2018 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
Vehicle Electronics and Architecture (VEA) & Ground Systems Cyber Engineering (GSCE) Technical Session
AUGUST 7-9, 2018 - NOVI, MICHIGAN

A TWO-STAGE DEEP LEARNING APPROACH FOR CAN INTRUSION DETECTION

Linxi Zhang
University of Michigan-Dearborn
Computer and Information Science Department
Dearborn, MI

Lyndon Shi
University of Michigan Ann Arbor
Electrical Engineering and Computer Science Department
Ann Arbor, MI

Nevrus Kaja
University of Michigan-Dearborn
Electrical and Computer Engineering Department
Dearborn, MI

Di Ma, PhD
University of Michigan-Dearborn
Computer and Information Science Department
Dearborn, MI

ABSTRACT

With recent advancements in the automotive world and the introductions of autonomous vehicles, automotive cybersecurity has become a main and primary issue for every automaker. In order to come up with measures to detect and protect against malicious attacks, intrusion detection systems (IDS) are commonly used. These systems identify attacks while comparing normal behavior with abnormalities. In this paper, we propose a novel, two-stage IDS based on deep-learning and rule-based systems. The objective of this IDS is to detect malicious attacks and ensure CAN security in real time. Deep Learning has already been used in CAN IDS and is already proven to be a successful algorithm when it comes to extensive datasets but comes with the cost of high computational requirements. The novelty of this paper is to use Deep Learning to achieve high predictability results while keeping low computational requirements by offsetting it with rule-based systems. In addition, we examine the performance of proposed IDS with the objective for using it in real-time situations.

I. INTRODUCTION

The transportation ecosystem is going through a revolutionary transformation with automation and connectivity as its main drivers. These services increase mobility and promise to virtually eliminate crashes and fatalities which are a

chronic problem to the current landscape of the automotive world [1]. In order to deliver these promising services, automotive manufacturers have to first eliminate malicious actors in such ecosystem and minimize their impact. The automotive cybsersecurity is a significant problem in today's industry. Securing vehicles includes securing in-vehicle networks which connect various Electric Control Units (ECU) for different subsystems in the vehicle. One of such networks is the Controller Area Network (CAN). CAN is one of the most predominant in-vehicle bus communication protocols. This protocol was designed from Bosch in 1985 with goals of efficiency and reliability, but security was not its primary objective.

The CAN bus is essentially a two-wire broadcast bus with frames carrying 64-bits of data. The main components in a CAN Frame include: ID (11 bits), Control Field (6 bits), Data Field (0-8 bytes), CRC Field (15 bits), End of Frame and other bits (13 bits). Newer versions such as CAN-FD extend this limit up to 512-bits while adding capabilities for security solutions at the application layer. The topology of a typical CAN bus system is provided in Figure 1.

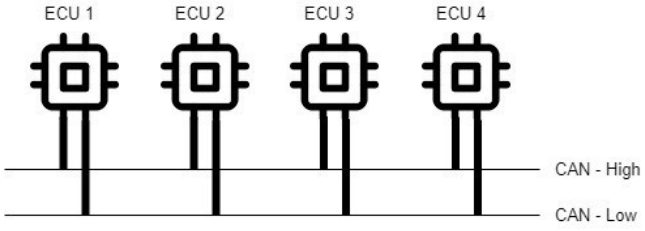

Figure 1: CAN Bus Topology.

CAN is message oriented. Instead of containing addresses of the transmitter and receiver, each CAN frame has a predefined ID and message structure defined in the DBC file, a database-like file in a proprietary format that contains all the specifications of every ECU of a specific vehicle configuration. (The DBC file is usually kept strictly confidential by car manufacturers.) An ECU is configured at compile time to receive CAN messages with specific CAN IDs and disregard others.

