RAJAPAKSHA, S., KALUTARAGE, H., AL-KADRI, M.O., MADZUDZO, G. and PETROVSKI, A.V. 2022. Keep the moving vehicle secure: context-aware intrusion detection system for in-vehicle CAN bus security. In *Jančárková, T., Visky, G. and Winther, I. (eds.). Proceedings of 14th International conference on Cyber conflict 2022 (CyCon 2022): keep moving, 31 May - 3 June 2022, Tallinn, Estonia.* Tallinn: CCDCOE, pages 309-330. Hosted on IEEE Xplore [online]. Available from: https://doi.org/10.23919/CyCon55549.2022.9811048

Keep the moving vehicle secure: context-aware intrusion detection system for in-vehicle CAN bus security.

RAJAPAKSHA, S., KALUTARAGE, H., AL-KADRI, M.O., MADZUDZO, G. and PETROVSKI, A.V.

2022

Keep the Moving Vehicle Secure: Context-Aware Intrusion Detection System for In-Vehicle CAN Bus Security

Sampath Rajapaksha
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
s.rajapaksha@rgu.ac.uk

Harsha Kalutarage
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
h.kalutarage@rgu.ac.uk

M. Omar Al-Kadri
School of Computing and Digital Technologies
Birmingham City University
Birmingham, United Kingdom
Omar.alkadri@bcu.ac.uk

Garikayi Madzudzo
Horiba-MIRA
Coventry, United Kingdom
Garikayi.madzudzo@horiba-mira.com

Andrei V. Petrovski
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
a.petrovski@rgu.ac.uk

Abstract: The growth of information technologies has driven the development of the transportation sector, including connected and autonomous vehicles. Due to its communication capabilities, the controller area network (CAN) is the most widely used in-vehicle communication protocol. However, CAN lacks suitable security mechanisms such as message authentication and encryption. This makes the CAN bus vulnerable to numerous cyberattacks. Not only are these attacks a threat the information security and privacy, but they can also directly affect the safety of drivers, passengers and the surrounding environment of the moving vehicles. This paper presents CAN-CID, a context-aware intrusion detection system (IDS) to detect cyberattacks on the CAN bus, which would be suitable for deployment in automobiles including military vehicles, passenger cars, commercial vehicles and other CAN-based applications such as aerospace, industrial automation and medical equipment. CAN-CID is an ensemble model of a gated recurrent unit (GRU) network and a time-based model. A GRU algorithm works by learning to predict the centre ID of a CAN ID sequence, and ID-based probabilistic thresholds are used to identify anomalous IDs, whereas the time-based model identifies anomalous IDs using time-based thresholds. The number of anomalies compared to the total number of IDs over an observation window is used to classify the window status as anomalous or benign. The proposed model uses only benign data for training and threshold estimation, avoiding the need to collect realistic attack data to train the algorithm. The performance of the CAN-CID model was tested against three datasets over a range of 16 attacks, including fabrication and more sophisticated masquerade attacks. The CAN-CID model achieved an F1-Score of over 99% for 13 of those attacks and outperformed benchmark models from the literature for all attacks, with near real-time detection latency.

Keywords: *controller area network, anomaly detection, vehicle networks, CAN bus*

1. Introduction

Modern automobiles are becoming complex and highly connected to provide safe, efficient and intelligent services to users. To facilitate these services, automobiles are equipped with multiple networks and communication devices and a range of sensors, actuators, cameras and microprocessor-based electronic control units (ECUs) [1]. Modern vehicles run software that exceed 100 million lines of code, and future vehicles will require 200 to 300 million lines of code [2]. These software run on up to 100 ECUs [3] that are connected to a controller area network (CAN) which is considered to be the de-facto network protocol for in-vehicle communication [1]. The CAN bus is a message-based protocol commonly used in vehicles, aerospace, industrial automation and medical equipment due to several benefits such as being low cost, speedy, lightweight and robust [4]. Despite these benefits, the CAN bus lacks security measures, especially given the absence of authentication, an ID-based priority system, broadcast transmission and lack of encryption. Increased connectivity and complexity and CAN bus security flaws have made modern vehicles vulnerable to cyberattacks. In fact, security researchers have demonstrated the capability of attacks against modern vehicles by compromising the CAN networks of various vehicle brands [5], [6], [7]. These researchers have shown that it is possible to implement CAN message injection attacks remotely and take physical control of these vehicles. An attacker obtaining physical control of a moving vehicle will directly affect the safety of drivers, passengers and the surrounding environment of the vehicle. The security of modern automobiles is a major concern for automotive manufacturers; therefore, they are seeking security measures to protect against such attacks [8], [9].

Developing an in-vehicle IDS for widespread adoption with a high detection capability is challenging due to the lack of knowledge about the CAN data specifications [10]. Generally, specifications of CAN messages are stored in a database-like file known as the database CAN (DBC), a confidential source of proprietary information, access to which is usually restricted to the vehicle manufacturer. Depending on the number of ECUs, the CAN bus transmits about 2000 frames per second [11]. This demands an IDS with real-time or near real-time detection capability under a computationally constrained environment. Cyberattackers could use various types of attacks (e.g. injection and masquerade attacks) that alter the different data fields of CAN messages to compromise the in-vehicle network. This is another challenge that limits the detection and generalization capabilities of an IDS. In addition, many events that arise in a vehicle could be considered anomalies despite being legitimate driving scenarios. For example, an emergency brake or sudden steering wheel turn while driving at 70 mph would be considered anomalous in normal driving scenarios. These kinds of benign anomalous behaviours could produce a significant number of false positives. Hence, knowledge of the context of the CAN sequences is vital to distinguish benign anomalies from potential attack scenarios. To successfully deal with the aforementioned challenges, this paper proposes CAN-CID (CAN Centre ID prediction) a novel context-aware ensemble IDS for the CAN bus based on natural language processing (NLP) and time-based techniques.

The main contributions of this paper can be summarized as follows.

1. CAN-CID uses only benign data to train the model and estimate thresholds. This avoids the need to collect real attack data to train the algorithm. It is significantly easier and safer to collect benign CAN data from real vehicles than to collect attack data. Further, using only benign data (one-class) during the training process improves the generalization capability of the algorithm.
2. Probability-based thresholds were estimated for each ID using only benign training data. Minimum thresholds were selected with the aim to minimize false positives which will help to improve the overall accuracy of the ensemble model.
3. CAN-CID uses a one-layer shallow GRU network to detect anomalous ID sequences. Hence it is lightweight, and detection latency is very low (10 ms for a 100 ms window). This makes the proposed solution suitable to deploy in real vehicles.

The rest of this paper is structured as follows: Section 2 presents the related work. Section 3 provides the background, including CAN data analysis. The proposed algorithm is explained in section 4. In section 5, the experiment results and performance evaluations are presented. Finally, section 6 concludes the paper.

