ID Sequence Analysis for Intrusion Detection in the CAN bus using Long Short Term Memory Networks

Araya Kibrom Desta, Shuji Ohira
Graduate School of Science and Technology
Nara Institute of Science and Technology
Nara, Japan
kibrom_desta.araya.js3@is.naist.jp, ohira.shuji.ok2@is.naist.jp

Ismail Arai, Kazutoshi Fujikawa
Information iniTiative Center
Nara Institute of Science and Technology
Nara, Japan
ismail@itc.naist.jp, fujikawa@itc.naist.jp

***Abstract*—The number of computer controlled vehicles throughout the world is rising at a staggering speed. Even though this enhances the driving experience, it opens a new security hole in the automotive industry. To alleviate this issue, we are proposing an intrusion detection system (IDS) to the controller area network (CAN), which is the de facto communication standard of present-day vehicles. We implemented an IDS based on the analysis of ID sequences. The IDS uses a trained Long-Short Term Memory (LSTM) to predict an arbitration ID that will appear in the future by looking back to the last 20 packet arbitration IDs. The output from the LSTM network is a softmax probability of all the 42 arbitration IDs in our test car. The softmax probability is used in two approaches for IDS. In the first approach, a single arbitration ID is predicted by taking the class which has the highest softmax probability. This method only gave us an accuracy of 0.6. Applying this result in a real vehicle would give us a lot of false negatives, hence we devised a second approach that uses log loss as an anomaly signal. The evaluated log loss is compared with a predefined threshold to see if the result is in the anomaly boundary. Furthermore, We have tested our approach using insertion, drop and illegal ID attacks which greatly outperform the conventional method with practical F1 scores of 0.9, 0.84, and 1.0 respectively.**

***Index Terms*—LSTM, In-vehicle Network Security, Automotive, Intrusion Detection, CAN bus**

I. Introduction

Since recently vehicles were enclosed devices that were only used for the sole purpose of transportation. Computers hackers were mainly focusing on getting unauthorized access to computer systems because it was impossible to remotely attack vehicles. But nowadays modern vehicles have several embedded Electronic Control Units (ECU) integrated internally. These ECUs communicate using a networking standard called Controller Area Network (CAN). Even after the introduction of this standard, attackers needed to get physical access to a target vehicle to manipulate these ECUs. These types of attacks might have a hazardous effect on a target vehicle but it would be practically impossible to attack in a large quantity due to cost and energy expenses.

Now vehicles have started to incorporate a higher number of computing devices that can also communicate with a remote server to do different activities including over-the-air (OTA) software updates [2]. This technology has changed the way we drive vehicles. Due to the fast growth of computer technology, the vision of connected cars is also becoming a reality. Drivers can easily access information about the ongoing environment without further looking outside of their vehicles. It became easy for a vehicle to connect to internet for entertainment purposes. These features of automobiles have brought along the security issue of computer technology. Hackers can use the same techniques to get remote access to the internals of the in-vehicle networks.

Researchers have now started to investigate the security vulnerabilities of the automotive industry to further enhance safe driving [1]. Most vehicles nowadays use a CAN bus, an International Standardization Organization (ISO) serial communication bus, that monitors most of the the car's systems and sensors [3]. CAN bus is a broadcast communication protocol with no source and destination address. This property of the CAN bus makes it vulnerable to insertion attacks. CAN bus also uses no authentication and encryption techniques for secure communication. Since it is used in a real-time communication environment, applying the common computer security techniques like encryption and decryption would slow it down. The other security issue of the CAN bus is its vulnerability to DoS attacks. CAN uses arbitration ID to prioritize bus usage, the lower the ID the higher priority it will be given to use the CAN bus. This property of the CAN bus makes it vulnerable to DoS attacks. Attackers can create DoS attack inside the CAN bus by continuously injecting a higher priority CAN packet so as for all the other arbitration IDs be backed off from using the bus resources.