A. CAN Vulnerabilities

Although CAN physical layer has strong error detection through CRC, bit stuffing, etc., it has no security protection. Most of the CAN vulnerabilities come from the facts that it is a broadcast bus, all packets broadcast to all nodes, and each node decides if it should process the packet [2]. Such a design results in many potential vulnerabilities for CAN communication: all nodes see all traffic which allows eavesdropping and learning patterns of target ECUs; any (compromised) node can send any arbitrary packet and no one knows who send that packet; the broadcast nature and relying on arbitration to win bus access also makes DoS attack possible; answers to standard challenges for authentication when doing sensitive things, such as reflashing components or firmware update, are stored in memory; etc. Besides the aforementioned typical vulnerabilities, attacks on CAN can also result in physical attacks. A CAN node can slightly alter its baud rate which will result in the node sending error frames constantly. This can disrupt real communication by causing latencies and system faults.

B. CAN Intrusion Detection Systems

The problem of CAN security became more relevant with the recent advancements in the automotive world and the introductions of autonomous and connected vehicles. Today's car is no longer isolated or driven primary from mechanical systems. Electronic controls and x-by-wire systems now control almost every aspect of the car. Adding to this includes the fact that soon a driver will not be required to control the car and the fact that every car is going to be connected and talk to other cars and other things (V2X). These developments make automotive cybersecurity become a primary issue for every automaker. Consequences that a cybersecurity breach can do

now for a car, are much higher than before. Several recent research works demonstrate an attacker is able to access the CAN network via a variety of attacking interfaces such as TPMS, Bluetooth, telematics, and OBD2 and take the whole control of victim vehicle. Securing CAN becomes an important security challenge.

Message authentication and sender identification schemes have been proposed to protect the integrity and verify the originality of CAN messages [3]-[9], [9]-[13]. Although these schemes are effective to defend against attacks originated from unauthorized devices which have access to the CAN, they are limited to defense against attacks originated from compromised ECUs since these ECUs are parts of the vehicle and usually configured with the right credential to conduct secure communication. Intrusion detection systems (IDSs) are another major defense mechanism commonly used to secure CAN from attacks. These systems identify attacks while comparing normal behavior with abnormalities. Various IDSs exploiting different characteristics of CAN messages have been proposed. Many existing schemes exploit the regularity and periodicity of CAN traffic [13]-[16]. They usually result in efficient IDS but do not work well with non-periodic attack messages. Researchers also exploit the sequence pattern of CAN traffic for intrusion detection [17]. However, it does not work well with replay attacks and single message injection attacks involving frequent messages. In summary, these systems usually only work well in specific threat models that have been already considered in the design and are susceptible to miss sophisticated attacks that do not fall in the specific threat models.

Giving attacks are becoming more sophisticated and cars become more connected and complicated, advanced data analytics such as deep neural network (DNN) are considered to improve detection rate, especially in detecting more sophisticated attacks that can escape free from being caught by detection methods exploiting regularity, periodicity, and simple sequence patterns [18]-[21]. However, machine learning based schemes usually involve high computation cost.

Although all proposed IDSs are supposed to work in real time, no one reports an evaluation on processing delay time except the work of [19] which reports a real-time processing delay of 2.05ms~3.78ms depending on the number of layers used in the DNN model.

In this paper, we propose a novel two-stage IDS to strike a balance between efficiency and detection rate. The first stage is rule-based detection while the second stage is DNN-based. Our idea is to use the lightweight rule-based approach to quickly detect common attacks that violate the regularity and periodicity of major CAN traffic and use DNN to catch missed attacks from the first stage and achieve high detection rate.

Contributions

With an objective to build an IDS that can detect malicious attacks and ensure CAN security in real time, this work makes the following contributions:

- Efficiency and detection accuracy. We achieve a balance between efficiency and detection accuracy through a 2-stage detection design. We choose robust and efficient rules in the first stage to minimize the work in the time-consuming second stage. We use advanced machine learning algorithms to detect sophisticated attacks to achieve high detection rate.
- Realistic attack model. We consider a realistic attack model which cover most known attack types. This is in contrast with previous work which usually use adversary models with specific attacks.
- Comprehensive evaluation with real traces collected from three car models. Previous work either use simulated data or data collected from one vehicle. We might be the

first who evaluate IDS over three different datasets collected from three real cars.