2. Related Works

Recent experiments focusing on attacks against modern automobiles [5], [6] have motivated research into countermeasures against in-vehicle network attacks. The majority of these works have focused on securing the CAN bus, as notable experimental attacks targeted the vulnerabilities of the CAN bus [6], [7]. In [11], the authors proposed a specification-based IDS for in-vehicle network intrusion detection by extracting design specifications of CAN messages. The IDS proposed in [12] used unique voltage signals generated by ECUs as features of the deep support vector domain description model. However, both of these models [11], [12] have a low generalization capability as they require specific knowledge of CAN data. A one-class compound classifier was used in [13] to detect CAN bus attacks. But this detected only 45–65% of attacks. The authors then suggested an ensemble of detection methods to overcome the problems that arise when using only one classifier. In [14], the authors proposed a long short-term memory (LSTM) autoencoder to detect CAN bus anomalies. This was trained using a payload of legitimate CAN frames. Reconstruction error was used to distinguish benign from malicious frames. A major limitation of this model, however, was the slow computation time due to the complex model architecture. LSTM-based deep learning model, which utilized the linear embedding of the CAN payload, was used in [15] to detect contextual anomalies in the payload. The authors examined the effect of context by removing the embedding layer of the proposed model. They observed that context and embedding helped to slightly improve the performance. However, this model recorded only around 95% accuracy for all attacks on one dataset. CAN payload signals were selected as the features of the deep learning-based IDS proposed in [16]. Similarly, the authors in [17] used sensor values in their deep neural network-based IDS. However, both of these approaches [16], [17] require the DBC files or knowledge about the CAN payload, which could limit the generalization capability of the proposed algorithms.

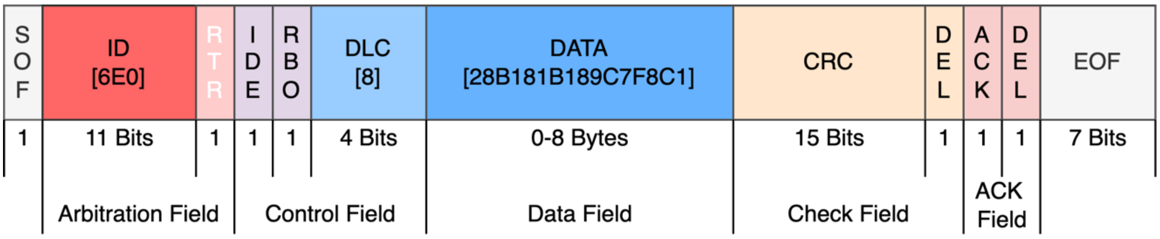
Frequency or time-based IDSs utilize the timing of CAN frames or the sequential nature of the IDs. In [18], the authors developed a context-aware anomaly detector for monitoring cyberattacks on the CAN bus using sequence modelling. The authors of [19] proposed an anomaly detection algorithm by modelling the normal behaviour of the CAN bus considering the recurring pattern of CAN IDs. This is equivalent to 2-grams in the N-gram based model used in [18]. While N-gram based algorithms can capture the context, this often leads to high computational overhead as N increases. A time-based IDS was proposed by [20] to detect CAN injection attacks. In [21], the authors used an LSTM model to predict the next ID and compare it with the actual ID to identify anomalous frames. However, this approach achieved only 60% accuracy. A CAN bus attack detection framework was proposed by [22] utilising both a rule-based and a supervised LSTM model. This ensemble model outperformed the individual models. In general, the deep learning-based IDSs discussed above demonstrated a higher detection capability than the other models. However, supervised learning-based models might have low generalization capability to other attacks and vehicles as they learn the attack pattern of the particular dataset. Further, these models have high detection latency due to their complex deep learning architecture. To address these problems, this work presents a lightweight ensemble model that uses a shallow neural network.

3. Background

A. Controller Area Network (CAN Bus)

CAN is a broadcast-based communication protocol developed by Bosch for in-vehicle communication [23]. ECUs of modern vehicles communicate using high-speed and low-speed CAN buses as their network protocol. Time-critical modules such as engine control and transmission control are connected to a high-speed CAN bus, whereas less time-critical modules such as door control and light control are connected to a low-speed CAN bus. A CAN bus data frame includes several fields: the CAN ID (arbitration field) is used to prioritize the messages and is capable of handling concurrent messages; a CAN payload contains the actual information (data) that is to be transmitted over the network; and other fields include start of frame (SOF), control field (DLC), cyclic redundancy code (CRC), acknowledge field (ACK), and end of frame (EOF). These fields are depicted in Figure 1, with their respective bit-lengths. When a node (ECU) is ready to transmit a frame, it checks the status of the bus, and if the bus is idle, it transmits the frame. The addresses of the transmitting node and the receiving node are not included in the frame. Instead, it uses CAN IDs unique to the transmitting nodes. As a result of the broadcast nature of the network, all nodes in the CAN network can receive the frame. Based on the ID of the frame, other nodes in the network decide to accept or ignore the frame. The priority-based arbitration scheme ensures that the highest priority IDs (lower IDs) get bus access when multiple nodes simultaneously transmit frames onto the CAN bus. The lowest priority IDs, on the other hand, must wait until the bus becomes idle.

Figure 1: CAN bus data frame. Example values are given for ID, DLC and data fields



B. CAN Bus Vulnerabilities and Attacks

The CAN bus is designed to provide robust, efficient, simple and low-cost in-vehicle communication without paying much attention to security-related features [13]. Therefore, by design, it is vulnerable to cyberattacks. Since the CAN bus uses no authentication, any node could transmit a message with an ID that belongs to another node. In addition, CAN frames are not encrypted due to real-time communication requirements. This allows attackers to collect and analyse CAN data (via sniffing). The broadcast nature of the CAN bus also transmits frames to all nodes connected to the CAN bus. Therefore, by utilising this property, a compromised ECU can not only monitor and listen to all CAN frames transmitted through the CAN bus but also send any frame to the network. Furthermore, attackers can use the ID-based priority scheme to inject their messages with the highest priority IDs to create a denial-of-service (DoS) attack, consequently making communication services unavailable to other IDs.

Some of the common CAN attack types are: DoS [24], fuzzing [25], replay [24], spoofing [26] and masquerade attacks [27]. In a fuzzing attack, a large number of random frames are injected into the CAN bus. Replay attacks re-send previously recorded frames at different times. When an attacker targets (injects) frames with specific CAN IDs, it is called a spoofing attack. In a masquerade attack, a compromised node impersonates another node to send malicious frames. All of these attacks have the potential to cause unexpected or harmful effects to a vehicle depending on the attacker's purpose.

C. CAN Bus Data Analysis

We analysed CAN ID data to understand the anomalous traffic patterns, both benign and malicious. To do this, we used a publicly available dataset, the Real ORNL Automotive Dynamometer (ROAD) CAN Intrusion dataset [10]. Figure 2 shows a five-second snapshot of a targeted ID attack. In this attack, the targeted ID is 0D0. What stands out in this figure is the periodic behaviour of the IDs. Each node transmits frames at a fixed interval, as observed in [6]. In the ROAD dataset, 104 out of 106 IDs exhibit similar behaviour. However, the injected ID causes a change in this pattern during the targeted ID attack. This can be observed in the shaded area (attack period) for 0D0. This changes the fixed transmission interval of IDs compared to that of the period of normal driving. In addition, it could create new ID sequences resulting from new frames appearing in an unusual context. For example, it might introduce a new ID sequence, such as '6E0 0D0 0D0', which was not observed during normal driving conditions.