By taking advantage of the aforementioned security holes of the CAN bus, attackers have found a way to gain unauthorized access to the internal networks of vehicles [4]. To improve the security drawbacks of the CAN bus, in this paper we are proposing an IDS by analyzing the sequence of CAN packets' arbitration IDs. In addition to other information, CAN packets have an arbitration ID that is used to control the priority of CAN packets. Our intrusion detection method extracts the IDs to train Long Short-term Memory Networks (LSTM). Once, the LSTM network learns the pattern of the IDs, if there is any, it is used to predict an arbitration ID that might appear after a certain sequence of arbitration IDs. Relying on the predicted arbitration ID, an anomaly signal is prepared from a softmax probability of all the available classes (all the arbitration IDs). An anomaly is detected using two ways. The first approach compares the probability values of all the classes and selects the one with the highest probability as a predicted arbitration ID. The predicted ID is then compared with the true ID for anomaly detection. The second approach gets an aggregated log loss value of the predicted classes that will be later compared with a predefined anomaly signal threshold to detect for intrusions.

The conventional method proposed by [15] trains a single transition matrix that will be used to test the possible transitions between two different IDs. Even if this performs with near perfect precision value (0.999), it has a very low recall value (0.4) as it is impossible to grab millions of arbitration ID sequences in a single transition matrix. In this paper, we are proposing an IDS system using LSTM that greatly outperforms the conventional method. The main contribution of this research study is to improve the anomaly detection accuracy of the conventional method. In addition, we tested both the conventional and trained network against insertion, drop and illegal ID attacks to further support our study. F1 scores of the proposed method are 0.9, 0.84 and 1.0 in order of insertion, drop and illegal ID attacks.

II. Related Work

The current state of the art intrusion detection methods in the CAN bus can be summarized into 4 categories [5]. The first category is fingerprints-based methods. Fingerprints-based methods take into consideration the fact that different ECU on the in-vehicle networks usually have unique hardware fingerprint information, like electric signals, and using this information it analyzes the change of this signals for intrusion detection. [6], [7], [8] are among the methods that are implemented by extracting fingerprint information of ECUs using different approaches. This is a physical level approach, but attacks can be bypassed if they are applied in the application level. In this case, a second category called parameter monitoring based methods are used. [9], [10], [11], these approaches collect different static values (frequency, mean, and variance etc) that will later be compared with a predefined values for intrusion detection. These approaches also have the drawback of heavily depending on periodic packets and being ineffective for unknown security threats. The other two categories, information-theory based and machine learning based, come here to solve this issue. The information-theory based method is based on the fact that malicious messages injected into the normal communication will affect the network stability, and the information entropy can reflect the anomaly, [12], [13]. Even if this has a small computational overhead, but it is mostly ineffective in attacks that modify the data portion of CAN packets.

Machine learning approaches have a better way of identifying anomalies that have never seen before. Our IDS is also in this category. Anomalies are detected by training a machine learning algorithm to learn about the true pattern of benign packet sequences, [19] [15] [20], to see if there is any deviation from the normal sequence of packets arriving in the CAN bus. [15] have proposed an anomaly detection algorithm that identifies anomalies in the sequence of messages that flow in the CAN bus. A complete survey of the current intrusion detection methods can be found in [5].

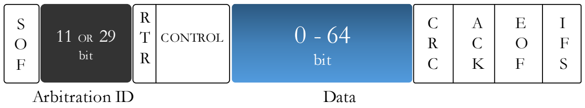


Fig. 1: CAN Packet

III. Attacking Controller area network

A. Controller Area Network

CAN is a serial communication protocol used in vehicles for connecting automotive electronics, ECUs, anti-skid-systems, etc [14]. CAN allows the implementation of peer-to-peer and broadcast or multi-cast communication functions with lean bus bandwidth use. CAN has two standards, standard and extended. Both of these standards are similar except the arbitration field has 11-bit identifier in the standard and 29-bit in the extended standard. There is also a little variation in the rest of the fields between the two standards, Fig. 1 shows CAN packet. For this research, we are proposing an LSTM approach that predict the arbitration ID of the CAN packet for intrusion detection.

When data is transmitted over a CAN network, no individual nodes are addressed. Instead, the message is assigned an identifier that works as a unique tag on its data content. The identifier not only defines the message contents but also the message priority. When a node wishes to transmit information it simply passes the data and the identifier to its CAN controller and sets the relevant transmit request. It is then up to the CAN controller to format the message contents and transmit the data in the form of a CAN frame. Once the node has gained access to the bus and is transmitting its message, all other nodes become receivers. Having received the message correctly, these nodes then perform an acceptance test to determine if the data is relevant to that particular device, based on the identifier of the message [16]. Our research uses arbitration IDs drawn from CAN packets for training LSTM network.

