Unsupervised real-time anomaly detection system for vehicular network security

Chundong Wang[1,2], Zhentang Zhao[1,2*], Likun Zhu[1,2], Zheli Liu[3], and Xiaochun Cheng[4]
[1]Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology
Tianjin University of Technology, Tianjin, China, 300384
[2]Key Laboratory of Computer Vision and System, Ministry of Education
Tianjin University of Technology, Tianjin, China, 300384
[3]School of Computer Science and Engineering, Nankai University, Tianjin, China, 300350
[4]Department of Computer Science, Middlesex University, London NW4 4BT, UK
michael3769@tjut.edu.cn

Abstract

An unsupervised machine learning based anomaly detection system by hierarchical temporal memory (HTM) based learning algorithm is proposed to enhance the security of vehicular network. Firstly, the frequency distribution of Controller Area Network (CAN) packets is extracted as a meaningful feature to detect attacks in the CAN traffic. Then the features of CAN packets are learned by HTM-based module to predict what it expects to happen next. Furthermore, a novel anomaly score is calculated to analyze the probability of each class to discriminate normal and attack status. The system protects vehicles by monitoring the CAN bus to detect threats in real time, including detecting anomalies that might indicate a sophisticated adversary hiding in the vehicle's systems. Finally, it is demonstrated with experimental results that the proposed method can provide a real-time anomaly detection to the attack in vehicular network.

Keywords: vehicular network security, real-time anomaly detection, HTM learning algorithm

1 Introduction

Recently, with increasing popularity of vehicular network applications, automobiles are referred to as ”computers on wheels” with newer models containing more than 100 million lines of code. All this code provides features such as forward collision warning systems and automatic emergency braking to keep drivers safe. This code offers other benefits such as traffic detection, smartphone integration, and enhanced navigation. The millions of lines of code in vehicles operate dozens of electronic control units (ECUs), such as the brakes, wipers and steering, that control automotive systems. Today's vehicles come with hundreds of sensors and ECUs that control everything from vehicle operations (steering, braking, and accelerating) to peripheral functions such as door locks or infotainment systems. These ECUs interact in simple networks called Controller Area Network (CAN) that allow them to communicate at high speeds.

The on-board diagnostics (OBD-II) port, which has been mandated in vehicles since 1996, is enabled car manufacturers to test for emissions and conduct inspections. The current design of the OBD-II port gives it unlimited access to some or all of a car's internal networks. An external interface also allows access from outside the car via WiFi, Bluetooth, or cellular. In modern vehicles the CAN bus is connected to the OBD-II port and can often receive and relay all traffic to the external side of the interface. Unfortunately, most of the ECUs were designed when access to the bus required physical access to the vehicle

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 3, Article No. 5 (October 15, 2017)

[*]Corresponding author: Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, 300384 Tianjin, China, Email: 1194576202@qq.com

and therefore security was not a primary concern. Most of the smart devices such as smartphone, tablet and laptop computers can have security or privacy problems when they are compromised by malicious attacks. Therefore, we need to develop detection and prevention designs to react the emerging threats on vehicles.

In order to enhance the security of vehicular network, there are two methods. One is to prevent the attack [11], [5], [7] and second is to detect and mitigate potential risks [16], [9], [19]. Some security protocols were designed considering the limited data payload of the CAN data frame. However, these protocols are not suitable for deployment in the vehicle environment since they do not support real-time data processing and do not consider connection with external devices such as an automotive diagnostic tool. Considering the security vulnerabilities of current CAN, a newer protocol can secure technologies in future cars, but they may not be useful for on-road vehicles.

In this paper, we propose a novel anomaly detection system exploiting a recent progress of hierarchical temporal memory (HTM) networks. HTM provides a flexible and biologically accurate framework for solving prediction, classification, and anomaly detection problems for a broad range of data types [1], [6]. HTM networks can learn time-based sequences in a continuous online fashion using realistic neuron models that incorporate nonlinear active dendrites [2], [12]. Our system will real-time monitor CAN messages to detect anomaly status and generate alert system. The following are the main contributions of this paper.

