Hidden Markov Model based Anomaly Detection Method for In-vehicle Network

Ye Neung Kim, Seok Min Ko, and TaeGuen Kim*

Soonchunhyang University, Asan City, Republic of Korea
{kyn2423, 20164620, tg.kim}@sch.ac.kr

Received: March 10, 2022; Accepted: May 10, 2022; Published: May 31, 2022

Abstract

CAN protocol is a serial bus protocol that complements the previously existing shortcomings in the point-to-point network topology, and it provides full-duplex communications for transmitting data between the host nodes consisting of the network. In addition, the CAN protocol has many advantages in terms of scalability and efficiency for the cost to wire the network devices. Due to this fact, many car manufacturers have adapted the CAN protocol for implementing their in-vehicle networks. Even though the CAN protocol is widely used for in-vehicle networks, it still does not support any security mechanism to provide safe data transmission, because the size of CAN message is limited to 8 bytes which is insufficient to contain the fields for the security. The network nodes, ECUs using the CAN protocol basically transmit the data in a broadcast way while not applying encryption or authentication to the transmitted data. Therefore, the attackers can sniff and analyze the data transmitted through the CAN bus, and also they can inject their malformed data to control the in-vehicle network. In this paper, we propose a novel anomaly detection framework to protect the in-vehicle network that uses CAN bus protocol. Our proposed framework uses many hidden markov models to represent the normality of the network, and the models are generated using two types of network information; the transmission time interval and the payload data changes. In evaluation, we had several experiments, and it was found that the proposed framework can detect abnormal network behaviors accurately.

Keywords: Controller Area Network, In-Vehicle Network, Hidden Markov Model, Anomaly Detection, Intrusion Detection System

1 Introduction

With the rapid development of information technology(IT), the electronic systems in vehicles have evolved to provide many diverse functionalities to the drivers. Many sensors and ECUs(Electronic Control Units) are equipped in modern vehicles such as the connected car and autonomous driving car, and the many data are transmitted among these electronic devices to share the essential information that each device needs, to provide many functions.

CAN(Controller Area Network) protocol which is a serial bus protocol is very widely used to provide in-vehicle communications. Since the CAN bus protocol is firstly introduced in ISO(International Standardization Organization) 11898 standard [1], it has been adopted by many car manufacturers because it was proven that the CAN protocol spends relatively low costs to implement and it is reliable and scalable .

Journal of Internet Services and Information Security (JISIS), volume: 12, number: 2 (May), pp. 115-125
DOI:*10.22667/JISIS.2022.05.31.115*

*Corresponding author: Department of Information Security Engineering, Soonchunhyang University, Asan 31538, Republic of Korea. Tel: 041-530-1114, Fax: 041-542-4615

Even the CAN bus protocol has many advantages in the network performance and the cost efficiency for the development, it has the critical issues regarding the secure communications[2]. For many years, many researchers have emphasized the need for security of the CAN bus by introducing attacks that compromise the CAN bus. Most of the attacks demonstrated by many researchers are conducted by analyzing the CAN messages normally transmitted through CAN bus and injecting malicious CAN messages to control the network. it is because the CAN bus protocol does not support any security mechanisms such as encryption and authentication. The size of the CAN message is limited to only 8 bytes, and even in the case of CAN-FD(CAN with Flexible DataRate), the maximum size of the message is only 64 bytes. In addition, the requirements for time overhead to process the messages are strict. Therefore, it is infeasible to make the message includes the security information or to perform the encryption to secure the message.

In this paper, we propose an anomaly detection framework that detects the abnormal network behaviors without modifying the original protocol to apply the security mechanisms. To distinguish the normal behaviors and the abnormal behaviors, our framework extracts two kinds of sequential features from normal CAN message dataset such as [3][4] and uses them to generate the HMM(Hidden Markov Model)[5][6]. The sequential features used are the hamming distance sequence and the transmission time interval sequences extracted from the message sequence for each CAN ID. The HMM is used for the anomaly detection model since the algorithm can accurately represent the sequential data within a relatively short time. The two types of HMM models per each CAN Id message sequence are used for the detection, and in detail, each HMM produces the score that represents the normality of the given input CAN message sequence.