B. CAN attack surfaces

Attackers should always find a way to send their attack packets to the internals of the vehicle so as for them to manipulate the network of vehicles. According to [18], attackers might gain access to a car's internals in two ways. The first is to physically approach the target vehicle and insert a malicious component into a car's internal network via the ubiquitous OBD-II port. The other is through the various wireless interfaces available in present-day vehicles. It can be by first intruding into the drivers phone that the driver might later connect it to their vehicle for entertainment purposes or gaining access to GPS of the vehicle.

IV. ID Sequence Analysis Using LSTM for Intrusion Detection

This section describes the approach we used for detecting intrusions through ID sequence prediction. LSTM network

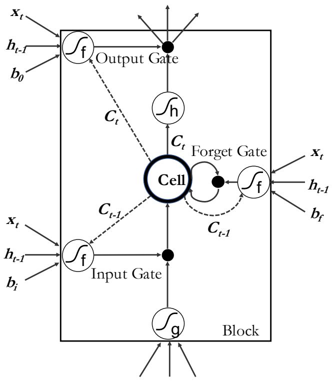
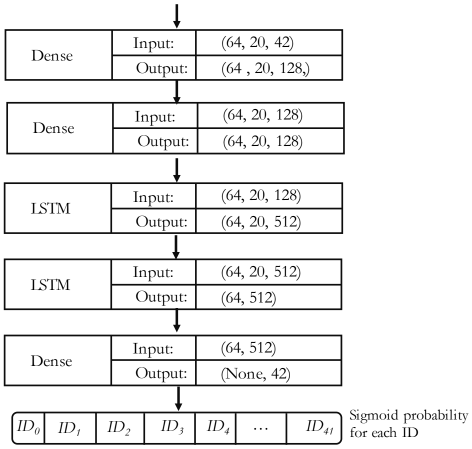


Fig. 2: LSTM memory block with one cell



Dense	Input:	(64, 20, 42)
	Output:	(64 , 20, 128,)
Dense	Input:	(64, 20, 128)
	Output:	(64, 20, 128)
LSTM	Input:	(64, 20, 128)
	Output:	(64, 20, 512)
LSTM	Input:	(64, 20, 512)
	Output:	(64, 512)
Dense	Input:	(64, 512)
	Output:	(None, 42)

Fig. 3: ID prediction network architecture

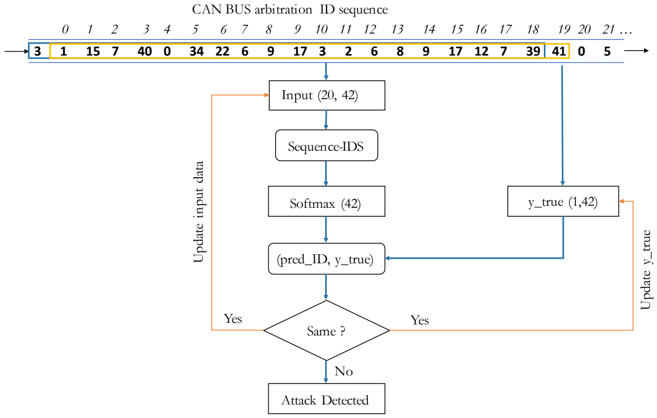


Fig. 4: ID Sequence Prediction IDS process

is trained to predict a subsequent arbitration ID by looking back at previously seen arbitration IDs. Fig. 2 provides an illustration of an LSTM memory block with a single cell. An LSTM network is the same as a standard RNN, except that the summation units in the hidden layer are replaced by memory blocks.

The LSTM architecture consists of a set of recurrently connected subnets, known as memory blocks [21]. Each block contains one or more self-connected memory cells and three multiplicative units (the input i_t, output o_t and forget gates f_t) that provide continuous analogues of write, read and reset operations for the cells. Each of the gates are updates in each step according to the equations shows in Equations 1 - 5.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_t + W_{ci}c_t + b_i) \tag{1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_t + W_{cf}c_t + b_f) \tag{2}$$

$$c_t = \sigma(f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{4}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{5}$$

The multiplicative gates allow LSTM memory cells to store and access information over long periods of time, thereby mitigating the vanishing gradient problem of standard RNNs.