1. We directly extracted features based on CAN identifier in the network. It requires only little computational complexity for the feature extraction. Any features from the analyzed data field may yield a better performance, but the complexity is intractable to the real-time detection in the anomaly detection system of in-vehicle networks.

2. Anomaly detection system HTM-based can learn continuously, which is also called to as ”on-line learning”. With each change in the inputs the memory of the system is updated. There are no batch learning data sets and no batch testing sets as is the norm for most machine learning algorithms. HTM builds a predictive model for the feature of CAN packets, which means that at every point in time the HTM-based system is predicting what it expects will happen next. The prediction is compared to what actually happens and forms the basis of learning. Another advantage of continuous learning is that the system will constantly adapt to the patterns in the CAN networks.

3. Finally, a novel judgment using anomaly score is proposed to detect whether network is normal status between the actual and predicted binary vectors. As long as the vectors are sufficiently sparse and of sufficient dimensionality, the score is that branching sequences are handled correctly.

The rest of the paper is organized as follows. In Section 2, we introduce different approaches and techniques about research of in-vehicle networks security. In Section 3, we discuss some model and assumption. We specifically introduce our anomaly detection system for in-vehicle networks in Section 4. Further, we describe the results of the experiment and performance analysis in Section 5. Finally, we conclude the paper in Section 6.

2 Related work

2.1 Adventures in vehicular network

Recently researchers were able to exploit the weakness in a car system build on top of basic CAN network by injecting malicious data into the internal bus. Effects of their hack ranged from acts like switching the lights on, to potentially fatal acts like applying brakes. Charlie Miller and Chris Valasek demonstrated a hack by attaching an external device [13]. They demonstrated their recent work at Blackhat 2015, in which they hacked a Jeep Cherokee [15] remotely without even attaching any external device. More reports [14] have come out listing various vehicles from popular car makers like Volkswagen, Skoda, Volvo, etc. that are vulnerable to another kind of crypto attack on keyless entry.

Every research report published thus far points out the vulnerability of in-vehicle CAN as a fundamental cause of car hacking. That is, a secure vehicle environment against hacking may be created by eliminating the vulnerability of vehicular networks.

2.2 Research on prevention method

To provide a in-vehicle CAN communication environment secure against replay attacks, Lin et al. [11] proposed a MAC generation technique using a message counter and a pair-wise symmetric key (PWSK). PWSK utilization means that a sender must generate as many MACs as receivers in the communication group and transmit them to each individual receiver. This technique will significantly increase the bus load, and therefore, is impractical. In addition, their security technique does not consider data confidentiality and connectivity with external devices. Therefore, in the studies presented, both security and availability are not ensured because of the limited data payload of the CAN data frame. Groza et al. [4] suggested a CAN data authentication protocol using a TESLA-like protocol. In TESLA, a sender attaches a MAC computed with a key k known only to the sender to each data packet. A short time later, the sender sends k to the receiver, who can then authenticate the data. To ensure real-time processing in CAN, key disclosure delay in the TESLA-like protocol should be minimized. However, as the delay decreases, the bus load increases. Therefore, a TESLA-like protocol is not available to a communicative environment requiring real-time processing.

Woo et al. [5] suggested a CAN data frame authentication technique considering the limited data payload of the CAN data frame. They proposed a security protocol using a truncated 32-bit MAC for data frame authentication. However, the extended ID field and the CRC field cannot be used for data transmission in an actual CAN protocol. Therefore, the security protocol cannot be applied to an actual in-vehicle CAN environment.

2.3 Research on attack detection

There have been several researches to detect attacks targeted on vehicles. Muter and Asaj proposed an entropy-base anomaly detection method [16]. They defined the notion of entropy on CAN bus and detected the intrusion by comparing entropy to a reference set. Song et al. [20] propose the light-weight IDS based on analysis of time intervals of CAN messages for in-vehicle networks. If time interval of a new message is shorter than normal, they judge the message as a injected message. Nair et al. [17] proposed a Hidden Markov Model for the prediction of anomalous status in vehicles. Kang et al. [9] proposed a intrusion detection method using a deep neural network. They trained high-dimensional CAN packet data to figure out the underlying statistical properties of normal and hacking CAN packets and extract the corresponding features.

