Context-aware Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus

Harsha Kumara Kalutarage
Omar Al-Kadri
h.kalutarage@rgu.ac.uk
o.alkadri@rgu.ac.uk
Cyber Security Group, School of Computing Science and Digital Media, Robert Gordon University
Aberdeen, United Kingdom

Madeline Cheah
Garikayi Madzudzo
madeline.cheah@horiba-mira.com
garikayi.madzudzo@horiba-mira.com
HORIBA-MIRA Ltd
Nuneaton, United Kingdom

ABSTRACT

Automotive electronics is rapidly expanding. An average vehicle contains million lines of software codes, running on 100 of electronic control units (ECUs), in supporting number of safety, driver assistance and infotainment functions. These ECUs are networked using a Controller Area Network (CAN). Security of the CAN bus has not historically been a major concern, however, recent research demonstrate that CAN has many vulnerabilities to cyber attacks. This paper presents a contextualised anomaly detector for monitoring cyber attacks on the CAN bus. Proposed algorithm is based on message sequence modelling, using so called N-grams distributions. It utilises only benign data (one class) for training and threshold estimation. Performance of the algorithm was tested against two different attack scenarios, RPM and gear gauge messages spoofing, using data captured from a real vehicle. Experimental outcomes demonstrate that proposed algorithm is capable of detecting both attacks with %100 accuracy, using far smaller time windows (100ms) which is essential for a practically deployable automotive cyber security solution.

KEYWORDS

In-Vehicle Networks, CAN bus, Automotive Cyber Security, Context-aware Anomaly Detection

ACM Reference Format:
Harsha Kumara Kalutarage, Omar Al-Kadri, Madeline Cheah, and Garikayi Madzudzo. 2018. Context-aware Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus. In *Proceedings of CSCS '19: ACM COMPUTER SCIENCE IN CARS Symposium (CSCS '19).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Modern automobiles are increasingly becoming intelligent and smarter, offering range of exciting new features such as telematics, advanced driver assistance and augmented reality displays. An average vehicle contains a million lines of software codes running on 100 of micro computers (known as ECUs) to facilitate these services [28]. These ECUs spread over the entire vehicle and largely connected to one another using bus-based network called CAN, low latency, low overhead high performance bus standard. Moreover modern vehicles have number of external communication interfaces to communicate with the outside world, for example, with personal devices, vehicular ad-hoc networks and the Internet. Estimates show that 75% of cars shipped globally by 2020 will be built with the necessary hardware to connect to the internet [1]. Despite the fact that security of some of these connections and software codes may be strengthened by automotive manufactures or original equipment manufacturers (OEMs), having so many lines of codes and increased connectivity extends the potential attack surface that can be exploited by a cyber criminal. Security researchers demonstrate that their ability to implement attacks to real vehicles [30]. Vehicle hacks are potentially disastrous. Illegitimately accessing and modifying data in a vehicle is not only a security issue but also a safety issue. For example, corrupted ECU driving the brakes can lead to an accident with serious consequences for passengers, people and goods in the surrounding environment. Therefore security of connected and autonomous vehicles is a big concern for automotive manufacturers and OEMs who are now seeking methods to secure their products against Cyberattacks.

Security research in this area has taken many forms, encompassing anything from hardware security to encryption of various aspects of the vehicle (see Section 2.2). One of the larger areas of research identified was the need for the traffic stream of the internal vehicle to be in some way monitored for potentially malicious behaviour. This paper focuses on contextualising anomaly detection on the intra-vehicular network bus (see Section 2.1). Anomaly detection for security monitoring on the CAN bus has been difficult due to the fact that many actions or reactions on a vehicle can be construed as anomalous; for example, an emergency braking event carried out by the driver, whilst legitimate, is always anomalous in day-to-day driving scenarios. To mitigate or avoid false positives, context is required to tell between a legitimate anomaly and one that could be interpreted as a potentially malicious action.

The contribution of this paper starts by modelling of normal CAN behaviour, then we propose a novel context-aware anomaly detector using n-gram distributions. The main features of the proposed algorithm can be summarised as follows.

(1) The algorithm depends only on benign data (one class) for the training purpose and threshold estimation. This avoids the need of large amount of realistic attack data for model

CSCS '19, October 08, 2019, Kaiserslautern, Germany

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

building. Collecting benign data for training purpose is relatively easier in our problem than collecting attack data.

(2) It is lightweight and can be used to detect anomalous CAN messages in real time. Algorithm is capable of picking up anomalies using far smaller time windows (e.g. 100ms). This is an essential feature for a practically deployable automotive cyber security solution.

(3) Threshold is estimated in a systematic way only using benign training data. With the estimated threshold, attack messages are assigned higher anomaly certainty scores, resulting in 100% accuracy rates of attack detection.

(4) It does not depend on prior knowledge of the CAN ID and its purpose which is not generally shared by the vehicle manufacturer. This feature enables the proposed algorithm to be retrofitted to existing vehicle infrastructure as shown in Fig. 1.