The rest of the paper is organized as follows: Section.2 presents the background behind the CAN bus protocol and explains the security problems that exist in the protocol, and Section.3 describes previous researches to secure the CAN bus protocols. Section.4 explains our HMM-based anomaly detection framework in detail, and Section.5 shows the experimental results to evaluate our proposed framework in terms of detection accuracy. Lastly, Section.6 summarizes our research and provides future work.

2 Background

2.1 CAN Overview

CAN bus protocol is a communication protocol designed to support the full-duplex communication to transmit non-host bus messages between network nodes such as ECUs in an in-vehicle network. The CAN messages are transmitted in a broadcasting manner through the CAN bus channel so that all network nodes, ECUs connected to the bus can send and receive all messages on the CAN bus channel.

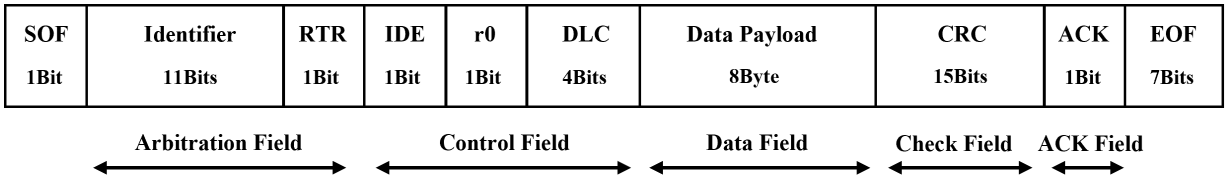


Figure 1: The format of standard CAN data frame

As shown in Fig.1, the CAN message is composed of several fields such as the arbitration field, control field, data field, CRC(Cyclic Redundancy Check) field, and ACK(Acknowledgement) field. In our method, the information contained in the arbitration field and the data field is used among the all information in many fields in the CAN message, with the transmission time that does not appear in the message.

The arbitration field contains the CAN Id which is a unique identifier indicating the transmission priority and type of the message[1], and it is originally used to perform the congestion control. For example, the CAN message includes the lowest Id selected by comparing each bit with the one in the other message, and it is transmitted firstly [1]. In our method, the CAN Id of the message is used to generate HMMs separately according to the Id value. In detail, Two different types of HMM (Time interval-based HMM and Hamming distance-based HMM) are generated per each CAN ID. The data field contains the payload data that the sender node tries to transmit to the other nodes, and in our method, the payload data of the CAN message for each different CAN Id is analyzed to check the degree of the data change compared to the previously transmitted CAN message that has the same CAN ID.

2.2 Threats on CAN Bus Protocol

The CAN bus protocol basically uses the broadcast transmission, so any network nodes in the bus channel can receive all messages which are not encrypted. If there are some malicious nodes in the network, they can analyze them to perform additional attacks. In addition, since the CAN message does not include the sender's address(or identifier), it is impossible to seize which node has sent the CAN message. If an attacker injects the malicious messages into the CAN bus, all malicious messages will be treated as normal messages by all nodes. Due to the property of the protocol, the following attacks can threaten the CAN bus communication.

(i) spoofing attack - An unauthorized attacker can send malicious messages by disguising a malicious node as a normal node.

(ii) fuzzing attack - Malicious nodes can inject invalid data into the vehicle network and disrupt the functioning of normal ECUs.

(iii) sniffing attack - If the message is delivered as an unencrypted plaintext message, the message can be intercepted by a passive attacker.

(iv) replay attack - An attacker can capture the message sent in plaintext from the ECU responsible for certain important functions (e.g, turning off the engine, operating the brake) and the attacker can re-transmit the collected message to control the vehicle.

3 Related Work

The CAN bus protocol has the advantage of being able to broadcast messages efficiently between many ECUs consisting of the in-vehicle network, but there are security vulnerabilities because it was not designed with security in mind[2]. Therefore, various kinds of methods to protect CAN bus communication have been researched. C.Miller and C.Valasek[3] hacked the Jeep Cherokee vehicle by exploiting the CAN bus communication and demonstrated that not only the air conditioner and radio but also the important functions of the vehicle, such as brake, accelerator, and ignition, were remotely controlled by the attackers. By showing that it is feasible to control the network by injecting the malformed messages using the remotely compromised telematics system, the researchers proved that the vehicle physically non-reachable is considered safe no longer. Olufowobi et al. [7] analyzed the transmission period and response time of the CAN message, and they proposed a method to generate a detection model that can set the threshold value range for the arrival time of a normal CAN message. Using the model, the arrival time of the CAN message per CAN Id is examined to check if it exceeds the threshold range specified in the model. If the arrival time is not in the pre-defined threshold range, then the model flags that the given CAN message dataset is abnormal. Wang et al. [8] use the property that the dominant bits (logic 0) of the