3 Model and assumption

3.1 Attacker abilities

An adversary has access to an automotive diagnostic tool to acquire a CAN data frame to force control of an ECU before launching an actual attack. The attacker can eavesdrop and inject the CAN data frame using a malicious self-diagnostic app into the in-vehicle CAN in the connected car environment. Thus, the attacker does not have to attack the target from a short range. The app may be widely spread through the app markets by masquerading as a legitimate self-diagnostic app for a vehicle.

3.2 Target vulnerabilities

The target vehicle uses CAN to communicate among ECUs. As mentioned in [10], [8], CAN does not offer security services such as encryption or data frame authentication. This means that eavesdropping and replay attack in CAN are possible. The unauthorized use of automotive diagnostic tools is also a security hole since the tool stores control commands for the ECUs.

4 Anomaly detection system using HTM algorithm

4.1 Overview of proposed anomaly detection system

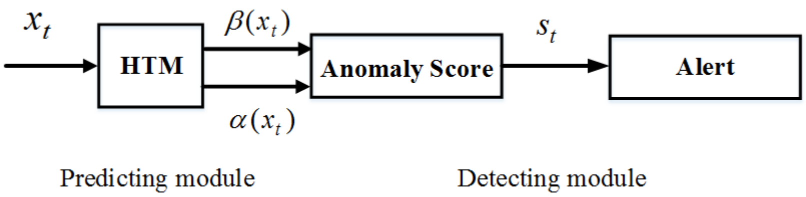

Figure 1: A predicting module and a detection module in the proposed anomaly detection system

The proposed anomaly detection system includes a predicting module and a detecting module as show in Fig 1. The predicting module is implemented based on HTM learning algorithm. HTM networks are continuously learning and model the spatiotemporal characteristics of their inputs. We can predict current state which is dependent on its previous states by using the predicting module in vehicle network. A given input will lead to different predictions depending on the current detected sequence and the current inferred position of the input within the sequence. The quality of the prediction is dependent on how well the HTM is modeling the current data stream. However, HTM algorithm do not directly outputs an anomaly score. We need compute a raw anomaly score from the two sparse vectors. Our module will generate an alert, if the anomaly score in the sequence is going below a set threshold value. Detailed explanations on mechanisms of the modules will be given below.

4.2 CAN packet feature

The proposed method extracts a CAN packet feature as an abstract representation of the system status. In other words, at each point in time t we would like to determine whether the behavior of the system up to that point is unusual by CAN packet feature. One of the key challenges is the performance of an anomaly detection. And another key factor is that the determination must be made in real-time, i.e. before time $t+1$ and without any look ahead.

To fulfill the goal, we create features directly extracted from CAN identifier in the network. Additionally, the frequency of normal CAN packets is very predictable. For example, we made a capture over 30 minutes in the Ford Escape on the high speed CAN bus. This included starting and stopping the engine, driving, braking, etc. In particular, the speed is controlled by packet with identifier 0201 on the high speed CAN network. The CAN ID 0201 had the following frequency distribution as shown in Fig 2.