- Achieve high detection rate and low latency. Compared with previous work, our experiments show the proposed IDS achieves high detection rate, low false positive rate, and low latency.

Organization

The organization of the paper is as follows. Section II reviews related work. Section III summarizes the adversary capabilities. Section IV will provide the methodology of our design. Section V will provide insights into the implementation and evaluation of our approach. And at the end we will conclude with Section VI.

II. RELATED WORK

Message authentication and intrusion detection are two main lines of research to secure CAN.

Message Authentication and Sender Identification. Several cryptographic protocol proposals, most of them are based on the use of message authentication code (MAC), have been proposed for **CAN message authentication** [3]-[12]. However, due to the highly restricting space in CAN message (a CAN packet is at most 8 bytes in length) and the demanding real-time requirement, to have a practical deployable solution for CAN authentication is still a challenging job.

To improve efficiency for real-time detection, an anonymous ID scheme is proposed to provide *implicit* **sender identification** (rather than message authentication) to prevent broadcasting from unauthorized senders in [22]. Since both the sender and target receiver can generate the anonymous IDs beforehand, this scheme is efficient and only adds negligible delay to the identification process. More recently, an *explicit* **sender identification** scheme is proposed based on the fingerprinting of ECUs by using clock skews [13]. However, both message authentication and sender identification are limited to detect attacks originated from a compromised ECU which can send any arbitrary message to the network.

Intrusion Detection. Parallel to message authentication and sender identification, various intrusion detection schemes (IDSs) have been proposed for CAN to curb cyber attacks. Some recent proposals exploit the fact that most CAN messages are sent at fixed intervals, or periodically [13]-[16]. The work of [14], [16] monitor the intervals of CAN messages and calculate the system entropy. Changes in system entropy and relative entropy are used for intrusion detection. The work of [15] detects message injection attacks by analyzing traffic anomalies based on message frequency with an assumption that *all* CAN messages are generated at regular frequency or interval. The clock-based IDS proposed in [13] uses the periodical nature of many CAN messages to detect anomalies as well as fingerprint ECUs. Although light weight, all these time-interval approaches do not work for attacks with aperiodic messages.

Besides message frequency, researchers also exploit other information for intrusion detection. The work of [23] suggests the use of a set of different in-vehicle sensors to verify message formality, location, data range, data plausibility, etc. However, it does not have an implementation. The work of [17] identifies anomalies in the sequence of messages that flow in the CAN bus based on the recurring patterns within the sequence of message IDs. This method does not work very well in detecting replay attacks since a replay of message sequences which have already seen during training. Also, it only exploits recurring patterns of two consecutive messages and does not work well with single message injection attack.

To detect unknown attacks and also improve detection rates, machine learning based IDS for in-vehicle networks have been proposed to exploit hidden patterns in IDS [18]-[21]. These schemes use different machine learning algorithms and data

features to train the model and detect anomalies. OBD-II port extracted data are used by [18], [20] to detect anomalous activities in vehicles. [18] uses the Hidden Markov Model while [20] uses artificial neural network (ANN). As OBD-II port extracted data are interpreted CAN data and usually only a limited set of message types can be extracted, it adds delay in data processing and has limited information to rely on for anomaly detection. The work of [19] uses bit pattern in the data field (64-bit) in the CAN packet and DNN for anomaly detection. The authors show the DNN-based approach outperforms traditional ANN-based approaches in detection accuracy. The authors of [21] propose a vehicular intrusion detection system, named as VIDS. VIDS includes two parts: lightweight domain-based model and crossdomain-based model. The lightweight domain-based model utilizes LSTM, a Recurrent Neural Network, and takes time frequency difference between CAN messages as input to learn the hidden logic. The comprehensive crossdomain-model uses the data field (64-bit) values in the CAN packet as input and ANN for anomaly detection.