CAN Id have a higher transmission priority over the recessive bits (logic 1). In detail, they assumed that the attacker will inject the CAN message that has the low CAN Id to make the bus being occupied by their malicious messages, and their proposed method checks the entropy of the probability of occurrence of recessive bit in each bit of the CAN ID. If the entropy is not in range of the case when only normal messages exist, then the detection method notifies that the attack has occurred. Song et al. [9] proposed an intrusion detection system that detects message injection attacks using features of transmission time intervals of CAN messages. After capturing messages from the CAN bus and analyzing the transmission time interval data per the CAN ID, if some messages that are transmitted within the too short interval or too long interval are found, then it is considered that there were injection attacks on the in-vehicle network. Delwar et al.[10] proposed a LSTM (Long Short-Term Memory) based CAN IDS (Intrusion detection system). LSTM classifier is trained using the normal CAN messages and the simulated CAN messages for the attacks such as DoS (Denial of Service), fuzzing, and spoofing attacks. In detail, the CAN Id sequence, DLC (Data length code) sequence, and payload data sequence are extracted from the training data set and the extracted features are used to generate the LSTM model. The researchers used two kinds of LSTM models; the Vanilla LSTM model and the Stacked LSTM model, and they checked the performance for each. As a result, they found that the Vanilla LSTM model with a single hidden layer showed the highest attack detection rate. Jerin et al.[11] proposed a hybrId approach that detects the CAN bus attacks. Firstly, the proposed method extracts the whole CAN Id sequence from the whole CAN messages collected in a normal situation, and the method extracts the small CAN Id patterns by moving the sliding window whose size is set to 4. The CAN Id patterns collected from the normal dataset are examined with the patterns extracted in real-time. By checking whether the currently extracted pattern is included in the normal pattern set, the method detects the abnormal behaviors in the current network. In addition to this, the method also measured the time intervals of the transmitted messages . The normal time interval for each CAN Id is first calculated and used to compare with the interval measured in real-time. When the timestamp value of the most recently transmitted message for a certain CAN Id is out of the pre-extracted normal time interval, then the method considers that the attack has occurred. Boumiza et al.[12] proposed Hidden Markov Model-based method to detect the anomaly in the in-vehicle network. The proposed method uses the CAN messages that have constant bit values or bit values increasing or decreasing within a certain range, in their payload data, and the bit value changes that can be extracted by analyzing CAN messages are used as features for the HMM.

4 Proposed framework

4.1 HMM based Anomaly Detection System

Our HMM-based anomaly detection framework consists of a training phase and a detection phase, and the overall architecture of the framework is shown in Fig.2. Our framework uses multiple HMMs to check the abnormality in various aspects. There are two types of HMMs, called time interval-based HMM and hamming distance-based HMM, and these two types of HMMs are generated separately per CAN ID. For the training phase, only a normal CAN message dataset is used to generate the multiple HMMs. More specifically, the sequences of the time intervals of normal CAN messages are used to generate both the time interval-based HMMs, and the sequences of the hamming distances of normal CAN messages are used to generate the hamming distance-based HMMs. After training the HMMs is done, the log-likelihood which is used as a score that represents the normality of given sequential data is calculated by using the training data again. Each score is used to define the normal threshold range. In the detection phase, the time interval sequence per each CAN Id and the hamming distance sequence per each CAN Id are extracted from the CAN message sequence first, and the sequences are inputted to each matched HMM to calculate the log-likelihood. Then, the framework checks whether the calculated values are in

the predefined normal ranges.