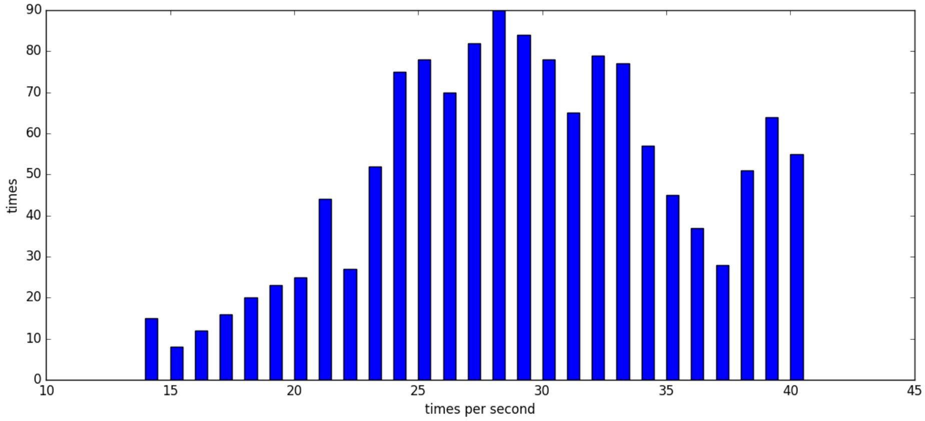

Figure 2: Ford CAN ID 0201 frequency distribution

The 0201 packet showed up 28 times in a second 90 times. Likewise, it showed up only 15 times in a second only 8 times. While messages being injected by attackers, ECUs still send their messages cyclically, Eventually, the rate of messages on the network can be increased more than two times (typically 20-100 times higher).

And as shown in Fig 3, the 0420 packet showed up only 2 times per second over 300 different times. It never showed up more than 7 times per second. Some attacks stand out greatly from normal CAN traffic and could easily be detected.

Therefore we propose that the system can detect CAN anomalies based on the known frequency of certain traffic and can alert a system or user if frequency levels vary drastically from what is well known.

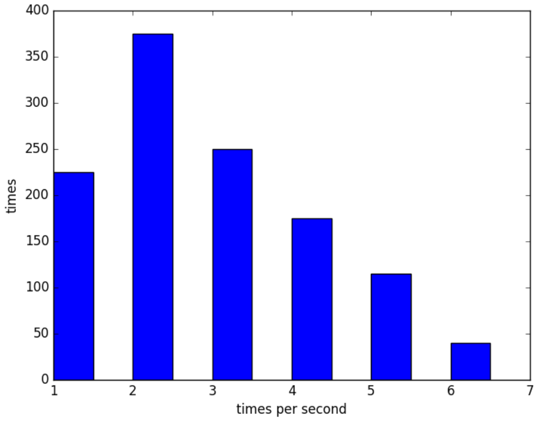

Figure 3: Ford CAN ID 0420 frequency distribution

4.3 Predicting based on HTM learning algorithm

Hierarchical Temporal Memory (HTM) is a biologically inspired machine intelligence technology that mimics the architecture and processes of the neocortex. Fig 4 shows the core algorithm components and representations within a typical HTM system. The current input x_t is fed to an encoder and then a sparse

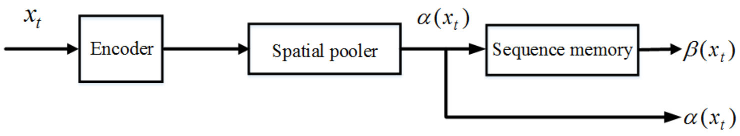

Figure 4: HTM core components and representations

spatial pooling process. The resulting vector, $\vec{\alpha}(x_t)$, is a sparse binary vector representing the current input. The heart of the system is the sequence memory component. This component models temporal patterns in and outputs a prediction in the form of another sparse vector $\vec{\beta}(x_t)$. $\vec{\beta}(x_t)$ is thus a prediction for $\vec{\alpha}(x_t)$.

The details of the encoding data and the spatial pooling process are described in depth in [18], [3], respectively. Here we describe the activation and learning rules for an HTM sequence memory network. There are three basic aspects to the rules: initialization, computing cell states, and updating synapses on dendritic segments.