Figure 2: Frame transmission of a targeted ID (0D0) attack. The shaded area represents the attack period. This represents only a subset of the 106 CAN IDs

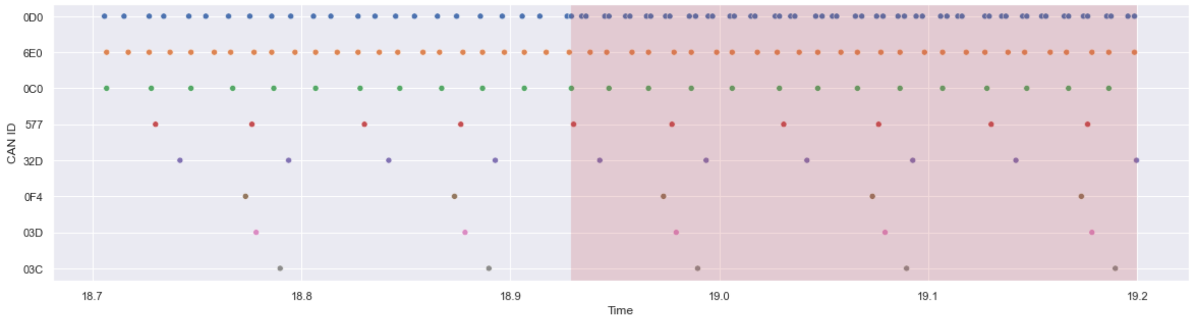
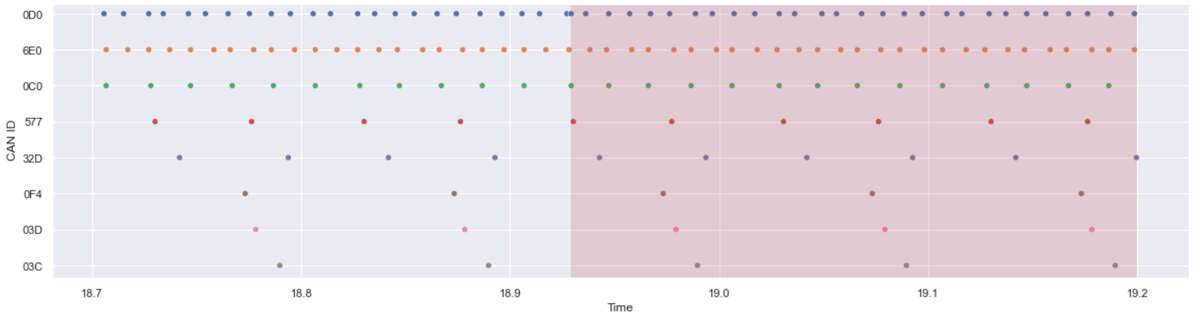


Figure 3 shows a five-second snapshot of a masquerade attack for the same ID (0D0). In this case, it does not significantly change the ID transmission frequency [10]. However, a masquerade attack might cause a slight deviation (shift of time) in the frame transmission time due to the difficulty of time synchronization with the legitimate ECU [28]. In addition, since a masquerade attack stops the frame transmission of a legitimate ECU, there might be a brief period where there is no frame transmitted with the targeted ID. A CAN bus transmits a large number of messages per second. Therefore, even a slight deviation from the normal driving scenario could create new ID sequences. For example, ID 0D0 might have the sequence 'ODO 6E0 0C0' during normal driving, whereas a slight time shift or absence of frames could create a new sequence of '6E0 0D0 0C0'. This behaviour (frequency and sequence change) can be observed for all injection and masquerade attacks for the ROAD dataset.

Figure 3: Frame transmission of a masquerade attack (0D0). The shaded area represents the attack period. This represents only a subset of the 106 CAN IDs



After analysing both benign and attack CAN traffic, our main finding is that most CAN IDs exhibit periodic behaviour that creates a finite set of ID sequences for a fixed window size (e.g. a window of ten consecutive IDs). Attacks on the CAN bus are likely to change the periodic behaviour of the IDs and hence create new sequences. In addition, injection attacks change the time between the consecutive attack IDs. Carefully trained machine learning algorithms can detect these subtle changes in CAN ID streams. These findings provide the basis for the proposed IDS.

4. Proposed CAN-CID Model

A. Threat Model and Datasets

In this work, we used the ROAD [10] dataset to test the proposed model. Additionally, to evaluate the generalization capability of the model, we used two other publicly available datasets, the car-hacking dataset for intrusion detection (HCRL CH) [29] and the survival analysis dataset for automobile IDS (HCRL SA) [30]. The ROAD dataset is considered the first open CAN bus dataset with advanced types of real attacks that have physically verified effects on the vehicle [10]. Data was collected through the OBD-II port in a fully compromised ECU mode while driving the vehicle on a dynamometer or on the road. The dataset includes 12 ambient (benign) datasets representing different driving activities, including drive, accelerate, decelerate, reverse, brake, cruise control, turn signals and anomalous but benign driving activities such as unbuckling a seatbelt and opening doors while driving. Attacks are categorized as fabrication attacks, suspension attacks and masquerade attacks. Fabrication attacks include fuzzing and targeted ID attacks. Attacks shown in Table I were selected to investigate the algorithm’s detection capability.

Table I: High-frequency injection (fabrication) attacks on the ROAD dataset

Attack	Attack technique	Consequence
Fuzzing	Inject random IDs and arbitrary payloads	Wide variety of unexpected results
Correlated signal attack	Inject false wheel speed values (ID - 6E0)	Stop the car due to different pairwise wheel speeds
Max speedometer attack	Change one byte of payload to maximum (FF) value (ID-0D0)	Display false speedometer value
Reverse light on attack	Change one bit of payload (ID-0D0)	Reverse lights do not reflect what gear the car is using
Reverse light off attack	Change one bit of payload (ID-0D0)	Reverse lights do not reflect what gear the car is using

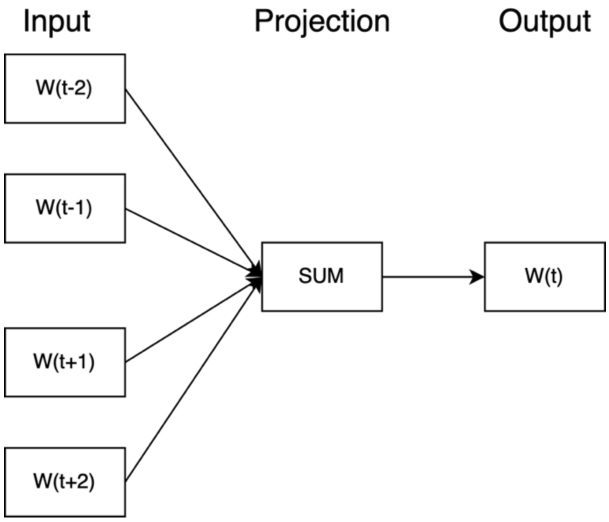
For fabrication attacks, the attacker injects a frame with a targeted ID immediately after a legitimate frame appears. The aim of this is to get the vehicle to ignore the legitimate message and accept the injected frame to change the vehicle state. In addition to the above attacks, this dataset includes a masquerade version of each ID fabrication attack. The masquerade version removes the legitimate target ID frames relevant to each injected frame in post-processing to simulate a masquerade attack. These realistic attacks required message injections for each attack. Such an approach was trialled by [6] in their experimental attacks. Message injections lead to changes in the transmission interval of the targeted ID and new sequences being created. Hence, even a payload attack might require an attacker to inject frames into the CAN bus [6]. This requirement makes an ID sequence or frequency-based model suitable for detecting the majority of payload attacks without using the payload related features. The HCRL CH dataset includes DoS, fuzzy and spoofing (RPM and gear) attacks, whereas the HCRL SA (KIA Soul) dataset includes flooding, fuzzy and malfunction (targeted ID) attacks.

