3,845,890 (86.56%)	597,252 (13.44%)
Spoofing RPM Gauge	4,621,702	3,966,805 (85.83%)	654,897 (14.17%)
Attack-free (Normal)	988,987	988,987 (100%)	-

Evaluation Metrics

To assess the proposed model's intrusion detection performance, we use four key evaluation metrics widely used in the field of machine learning for classification tasks. Each metric provides unique insights into the model's effectiveness and reliability in classification tasks.

Precision measures the how the model's positive predictions is accurate. It is defined as the ratio of true positives to the total number of positives (i.e., the sum of true positives and false positives). High precision indicates a low rate of false positives. This is important to prevent disrupting by incorrect intrusion alerts.

Recall (Sensitivity) measures the how model correctly identified all actual positives. It is calculated as the ratio of true positives to the total actual positives (i.e., the sum of true positives and false negatives). High recall is essential in security contexts as it reflects the model's capability to detect all potential threats without missing any.

F1-Score is the harmonic mean of precision and recall. This metric is useful because it provides a single score that balances the model's precision and recall. This is beneficial when comparing the overall performance of different models. A high F1-score indicates the model has a robust balance between precision and recall.

Accuracy measures the overall correctness of the model across all predictions made. It is the ratio of correctly predicted observations (both true positives and true negatives) to the total observations. While accuracy is a useful metric, its reliability can be affected in datasets where class distributions are imbalanced. Therefore, accuracy is best used in conjunction with other metrics such as precision, recall, and F1-score to provide a comprehensive overview of model performance.

These metrics collectively offer a thorough assessment of the model's performance in detecting intrusions. Precision and recall are especially pivotal in evaluating the model's effectiveness within a security context. They are essential for systems were overlooking an intrusion or mistakenly identifying legitimate behavior as an intrusion can lead to severe consequences. The equilibrium among these metrics, exemplified by the F1-score, together with overall accuracy, will guide further enhancements and modifications in the model's design and implementation.

Intrusion Detection Performance

The performance of the proposed intrusion detection model was extensively evaluated and compared against two benchmark models, DCNN (Song et al., 2020) and LSTM (Nayak et al., 2023), across various types of attacks. Table 2 summarizes the comparative results, representing the performance of each model in terms of precision, recall, F1-score, and accuracy.

Table 2: Detection Performance Comparison of Baseline Models

Attack Type	Model	Precision	Recall	F1-Score	Accuracy
DoS Attack	DCNN (Song et al., 2020)	0.9999	0.9995	0.9997	0.9998
	LSTM (Nayak et al., 2023)	**1.0000**	0.9995	**0.9998**	0.9998
	Ours	**1.0000**	**0.9997**	**0.9998**	**0.9999**
Fuzzing Attack	DCNN (Song et al., 2020)	0.9989	0.9977	0.9983	0.9988
	LSTM (Nayak et al., 2023)	0.9996	**0.9978**	**0.9987**	0.9991
	Ours	**1.0000**	0.9963	0.9982	0.9987
Spoofing Drive Gear	DCNN (Song et al., 2020)	0.9982	**0.9996**	0.9989	0.9989
	LSTM (Nayak et al., 2023)	**0.9998**	0.9994	**0.9996**	**0.9996**
	Ours	0.9996	0.9987	0.9991	0.9991
Spoofing RPM Gauge	DCNN (Song et al., 2020)	**0.9999**	**0.9993**	**0.9996**	**0.9995**
	LSTM (Nayak et al., 2023)	**0.9999**	0.9990	0.9994	0.9994
	Ours	0.9997	0.9988	0.9992	0.9991

In DoS attacks, our model outperforms both the LSTM and DCNN models on all evaluation metrics of precision, recall, F1 score, and accuracy. Especially, our model showed a higher recall of 0.9997 compared to the LSTM and DCNN of 0.9995 while achieving complete precision of 1.0000. This means that our model makes fewer false negatives without any false positives, resulting in an F1-score of 0.998 and an accuracy of 0.9999, indicating its robustness in identifying DoS attacks with minimal error.