III. ADVERSARY MODEL

We consider a general adversary model which can cover most known attacks targeting CAN communication. The attacker can send CAN packets to the bus directly (for example by plugging into the OBD-II port or compromising an ECU) or indirectly (for example by sending tire pressure info wirelessly to the RF receiver which is hooked directly into a car ECU which is connected to CAN). The attacker can be any of the following three attacker types with different capacity:

- *Weak*: A weak adversary has no idea about the semantic meaning of CAN messages of the victim vehicle and has no possession of previous CAN traffic trace. So, a weak adversary can only send random CAN packets or specific packets such as all-zero messages.
- *Medium powerful*: A medium powerful adversary has all the capabilities of the weak adversary. It can also have access to current or previous traffic traces. It also has knowledge about the specification of certain CAN IDs either through reverse engineering or learn from other sources such as the proprietary DBC file. It has access to a compromised ECU. It can send the network spoofed and replay messages with the ID configured for the compromised victim. However, it cannot masquerade by sending CAN messages on behalf of other ECUs rather than the compromised ECU.
- *Strong*: A strong adversary has all the capabilities of medium-powerful adversary. It also can block a victim ECU from transmitting and send masqueraded packets from another ECU.

More specifically, the attacker can mount the following attacks:

- *Random ID Attack*: A random attack, just like the name depicts, is when a random ID CAN message is generated and injected into the CAN bus.
- *All Zero ID Attack*: All zero attack is launched by messages with zero ID section. The ID section represents the priority of a CAN message, and the lower value of ID means the higher the priority. In zero ID Attack, the adversary uses the highest priority ID to launch attacks, usually DoS service.
- *Replay Attack*: A replay attack is executed by messages which are transmitted in a normal vehicle behavior. The attacker collects normal CAN messages showed on bus before and replays them back to the bus at a later time. Because CAN bus lacks freshness protection mechanisms, this attack can be realized easily by adversary.
- *Spoofing Attack*: Due to the lack of authentication in CAN bus, CAN messages

can be modified or spoofed on the bus. This attack may lead ECUs to mal-perform.

- *Drop Attack*: When an attacker has the ability to access a compromised ECU, it is easy to stop transmitting selected or all messages from this ECU. In this case, the attack could lead to some serious errors for the vehicle.

IV. OUR DESIGN

The objective of this research is to build a novel in-vehicle IDS to monitor malicious activity on the CAN bus and report deviations from normal or malicious activities in real time. To achieve the objective, we propose an IDS design consisting of two stages. The first stage is a robust rule-based system. The second stage uses DNN for anomaly detection. Figure 2 shows a high-level architecture for the proposed CAN-IDS. When this IDS is applied on the CAN bus, all the messages need to go through the first rule-based detection system for "validity check". CAN packages which pass the first stage are further processed by the second DNN-based detection system. Finally, messages which pass both stages are allowed to go to the vehicle network.

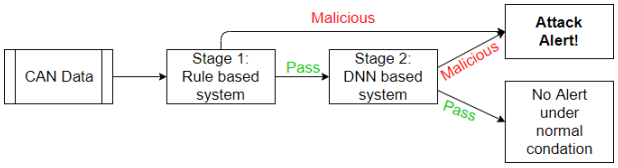

Figure 2: Proposed IDS Architecture.

A. Rule-based IDS Design

The first stage of this proposed IDS is a rule-based component that provides efficient detection prior to the second stage. Rule based systems are common in IDS because they can be designed based on characteristics of CAN traffic. There are many rules which can be used. The challenge here is to choose a set of rules that has very low false negative rate so that attack messages can be identified in the first stage without moving to the time-consuming second stage. Also, to achieve fast detection, a chosen rule must have small processing delay. Some typical rules used in IDS include:

Valid ID: This rule is established based on the fact that each message contains a unique ID specified in the OEM's DBC file. A valid CAN message must contain a valid unique ID. When applying the valid ID rule, a message containing an invalid ID will be detected and marked as a potentially malicious message. The valid ID list can be generated easily from normal CAN traffic data.