B. CAN Centre ID Prediction Task

This work is inspired by the work of [18] and the continuous bag-of-words (CBOW) model architecture proposed by [31]. In [18], the authors used N-gram distributions to build the CAN ID sequence model. The underlying concept of this work is a mathematical model (n-gram) that can be trained to learn the

CAN message sequences and predict subsequent elements in the sequence. The authors showed that the occurrence of an event (ID) can be determined based on a short history. However, this might depend on the number of nodes in the network (equivalent to the number of unique words in a language). Thus for a larger number of nodes, a longer history may be required as a larger number of unique sequences could be created for the selected window size. N-gram models are inefficient for higher values of N because this will result in more combinations. This approach [18] is similar to the next word prediction task of NLP given the previous words (context). The Word2vec model proposed by [31] learns the word vectors (word embeddings) by learning to predict the centre (target) word given the context. This model architecture is shown in Figure 4. The CBOW model is expected to learn the word vectors representing the middle word's meaning and the context words. However, the main objective of the CBOW model is not to predict the words but to learn accurate word vectors that encode semantic relationships for all the words in the corpus. Then, the learned word vectors can be used in many language models with specific deep learning architectures.

Figure 4: Continuous bag-of-words (CBOW) architecture to predict the centre word given the previous and next words as the context



Using the target word's historical and future words as the input words improved the centre word prediction [31]. We expect the same behaviour for CAN ID sequences. To elaborate on this, as an example, take a driving scenario of a right-hand turn at an intersection. Possible events in the vehicle are activate signal lights, decelerate (apply brake), stop, accelerate and turn right. If we want to predict the third event, which is stop in this case, we can use only previous events as the context or use previous and future events as the context. These two tasks can be formulated as follows:

$$x_1 = \{activate\ signal\ lights, decelerate\}, y = \{stop\} \tag{1}$$

$$x_2 = \{activate\ signal\ lights, decelerate, accelerate, turn\ right\}, y = \{stop\} \tag{2}$$

The second task (equation 2) can be used to make the prediction (stop) with higher accuracy, as the number of possible events for the centre (middle) event will be equal to or fewer compared to the first task. For example, 'accelerate' would be another probable prediction for equation 1. But in the second task, given accelerate in the context, it will make the prediction of 'stop' more accurate. Therefore, we use the CBOW architecture to infer the context for CAN ID sequences. One limitation of the CBOW approach is that it must wait for a few messages to see if the target (centre) ID is malicious. However, considering the CAN ID transmission rate, this will be a minimal amount of time (around a 0.005 s delay for 10 IDs). Additionally, continuous message injections are required to execute such an attack

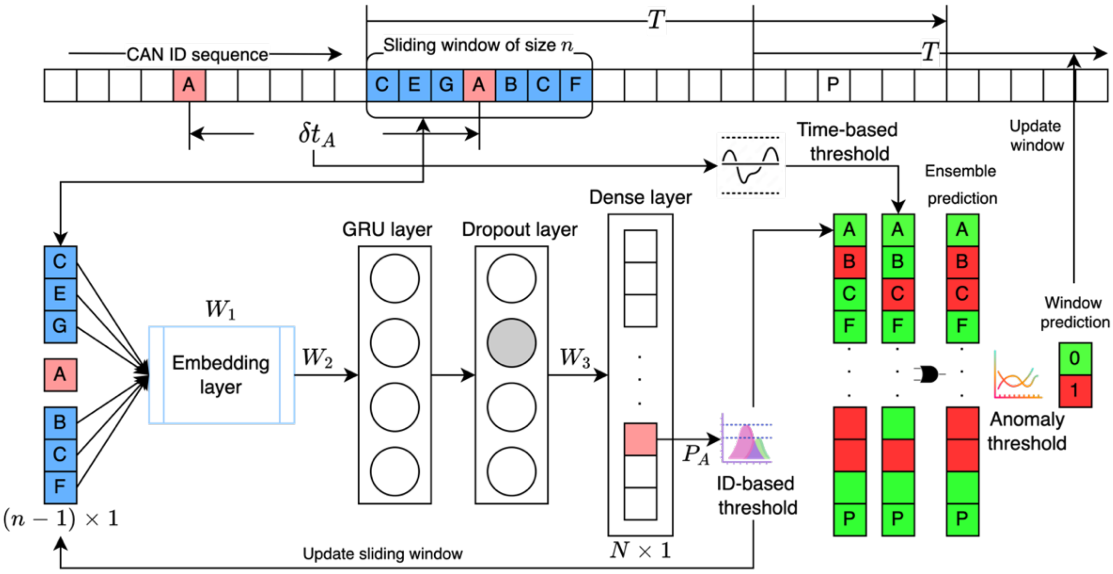
[6], which increases the chances of detecting the attack before the attacker can take physical control of the vehicle. Hence, CBOW is a viable option for detecting attacks in CAN ID sequences.

C. CAN-CID Architecture

The order of the words is not considered in the CBOW model. However, the order is highly important to identify anomalous frames in CAN ID sequences. A recurrent neural network (RNN) model can capture the temporal patterns of sequential data [32]. But RNN models do not have a long-term memory due to the vanishing gradient problem. To address this issue, LSTM was introduced. LSTM consists of three gates: input, forget, and output. In contrast, GRU is a variant of LSTM with only two gates: reset and update. The simple structure reduces the matrix multiplication, making GRU more computationally efficient with low memory overhead [32]. Due to these properties, which are ideal for resource-constrained environments, we use a GRU layer to capture the temporal pattern of CAN ID sequences.

Figure 5 represents the architecture of the proposed model. We use a sliding window of size n (number of IDs) within a large sliding window of size T (time), where n is an odd number. Let N be the total number of unique IDs. For the GRU-based model, the input to the embedding layer is a sequence of vectorised CAN IDs of size $(n-1)$. The centre (middle) ID $(n+1)/2$ is used as the target of the prediction. As mentioned earlier, a single GRU layer is used as the hidden layer to learn the temporal patterns. A dropout layer is used to reduce the overfitting and improve the model generalization capability. The output layer is a dense layer that outputs softmax probabilities for N IDs. During the training, W_1, W_2, and W_3 are updated using backpropagation. P_A which represents the softmax probability of the target ID (A) is compared with the pre-defined ID-based threshold. If the predicted probability is less than the threshold, the target ID is flagged as a weak anomaly; otherwise it is flagged as a benign ID. In the same way, the time-based model compares the time between two consecutive target IDs (δt_A) with the pre-defined time-based thresholds (minimum and maximum time). If δt_A is outside the thresholds, the current ID is flagged as a weak anomaly; otherwise it is flagged as a benign ID. This process continues for all IDs in window T. The OR operator is used to combine the two models as an ensemble model. Finally, an anomaly threshold is used to classify the window of time T as a malicious sequence or a benign sequence. This process continues for all IDs in the CAN ID stream by sliding the window of time T. The sliding window overlaps for $(n-1)$ IDs, to make predictions for the missing IDs from the previous window.

Figure 5: Ensemble model architecture



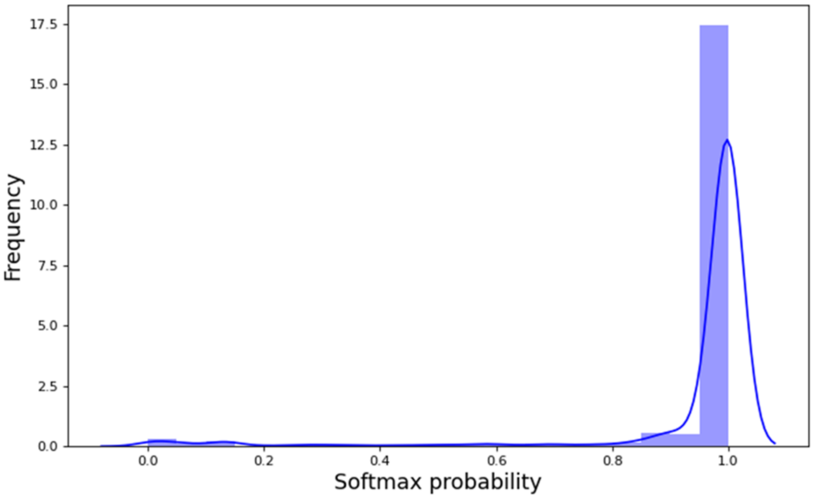
D. Threshold Estimation

The proposed model uses three thresholds.