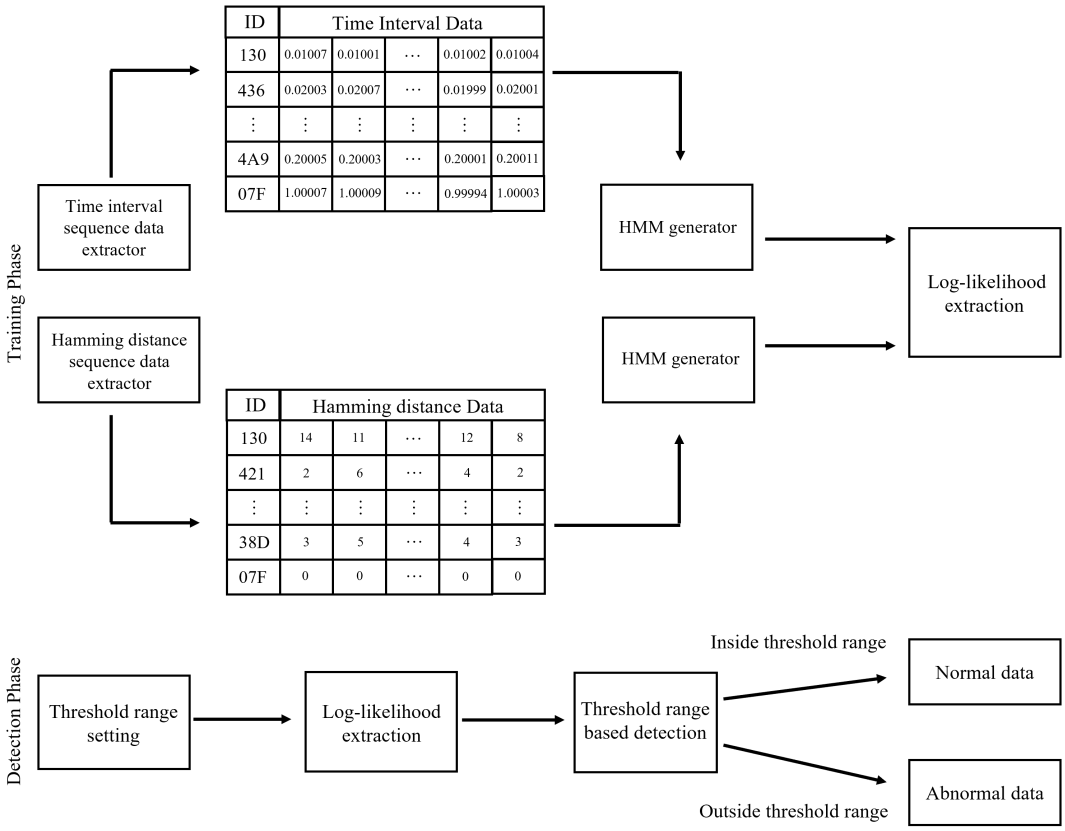


Figure 2: Architecture of Our Proposed Detection System

4.1.1 Time Interval Sequence Extractor

A CAN message can be classified into the periodic message and the event message, according to transmission type. Although there are event messages transmitted in CAN bus, most of the messages transmitted in the bus are periodic. If an attacker injects malicious messages or suspends normal messages, then the transmission period of the messages affected by the attack will be changed, compared with the normal period. Consequently, our framework uses the time interval information for anomaly detection. The time interval sequence extractor is a module that has a role to extract the time interval information. The Time interval sequence extractor first separates the whole CAN message sequence according to the CAN Id that each message includes. For example, if there is a sequence consists of the messages that includes different CAN Id (e.g., [Id=1, Id=2, Id=3, Id=1, Id=2, Id=3, Id=1, Id=2, Id=3...]), then the whole sequence will be separated into three sequences per CAN Id (e.g., [Id=1, Id=1, ...], [Id=2, Id=2, ...], [Id=3, Id=3, ...]). After all sequences are extracted, The Time interval sequence extractor calculates the transmission time interval between the consecutive CAN messages in each sequence, and the time interval values are listed to form a sequence.

4.1.2 Hamming Distance Sequence Extractor

CAN message with the same Id are usually used for the same functionality. The payload bytes included in the messages with the same Id have the same or similar value, or there are many messages with the same

CAN Id whose payload bytes increase or decrease by a certain unit. The gap between the payload bytes of consecutive messages with the same Id tends to be constant or similar. If an attacker injects malicious messages, then the value gap of the injected messages' payload data and the benign messages' payload data before and after injection will be different, compared with the gap among the benign messages. Therefore, our framework uses the data changes in payload bytes of consecutive frames in a transmission flow. Like the time interval sequence extractor, the hamming distance sequence extractor first extracts the sequences that consist of the CAN messages with the same ID. And the hamming distances[13] of consecutive frames in a transmission flow are calculated and arranged to form a sequence. The hamming distance means the number of different bits between two given bit sequences, and each bit in the same position is compared in turn to count the different bits. Hamming distance of two sequences, x and y can be computed as follows Eq.1:

$$D = \sum_{i=1}^{k} |x_i - y_i| \quad , where\ \ x_i,\ y_i\ are\ the\ i_{th}\ bit\ in\ x\ and\ y \tag{1}$$