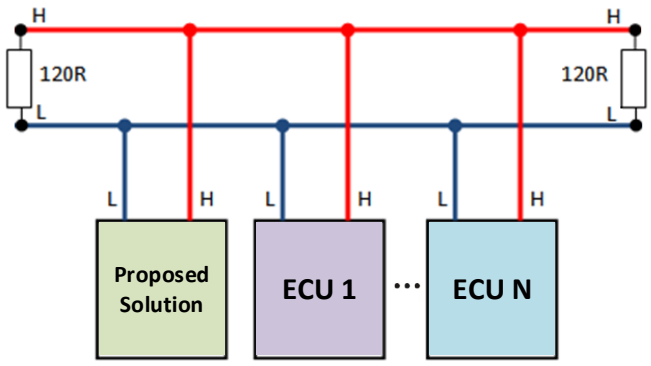


Figure 1: Demonstrative CAN network topology with the proposed solution fitted as an ECU.

The rest of this paper is organised as follows: Section 2 lays out the background including related work in this field. Section 3 provides a description of the proposed algorithm and its mathematical basis. Section 4.1 details the dataset used in this work. Section 4.2 provides a description of experimental design, with Section 4.3 exploring the results and discussions thereof. Conclusions and future work are discussed in Section 5.

2 BACKGROUND

In this section, we give the background on the ubiquitous automotive CAN protocol and then explore related work in this field.

2.1 Controller Area Network (CAN)

The CAN protocol is a ubiquitous and pervasive standard of intra-vehicular networking, the latest of which (CAN 2.0) is formally embodied in a specification document by Bosch in 1991. It is a broadcast protocol, with no addressing facilities. Instead, every ECU receives every message, but only acts on those that it is specified to recognise. Arbitration is bitwise; and every ECU in the vehicle needs to be synchronised to sample every bit at the same time, with dominant bits being the logical '0' and the recessive bit being '1'. Each message is prefixed with (amongst other things) with a CAN ID which identifies the priority of the message: the lower the CAN ID, the higher the priority. The CAN ID and its purpose is not generally shared by the vehicle manufacturer.

Due to its origin in non-connected vehicles, security of the CAN bus has not historically been a major concern, however, recent research demonstrate that CAN has many vulnerabilities to cyber attacks due to the broadcast transmission nature of communication, and the lack of encryption and authentication. This makes it susceptible to attacks, such as injection [10], replay [3], fuzzing [8], flooding [2] or through lack of space within the protocol to embed meaningful security such as encryption (although there have been proposals [15]). In many cases ECUs are exposed through external interfaces. For example, Miller et al. exploited a Wi-Fi port of a popular Jeep model to gain access and reprogram its ECUs. Authors managed to control a wide range of automotive functions remotely (e.g. disabling brake and stopping engine) [21]. The attack surface presented by the vehicle has grown due to the addition of Bluetooth, Wi-Fi and 4G in car technology exposing weaker CAN protocol.

2.2 Related work

An increased number of automotive malicious attacks have been noted in recent years. Consequently, research into anomaly detection on the automotive CAN bus has surged. This is primarily due to technological advances in vehicles manufacture and the insufficient security provisions offered by the CAN protocol. Several attempts aiming at enhancing the CAN bus security had been proposed. Authors in [16] have used a specification-based approach for intrusion detection on the vehicle network, while [19] proposed a method of blocking unauthorised data transmission on the CAN bus. In [9], authors have proposed three anomaly detection measures, which involve detection of an abnormal frequency of cyclic messages, characteristics observation of low level communication and detection of message IDs misuse. Additionally, frequency-based anomaly detection using network traffic characteristics was used for intrusion detection systems in [26]. This concept was adapted from the work conducted in [29], which statistically evaluates anomaly detection in industrial control system traffic. Such detection methods can result in good accuracy and low false-positive rate, however, their application is limited to periodic traffic. A detection model based on time interval anomalies of CAN messages is proposed in [27]. Moreover, authors in [23] have used generative adversarial networks for attack detection on the CAN bus. They have used randomly generated fake attacks to train their models, as opposed to data from real attacks. Entropy-based attack detection mechanism for CAN networks is presented in [22]. In [5] authors have proposed carefully constructed defence messages to block attackers permanently. Authors in [25] have applied time interval analysis for intrusion detection. Their system was designed to detect message injection attacks. In [18], authors have proposed an intrusion detection algorithm that aims to identify malicious injected messages. The algorithm's detection performance was demonstrated through experiments conducted on real traffic.

Though various studies have been published on anomaly detection on the automotive CAN bus, none of them focus on contextualising the anomaly detection. Inspecting alerts generated by

an anomaly detector is not sufficient to identify the cause. Identifying the context of the CAN messages can provide meanings behind events, which consequently would reduce false alarms in complex scenarios. A recent comprehensive survey on intrusion detection for in-vehicle networks had been published [30], none of the stated works focus on context-aware anomaly detection. This paper presents contextualised anomaly detection for monitoring cyber attacks on the CAN bus network, therefore, this work is different and unique.