1) *ID-Based Threshold*

A sample of the benign dataset was used to estimate thresholds. Softmax probabilities were calculated for all IDs in the benign sample. The minimum values of each ID were selected as the ID-based thresholds to minimize the false positives of the ensemble model. We assumed a zero probability of values less than the minimum values for benign data. Figure 6 shows a softmax probability distribution for a selected ID.

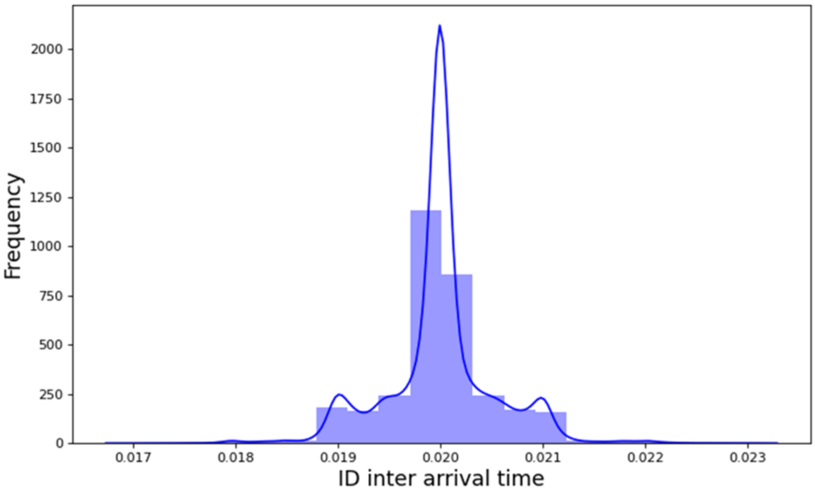
Figure 6: Softmax probability distribution of ID 580



2) *Time-Based Threshold*

We used the training dataset to define the time-based threshold. For each ID, we calculated the time difference between two consecutive frames for the benign dataset. Then, the minimum and maximum values for each ID were used as the minimum and maximum thresholds. Figure 7 shows an inter-arrival time distribution for a selected ID.

Figure 7: Inter-arrival time distribution for ID 580



3) *Anomaly Threshold*

We used ID-based and time-based thresholds to identify weak anomalies. Counting weak anomalies over a window (T) helps to minimize false positives. Hence, we defined the anomaly thresholds ($\mathcal{E}$) to identify the windows as attack or benign. We assigned labels for each window (0 for benign and 1 for attack) as the ground truth and used the same to evaluate the model performance. Equation 5 was used to calculate ground truth, and equation 6 was used for window prediction. The GRU-based model is likely to identify several frames besides the actual injected frame as weak anomalies because the injected frame might create several new (anomalous) CAN ID sequences.

$$X_g = \frac{Number\ of\ attack\ frames\ in\ T}{Total\ number\ of\ frames\ in\ T} \tag{3}$$

$$X_w = \frac{Number\ of\ weak\ anomalies\ in\ T}{Total\ number\ of\ frames\ in\ T} \tag{4}$$

$$Ground\ truth = \begin{cases} 1, X_g \geq \mathcal{E} \\ 0, X_g < \mathcal{E} \end{cases} \tag{5}$$

$$Window\ prediction = \begin{cases} 1, X_w \geq \mathcal{E} \\ 0, X_w < \mathcal{E} \end{cases} \tag{6}$$

5. EXPERIMENT RESULTS AND PERFORMANCE EVALUATION

A. *Experimental Setting*

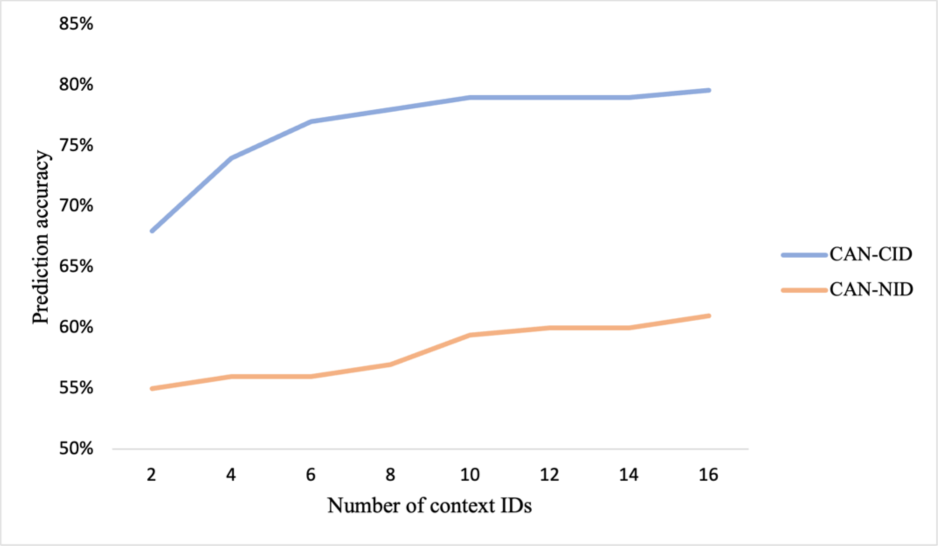
We selected ten benign datasets for training and two benign datasets for ID-based threshold estimation. To create the sliding window, ten IDs were selected from both sides of the target ID ($n = 10$). In addition, the attack datasets were split into 100-millisecond windows to identify attack windows, which can be considered smaller windows for near-real-time detection. This resulted in about 250 IDs per prediction window. To make the model more lightweight, only 32 GRU nodes were used in the hidden layer, followed by a 0.2 dropout layer. The ROAD, HCRL CH and HCRL SA datasets include 106, 27 and 45 nodes respectively (N). Based on a grid search, we observed that small anomaly thresholds (e.g. 0.01) work well with a large N and large anomaly thresholds (e.g. 0.1) work well with a small N. Hence, for the ROAD dataset, we set the anomaly threshold to 0.01, and for the HCRL datasets, the threshold was set to 0.1. A grid search was used for hyperparameter optimization, and the same parameters used in the ROAD dataset were also used for both HCRL datasets. We selected the best smallest hyperparameters for n, the number of GRU nodes and the embedding size. The proposed algorithm was implemented using Python 3.8 with TensorFlow and the Keras library. Experiments were run on a MacBook Pro 2.2 GHz Intel Core i7 with 16 GB RAM.

We compared CAN-CID with two baseline methods, that is, the N-gram-based model (N-gram) [18] and the transition matrix-based model (transition matrix) [19], where both models detect anomalies based on observed benign ID sequences. In addition, we used a variant of CAN-CID, referred to as CAN-NID (CAN Next ID prediction). CAN-NID is similar to CAN-CID, except the GRU model takes context IDs from one side (previous IDs). Optimized hyperparameters for the CAN-NID model include 16 previous IDs as the context, two hidden GRU layers with 128 nodes and a dense output layer with a softmax activation function. We also fine-tuned both baseline models for each dataset for a fair comparison with our model. To evaluate the model performance, we used F1-Score, false-positive rate (FPR) and false-negative rate (FNR) [33].