Let N represent the number of mini-columns in the layer, M the number of cells per column, and NM the total number of cells in the layer. Each cell can be in an active state, in a predictive state, or in a non-active state. At any time step t, the set of active cells is represented by the $M \times N$ binary matrix $\vec{\alpha}^t$, where α_{ij}^t is the activity of the i'th cell in the j'th column. Similarly, the $M \times N$ binary matrix $\vec{\beta}^t$ denotes cells in a predictive state for time t, where β_{ij}^t is the predictive state of the i'th cell in the j'th column. Each cell is associated with a set of distal segments, D_{ij}, such that D_{ij}^d represents the d'th segment of the i'th cell in the j'th column. We use $\overset{\sim d}{D_{ij}}$ to denote a binary matrix containing only the connected synapses.

1) Initialization: the network is initialized such that each segment contains a set of potential synapses (i.e., with non-zero permanence value) to a randomly chosen subset of cells in the layer. The permanence values of these potential synapses are chosen randomly: initially some are connected (above threshold) and some are unconnected.

2) Computing cell states: All the cells in a mini-column share the same feed forward receptive fields. We assume that an inhibitory process has already selected a set of k columns that best match the current feed forward input pattern. We denote this set as W^t. The active state for each cell is calculated as follows:

$$\alpha_{ij}^t = \begin{cases} 1 & if\ j \in W^t\ and\ \beta_{ij}^{t-1} = 1 \\ 1 & if\ j \in W^t\ and\ \sum_i \beta_{ij}^{t-1} = 0 \\ 0 & otherwise \end{cases} \tag{1}$$

The first line will activate a cell in a winning column if it was previously in a predictive state. If none of the cells in a winning column were in a predictive state, the second line will activate all cells in that column. The predictive state for the current time step is then calculated as follows:

$$\beta_{ij}^t = \begin{cases} 1 & if\ \exists_d \left\| \overset{\sim d}{D_{ij}} \cdot A^t \right\|_1 > \theta \\ 0 & otherwise \end{cases} \tag{2}$$

Threshold θ represents the spiking threshold. At a given point in time, if there are more than θ connected synapses with active presynaptic cells, then that segment will be active. A cell will be depolarized if at least one segment is active.

3) Updating segments and synapses: the HTM synaptic plasticity rule is a Hebbian-like rule. If a cell was correctly predicted, we reinforce the dendritic segment that was active and caused the depolarization.

Specifically, we choose those segments D_{ij}^{d} such that:

$$\beta_{ij}^{t-1}|_{\forall j\in w^{t}} > 0 \; and \left\|\overset{\sim d}{D}_{ij} \cdot A^{t-1}\right\|_{1} > \theta \tag{3}$$

The first term selects winning columns that contained correct predictions. The second term selects those segments specifically responsible for the prediction.

If a winning column was unpredicted, we need to select one cell that will represent the context in the future if the current sequence transition repeats. To do this we select the cell with the segment that was closest to being active, i.e., the segment that had the most input even though it was below threshold. Let $\overset{\circ d}{D}_{ij}$ denote a binary matrix containing only the positive entries in D_{ij}^{d}. We reinforce a segment where the following is true:

$$\sum_{\mathrm{i}} \beta_{ij}^{t-1}|_{\forall j\in w^{t}} =0 \; and \left\|\overset{\circ d}{D}_{ij} \cdot A^{t-1}\right\|_{1} =max_{\mathrm{i}}(\left\|\overset{\circ d}{D}_{ij} \cdot A^{t-1}\right\|_{1}) \tag{4}$$

4.4 Anomaly detection

We compute a raw anomaly score that measures the deviation between the model's predicted input and the actual input. It is computed from the intersection between the predicted and actual sparse vectors. At time t the raw anomaly score, st, is given as:

$$\mathrm{s}_{t} = 1 - \frac{\vec{\beta}\,(x_{t}) \cdot \vec{\alpha}\,(x_{t})}{\|\vec{\alpha}\,(x_{t})\|^{2}} \tag{5}$$

Where $\vec{\alpha}\,(x_{t})$ is the scalar norm, i.e. the total number of 1 bits in $\vec{\alpha}\,(x_{t})$. The raw anomaly score will be 0 if the current input is perfectly predicted, and 1 if it is completely unpredicted, or somewhere in between depending on the similarity between the input and the prediction. An interesting aspect of this score is that branching sequences are handled correctly. In HTMs, multiple predictions are represented in $\vec{\beta}\,(x_{t})$ as a binary union of each individual prediction. Similar to Bloom filters, as long as the vectors are sufficiently sparse and of sufficient dimensionality, a moderate number of predictions can be represented simultaneously with exponentially small chance of error. The anomaly score handles branching sequences gracefully in the following sense. If two completely different inputs are both possible and predicted, receiving either input will lead to a 0 anomaly score. Any other input will generate a positive anomaly score.