3 PROPOSED ALGORITHM

We inspired by the successful applications of sequence modelling techniques in other areas, particularly, use cases like speech recognition, language translation and sentiment analysis in natural language processing. Video activity recognition is another trending use case of successful application of sequence modelling techniques, in which the model predicts which activity is going on based on input sequence of frames. Nevertheless, the most of data in real life are in the form of sequences, for example, it can be a number sequence, character sequence, network traffic sequence, image pixel sequence, a video frame sequence or an audio sequence, and the mathematical models have to map these input data to an output which can also be a sequence or a scalar. In our problem, the input would be a sequence of CAN messages and proposed algorithm has to map them into a label, in streaming fashion as illustrated in figure 2. In the rest of this section, we will discuss how to achieve this task using n-grams distributions [11].

As mentioned in Section 2.1, CAN bus is a robust vehicle bus standard designed for connecting ECUs and embedded computing devices, referred as nodes hereafter, to communicate with each other. Two or more such nodes are required to communicate and the complexity of a node can range from a simple I/O device to an embedded computing device with sophisticated software. A modern automobile contains about 100 of such nodes supporting various safety critical subsystems in a vehicle, e.g. engine control, airbags, ABS, power windows etc. Though some of these subsystems function independently, communications with others are essential to control actuators and receive necessary feedback from other sensors. Thus some nodes constantly broadcast their message(s) while others broadcast in responding to an event (e.g. operator's command). This forms arbitrarily long message sequences on the CAN bus which can be mathematically modelled and used to predict subsequent element(s) in the sequence, and hence to anomaly detection.

3.1 N-gram modelling for the CAN bus

Let $E = \{e_1, e_2, e_3, ..., e_m\}$ be a finite set of nodes in a CAN network. A is the massage alphabet. J={1,2,3,...,m} is an index set. Node e_i broadcasts the message s_i. Then $A = \{s_i | \forall i \in \mathbb{J}\}$. As mentioned above CAN messages (i.e. $s_i \in A$) carry everything from operator commands (e.g. roll down the windows) to readouts from sensors (e.g. reporting engine temperature) and constantly flow on the CAN bus. This process forms arbitrarily long repetitive or non-repetitive message sequence $(s_i)_{i\in\mathbb{J}}$ over the alphabet A. Depending on statistical properties (see Figure 5), these sequences can be modelled using n-grams and used to anomaly detection as described in rest of the paper.

An n-gram is any sequence of contiguous messages. For a fixed value of n, based on the Markov assumption [20], a probability for each n-gram can be estimated using the Maximum Likelihood Estimator (MLE) method. Markov assumption describes a property in a stochastic process that the occurrence of an event(s) only depends on a short history. In other words, instead of computing the probability of occurrence a message given its entire history, we can approximate the history by just the last few messages. For example, in the bigram model

$$P(s_m|s_1 s_2 s_3 ... s_{(m-1)}) \approx P(s_m|s_{(m-1)}), \quad (1)$$

in the trigram model

$$P(s_m|s_1 s_2 s_3 ... s_{(m-1)}) \approx P(s_m|s_{(m-2)} s_{(m-1)}) \quad (2)$$

and, in general, the n-gram model

$$P(s_m|s_1 s_2 ... s_{(m-1)}) \approx P(s_m|s_{(m+1-n)} s_{(m+2-n)} ... s_{(m-1)}) \quad (3)$$

and so on. Equation 3 can be extended for any value of n, and by taking relative frequency counts (C), we can estimate MLE for each n-grams. For example in the trigram model,

$$P_{MLE}(s_m|s_{(m-2)} s_{(m-1)}) = \frac{C(s_{(m-2)} s_{(m-1)} s_m)}{C(s_{(m-2)} s_{(m-1)})} \quad (4)$$

In principal, with a large enough corpus of CAN messages for normal driving behaviour, we can compute these counts and estimate the probability from equation 4. But, in practice, there might be unseen n-grams to the system that does not appear in the training corpus. Different smoothing techniques have been proposed to overcome this situation [4]. We use a technique proposed in [14] for this purpose.

3.2 Context-aware anomaly detection

Since the CAN IDs and their purpose is not generally shared by the vehicle manufacturer, it is not possible to identify the exact function of a CAN message, therefore, the "context" learned from normal driving behaviour is essential to identify anomalous messages respect to the learned context.

For example, a benign context can consist of driving with a speed of 60 MPH, infotainment system is on, cruise control is on, and AC is operating on a certain temperature. Subject to a certain variation, this will produce a set of message sequences $(s_i)_{i\in\mathbb{J}}$ that can predict the expected behaviour with respect to the learned context. A message to open the door at this instance would be certainly an "out-of-context" message that would not occur in normal circumstances. In the proposed algorithm, even though we do not know the exact nature of the messages that are being sent, we can still contextualise the scenario to identify malicious behaviour by using a sequence of messages and its frequency of appearance, which allows to predict a set of benign messages that should follow as part of that context.