During detecting fuzzing attack evaluations, our model maintained a precision of 1.0000, equaling its performance in DoS attack scenarios, yet exhibited a lower recall of 0.9963. This was the lowest recall among the models, resulting in the lowest F1 score of 0.9982, but it still closely trailed the LSTM's 0.9987 and DCNN's 0.9983. The LSTM showed the best performance in detecting fuzzing attacks with a precision of 0.9996 and the highest recall of 0.9978. While our model is highly precise, there is room for improving its sensitivity to fuzzing attacks.

For spoofing drive gear attacks, our model achieved a precision of 0.9996 and a recall of 0.9987, resulting in an F1-score of 0.9991. It outperforms the DCNN model in precision, F1 score, and accuracy but underperforms in recall, for which the DCNN shows the highest score of 0.9996. Although our model is not superior in any metrics, it shows that competitive detection performance closely matches the LSTM's performance, which had the best precision, F1-score, and accuracy scores. These results indicate our model's capability to effectively detect complex spoofing attacks, though slight enhancements are required by model tuning for further improvements.

In the case of RPM spoofing attacks, the DCNN model shows the best detection performance in all metrics with an F1-score of 0.9996 and an accuracy of 0.9995. Our model recorded a precision of 0.9997, while both the DCNN and the LSTM models achieved 0.9999. Although it slightly lags compared to other deep learning models, our model shows an F1-score and accuracy of 0.9992 and 0.9991, respectively.

Across all tested scenarios, our model consistently achieved competitive detection performance compared to the DCNN and the LSTM models. Especially, our model outperforms the other models in detecting DoS attacks with the highest accuracy of 0.9999. However, in other attack scenarios, it underperforms compared to the DCNN and LSTM models. These comparative experiment results demonstrate the proposed model's capability for intrusion detection in in-vehicle networks, while the analysis also points to potential improvements, particularly in enhancing recall without compromising the precision to achieve a more balanced detection performance.

Computational Cost

Test Environment We performed all the experiments in a cloud computing environment provided by Google Colab. It helps us ensure a consistent and controlled setting. This platform provides both a CPU-only environment and a GPU-accelerated environment. The CPU used was an Intel Xeon CPU of 2.20GHz. The Nvidia Tesla T4 with 15.36 GB of GDDR6 memory was utilized for GPU-accelerated tests. In our comprehensive assessment of computational cost, we evaluated the models across various metrics: total parameters inside the neural network, memory usage, and computational times on both CPU and GPU environments.

Table 3: Comparative Computational Cost of Intrusion Detection Models

Model	Total Parameters	Memory Usage	CPU Time (ms)	GPU Time (ms)
DCNN (Song et al., 2020)	2,192,353	8.36 MB	108 ms ± 35.4	67.5 ms ± 9.34
LSTM (Nayak et al., 2023)	56,449	220.50 KB	131 ms ± 63.2	69.2 ms ± 12.5
Ours	5,409	21.13 KB	82.5 ms ± 12.6	63.8 ms ± 10.9

The DCNN model has 2,192,353 parameters and occupies 8.36 MB of memory. The LSTM model has 56,449 parameters and requires 220.50 KB of memory. Our model has only 5,409 parameters and requires a memory of just 21.13 KB. It has only about 10% more parameters and memory usage than the LSTM model.

To assess computational performance, we measured the average inference time of the models. Interestingly, the DCNN, the heaviest model, has over 2 million parameters, requires the most extensive memory, and processes data on the CPU in an average time of 108 ms. In contrast, the LSTM model is somewhat slower, averaging 131 ms. This is probably due to the nature of the LSTM's recurrent architecture. Our model shows an average interference time of 82.5 ms, relatively faster than the DCNN and LSTM.

In the GPU-accelerated environment, the average inference time of the DCNN and LSTM is 67.5 ms and 69.2 ms, respectively. Our model shows an average inference time of 63.8 ms. Although it still shows the highest inference time, there is no notable difference in the GPU-accelerated computing environment. These performance metrics are summarized in Table 3, clearly illustrating that our model uses fewer resources and achieves faster inference speeds on both computational platforms. Such efficiency is particularly suitable for the automotive environment, where rapid response times and minimal resource consumption are essential for effective operation.