4.1.3 HMM Generator

HMM [14][5] is an algorithm that is widely used to model sequential data for various applications such as speech recognition, bioinformatics, and so on. The HMM algorithm can estimate the degree of matches between the given sequential data and the pre-trained model. More in detail, the log-likelihood which means the degree of matches can be computed from the forward algorithm of HMM. In our framework, HMMs for time interval sequences and HMMs for hamming distance sequences are generated by HMM generator and their log-likelihood values are used for threshold-based detection. For the HMM generation, the parameters such as the number of states, emission probabilities, transition probabilities, and initial probabilities should be specified first. The number of states is set to five and the initial probabilities and emission probabilities are randomly selected.

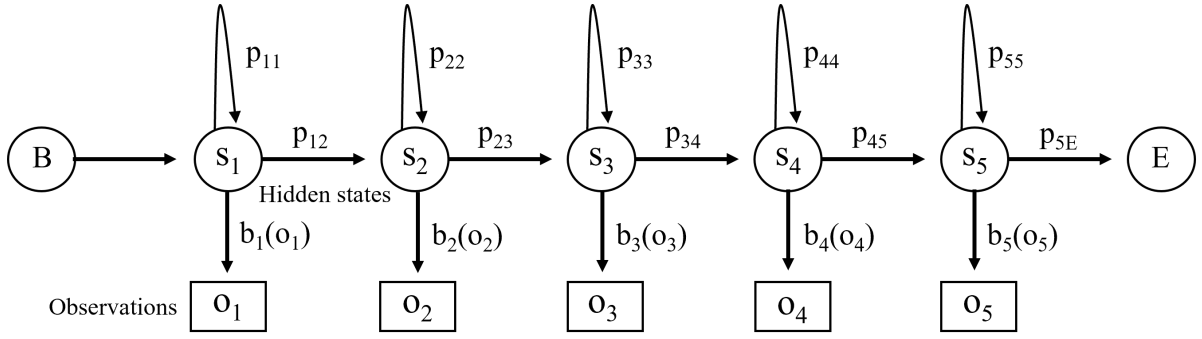


Figure 3: Example of HMM

4.1.4 Log-likelihood Extractor

Using multiple HMMs, it is possible to extract the log-likelihood values calculated by the forward algorithm. The log-likelihood is a measure that represents how the given input sequence fits well with the train HMM, and the higher value of the log-likelihood means that the given sequence is more well-matched with the HMM[15][16].

4.1.5 Threshold-based Detector

The threshold-based detector has the log-likelihood values calculated using the time interval sequences and the hamming distance sequences of the benign CAN messages in the training dataset. And each threshold is used to determine whether the given data is abnormal or not. In our framework, The only input data whose log-likelihood values from -10% to +10% of the threshold are considered normal. In the opposite case, the inputs are considered abnormal.

5 Evaluation

In this section, the experimental results for evaluating our proposed framework are described. HMM adapting time interval sequence and HMM adapting hamming distance sequence are evaluated separately, and the individual detection performance for each is described in turn. The public dataset that is used in previous works [3][4] is used in our experiments. The dataset contains the CAN messages collected from a Hyundai Avante(CN7) vehicle, and the CAN messages for the attack such as flooding, spoofing, replay, and fuzzing are also included in the dataset. Table 1 describes the configuration of the training set and test set. The training dataset consists of total of 179,347 normal CAN messages, and the test dataset to evaluate the true negative and the false positive consists of 180,687 normal CAN messages, and the test dataset to evaluate the true positive and the false negative consists of both of 941,535 normal messages and 107,040 attack messages.