When an attack progresses on the CAN bus, malicious activities can occur in an on-off pattern in the time line. This occurrence can be one time or a recurrence, and as a result, lack of agreement or harmony between points in $(s_i)_{i\in\mathbb{J}}$ can occur in a similar or different on-off fashion. However, in a Car attack, malicious CAN messages

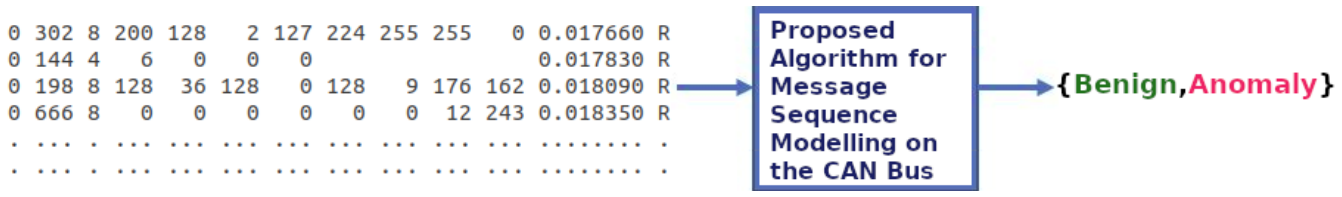


Figure 2: Illustrative diagram for the message sequence modelling on the CAN bus for anomaly detection: the input is a sequence of CAN messages while the output is a label assigned for the input.

can be exactly same as benign messages in terms of protocol data unit (PDU) structure, but the context they come into the scene may be different. To elaborate this further, as an example, take harsh braking scenario on a motorway which can be used either for malicious or benign purposes. An intruder can take over the control of a fast moving vehicle and apply harsh breaking with no driver intervention to instigate a collision. But in the benign case, driver will apply harsh breaking with other controls (e.g. steering, releasing accelerator, etc) in order to prevent a collision. We argue that these two scenarios are different in the context and produce two different message sequences, say $(s_i^m)_{i\in\mathbb{J}}$ and $(s_j^b)_{j\in\mathbb{J}}$ on the CAN bus. Though there can can be many overlaps between $(s_i^m)_{i\in\mathbb{J}}$ and $(s_j^b)_{j\in\mathbb{J}}$, however, there would be a very small chance both sequences to be identical, especially, for all possible n values in n-gram modelling. This difference can be captured via carefully trained mathematical models and used to distinguish two different scenarios. A recent work [7] supports towards this claim in which authors use in-vehicle sensor data on the CAN bus for driver fingerprinting. As per authors, using less than 8 minutes of training data, they were able to differentiate 15 drivers using the same vehicle (popular 2009 sedan) with 100% accuracy. The objective of our algorithm is to detect context dependent anomalies using CAN messages and hence to add security intelligence to the modern automobile.

3.3 Anomaly certainty ratio (λ_ω)

An anomaly is an irregularity of data patterns (or points) which deviates from expected (or normal) behaviour. The simple approach to anomaly detection is to define a region representing expected values and declare any observation lies outside that region as an anomaly [13]. To this end we define an anomaly certainty ratio (λ_ω) for a smaller observation window ω in the the time line as follows.

Given the history $h_i = \{s_1 s_2 ... s_{i-1}\}$ of i^{th} message s_i, using n-grams, we predict a set of expected values Θ for s_i. If the observed value for s_i lies outside the predicted values then a weak anomaly is declared at point of s_i. The process repeats for all messages in ω, and then λ_ω is computed as follows.

$$\lambda_\omega = \frac{C(\#\text{ weak anomalies in } \omega)}{C(\#\text{ benign messages in } \omega)} \tag{5}$$

Counting relatively weak anomalies over an observation window helps to reduce false positives. We use λ_ω as a measurement of suspicion for a ongoing attack during the ω. Algorithm 1 presents the procedure for computing the λ_ω. For the experiments presented in this paper, we set up $k = 5$ in the algorithm. Here k denotes the maximum length of n-grams to use in computing the λ_ω. However, it should be noted that k can be varied depending on the accuracy requirements and availability of the computational resource on the monitoring device.

Algorithm 1 Procedure for computing the λ_ω

```
global variables
    ν, Pre-built n-grams models
    T, Pre-defined threshold
    ω, Observation window size
    λ_ω, Anomaly certainty ratio
end global variables
procedure ComputeAnomalyCertaintyRatio(ν, ω)
    k, Maximum length of n-grams to use
    Θ, Best candidates for s_i
    if ω > 0 then
        read (s_i)_{i∈J} in ω
        foreach s_i ∈ ω do
            Take h_i for s_i, i.e. the last k-1 messages of the corresponding k-gram
            Using ν, predict the next message for h_i
            Returns the best candidate messages Θ for the s_i
            if s_i ∉ Θ then
                Declare a weak anomaly at point of s_i
            end
        end
        λ_ω ← C(Number of weak anomalies in ω) / C(Number of benign messages in ω)
        if λ_ω > T then
            return "Anomaly";
        else
            return "Benign";
        end
    end
end procedure
```

3.4 Kernel density for threshold estimation

In probability theory, the distribution of a continuous random variable can be characterised using its probability density function (pdf). The pdf describes the likelihood of taking a given value by a random variable. This is an important property which can be used for anomaly detection, as an observation lies in a very low density region of the pdf can be considered as an anomaly. We utilise this property to define a threshold (T) for the benign region of λ_ω values, and hence to anomaly detection. To this end we need

to obtain the pdf of our target variable λ_ω. A variety of approaches are available for this purpose [24], the most basic form is using a re-scaled histogram (see Figure 3).