A. Input Data Preprocessing

The input to the LSTM network is only a sequence of IDs. Like all types of neural networks LSTM also accept only numeric tensors. To convert the sequence of the IDs to a numeric tensor, we have vectorized each input ID. After we tokenized each arbitration ID to a numeric value, the sequence of numbers are one hot encoded before we fed it to the network. From here what the network has to do is give us a sigmoid probability for each of the classes in the tensor.

B. IDS Network Architecture

The input to the neural network is the sequence of IDs extracted from CAN packets. The network architecture of this IDS is similar to the architecture used by [22] other than the input tensors and similar attributes that we had to change to fit to this method's training data. The keras [23] implementation of the network architecture consists of three dense layers and two LSTM layers (Fig. 3).

First the input to the network is changed to a suitable format as explained in the last section. Then a tensor of high dimensional vector is fed to the two dense layers each with 128 units and tanh activation function. Later on, there is a dropout of 0.2 which we used to fight over-fitting during training. The output from this layer then passes to two LSTM layers with 512 units each. This two LSTM layers have tanh activation function and 0.2 dropout value. The final layer in this architecture is a dense layer with 42 units that provides us with sigmoid probability value of each arbitration ID. The ID sequence intrusion detection process is shown in Fig. 4. After we trained the network, each time a message appears in the CAN bus, we collect the first 20 IDs of these messages. Using these 20 messages as input to the trained network, we get a softmax probability to the prediction of the next arbitration ID.

The one given with the highest probability will be the predicted arbitration ID. Next, we compare the predicted arbitration ID with the one that has appeared after 20^{th} arbitration ID. If the predicted and true arbitration IDs are not the same, an anomaly signal is flagged. But, if both of these values are the same, we update the input and the true values in the next step. The input value will be a tensor that grabs 20 arbitration IDs again but this time the start pointer is updated by one to point to the ID next to the first one. And the last pointer will also be incremented by one to incorporate the last predicted arbitration ID. Using these 20 arbitration IDs we again go through the same process to monitor for intrusions. This process starts from when engine of the car is started and continues till the car's engine is stopped.

C. Attack packets and Anomaly Signal

We have simulated three types of attacks. insertion, drop and illegal ID attacks. When intruders attack a vehicle there will be some deflection from the normal sequence of the arbitration IDs as most attacks either remove a packet or add a new packet to the CAN bus. Hence, what this method does is, it checks if any ID has appeared in the CAN bus out of the benign sequence. When a new attack packet appears in the CAN bus, this IDS method will first grab the last 20 IDs sequences to predict the already appeared attack packet's ID. Since this was an attack packet, the IDS will predict a different arbitration ID value than this one. If the predicted one is different from the already appeared packet's ID, this IDS creates an alert message about this attack or further checks the log loss value evaluated for four consecutive predictions. In addition, we have also prepared an illegal ID attack. This attack is a kind of insertion attack but the arbitration ID in the packet is different from all the IDS in the training and test data.

Predicting for ID can give us more false negatives due to the randomness of the ID sequence. To solve this issue, we used log loss for multi-class prediction as our anomaly signal [24]. We selected log loss because log loss penalizes higher errors than low errors. After we get the softmax probability for each arbitration ID we calculated the log loss of the predicted ID and the true ID. Let the true labels for our predicted arbitration ID be encoded as 1-of-K binary indicator matrix Y, i.e.,$Y_{i,k} = 1$ if sample i has label k taken from the set of K arbitration ID labels. Let P be a matrix probability estimates, with $p_{i,k} = Pr(t_{i,k} = 1)$. Then the log loss of the whole set is calculated by using Equation, 6.

$$L_{\log}(Y, P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} (y_{i,k} \log P_{i,k}) \tag{6}$$

Using this value as a signal we further improved the detection accuracy. And unlike the first approach which provides us an anomaly signal value in every 20 ID sequences, this one gives an anomaly signal in every 100 ID sequences. The collective anomaly signal is generated by taking the average of 5 consecutive log loss values. This value is then compared with a predefined threshold to see if its above it or not.