5 Experiment

5.1 Attack construction

We assume an attacker to target an instrumental panel to deceive by showing a wrong value of the speed on the panel. We determine a syntax of a CAN packet such as an identifier, a data field, and so on, but also add some Gaussian noises if the corresponding fields present some analog values. After the speed control identifier is determined, we tamper and send it at 10 to 20 times inherent frequencies to deceive system.

In our experiment, the packets are generated to show almost a max speed value with a small random generated noise while the attacker keep with injecting an attack packet in the middle of the time to confuse a system. The attack packet aim to modify a value of the speed on the panel. So for example the

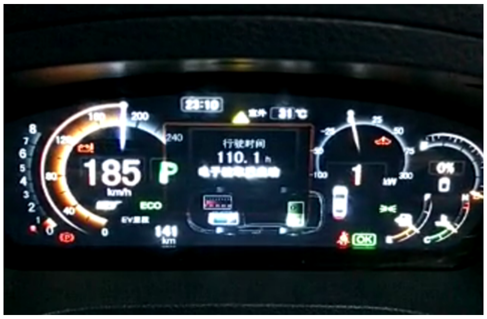

Figure 5: An instance of tampering speedometer

attack packet, when played continuously, will result in the speedometer reading 185 kilometers per hour in Fig 5.

5.2 Result

Fig 6 show the time intervals of the selected CAN ID at actual status and predictive state, respectively. We found that the HTM module can achieve precise forecasting through unsupervised learning. It provides a guarantee for the next step that we can provide real time detection of abnormal state in vehicle networks.

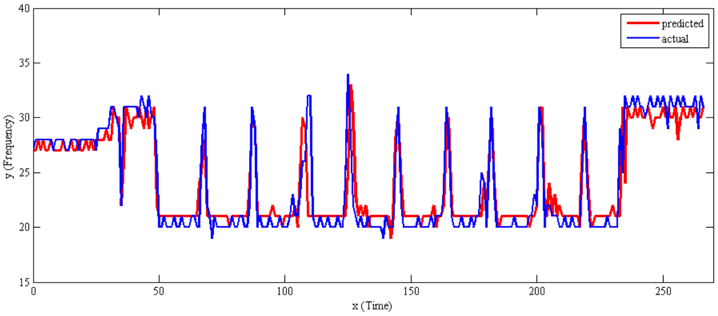

Figure 6: An instance of tampering speedometer

More after Fig 7 show an anomaly detection instance of tampering speedometer by injecting malicious data into the in-vehicle CAN. The malicious packets are consisted of CAN ID 0201 after tampering. And messages are injected faster than the own cycle of the CAN ID. We injected malicious message two times. The first injection started at 116 seconds. The second injection started at 164 seconds continued for about 4 seconds. Initially, the anomaly score is very high. This is expected. It happens because the model is still learning the patterns in the dataset. Prior to learning the patterns in the data, everything seems unfamilar to the model which leads it to output a high anomaly score. After the familiarizing itself with the patterns in the data, new and unseen patterns will trigger a high anomaly score.

In addition, we injected messages of selected different CAN IDs with double, quintuple, and decuple than original speed. Our anomaly detection system can classifies the all attack status and normal status. But we acknowledge the fact that we need to test our method with anomalous status of varying degrees.