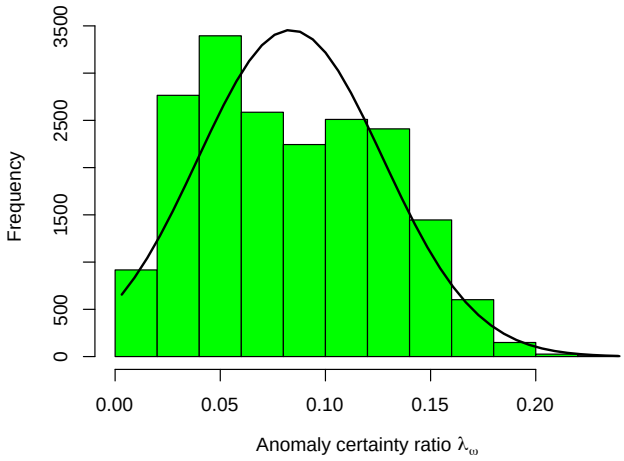


Figure 3: Re-scaled histogram for the training data with a normal curve fitted.

However, in histograms, the shape of a distribution is strongly affected by the number of bins used, hence not an effective way of estimating the density function. To avoid this weakness, Kernel density plots estimation was employed in this work. Kernel density estimation is a non-parametric way of constructing an estimate, based on observed data, of an unobserved underlying probability density function of a random variable. Figure 4 presents the Kernel density plot obtained for the random variable λ_ω, only using training (benign) data captured during a period of normal driving. Note that vertical axis in this graph represents densities (probabilities per unit), but not the actual probabilities, hence its values can exceed 1. As seen in figure 4, $p(\lambda_\omega > 0.26) = 0$. In other words, according to the pdf, there is a zero probability to take values greater than 0.26 by λ_ω if we computed it using benign data. Hence any observation taken $\lambda_\omega > 0.26$ can be considered as an anomaly, i.e. $T = 0.26$.

4 EXPERIMENTAL SETTING, RESULTS & DISCUSSION

In this section we apply our algorithm to a dataset captured from a real vehicle to demonstrate its validity in practice.

4.1 Dataset description

We use a publicly available dataset provided by Hacking and Countermeasures Research Lab (HCRL) [17]. According to the authors, the dataset was constructed by logging CAN traffic via the OBD-II port from a real vehicle while message injection attacks were being performed. HCRL provides dedicated datasets for the benign behaviour and each different attack types. Benign set contains 6082544 of CAN messages produced during a 50 minutes of normal driving

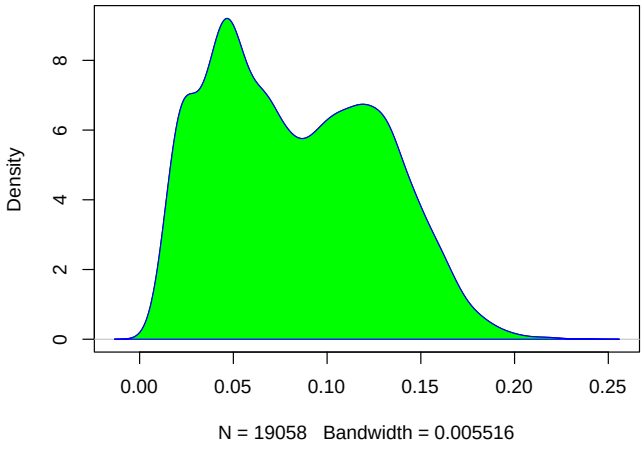


Figure 4: Threshold (T) estimation: Kernel density function obtained for the training data.

time. Each attack dataset contain 300 intrusions of message injection. Each intrusion is performed for 3-5 seconds, and each dataset has a collection of 30-40 minutes of CAN traffic.

We perform our tests and analysis using benign traffic and traffic traces from two spoofing attacks, which are spoofing engine round per minute (RPM) and gear gauge CAN messages. Spoofing attacks is defined as injecting messages of specific CAN IDs every 1 millisecond related to RPM and gear gauge information.