B. *Results and Discussion*

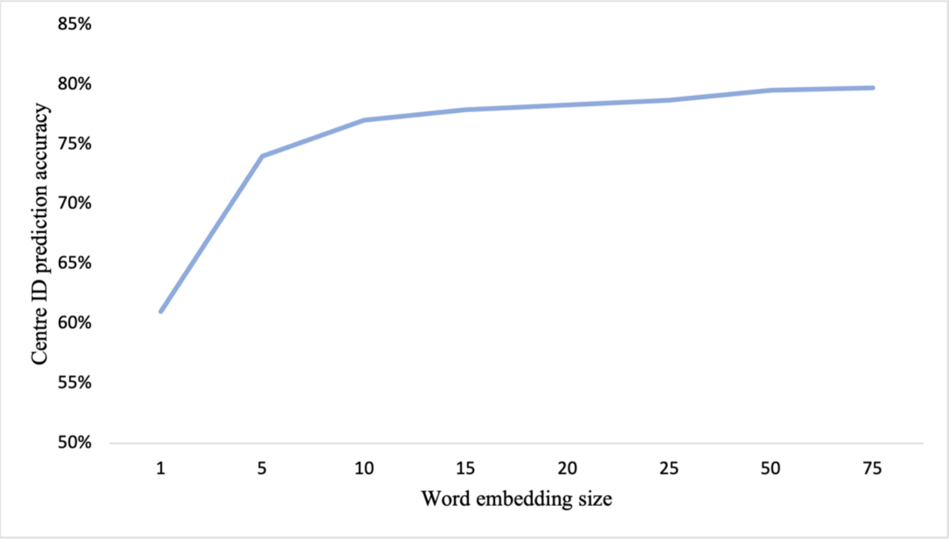
The detection accuracy of the GRU model of CAN-CID depends on the centre word prediction accuracy. We expect accurate predictions for benign frames and inaccurate predictions for attack frames to detect weak anomalies. To identify the optimum context from both sides of the centre ID, we tested various IDs by targeting the highest prediction accuracy for a sample from the benign dataset. Similarly, we tested a number of previous IDs for the CAN-NID GRU model. As shown in Figure 8, the CAN-CID model achieved 80% accuracy, whereas CAN-NID achieved a maximum of 61% accuracy for 16 context IDs. This highlights the effectiveness of the CBOW approach for CAN sequences. However, achieving 100% prediction accuracy is not realistic due to randomness incurred from jitters [28]. Considering the computational efficiency, we selected ten context words (79%) from each side for CAN-CID and 12 context words (60%) for CAN-NID.

Figure 8: Comparison of centre ID (CAN-CID model) and next ID (CAN-NID model) prediction accuracy



Word embedding size is another critical factor for accuracy and computational efficiency. Therefore, we tested the CAN-CID model with different embedding sizes as shown in Figure 9. We observed that accuracy improved up to an embedding size of 50. Therefore, we used 50 as the embedding size for both GRU models.

Figure 9: Accuracy improvement with word embedding size for the CAN-CID model



The F1-Scores, FPRs and FNRs of the CAN-CID and CAN-NID models and two baseline models are presented in Table II and III for the ROAD dataset. Table II and III report fabrication and masquerade attacks respectively, where the best performance (F1-Score) for each attack is shown in bold. As shown in the tables, the CAN-CID model outperforms the two baseline models for every attack and achieved a 100% F1-Score for six attacks. More importantly, this model achieved 0% or very small FPR and FNR values, which are critical aspects for an IDS. The CAN-NID model also outperformed baseline models for seven attacks. A fuzzing attack is relatively easy to detect due to illegal ID injection, and therefore, all models except the transition model achieved an F1-Score of 100%. However, correlated signal and correlated signal masquerade attack detection rates are low compared to other attacks. This might be because they target the second most frequent ID which has a slightly random transmission rate compared to other IDs. Therefore, it creates more sequences, which results in more valid sequences being created, even for attack frames. This is a limitation of the proposed model whereby it achieves a lower detection rate for attacks that target IDs with random transmission rates. However, a greater number of CAN IDs have fixed transmission rates [6], and therefore, CAN-CID can detect the majority of injection attacks. Further, since CAN IDs have fixed transmission rates, most ID sequences are likely to be independent of driving behaviours. This makes the model resilient to such changes. However, one of the limitations of the proposed model is that the CAN-CID model requires greater variety in the benign data to minimise the unseen CAN ID sequences and time intervals.

Table II: Comparison of CAN-CID and CAN-NID models and baseline models detection performance of fabrication attacks (ROAD dataset)

Attack	Model	F1-Score	FPR	FNR
Fuzzing	Transition matrix	71%	48%	0%
	N-gram	**100%**	0%	0%
	CAN-NID	**100%**	0%	0%
	CAN-CID	**100%**	0%	0%
Correlated signal	Transition matrix	90%	10%	6%
	N-gram	27%	0%	100%
	CAN-NID	78%	21%	42%
	CAN-CID	**91%**	2%	12%
Max speedometer	Transition matrix	79%	27%	0%
	N-gram	89%	0%	28%
	CAN-NID	**100%**	0%	0%
	CAN-CID	**100%**	0%	0%
Reverse light on	Transition matrix	63%	57%	0%
	N-gram	87%	0%	29%
	CAN-NID	94%	1%	2%
	CAN-CID	**100%**	0%	0%
Reverse light off	Transition matrix	92%	9%	0%
	N-gram	94%	0%	16%
	CAN-NID	**100%**	0%	7%
	CAN-CID	**100%**	0%	0%

Table III: Comparison of CAN-CID and CAN-NID models and baseline models detection performance of masquerade attacks (ROAD dataset)

Attack	Model	F1-Score	FPR	FNR
Correlated signal masquerade	Transition matrix	38%	10%	86%
	N-gram	27%	0%	100%
	CAN-NID	64%	22%	57%
	CAN-CID	**89%**	4%	10%
Max speedometer masquerade	Transition matrix	79%	27%	0%
	N-gram	99%	0%	1%
	CAN-NID	86%	0%	36%
	CAN-CID	**100%**	0%	0%
Reverse light on masquerade	Transition matrix	63%	57%	0%
	N-gram	87%	0%	29%
	CAN-NID	94%	1%	2%
	CAN-CID	**99%**	0%	1%
Reverse light off masquerade	Transition matrix	92%	9%	0%
	N-gram	94%	0%	16%
	CAN-NID	95%	0%	7%
	CAN-CID	**100%**	0%	0%

Figure 10 and 11 present comparisons of the attack detection performance of time-based and GRU models. Figure 10 shows fabrication attacks, whereas Figure 11 shows masquerade attacks. Typically, the time-based model is capable of detecting fabrication attacks with a higher F1-Score, whereas it fails to detect masquerade attacks. In contrast, the GRU model is capable of detecting both types of attacks with a higher F1-Score.