5 Conclusion

This study proposed the lightweight Intrusion Detection System (IDS) for securing Controller Area Network (CAN) in automotive systems. It utilizes Natural Language Processing (NLP) techniques, specifically word embedding, to create the embedding matrix of CAN IDs that contextually converts each CAN ID into a multi-dimensional embedding vector. By leveraging the pre-trained embedding matrix, the proposed intrusion detection system could use a shallow neural network with a single hidden layer to achieve low memory usage and fast inference speed while keeping competitive intrusion detection performance.

While our model shows strengths in several aspects, it has some limitations. In some attack scenarios, the proposed model showed lower recall scores, requiring further sensitivity improvement. Moreover, pre-trained word embeddings may limit its detection capabilities to entirely novel attacks, which are not in the training data.

To overcome these limitations, future research will concentrate on the following aspects:

Expansion of Datasets Collecting more comprehensive attack scenario data can help the model be robust against novel attacks, accomplishing generalized attack detection performance.

Advanced Models Combining word embeddings with other types of neural network architectures, such as transformers, which show promising performance, might provide a richer understanding of the sequential data in CAN traffic, potentially improving precision and recall.

Real-World Testing Conducting a test in a real-world automotive environment would help validate the model's effectiveness in practical settings.

Acknowledgement

This work was supported by the Technology Innovation Program (P0023522, HRD Program for Industrial Innovation) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

[1] Ahmed, I., Jeon, G., & Ahmad, A. (2021). Deep learning-based intrusion detection system for internet of vehicles. *IEEE Consumer Electronics Magazine*, *12*(1), 117-123.

[2] Alsarhan, A., Al-Ghuwairi, A. R., Almalkawi, I. T., Alauthman, M., & Al-Dubai, A. (2021). Machine learning-driven optimization for intrusion detection in smart vehicular networks. *Wireless Personal Communications*, *117*, 3129-3152.

[3] Bobir, A. O., Askariy, M., Otabek, Y. Y., Nodir, R. K., Rakhima, A., Zukhra, Z. Y., & Sherzod, A. A. (2024). Utilizing Deep Learning and the Internet of Things to Monitor the Health of Aquatic Ecosystems to Conserve Biodiversity. *Natural and Engineering Sciences, 9*(1), 72-83.

[4] Chandrababu, M., & Senthilkumar, K. (2023). Strengthening IoT Intrusion Detection through the HOPNET Model. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(3), 89-102.

[5] Elshrkawey, M., Alalfi, M., & Al-Mahdi, H. (2021). An enhanced intrusion detection system based on multi-layer feature reduction for probe and dos attacks. *Journal of Internet Services and Information Security, 11*(4), 61-78.

[6] Gou, W., Zhang, H., & Zhang, R. (2023). Multi-Classification and Tree-Based Ensemble Network for the Intrusion Detection System in the Internet of Vehicles. *Sensors*, *23*(21), 8788. https://doi.org/10.3390/s23218788

[7] Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). Long short-term memory-based intrusion detection system for in-vehicle controller area network bus. *In IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 10-17.

[8] Islam, R., Refat, R. U. D., Yerram, S. M., & Malik, H. (2020). Graph-based intrusion detection system for controller area networks. *IEEE Transactions on Intelligent Transportation Systems, 23*(3), 1727-1736.

[9] Javed, A. R., Ur Rehman, S., Khan, M. U., Alazab, M., & Reddy, T. (2021). CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE transactions on network science and engineering*, *8*(2), 1456-1466.

[10] Kalkan, S. C., & Sahingoz, O. K. (2020). In-vehicle intrusion detection system on controller area network with machine learning models. *In IEEE 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-6.

[11] Kongjian, Q., Hao, S., Kexun, H., & Baizheng, W. (2020). Machine-based automotive intrusion detection Technology. *In IEEE International Conference on Computer Science and Management Technology (ICCSMT)*, 65-68.

[12] Kostić, S., Nikolić, N., & Malbašić, V. (2021). Development of A Methodology for Stability Monitoring of a Defense Embankment Loaded with Frequent Traffic: The Example of the Kovin Mine. *Archives for Technical Sciences*, *2*(25), 29–42.