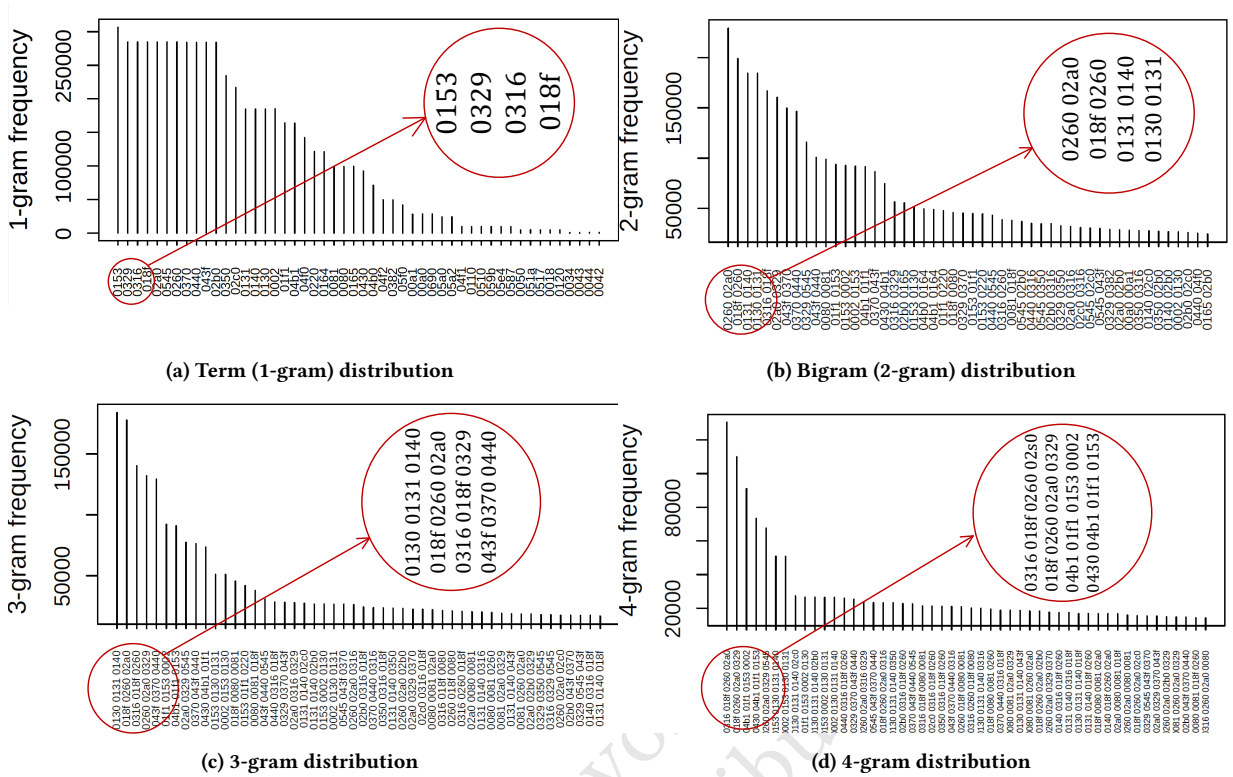
There are multiple attributes of the HCRL dataset, which are Timestamp, CAN ID, DLC, DATA[0-7], and Flag. Those attributes are defined as follows

(1) Timestamp : is the recorded time in seconds.
(2) CAN ID : is the identifier of CAN message in HEX.
(3) DLC : is the number of data bytes, ranging from 0 to 8.
(4) DATA[0-7] : is the data value (byte)
(5) Flag : There are two distinct flags, T and R, representing injected messages and normal messages respectively.

4.2 Experimental design

As mentioned earlier, we train our models only using benign traffic traces, taking so called one class based modelling approach [6, 12]. The idea here is to create the model only using benign instances, and then use the trained model to identify new/unknown instances. If the target-data is too different from the training instances then it is reported as suspicious. To this end we split benign traffic traces into two parts: training (80%) and testing (20%), but keep unchanged the temporal order of message sequences. Training set was used to build n-gram models and also to estimate the threshold value (T) which was at $\lambda = 0.26$.

In order to create the test set, which includes both benign and malicious traffic instances, above benign test set (i.e. remaining 20%) was merged with malicious traffic traces, and then resulted data set was split into 100 milliseconds smaller windows (ω). In our



(a) Term (1-gram) distribution

(b) Bigram (2-gram) distribution

(c) 3-gram distribution

(d) 4-gram distribution

Figure 5: The top 50 n-gram distributions for n=1, 2, 3, 4.

test set, attacks start at window number 676 in the time line and then continue as time progresses (see figures 6 and 7). We created two different test sets for RMP message spoofing attack and Gear gauge message spoofing attack in order to investigate algorithm's ability to detect them in isolation.

We implemented the monitoring algorithm using R and Bash scripts and run our experiments on a mobile workstation with an Intel Core^{TM} i7-6820HQ CPU and 32GB of RAM, by replaying the traffic traces in the test set,

4.3 Results & discussion

Figure 5 shows histograms of top 50 n-grams for n=1,2,3,4. Notice the case of 1-grams, a few message IDs have very high relative frequencies, for example, the one-gram "0153" appears 320000 times, followed by a set of 10 message IDs which appear 280000 times. The most common messages rapidly decrease in frequency. After the top 30 terms, the frequency of the next common 1-gram is below 10000. As we move forward in terms of the order of the n-grams, the frequencies drop but the distributions become more skewed to the left. For instance, the top 4-gram "0316 018f 0260 02a0"appears 130476 times. As we see in the plot, the top 50 4-grams occur between 14333 and 130476 times. This exploratory analysis discloses an interesting property in the CAN data, i.e. certain n-grams have higher probabilities to occur while others have lower probabilities to occur. This is an interesting property to observe as it would be useful in our probabilistic modelling, and also occurrence of an event(s) depends only on a short history.