Table 1: Dataset Configuration

Usage	# of normal messages	# of attack messages
Training dataset	179,347	0
Test dataset(TN,FP)	180,687	0
Test dataset(TP,FN)	941,535	107,040

5.0.1 Detection Performance of HMMs for time interval sequence data

Our framework generates the time interval-based HMMs to check whether there are CAN messages that violate the normal transmission period. From the training set shown in Table 1, the 73 time interval-based HMMs which are the models representing 73 CAN Ids' flows were generated, and the generated HMMs were evaluated using the two different test dataset: one for the true negative(or false positive) and one for the true positive(or false negative). Fig 4 and Fig 5 show the evaluation result. The graph in each figure describes the threshold ranges for the 73 HMMs and the log-likelihood values that were obtained by applying each test dataset to the HMMs. As a result, all HMMs accurately were able to detect the attack messages without the false negative. In the case of the test using only the normal messages, the 72 time interval-based HMMs were able to identify the normal messages accurately, a single HMM misjudged that a normal sequence is abnormal because the log-likelihood value was not in the threshold range.

5.0.2 Detection Performance of HMMs for hamming distance sequence data

Our framework generates the hamming distance-based HMMs to check whether there are CAN messages whose payload contains exceptional values in bits. The training set shown in Table 1 was used to generate the 73 hamming distance-based HMMs for 73 CAN Id. Fig 6 and Fig 7 show the evaluation result. As a result, the true negative, the false positive, and the true positive were 49, 24, and 73, respectively, and there was no false negative.