D. Training and Test Set Description

The training data is extracted from the sequence of data we collected from a real car. We used 36 million packets that are collected in the first 100 hours of driving. Out of this data we used 70% for training, 15% for validation and the last 15% is used as a testing data. We used 2-fold cross validation to further validate our proposed network architecture. Fig. 5 shows the scattered sequence of the first 1000 IDs, each color represents a single arbitration ID of the 42 IDs. As we can see it in the figure some of the IDs appear periodically and some appear randomly but there seems to be some pattern in the sequence of the IDs. Hence, the target of this method would be to check if the LSTM would be able to learn this pattern of sequences. When the engine of the car starts some of the the arbitration IDs (e.g. 1, 2, 3, and 4) start to appear a bit late by around 0.2 seconds than the others. This is not considered in the training because this will not have a significant effect in a data size of 36 million sequences.

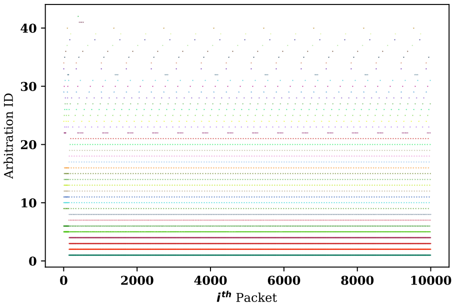


Fig. 5: Scattered plot of the first 10000 packet sequences of all the 42 arbitration IDs.

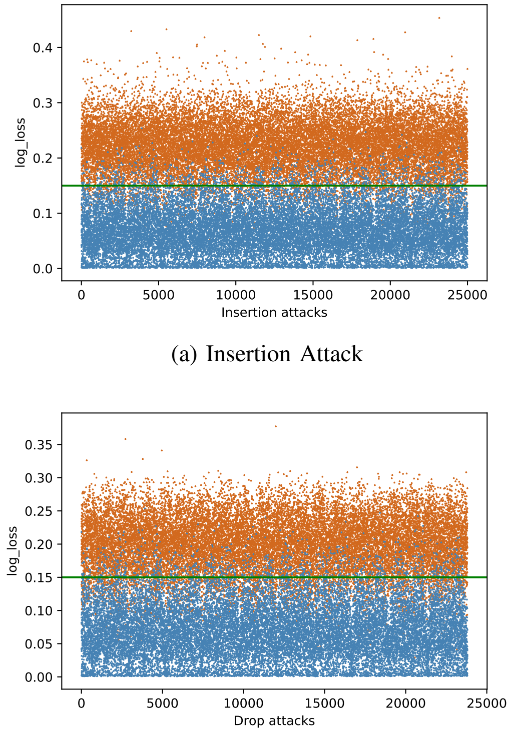
TABLE I: Results of ID prediction, predicting a single subsequent arbitration ID for every 20 ID sequences

correctly predicted	incorrectly predicted
153927	106073

V. Experiment results and Discussion

In this section we elaborate the performance of our intrusion detection system for the two approaches. The first approach is to hard code an ID prediction neural network. The approach tests for anomalies by considering 20 sequence of arbitration IDs of the CAN bus with 680 messages in a second. From here we take the true value from the CAN bus and compare both of these values, results are shown in table I. As we can see it from the results, out of 160 thousand sequence of size 20 it only detected around 60% of the total. This result might help in identifying some attacks but deploying it in a real vehicle would be inapplicable.

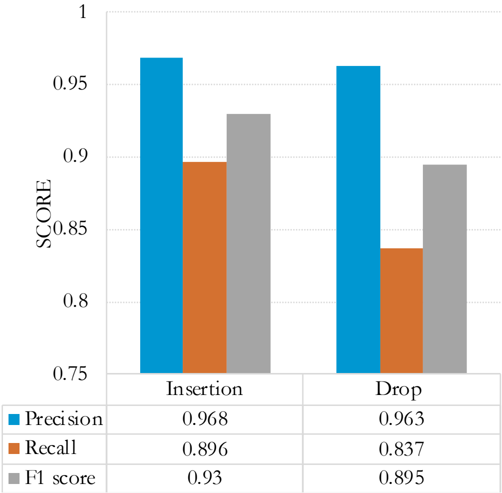
In the second method, a log loss is used as anomaly signal. We calculate log loss value of the softmax output and the true value of five consecutive predictions for both the anomalous sequence and sane sequence. By making the softmax a bit flexible and identifying a single anomaly in a every 5 predictions it gives us better results. Fig. 6 shows the log loss scattered plot for both the insertion and benign results respectively. Fig. 7 also shows how relaxing the anomaly signal calculation can help us improve the detection capability. In the first approach a single prediction was being made in every 20 arbitration IDs. The first approach mainly focuses on taking the