Time Interval: This rule is based on the fact that most CAN messages are generated and sent to the network periodically. Time interval rule is a very powerful rule in detecting injected messages. The time-interval based scheme proposed in [15] achieves 100% detection rate on a data set collected from a certain vehicle model. Time intervals can be calculated easily from CAN traffic data.

Message frequency: This rule is established on the fact that most CAN packets are sent periodically and thus their frequency distribution is predictable. Figure 3 illustrates frequency distribution of three example ID packets from three different car models respectively. The 0x0D4 packet (in blue) has five frequencies f_1= 14, f_2= 22, f_3= 81 and f_4= 101 and f_5= 100, and f_5 occurs 310 times while others occur only about 3 times in the data set we have. The 0x916 packet (in yellow) has only one frequency f_1= 1 which occurs 721 times in the data set we have. The 0x465 packet (in green) has two frequencies f_1= 10, f_2= 11 and f_1 occurs 60 times while f_2 occurs only 2 times in the dataset we have. This rule is efficient to mitigate the DDoS or flooding attacks. The implementation of frequency rule is a little more complicated than the time interval rule.

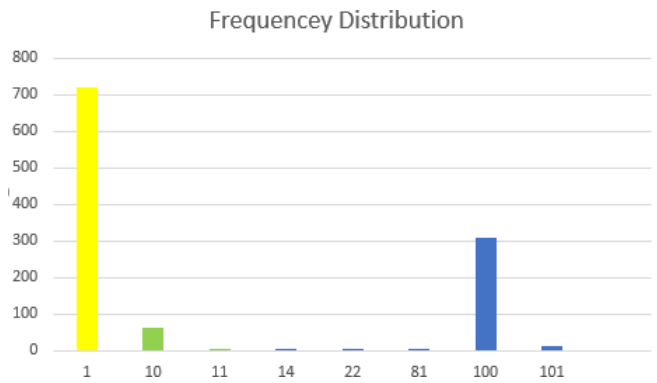

Figure 3: Predictable Frequency Distribution Examples.

Message sequence: The sequence rule is based on a characteristic that an ID always followed by some specific IDs. Valid ID sequences serve as the sequence whitelist. However, it is difficult to enumerate all possible sequences and it does not work well with replay attacks.

Based on the discussion above, we choose the valid ID rule and time interval rule in our rule-based IDS design for efficiency and high detection rate.

B. DNN-based IDS Design

Artificial Neural Network (ANN) are machine learning algorithms which have gained a lot of praise in data heavy applications lately. ANN models a human brain and try to solve problems in a similar manner as the brain. They are widely used for intrusion detection. When an ANN has two or more hidden layers, it is known as deep neural network (DNN). Previous work has shown DNN outperforms other machine learning algorithms such as support vector machine (SVM) and traditional ANN in anomaly detection [19], [24]. Based on our experience with a DNN-based network IDS [25] and other previous works [26], [27], we chose to use sequential forward selection algorithms which starts by finding the best single individual feature. This algorithm helps to select features from features, including message ID, number of occurrences in the past second, ID sequence, relative distance of ID entropy, changes in system ID entropy, relative distance of data entropy, changes in system data entropy, the occurrences of bit-symbol "1" in the CAN message data filed. After a thorough analysis, the following features are selected:

- **Message ID**: Message ID plays a crucial role in identifying malicious messages alone. For example, diagnostic packets should not be observed while driving. It also serves as the link among packets with the same ID and assists DNN to find hidden patterns.
- **Number of occurrences in the past second**: We use the number of occurrences in the past second of an incoming message ID as a feature to enhance the DNN classifier.
- **Relative distance**: Another important training feature is the relative distance between message ID's. It is based on the observation that a higher injection rate leads to an increase in relative distance. The usage of relative distance help to identify injected messages in CAN bus [14]. Previous research has also showed that malicious messages must be injected 20-100 times the normal rate to be properly received by a target ECU [28].
- **Change in system entropy**: Based on the observation that flooding attacks lead to a dramatic decrease in system entropy [14]. We include the change in system entropy from receipt of the previous message to receipt of the incoming message as a feature in our DNN.