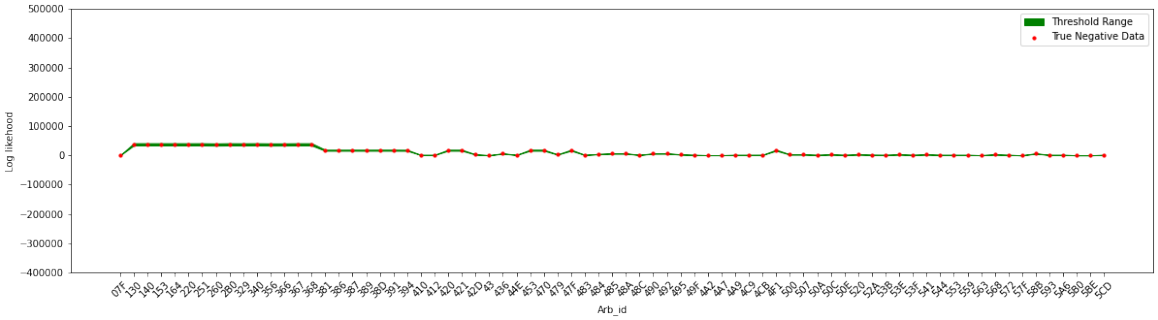


Figure 4: True Negative Test for Time Interval-based HMMs

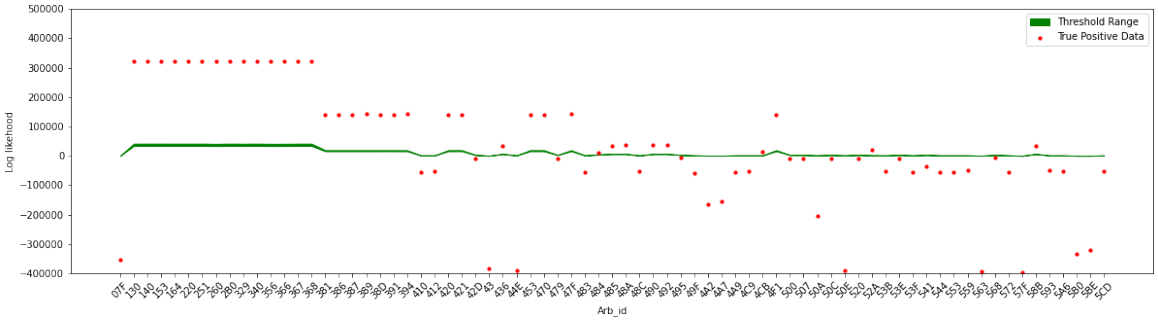


Figure 5: True Positive Test for Time Interval-based HMMs

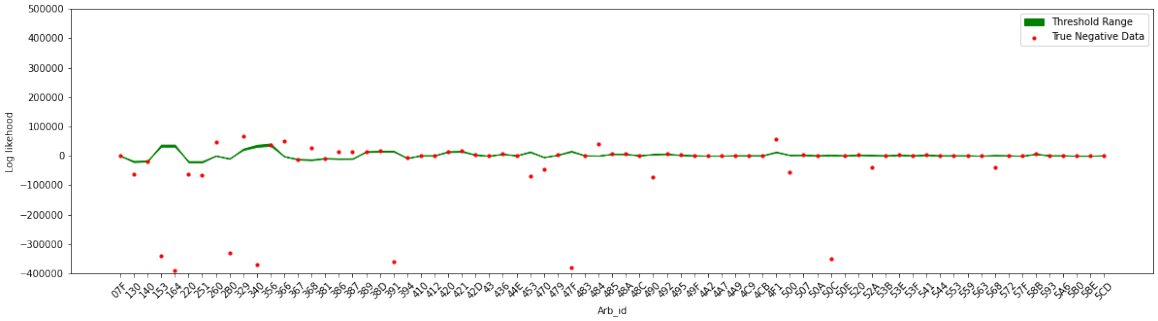


Figure 6: True Negative Test for Hamming Distance-based HMMs

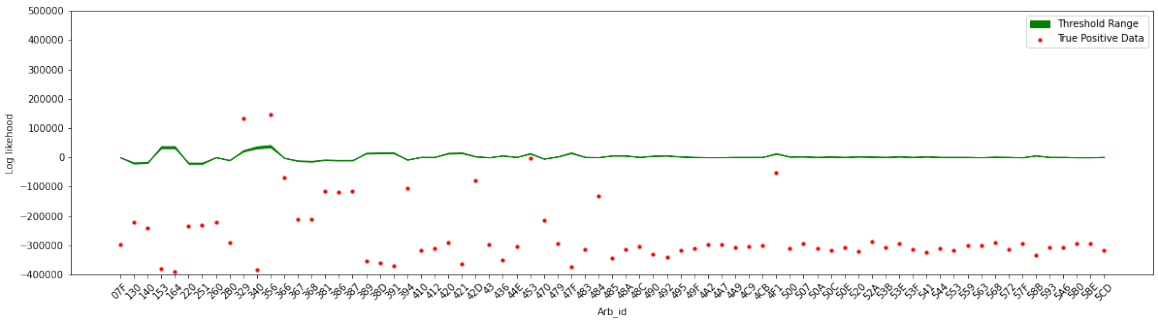


Figure 7: True Positive Test for Hamming Distance-based HMMs

5.1 Overall Performance Metrics

To provide the overall performance of our proposed framework, Accuracy, Precision, Recall, and F1 Score were calculated using the true positive (TP), the true negative (TN), the false positive (FP), and the false negative (FN) measured through the experiments. The equations for the performance metrics are listed in Eq.2345

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$Precision = \frac{TP}{FP+TP} \tag{3}$$

$$Recall = \frac{TP+FN}{TP} \tag{4}$$

$$F_1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

As a result, the overall performance of our proposed framework utilizing various HMMs is summarized in Table.2. The accuracy and F1 score of the detection using the time interval-based HMMs were all about 0.993, and the hamming distance-based HMMs produced an accuracy of 0.836 and F1 score of 0.860 respectively. The performance of the hamming distance-based HMMs was not high enough because of the high false positive cases. The false positive has occurred by the HMMs generated with the CAN messages that don't have any constant or tendency data. It is possible to use the time interval-based HMMs and the hamming distance-based HMMs together. Therefore, if the framework excludes the HMMs that occur false positive, then our framework can detect all attacks accurately without false positives.

Table 2: Final Experimental Result

Sequence Data	Accuracy	Precision	Recall	F1 Score
Time Interval	0.993	0.986	1.0	0.993
Hamming Distance	0.836	0.753	1.0	0.860

6 Conclusion

In the paper, we proposed a framework to detect abnormal network behaviors for attacks on CAN bus communication. Our proposed framework uses two kinds of HMMs: the time interval-based HMMs and the hamming distance-based HMMs. Each HMM is generated with each CAN message flow consisting of the messages with the same Id. With the dataset collected from the vehicle(Avante), we evaluated our time interval-based HMMs and hamming distance-based HMMs, and their accuracy values measured as 0.993 and 0.836 in turn. And if the time interval-based HMMs and hamming distance-based HMMs are used together or used exclusively, it is possible to achive the 100% accuracy . In the future, we will continue to perform research to improve our proposed framework by maximizing the number of the useful features for the anomaly detection. In addition, we have a plan to develop a ensemble method that combines the output of the multiple HMM to produce one solid normality score.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(No. 2022-0-01197, Convergence security core talent training business(SoonChunHyangUniversity)). This work was supported by the Soonchunhyang University Research Fund.