(a) Insertion Attack

(b) Drop Attack

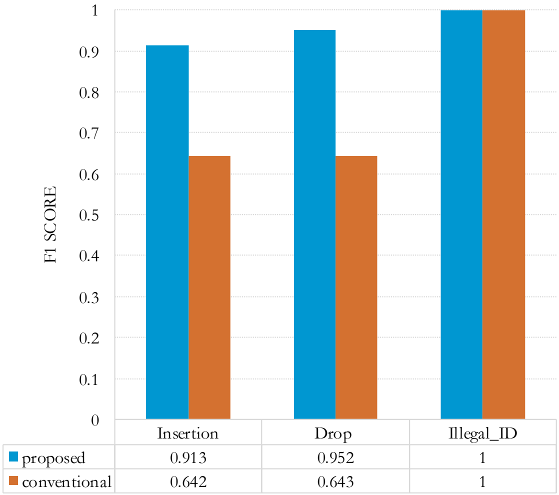
Fig. 6: scattered results of Insertion and Drop attacks for the first 8000 mix of attack and benign sequences



	Insertion	Drop
■ Precision	0.968	0.963
■ Recall	0.896	0.837
■ F1 score	0.93	0.895

Fig. 7: Detection scores for both insertion and drop attacks

maximum softmax probability of a class to make a prediction. But, this method calculates the average log loss value of the softmax values and true values to test if the evaluated value is above a predefined value. We make a single decision threshold for every 100 frames, unlike the first one which only considers the first 20 frames.



	Insertion	Drop	Illegal_ID
■ proposed	0.913	0.952	1
■ conventional	0.642	0.643	1

Fig. 8: F1 score comparison of the proposed and conventional methods

As it is shown in fig. 8, the F1 score of the proposed method greatly outperforms the conventional method except in the illegal ID attack case. Illegal ID attack is easier to identify for both the conventional and the proposed method. For both of the methods, way before the intrusion detection process, the arbitration IDs of the packets are checked to see if the IDs are in the class of arbitration IDs. If so, we go to the detection process but if not an anomaly signal is initiated with our further going to the IDS process.

Implementation of the IDS in the in-vehicle network would require a fast processor that is capable of collecting packets in n time window and make a prediction in as short time as possible. But, there is always a significant delay from the time when the packets appear in the CAN bus to the time when these packets are collected for analysis. The IDS has an average execution time of 24ms. The shown execution time is evaluated in an environment with Ubuntu 18.04 OS, Intel Xeon E5-1620 CPUand GM200 (GeForce GTX TITAN X) CUDA 6144 cores GPU that has a clock speed 1000MHz and a RAM size of 16GB. We played the dumped file in the terminal with vcan0 interface and created a program that collects the packets through SocketCAN API to do the intrusion detection.

The approach can only detect specific types of attacks. Attacks that can be identified by this method are those that can deviate the flow of messages disturbing the normal sequence of arbitration IDs. If attackers can compromise a single ECU, they might bypass the IDS by sending normal sequence of arbitration IDs with spoiled data portion of the packets. The other limitation is it leaves some messages behind when enough messages are collected for processing. During processing for detecting intrusion there is an average delay of 24ms. During this delay, some messages can appear in the CAN bus that the method will skip. Therefore, we either need a fast computing processor that can cope up with the frequency of messages or we should work on improving the execution time of the neural network.

VI. Conclusion

In this paper, we proposed an intrusion detection system using LSTM based on analysis of arbitration ID sequences. Our experiment focused on intrusion detection in in-vehicle networks, but the idea can be more extended to anomaly detection of other types of sequential data. The model is based on two approaches. Once the LSTM network is trained, the first approach uses the highest softmax probability to select the next arbitration ID. The predicted arbitration ID is then compared with the true ID for detecting anomalies. The second approach is an improvement to the first one by using log loss anomaly signal. After we get the softmax probability for each arbitration ID, we calculated the log loss of the predicted ID and the true ID. The log loss is then compared with a predefined threshold for intrusion detection. The first approach doesn't give practical results but the experimental results from the second approach show that our model can be implemented in a real vehicle. Attacks that can be detected using these approaches are the kind of attacks that might alter the normal sequence of arbitration IDs. However, attacks that don't alter the sequence (e.g. impersonation attacks) will not be detected using these approaches. For the future, we will incorporate more features from the data sequence so as for our system to identify these kinds of attacks and improve the detection performance of the selected attack types.