V. IMPLEMENTATION AND EVALUATION

A. System Environment

We implemented the proposed two stage IDS in a ThinkPad T440s notebook with Intel Core i5-4200U CPU @ 1.60GHz processor. The software environment is Python version 3 and TensorFlow is used to build the deep neural network.

Generally, detection accuracy increases with more hidden layers used in the DNN model at a cost of increased processing time [19], [29]. After considering a balance between time complexity and accuracy, we choose to use a DNN model

with five hidden layers with 100, 100, 80, 60, and 40 neurons respectively.

We use three datasets in our evaluation and they are collected from three real vehicles under normal situation. The first dataset collected from a 2006 Honda Accord consists of 243,762 messages. The second dataset collected from an Asia brand vehicle has 1,152,394 messages. The third dataset collected from a US brand vehicle contains 2,886,338 messages. The training data contains only normal CAN messages from the three datasets. The testing data contains normal CAN messages as well as malicious messages we inject. Five types of malicious messages are randomly injected into and mixed with normal traffic.

B. Attack Strategy

As defined in our adversary model, we evaluate the performance of the proposed IDS with five types of attacks:

- Random Attack: inject random messages with random ID.
- Zero ID Messages Attack: launch flooding attack.
- Replay Attack: repeat previous messages.
- Spoofing Attack: inject malicious messages which are generated based on the knowledge of CAN message specification.
- Drop Attack: drop normal messages.

After each normal message is read from the dataset, our program decides whether to inject a message, and what type of attack to generate. A random number (*rand_num*) in [0, 24] is selected, so each malicious message type has a 4% chance of being generated for each valid message parsed.

It is noted that the spoofing messages are generated from reverse engineering. For example, we found instrument cluster speedometer's corresponding message ID is 0x0C8 in the Honda dataset. For ID 0x0C8, only the fifth and sixth bytes in data section are changed under normal status, shown as the red part in Figure 4. By changing fifth and sixth bytes, the attacker is able to generate malicious messages for spoofing attack, such as ID: 0x0C8, Data: 0x00 0x00 0x00 0x00 0x01 0x02 0x00 0x00.

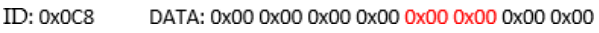

Figure 4: Spoofing Malicious Message Example.

C. Experiment Results

With each dataset is split with 60 percent used for training and 40 percent used for testing, preliminary performance results are provided in Table 1. We achieve a detection rate of over 99.9% with slight difference among all three vehicle datasets. The false positive error is about 0.3% and the processing time for each message is about 0.55 ms.

	Honda Accord	US Brand	Asia Brand
Detection rate	99.91%	99.97%	99.92%
False positive	0.090%	0.029%	0.018%
Time per msg	0.56 ms	0.53 ms	0.61 ms

Table 1: Performance Results.

The work of [21] reports a detection rate of 95% and the work of [19] reports a 1.6% false positive error and about 97.8% total accuracy. Compared with these two machine learning based schemes, our model achieves better performance in attack detection with low false positive error. The work of [13] reports a similar false positive rate of 0.055%.

To further evaluate the performance, we use different percentage of data for training. We note that detection rate is relatively high even with small amount of training data. In general, the false positive is a big issue when applying machine learning. However, our experimental results show that with the appropriate increasing of training data, we may decrease the false positive rate.

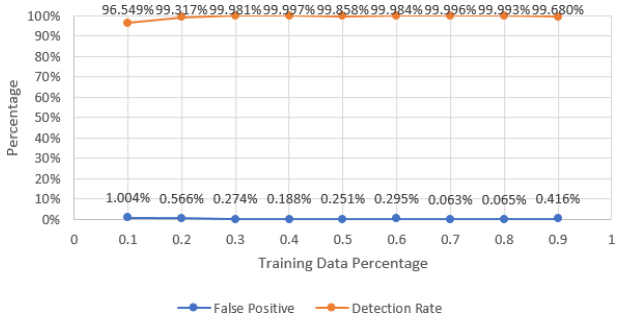

Figure 5: The Honda Dataset Experimental Results.