Figure 10: Time-based and GRU model detection performance for fabrication attacks

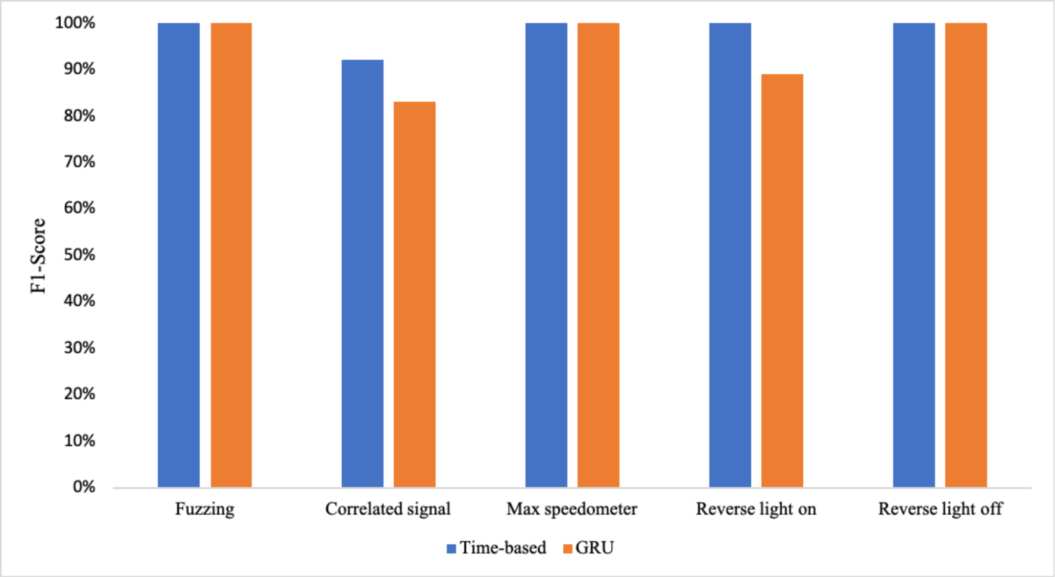
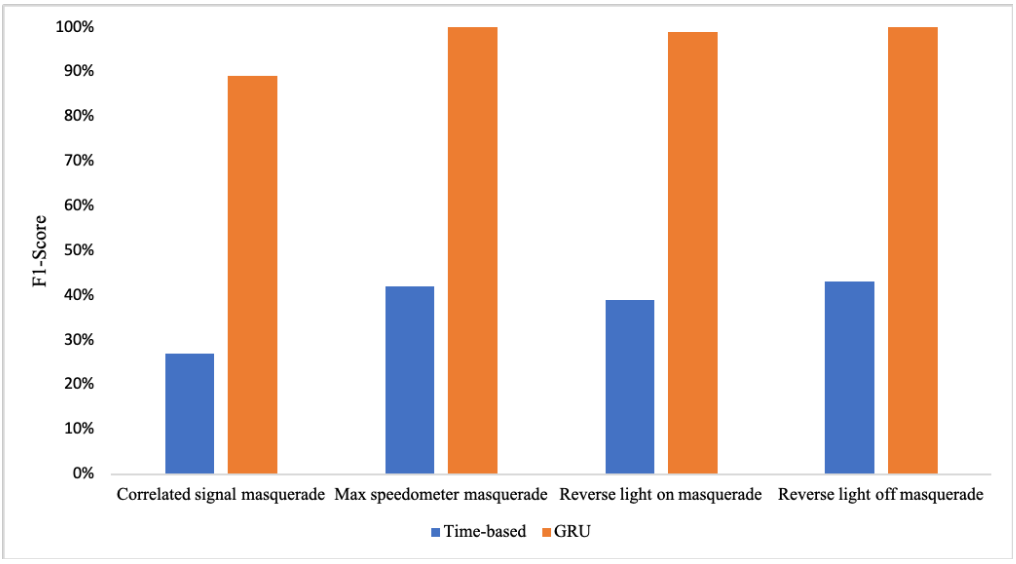


Figure 11: Time-based and GRU model detection performance for masquerade attacks



As mentioned earlier, we used two HCRL datasets to evaluate the generalization capability of the proposed model. The results from these two datasets are similar to the results we achieved from the ROAD dataset (see Table IV and V). The CAN-CID and CAN-NID models outperformed both baseline models. However, both baseline models showed comparatively better results for the HCRL datasets. This might be due to the HCRL datasets having a limited number of IDs, thus limiting the number of CAN ID sequences created compared to the ROAD dataset, which would help achieve higher predictability.

Table IV: Comparison of attack detection performance of the CAN-CID and CAN-NID models and the baseline models for the HCRL CH dataset

Attack	Model	F1-Score	FPR	FNR
DoS	Transition matrix	75%	52%	0%
	N-gram	96%	10%	0%
	CAN-NID	97%	6%	0%
	CAN-CID	**99%**	1%	0%
Fuzzy	Transition matrix	91%	20%	0%
	N-gram	94%	14%	0%
	CAN-NID	97%	6%	0%
	CAN-CID	**100%**	0%	0%
Gear Spoofing	Transition matrix	98%	4%	0%
	N-gram	98%	4%	0%
	CAN-NID	99%	1%	1%
	CAN-CID	**100%**	0%	0%
RPM Spoofing	Transition matrix	86%	28%	0%
	N-gram	98%	4%	0%
	CAN-NID	**99%**	0%	2%
	CAN-CID	**99%**	0%	2%

Table V: Comparison of attack detection performance of the CAN-CID and CAN-NID models and the baseline models for the HCRL SA dataset

Attack	Model	F1-Score	FPR	FNR
Flooding	Transition matrix	89%	28%	0%
	N-gram	99%	2%	0%
	CAN-NID	**100%**	0%	0%
	CAN-CID	**100%**	0%	0%
Fuzzy	Transition matrix	85%	28%	0%
	N-gram	99%	1%	0%
	CAN-NID	99%	1%	0%
	CAN-CID	**100%**	0%	0%
Malfunction	Transition matrix	68%	54%	0%
	N-gram	84%	28%	0%
	CAN-NID	91%	2%	17%
	CAN-CID	**96%**	0%	4%

Detection latency is another criterion that we focused on improving as it is vital for moving vehicles. Table VI compares average detection latency for CAN-CID, CAN-NID and the two baseline models. The IDS monitors CAN traffic for 100 ms and gives the prediction in 10 ms. CAN-CID outperformed CAN-NID and the two baseline models. The small amount of time required for monitoring and prediction allows the vehicle driver or the vehicle itself to take appropriate countermeasures. Therefore, considering the detection capability and latency, the proposed algorithm is a practically deployable solution to detect cyberattacks on the CAN bus. Furthermore, using the CAN ID and time as the only features of the ensemble model improves the detection latency in a resource-constrained environment. Additionally, this model is likely to have a better generalization capability than a payload-based model as data (payload) specifications might change significantly across different vehicle makes and models.

Table VI: Average detection latency comparison for a 100 ms prediction window

Model	Detection latency (ms)
Transition matrix	36
N-gram	452
CAN-NID	12
CAN-CID	**10**

6. Conclusion and Future Works

Increased connectivity and complexity in modern automobiles create more attack surfaces that could allow attackers to take control of automobiles. Cyberattacks on moving vehicles are highly dangerous and could result in serious injury or even deadly consequences. Therefore, there is a dire need to implement defence mechanisms against these attacks. Due to the complexity of CAN data and the different characteristics of different types of potential attacks, this work demonstrates that the solution requires an ensemble model with an optimized model for each field of CAN data.

Hence, we proposed CAN-CID, a novel context-aware ensemble IDS for CAN bus security. Our experiments showed that the ensemble model improved the overall attack detection performance and outperformed two baselines and a variant of the proposed model. Additionally, the proposed CAN-CID model has a low detection latency, which is necessary for a deployable in-vehicle IDS. We also identified potential future work to improve the model. We propose adding another model to the ensemble model to monitor the CAN payload and thus detect more advanced attacks, which would not change ID sequences or frequencies. Secondly, the IDS should be capable of adapting to new data. Therefore, we plan to work on introducing streaming learning capability. Finally, we plan to deploy the IDS and test it under real-world conditions. These additions to the proposed model will help keep moving vehicles secure from an even wider range of in-vehicle network cyberattacks.