Figures 6 and 7 present the outcomes which includes 3.3 minutes duration snapshot of the monitoring period. As seen in graphs, while both attacks progress, anomaly certainty ratio λ_ω is fluctuating far above the threshold value T. This may be due to, during the attack period, message spoofing packets have been injected in regular time intervals. If $\lambda_\omega > T$ we alert suspicious activity in a smaller window ω, otherwise benign behaviour.

It would be worth to mention here that the two second rule in driving - the safe trailing distance at any speed which a driver should maintain. Ability to detect suspicious activities using far smaller time windows (100 milliseconds in our case) by our algorithm should be acknowledged. It allows either vehicle operator or vehicle itself enough time to react in advance when an incident is alerted by the monitoring system. This is essential for a practically deployable automotive cyber security solution. Thus proposed algorithm is a practical automated solution for anomaly detection on the CAN bus.

An intrusion is different from the normal behaviour of a system, and hence anomaly detection techniques are applicable in intrusion detection domain. When there is an intruder who has no idea of the legitimate user's activity patterns (i.e. driving pattern in our problem), the probability that the intruder's activity is detected as anomalous is high. It should be noted that, however, an anomalous may not always be coincided with an intrusive activity. There may

be abnormal activities that are not intrusive, for example, an emergency breaking event on the highway. This could be alerted as a false positive by the proposed systems. Further research needs to identify such activities, which is left as a future work.

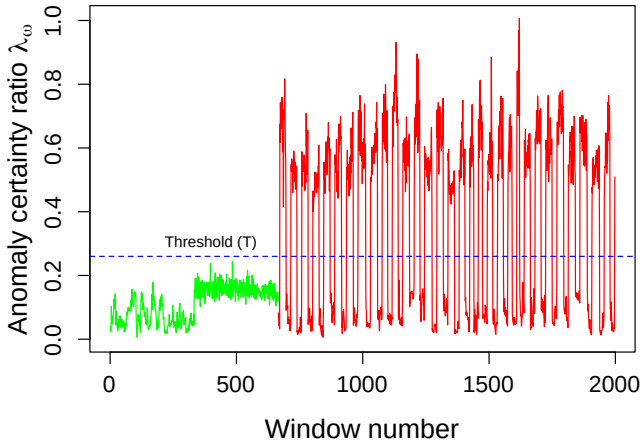


Figure 6: Monitoring for RPM message spoofing attack: Green line denotes λ_ω values during the benign data. Red line denotes λ_ω values during the attack data. Blue dotted line denotes the threshold T at $\lambda_\omega = 0.26$.

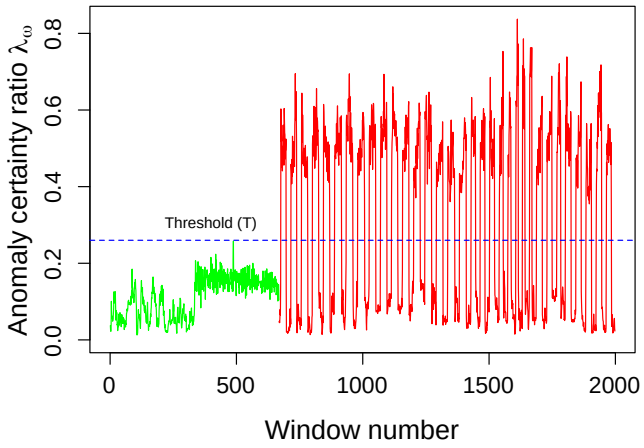


Figure 7: Monitoring for Gear gauge message spoofing attack: Green line denotes λ_ω values during the benign data. Red line denotes λ_ω values during the attack data. Blue dotted line denotes the threshold T at $\lambda_\omega = 0.26$.

5 CONCLUDING REMARKS

Securing automotive controller area networks has proven to be a necessity considering how cyber attacks can have highly disruptive or even deadly consequences. In this paper, we have proposed a context-aware anomaly detection algorithm where a regularly updated anomaly certainty ratio λ_ω is computed to determine the level of suspicious to identify the out of context messages. Results illustrate how the proposed algorithm is capable of successfully identify the deployed attacks with 100% accuracy, with zero false positive and negative rates (see Figures 6 & 7) .

In this paper, we have used the data set presented by the Hacking and Countermeasures Research Lab (HCRL), where benign and attack data is collected from a single vehicle. Further investigation will explore the performance of the proposed algorithm using different data sets with data collected from multiple vehicles within multiple scenarios. Additionally, more attack scenarios will be investigated in addition to the currently deployed message spoofing attacks to provide wider comprehension of the performance of the proposed algorithm under different circumstances.

Our goal is to build a fully automated and effective anomaly detection system for automotive Cyber Security. The performance of the algorithm will be central to address scale. It is not clear at this stage where the bottleneck will be when we deploy our approach to real life applications in production environments. For example, the current implementation was done on a Intel Core i7-6820HQ CPU, which would not be the case in real world setting as such processors usually do not exist in automotive hardware. With help from our industry partner HORIBA MIRA Ltd., further work will focus on identifying these bottlenecks to optimise the proposed algorithm. As discussed in section 3.2, distinguishing unique scenarios like emergency breaks under all conditions would be challenging as modelling such benign out of context scenarios will not be trivial. We hope to investigate this extensively in the future.