References

[1] S.C. HPLorrigan. Introduction to the controller area network (can). *Application Report SLOA101*, SLOA101:1–17, August 2002.

[2] P. Carsten, T.R. Andel, M. Yampolskiy, and J.T. McDonald. In-vehicle networks: Attacks, vulnerabilities, and proposed solutions. In *Proc. of the 10th Annual Cyber and Information Security Research Conference (CISR'15), Oak Ridge, TN, USA*, pages 1–8. ACM, April 2015.

[3] C. Miller and C. Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015(S91):1–92, August 2015.

[4] H. Kang, B.I., Y.H. Lee, H. Lee, H. Lee, and H.K. Kim. Car hacking and defense competition on in-vehicle network. In *Proc. of the 2021 Workshop on Automotive and Autonomous Vehicle Security (AutoSec'21), Virtual*, page 25. Internet Society, February 2021.

[5] S.R. Eddy. What is a hidden markov model? *Nature biotechnology*, 22(10):1315–1316, October 2004.

[6] A. Salaün, Y. Petetin, and F. Desbouvries. Comparing the modeling powers of rnn and hmm. In *Proc. of the 18th IEEE International Conference On Machine Learning And Applications (ICMLA'19), Boca Raton, FL, USA*, pages 1496–1499. IEEE, December 2019.

[7] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom. Saiducant: Specification-based automotive intrusion detection using controller area network (can) timing. *IEEE Transactions on Vehicular Technology*, 69(2):1484–1494, February 2020.

[8] Q. Wang, Z. Lu, and G. Qu. An entropy analysis based intrusion detection system for controller area network in vehicles. In *Proc. of the 31th IEEE International System-on-Chip Conference (SOCC'31), washington DC, USA*, pages 90–95. IEEE, September 2018.

[9] H.M. Song, H.R. Kim, and H.K. Kim. Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network. In *Proc. of the 2016 International Conference on Information Networking (ICOIN'16), Kota Kinabalu, Malaysia*, pages 63–68. IEEE, January 2016.

[10] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi. Long short-term memory-based intrusion detection system for in-vehicle controller area network bus. In *Proc. of the 44th Annual Computers, Software, and Applications Conference (COMPSAC'20), Madrid, Spain*, pages 10–17. IEEE, July 2020.

[11] J. Sunny, S. Sankaran, and V. Saraswat. A hybrid approach for fast anomaly detection in controller area networks. In *Proc. of the 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS'20), Delhi, India*, pages 1–6. IEEE, December 2020.

[12] S. Boumiza and R. Braham. An efficient hidden markov model for anomaly detection in can bus networks. In *Proc. of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM'19), Split, Croatia*, pages 1–6. IEEE, September 2019.

[13] R.W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950.

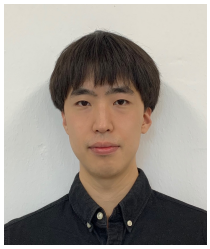
[14] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[15] I.J. Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, February 2003.

[16] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, January 2003.

Author Biography

Ye Neung Kim Ye Neung Kim received a Bachelor's degree in Information Security Engineering from SoonChunHyang University, South Korea, in 2021. he is currently Master's course student of the Department of Information Security Engineering of SoonChunHyang University, Asan-si, South Korea. His research topics are Controller Area Network Security, IoT Security Protocol, and Cryptographic Protocol.



Seok Min Ko Seok-Min Ko is currently a Master's course student of the Department of Mobility Convergence Security of SoonChunHyang University, Asan-si, South Korea. His research topics are Fine-grained Classification, Vulnerability Detection, and Malware Detection.

TaeGuen Kim TaeGuen Kim received the B.S. degree in electronics and computer engineering and the M.S. degree in computer and software from Hanyang University, South Korea, in 2011 and 2013, respectively. He also received the Ph.D degree in computer and software from Hanyang University, in 2018, and he worked at Hyundai Motor company as a senior research engineer. Currently, he is with Soonchunhyang University as an assistant professor since March 2021. His research interests include malware analysis, artificial intelligence, and automotive security.