REFERENCES

[1] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby and A. Mouzakitis, ‘Intrusion detection systems for intra-vehicle networks’, *IEEE Access*, vol. 7, pp. 21266–21289, 2019.

[2] R. N. Charette, ‘This Car Runs on Code’. Accessed: Nov. 28, 2021. [Online]. Available: https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code

[3] D. Moller and R. Hass, *Guide to Automotive Connectivity and Cybersecurity: Trends, Technologies, Innovations and Applications*, Springer, Cham, 2019.

[4] O. Avatefipour and H. Malik, ‘State-of-the-art survey on in-vehicle network communication (CAN-Bus) security and vulnerabilities’, *IJCSN*, vol. 6, no. 6, pp. 720–727, 2017.

[5] Z. Cai, A. Wang, W. Zhang, M. Gruffke and H. Schweppe, ‘0-days & Mitigations: Roadways to Exploit and Secure Connected BMW Cars’, in *Black Hat USA*, 2019.

[6] C. Miller and C. Valasek, ‘CAN Message Injection’, 28 June 2016. Accessed: Nov. 29, 2021. [Online]. Available: http://illmatics.com/can%20message%20injection.pdf

[7] S. Nie, L. Liu and Y. Du, ‘Free-fall: Hacking Tesla from wireless to CAN bus’, in *Black Hat USA*, 2017.

[8] M. Engstler, ‘Heavy On Connectivity, Light On Security: The Challenges Of Vehicle Manufacturers’, Jan. 15, 2021. Accessed: Nov. 29, 2021. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2021/01/15/heavy-on-connectivity-light-on-security-the-challenges-of-vehicle-manufacturers/?sh=68dda9247fc7

[9] F. Lambert, ‘Tesla is challenging hackers to crack its car, and it is putting ~$1 million on the line’, Jan. 10, 2020. Accessed: Nov. 29, 2021. [Online]. Available: https://electrek.co/2020/01/10/tesla-hacking-challenge/

[10] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay and F. L. Combs, ‘ROAD: The Real ORNL Automotive Dynamometer Controller Area Network Intrusion Detection Dataset (with a comprehensive CAN IDS dataset survey & guide)’, *arXiv preprint arXiv:2012*, vol. 14600, 2020.

[11] N. Salman and M. Bresch, ‘Design and implementation of an intrusion detection system (IDS) for in-vehicle networks’, M.S. thesis, Dept. Comp. Sci. Eng., Univ. Gothenburg, Sweden, 2017.

[12] Y. Xun, Y. Zhao and J. Liu, ‘VehicleEIDS: A Novel External Intrusion Detection System Based on Vehicle Voltage Signals’, *IEEE Internet of Things Journal*, 2021.

[13] A. Tomlinson, J. Bryans and S. A. Shaikh, ‘Using a one-class compound classifier to detect in-vehicle network attacks’, in *Proc. Genet. Evol. Comput. Conf. Companion*, 2018.

[14] S. Longari, M. Zago and S. Zanero, ‘CANnolo: An Anomaly Detection System Based on LSTM Autoencoders for Controller Area Network’, *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1913–1924, 2021.

[15] P. Balaji and M. Ghaderi, ‘NeuroCAN: Contextual Anomaly Detection in Controller Area Networks’, in *IEEE Int. Smart Cities Conf. (ISC2)*, 2021.

[16] 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, 2016, Art. no. e0155781.

[17] J. Zhang, F. Li, H. Zhang, R. Li and Y. Li, ‘Intrusion detection system using deep learning for in-vehicle security’, *Ad Hoc Netw.*, vol. 95, 2019, Art. no. 101974.

[18] H. K. Kalutarage, O. M. Al-Kadri, M. Cheah and G. Madzudzo, ‘Context-aware anomaly detector for monitoring cyber attacks on automotive CAN bus’, in *Proc. – CSCS 2019: ACM Comp. Sci. Cars Symp.*, 2019.

[19] M. Marchetti and D. Stabili, ‘Anomaly detection of CAN bus messages through analysis of ID sequences’, in *IEEE Intell. Veh. Symp. Proc. (IVS)*, 2017.

[20] D. H. Blevins, P. Moriano, R. A. Bridges, M. E. Verma, M. D. Iannacone and S. C. Hollifiel, ‘Time-Based CAN Intrusion Detection Benchmark’, *arXiv preprint arXiv:2101*, vol. 05781, 2021.

[21] A. K. Desta, S. Ohira, I. Arai and K. Fujikawa, ‘ID Sequence Analysis for Intrusion Detection in the CAN bus using Long Short Term Memory Networks’, in *2020 IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, *PerCom Workshops 2020*.

[22] S. Tariq, S. Lee, K. H. Kim and S. S. Woo, ‘CAN-ADF: The controller area network attack detection framework’, *Comput. Secur.*, vol. 94, 2020, Art. no. 101857.

[23] E. Aliwa, O. Rana, C. Perera and P. Burnap, ‘Cyberattacks and Countermeasures for In-Vehicle Networks’, *ACM Comput. Surv. (CSUR)*, vol. 54, no. 1, pp. 1–37, 2021.

[24] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson and H. Shacham, ‘Experimental security analysis of a modern automobile’, in *Proc. 31st IEEE Symp. Secur. and Priv.*, pp. 447–462, 2010.

[25] V. Chockalingam, I. Larson, D. Lin and S. Nofzinger, ‘Detecting Attacks on the CAN Protocol With Machine Learning’, in *Annu. EECS*, 2016.

[26] J. Dürrwang, J. Braun, M. Rumez, R. Kriesten and A. Pretschner, ‘Enhancement of automotive penetration testing with threat analyses results’, *SAE Int. J. Transp. Cybersecur. Priv.*, pp. 91–112, 2018.

[27] S. Woo, H. J. Jo and D. H. Lee, ‘A practical wireless attack on the connected car and security protocol for in-vehicle CAN’, *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, 2014.

[28] K.-T. Cho and K. G. Shin, ‘Error handling of in-vehicle networks makes them vulnerable’, in *Proc. ACM Conf. Comput. Comm. Secur.*, 2016.

[29] H. M. Song, J. Woo and H. K. Kim, ‘In-vehicle network intrusion detection using deep convolutional neural network’, *Veh. Commun.*, vol. 21, 2020, Art. no. 100198.

[30] M. L. Han, B. I. Kwak and H. K. Kim, ‘Anomaly intrusion detection method for vehicular networks based on survival analysis’, *Veh. Commun.*, vol. 14, pp. 52–63, 2018.

[31] T. Mikolov, K. Chen, G. Corrado and J. Dean, ‘Efficient estimation of word representations in vector space’, in *1st Int. Conf. Learn. Represent.*, *ICLR 2013 – Workshop Track Proc.*, 2013.

[32] S. Yang, X. Yu and Y. Zhou, ‘LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example’, in *Proc. – 2020 Int. Workshop Electron. Commun. Art. Intell.*, *IWECAI 2020*.

[33] O. Minawi, J. Whelan, A. Almehmadi and K. El-Khatib, ‘Machine Learning-Based Intrusion Detection System for Controller Area Networks’, in *DIVANet 2020 – Proc. 10th ACM Symp. Des. Anal. Intell. Veh. Netw. Appl.*, 2020.