ACKNOWLEDGMENTS

This work has been partly funded by Robert Gordon University Pump priming grant ID MT104 612, we are excited to work on this challenging piece of research.

REFERENCES

[1] [n. d.]. Connecting cars to the internet has created a massive new business opportunity. https://www.businessinsider.com.au/connected-car-market-forecast-report-2015-5. ([n. d.]). Accessed: 2019-06-02.
[2] Shaikh S. Haas O. Ruddle A Cheah, M. 2017. Towards a systematic security evaluation of the automotive Bluetooth interface. (2017), 8–18.
[3] Stephen Checkoway, Damon Mccoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *20th USENIX Security Symposium*, Vol. August. The USENIX Association, San Francisco, 77–92. http://dl.acm.org/citation.cfm?id=2028067.2028073
[4] Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language* 13, 4 (1999), 359–394.
[5] Tsvika Dagan and Avishai Wool. 2016. Parrot, a software-only anti-spoofing defense system for the CAN bus. *ESCAR EUROPE* (2016).
[6] Konstantinos Demertzis, Lazaros Iliadis, and Stefanos Spartalis. 2017. A spiking one-class anomaly detection framework for cyber-security on industrial control systems. In *International Conference on Engineering Applications of Neural Networks*. Springer, 122–134.
[7] Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno. 2016. Automobile driver fingerprinting. *Proceedings on Privacy Enhancing Technologies* 2016, 1

(2016), 34–50.
[8] Bryans J. Shaikh S.A. Wooderson P Fowler, D.S. 2018. Fuzz Testing for Automotive Cyber-Security. (2018), 239–246.
[9] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. 2008. Security threats to automotive CAN networks–practical examples and selected short-term countermeasures. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 235–248.
[10] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. 2011. Security threats to automotive CAN networksPractical examples and selected short-term countermeasures. *Reliability Engineering and System Safety* 96, 1 (2011), 11–25. https://doi.org/10.1016/j.ress.2010.06.026
[11] Dan Jurafsky and James H Martin. 2014. *Speech and language processing*. Vol. 3. Pearson London.
[12] Harsha Kalutarage, Bhargav Mitra, and Robert McCausland. 2018. Modelling IoT Anomaly Detection. *ITNOW* 60, 3 (2018), 44–45.
[13] Harsha K Kalutarage. 2013. *Effective monitoring of slow suspicious activities on computer networks*. Ph.D. Dissertation. Coventry University.
[14] Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing* 35, 3 (1987), 400–401.
[15] Pierre Kleberger, Tomas Olovsson, and Erland Jonsson. 2011. Security aspects of the in-vehicle network in the connected car. In *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 528–533.
[16] U.E Larson, D.K Nilsson, and E Jonsson. 2008. . An approach to specification-based attack detection for in-vehicle network. In *IEEE Intelligent Vehicles Symposium*. IEEE, Netherlands, 220–225.
[17] H. Lee, S. H. Jeong, and H. K. Kim. 2017. OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, Vol. 00. 57–5709. https://doi.org/10.1109/PST.2017.00017
[18] M Marchetti and D Stabili. 2017. Anomaly detection of CAN bus messages through analysis of ID sequences. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, USA, 1577–1583.
[19] T Matsumoto, H Masato, T Masato, Y Katsunari, and O Kazuomi. 2012. A method of preventing unauthorized data transmission in controller area network. In *IEEE 75th Vehicular Technology Conference (VTC Spring)*. IEEE, Yokohoma, Japan, 1–5.
[20] Sean P Meyn and Richard L Tweedie. 2012. *Markov chains and stochastic stability*. Springer Science & Business Media.
[21] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* 2015 (2015), 91.
[22] Michael Müter and Naim Asaj. 2011. Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1110–1115.
[23] E. Seo, H. M. Song, and H. K. Kim. 2018. GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. 1–6. https://doi.org/10.1109/PST.2018.8514157
[24] Bernard W Silverman. 2018. *Density estimation for statistics and data analysis*. Routledge.
[25] H.M Song, H.K Kim, and H.R Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *international conference on information networking (ICOIN)*. Kota Kinabalu, Malaysia, 63–68.
[26] A Taylor, N Japkowicz, and S Leblanc. 2015. Frequency-based anomaly detection for the automotive CAN bus. In *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, London, 45–49.
[27] Andrew Tomlinson, Jeremy Bryans, Siraj Ahmed Shaikh, and Harsha Kumara Kalutarage. 2018. Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 231–238.
[28] Shane Tuohy, Martin Glavin, Ciarán Hughes, Edward Jones, Mohan Trivedi, and Liam Kilmartin. 2015. Intra-Vehicle Networks : A Review. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* 16, 2 (2015), 534–545.
[29] A Valdes and S Cheung. 2009. Communication pattern anomaly detection in process control systems. In *2009 IEEE Conference on Technologies for Homeland Security*. IEEE, Massachusetts, USA, 22–29.
[30] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li. 2019. A Survey of Intrusion Detection for In-Vehicle Networks. *IEEE Transactions on Intelligent Transportation Systems* (2019), 1–15. https://doi.org/10.1109/TITS.2019.2908074