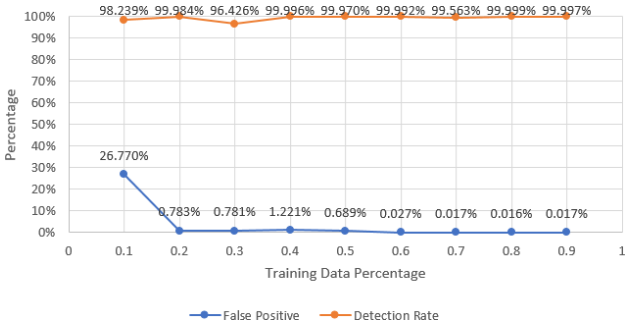

Figure 6: The Asia Dataset Experimental Results.

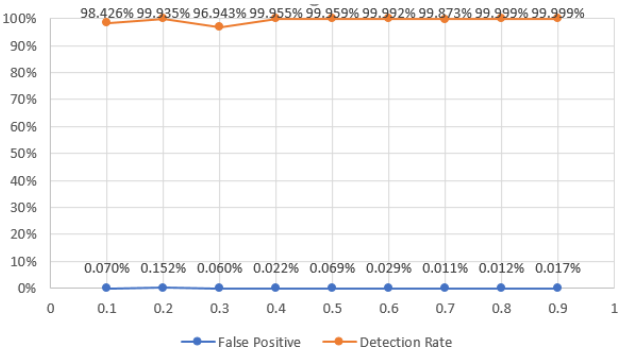

Figure 7: The US Dataset Experimental Results.

Figure 5, 6 and 7 show our experimental results on the detection accuracy and false positive rate for Honda Accord (2016), the Asia brand vehicle and the US brand vehicle with different percentage of data for training. The horizon axis means the percentage of datasets used for training. And the vertical axis represents the percentage of false positive and detection rate. Those results show: 1) Small amount of training data (about 10% to 30%) leads to high false positive; 2) Acceptable false positive gained by larger amount of data (about 50% to 80%). And the performance of the third datasets should be counted as an exception: The performance is good starting at 10%. Due to large amount of datasets, 10% messages for training is enough to train a classifier. Based on our experiment results and fact that the training process time complexity will increase tremendously with the increasing of the amount of training data, we can draw a conclusion that 60% to 70% is the reasonable percentage of data for DNN training. It is noted that the result may vary on different vehicle models.

We also evaluate the time performance of our design. In general, more hidden layer means higher computation cost. Fortunately, the training process can be completed offline. After the training process, the classifier generated by the offline training can be applied directly for real-time detection. Our experiment shows about 0.6 ms an average processing delay per message. It is better than the reported processing delay of 2-5 ms in [19]. By considering the CAN bus speed and CAN packet size, the proposed IDS model can be used in real-time environment.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a two-stage intrusion detection system for in-vehicle CAN bus. In the first stage, we use a rule-based component to check injected CAN messages while in the second stage we use a DNN based component. Experimental results show that this two-stage CAN IDS has a good performance with high detection rate, low false positives, and small processing delay. There is potential for real-time implementation. We also prove that this system can work for different vehicle models with necessary training. Another contribution of this paper is that it provides a feasible architecture that leverages emerging and performance-proven algorithms such as deep neural network to be applied into CAN Intrusion Detection Systems.

The experimental results provide the performance comparison between three different vehicle brands. The results affirm that the proposed IDS could work for different models of vehicles.

Future works include continuing to determine the implementation feasibility of this algorithm in real time (with ECU spec), applying pre-process to improve performance, reducing DNN hidden layer without impacting the performance and reducing the false positives, and adding more robust rules to improve the first component. Another direction for future work is to develop more complex attack strategies such as event-based attacks and evaluate the effectiveness of the proposed IDS against them.