[13] Kutlu, Y., & Camgözlü, Y. (2021). Detection of coronavirus disease (COVID-19) from X-ray images using deep convolutional neural networks. *Natural and Engineering Sciences*, *6*(1), 60-74.

[14] Lampe, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications. *Expert Systems with Applications*, *221*, 119771. https://doi.org/10.1016/j.eswa.2023.119771

[15] Longari, S., Valcarcel, D. H. N., Zago, M., Carminati, M., & Zanero, S. (2020). CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network. *IEEE Transactions on Network and Service Management*, *18*(2), 1913-1924.

[16] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[17] Muralidharan, J. (2020). Wideband Patch Antenna for Military Applications. *National Journal of Antennas and Propagation (NJAP), 2*(1), 25-30.

[18] Nayak, P., Vanditha, M., & Tippannavar, S. (2023). Smart Intrusion Detection System for CAN Network Implemented using LSTM Strategy. *International Journal of Innovations in Research, Engineering and Advanced Technology*, *10*(03), 98-105.

[19] Nikitina, V., Raúl, A. S., Miguel, A. T. R., Walter, A. C., Anibal, M. B., Maria, D. R. H., & Jacqueline, C. P. (2023). Enhancing Security in Mobile Ad Hoc Networks: Enhanced Particle Swarm Optimization-driven Intrusion Detection and Secure Routing Algorithm. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *14*(3), 77-88.

[20] Prasad Babu, P., & Vasumathi, A. (2023). Role of Artificial Intelligence in Project Efficiency Mediating with Perceived Organizational Support in the Indian IT Sector. *Indian Journal of Information Sources and Services*, *13*(2), 39–45.

[21] Rajesh, D., Giji Kiruba, D., & Ramesh, D. (2023). Energy Proficient Secure Clustered Protocol in Mobile Wireless Sensor Network Utilizing Blue Brain Technology. *Indian Journal of Information Sources and Services*, *13*(2), 30–38.

[22] Song, H. M., Woo, J., & Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications*, *21*, 100198.

[23] Sreenivasu, M., Kumar, U. V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. *Journal of VLSI Circuits and Systems*, *4*(2), 1-4.

[24] Srivastava, K. K., Tripathi, A., & Tiwari, A. K. (2013). Secure Data Transmission in AODV Routing Protocol. *International Journal of Communication and Computer Technologies (IJCCTS)*, *1*(2), 111-113.

[25] Stevovic, S., Jovanovic, J., & Djuric, D. (2018). Energy Efficiency in Urban Areas by Innovative Permacultural Design. *Archives for Technical Sciences*, *2*(19), 65–74.

[26] Tanksale, V. (2020). Anomaly detection for controller area networks using long short-term memory. *IEEE Open Journal of Intelligent Transportation Systems*, *1*, 253-265.

[27] Yang, J., Wang, L., & Shakya, S. (2022). Modelling Network Traffic and Exploiting Encrypted Packets to Detect Stepping-stone Intrusions. *Journal of Internet Services and Information Security*, *12*(1), 2-25.

[28] Yu, Y., Zeng, X., Xue, X., & Ma, J. (2022). LSTM-based intrusion detection system for VANETs: A time series classification approach to false message detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 23906-23918.

[29] Yu, Z., Liu, Y., Xie, G., Li, R., Liu, S., & Yang, L. T. (2022). TCE-IDS: Time interval conditional entropy-based intrusion detection system for automotive controller area networks. *IEEE Transactions on Industrial Informatics*, *19*(2), 1185-1195.

Authors Biography

Harang Kim is a dedicated researcher and PhD student in the Department of Computer Science at Dankook University, Yoingin, South Korea. He received his master’s degree in information security, Graduate School of Information Security, Korea University. His research interests are automotive cybersecurity, artificial intelligence, and data analysis-based cybersecurity applications.

Hyun Min Song is an Assistant Professor at the Department of Cybersecurity at Dankook University, Yongin, South Korea. He received his PhD degree in information security from Graduate School of Information Security, Korea University, Seoul, South Korea. His research interests include intrusion and anomaly detection, solving security problems in various IT systems based on the data analysis and artificial intelligence.