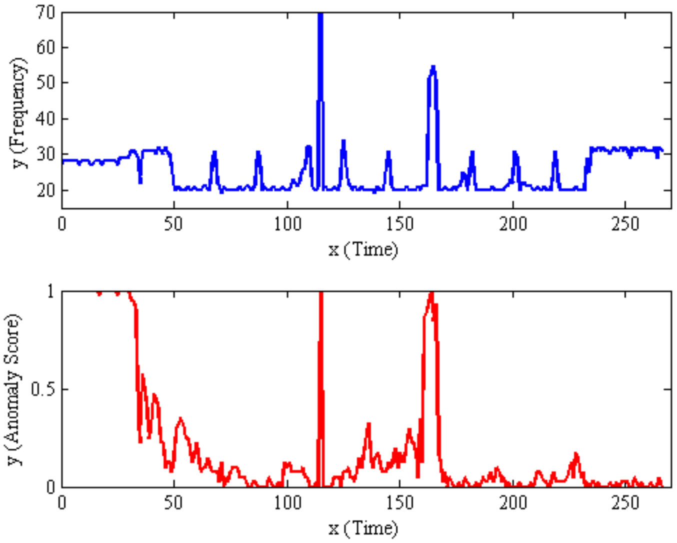

Figure 7: An anomaly detection instance of tampering speedometer

6 Conclusion

With the increase in connected real-time sensors, the detection of anomalies in vehicle networks is becoming increasingly important. In this paper we introduced a system of HTM-based anomaly detection for in-vehicle networks. We successfully extract data from various manufactures like Ford Escape, Toyota and BYD by attaching a device to their OBD port. Firstly, we extracted the frequency distribution of CAN packets as feature to recognize anomaly status. Then the feature of CAN packets are learned with HTM-based module to predict what it expects will happen next. At the last, a novel anomaly score provides the detection of each class to discriminate normal and hacking status, and, thus the system can identify any malicious attack to the vehicle as a result. It is also demonstrated with experimental results that the proposed method can provide a real-time anomaly detection to the attack in vehicle network.

The approach includes several advantages with respect to a practical deployment in the automotive industry. At first, it only requires a record of in-vehicle network traffic as input for the normal behavior. No details about the specification of allowed messages, cycle times, header sizes, etc. are required. Moreover, it allows an easy extension by learning other patterns in the CAN networks, such as the value changing information of other sensors. Besides, unlike some other methods, our method could successfully be utilized in both older and newer vehicles based on a new record of benign in-vehicle network traffic. We plan to extend our work by analyzing more attacks for further evaluations.

Acknowledgments

Our work was supported by the Foundation of the Educational Commission of Tianjin, China (Grant No.2013080), the General Project of Tianjin Municipal Science and Technology Commission under Grant (No.15JCYBJC15600), the Major Project of Tianjin Municipal Science and Technology Commission under Grant(No.15ZXDSGX00030), and NSFC:The United Foundation of General Technology and Fundamental Research (No. U1536122).

References

[1] S. Ahmad and J. Hawkins. Properties of sparse distributed representations and their application to hierarchical temporal memory. *arXiv*, (1503.07469):1–18, March 2015.

[2] S. D. Antic, W.-L. Zhou, A. R. Moore, S. M. Short, and K. D. Ikonomu. The decade of the dendritic nmda spike. *Journal of neuroscience research*, 88(14):2991–3001, November 2010.

[3] Y. Cui, S. Ahmad, and J. Hawkins. The htm spatial pooler: a neocortical algorithm for online sparse distributed coding. *bioRxiv*, page 085035, March 2016.

[4] B. Groza and S. Murvay. Efficient protocols for secure broadcast in controller area networks. *IEEE Transactions on Industrial Informatics*, 9(4):2034–2042, March 2013.

[5] B. Groza and S. Murvay. A practical wireless attack on the connected car and security protocol for in-vehicle can. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):993–1006, March 2015.

[6] J. Hawkins and S. Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in neural circuits*, 10:085035, March 2016.

[7] K.-D. Kang, Y. Baek, S. Lee, and S. H. Son. An attack-resilient source authentication protocol in controller area network. In *Proc. of the 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS'17), Beijing, China*, pages 109–118. IEEE, May 2017.