REFERENCES

[1] S. Abuelsamid, "Autonomous Automotive Cybersecurity, the need to protect automated and connected vehicles," 2016

[2] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, and K. Oishi, "A method of preventing unauthorized data transmission in Controller Area Network," in 2012 IEEE 75th Vehicular Technology Conference (VTC Spring), pp. 1–5, May 2012.

[3] D. Nilsson, U. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound Message Authentication Codes," in IEEE 68th Vehicular Technology Conference (VTC), 2008.

[4] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in IEEE/IFIP International Conference on Dependable Systems Networks (DSN'09), pp. 165–174, 2009.

[5] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered Embedded Control Networks," in the 5th Workshop on Embedded Systems Security, 2010.

[6] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "CAR2X communication: Securing the last meter," in 4th IEEE International Symposium on Wireless Vehicular Communications, 2011.

[7] A. Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth - a simple, backward compatible broadcast authentication protocol for CAN bus," in 10th escar Embedded Security in Cars Confer- ence, 2011.

[8] C. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the Controller Area Network (CAN) communication protocol," ASE Science Journal, vol. 1, no. 2, pp. 80–92, 2012.

[9] O. Hartkopp, C. Reuber, and R. Schilling, "Macan - message authenticated CAN," in Embedded Security in Cars Conference (escar), 2012.

[10] B. Groza and P.-S. Murvay, "Broadcast authentication in a low speed Controller Area Network," in E-Business and Telecommu- nications, International Joint Conference, ICETE 2011, 2012.

[11] B.Groza, S. Murvay, A. Herrewege, and I. Verbauwhede, "LiBrA-CAN: a lightweight broadcast authentication protocol for Controller Area Network," in 11th International Conference on Cryptology and Network Security (CANS), 2012.

[12] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security authentication system for in-vehicle network," pp. 5–9, 10 2015.

[13] K.-T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for vehicle intrusion detection," in 25th USENIX Security Symposium (USENIX Security 2016), pp. 911–927, 2016.

[14] M. Mter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in 2011 IEEE Intelligent Vehicles Symposium (IV), 2011.

[15] 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 2016 International Conference on Information Networking (ICOIN), pp. 63–68, Jan 2016.

[16] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), pp. 1–6, Sept 2016.

[17] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1577– 1583, June 2017.

[18] S. N. Narayanan, S. Mittal, and A. Joshi, "Using data analytics to detect anomalous states in vehicles," CoRR, vol. abs/1512.08048, 2015.

[19] M.-J. Kang and J.-W. Kang, "Intrusion Detection System using Deep Neural Network for in-vehicle network security," in PloS one, 2016.

[20] A. Wasicek, M. D. Pes, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in 5th escar 2017, (Ypsilanti, MI, USA), 2017.

[21] Z. Wei, Y. Yang, Y. Rehana, Y. Wu, J. Weng, and R. H. Deng, "IoVShield: An efficient vehicular Intrusion Detection System for self-driving," in Information Security Practice and Experience, pp. 638–647, Springer International Publishing, 2017.

[22] K. Han, S. Potluri, and K. Shin, "Real-time frame authentication using ID Anonymization in automotive networks," Mar. 26 2015. US Patent App. 14/494,141.

[23] M. Mter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in 2010 Sixth International Conference on Information Assurance and Security, pp. 92–98, Aug 2010.

[24] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of Intrusion Detection using Deep Neural Network," in 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 313–316, Feb 2017.

[25] N. Kaja, A. Shaout, and D. Ma, "A two stage Intrusion Detection Intelligent System," in The International Arab Conference on Information Technology, 2017.

[26] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, no. Mar, pp. 1157–1182, 2003.

[27] T. Kavzoglu and P. M. Mather, "The role of feature selection in artificial neural network applications," International Journal of Remote Sensing, vol. 23, no. 15, pp. 2919–2937, 2002.

[28] C. M. Chris Valasek, "A survey of remote automotive attack surfaces," in Black Hat, 2014.

[29] Y. B. Ian Goodfellow and A. Courville, Deep Learning Book. MIT Press, 2016.