References

[1] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H. and Savage, S., 2010, May. Experimental security analysis of a modern automobile. In 2010 IEEE Symposium on Security and Privacy (pp. 447-462). IEEE.

[2] Halder, S., Ghosal, A. and Conti, M., 2019. Secure OTA Software Updates in Connected Vehicles: A survey. arXiv preprint arXiv:1904.00685.

[3] Bosch CAN Specification, September 1991 Robert Bosch GmbH, Postach 50, D- 7000 Stugart

[4] C. Miller and C. Valasek. (2015) Remote exploitation of an unaltered passenger vehicle. White paper of Blackhat USA conference

[5] Wu, W., Li, R., Xie, G., An, J., Bai, Y., Zhou, J. and Li, K., 2019. A Survey of Intrusion Detection for In-Vehicle Networks. IEEE Transactions on Intelligent Transportation Systems.

[6] Cho, K.T. and Shin, K.G., 2016. Fingerprinting electronic control units for vehicle intrusion detection. In 25th USENIX Security Symposium (USENIX Security 16) (pp. 911-927).

[7] Kyong-Tak Cho and Kang G. Shin. 2017. Viden: Attacker Identification on In-Vehicle Networks. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). ACM, New York, NY, USA, 1109-1123.

[8] Kneib M, Huth C. Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security 2018 Oct 15 (pp. 787-800). ACM.

[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. 2016 International Conference on Information Networking (ICOIN), Jan. 2016, pp. 6368.

[10] A. Taylor, N. Japkowicz, and S. Leblanc, Frequency-based anomaly detection for the automotive CAN bus, in Proc. 2015 World Congress on Industrial Control Systems Security (WCICSS), Dec. 2015, pp. 4549.

[11] Kuwahara T, Baba Y, Kashima H, Kishikawa T, Tsurumi J, Haga T, Ujiie Y, Sasaki T, Matsushima H. Supervised and unsupervised intrusion detection based on CAN message frequencies for in-vehicle network. Journal of Information Processing. 2018;26:306-13.

[12] Marchetti, M., Stabili, D., Guido, A. and Colajanni, M., 2016, September. 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). IEEE.

[13] Wu, W., Huang, Y., Kurachi, R., Zeng, G., Xie, G., Li, R. and Li, K., 2018. Sliding Window Optimized Information Entropy Analysis Method for Intrusion Detection on In-Vehicle Networks. IEEE Access, 6, pp.45233-45245.

[14] ROBERT BOSCH GmbH, Postfach 50, D-7000 Stuttgart 1

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

[16] Farsi, M., Ratcliff, K. and Barbosa, M., 1999. An overview of controller area network. Computing & Control Engineering Journal, 10(3), pp.113-120.

[17] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaniche, et al.. A Survey of Security Threats and Protection Mechanisms in Embedded Automotive Networks. The 2nd Workshop on Open Resilient human-aware Cyber-physical Systems (WORCS-2013), co-located with the IEEE/IFIP Annual Symposium on Dependable Systems and Networks (DSN-2013), Jun 2013, Budapest, Hungary. pp.1-12

[18] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H. and Savage, S., 2010, May. Experimental security analysis of a modern automobile. In 2010 IEEE Symposium on Security and Privacy (pp. 447-462). IEEE.

[19] Nanduri, A. and Sherry, L., 2016, April. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In 2016 Integrated Communications Navigation and Surveillance (ICNS) (pp. 5C2-1). IEEE.

[20] Taylor, A., Leblanc, S. and Japkowicz, N., 2016, October. Anomaly detection in automobile control network data with long short-term memory networks. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 130-139). IEEE.

[21] Graves, A., 2012. Supervised sequence labelling with recurrent neural networks. 2012. http://books. google. com/books

[22] Taylor, A., Leblanc, S. and Japkowicz, N., 2016, October. Anomaly detection in automobile control network data with long short-term memory networks. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA) (pp. 130-139). IEEE.

[23] Chollet, Fran(c)ois and others 2015 Keras https://keras.io

[24] Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, VincentMichel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Pretten-hofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learningin python.Journal of machine learning research, 12(Oct):28252830,2011.