[8] M.-J. Kang and J.-W. Kang. Security threats to automotive can networks–practical examples and selected short-term countermeasures. *Computer Safety, Reliability, and Security*, pages 235–248, March 2008.

[9] M.-J. Kang and J.-W. Kang. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, 11(6):e0155781, March 2016.

[10] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. Experimental security analysis of a modern automobile. In *Proc. of the 2010 IEEE Symposium on Security and Privacy (SP'10), Berkeley/Oakland, California, USA*, pages 447–462. IEEE, May 2010.

[11] C.-W. Lin and A. Sangiovanni-Vincentelli. Cyber-security for the controller area network (can) communication protocol. In *Proc. of the 2012 International Conference on Cyber Security (CyberSecurity'12), Alexandria, Virginia, USA*, pages 1–7. IEEE, December 2012.

[12] G. Major, M. E. Larkum, and J. Schiller. Active properties of neocortical pyramidal neuron dendrites. *Annual review of neuroscience*, 36:1–24, July 2013.

[13] C. Miller and C. Valasek. Active properties of neocortical pyramidal neuron dendrites. *Annual Review of Neuroscience*, 21:260–264, March 2013.

[14] C. Miller and C. Valasek. Remote exploitation of an unaltered passenger vehicle. Black Hat USA, March 2015.

[15] C. Miller and C. Valasek. A survey of remote automotive attack surfaces. Black Hat USA, March 2015.

[16] M. Müter and N. Asaj. Entropy-based anomaly detection for in-vehicle networks. In *Proc. of the 2011 IEEE Intelligent Vehicles Symposium (IV'11), Baden-Baden, Germany*, pages 1110–1115. IEEE, June 2011.

[17] S. Nair, S. Mittal, and A. Joshi. Obd securealert: An anomaly detection system for vehicles. In *Proc. of the IEEE Workshop on Smart Service Systems (SmartSys 2016)*. IEEE, May 2016.

[18] S. Purdy. Encoding data for htm systems. *arXiv*, (1602.05925):235–248, March 2016.

[19] R. Rieke, M. Seidemann, E. K. Talla, D. Zelle, and B. Seeger. Behavior analysis for safety and security in automotive systems. In *Proc. of the 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP'17), St. Petersburg, Russia*, pages 381–385. IEEE, May 2017.

[20] 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.

Author Biography

Chundong Wang received the BSc degrees in computer science from Tianjin Normal University, China, in 1991, He received the MSc and PhD degrees in computer science from Nankai University, China, in 2002 and 2007, respectively. Currently, he works at Tianjin University of Technology as an Professor. His current research interests include network information security, pervasive computing, mobile computing and intelligent information processing.

Zhentang Zhao received her B.S. degree from Liaocheng University in 2015. He is currently studying for a master's degree at Tianjin University of Technology, Tianjin, China. His research interests includes Wireless security and Security of Vehicle networking.

Likun Zhu received his B.S. degree in communication engineering from North China University of Science and Technology in 2016, and he is currently working towards the MSc degree in Information and Communication Engineering at Tianjin University of Technology, China. His main research is directed to the architecture of wireless security, indoor location and wireless perception.

Zheli Liu received the BSc and MSc degrees in computer science from Jilin University, China, in 2002 and 2005, respectively. He received the PhD degree in computer application from Jilin University in 2009. After a postdoctoral fellowship in Nankai University, he joined the College of Computer and Control Engineering of Nankai University in 2011. Currently, he works at Nankai University as an Associate Professor. His current research interests include applied cryptography and data privacy protection.

Xiaochun Cheng Dr Xiaochun Cheng had his BEng on Computer Software in 1992 and his PhD on Artificial Intelligence in 1996. He has been a senior member of IEEE since 2004. He is the secretary for IEEE SMC UK&RI. He is a member of IEEE SMC: Technical Committee on Systems Safety and Security. He is also a committee member of European Systems Safety Society.