Using A One-Class Compound Classifier To Detect In-Vehicle Network Attacks

Andrew Tomlinson
Cyber Security Group, Institute for Future Transport and Cities, Coventry University, UK
tomlin25@uni.coventry.ac.uk

Jeremy Bryans
Cyber Security Group, Institute for Future Transport and Cities, Coventry University, UK
ac1126@coventry.ac.uk

Siraj Ahmed Shaikh
Cyber Security Group, Institute for Future Transport and Cities, Coventry University, UK
aa8135@coventry.ac.uk

ABSTRACT

The Controller Area Network (CAN) in vehicles provides serial communication between electronic control units that manage engine, transmission, steering and braking. Researchers have recently demonstrated the vulnerability of the network to cyber-attacks which can manipulate the operation of the vehicle and compromise its safety. Some proposals for CAN intrusion detection systems, that identify attacks by detecting packet anomalies, have drawn on one-class classification, whereby the system builds a decision surface based on a large number of normal instances. The one-class approach is discussed in this paper, together with initial results and observations from implementing a classifier new to this field. The Compound Classier has been used in image processing and medical analysis, and holds advantages that could be relevant to CAN intrusion detection.

KEYWORDS

intrusion detection, controller area network, one-class, anomaly detection, cybersecurity, classifier, nearest neighbour

ACM Reference Format:
Andrew Tomlinson, Jeremy Bryans, and Siraj Ahmed Shaikh. 2018. Using A One-Class Compound Classifier To Detect In-Vehicle Network Attacks. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3205651.3208223

1 INTRODUCTION

Vehicle cybersecurity will become more important with the rise of autonomous and connected vehicles [19]. The vulnerability of vehicles to cyber-attack has been demonstrated in staged attacks [4, 8, 9] that focussed on the in-vehicle Controller Area Network (CAN). An attack on this network could incapacitate the vehicle, control it, or compromise its safety.

Preventive measures have been proposed (e.g. [14, 15]), although full assurance from attack is difficult to achieve, especially considering the architecture and functionality of the CAN, which is designed for speed and robustness, rather than security and authentication. Therefore, CAN security also requires the development of a viable, practicable intrusion detection system (IDS) so that ameliorative action can be taken as soon as an attack is detected [10].

Attack detection on computer networks falls broadly into two approaches: signature matching, and anomaly detection [3]. Signature approaches match traffic patterns against known attacks. They can offer low error rates and can helpfully classify the attacks. However, they rely on the ability to capture an attack model, encode it, and update the client database; all before the newly identified attack has time to proliferate.

Anomaly approaches detect anomalies in the network traffic, the assumption being that these might indicate an attack. Such approaches are more prone to detecting false positives, however, their non-reliance on known attack signatures makes them appealing in novel or unpredictable situations, such as the emerging field of in-vehicle cybersecurity. Anomaly detection methods proposed for the automotive CAN include statistical approaches [5], legitimate state transitions [17], k-Nearest Neighbours [11], and Support Vector Machines and Neural Networks [18]. Because the unpredictability of future attack scenarios makes predicting an attack class difficult, some researchers have proposed a one-class classification approach [10, 18].

A classifier that might be beneficial is the Compound Classifier proposed by Batchelor [1, 2]. This is able to generalise, learn, make rapid decisions, and can be adapted for one-class problems [2]. Devised for machine vision and colour recognition, it has also been used in medical diagnostic devices [1].

This paper presents initial findings from building a Compound Classifier, and the early testing of it using CAN data. The rest of the paper is organised as follows: Section 2 discusses the CAN, its vulnerability and requirements for intrusion detection. Section 3 states the reasons for considering CAN intrusion detection as a one-class problem. The one-class Compound Classifier is explained in Section 4. Initial testing, results and observations from building the Compound Classifier are presented in section 5. Finally, conclusions and future work are discussed in Section 6.

2 VEHICLE CONTROLLER AREA NETWORK

Vehicles are increasingly connecting to external networks through Bluetooth, cellular, or internet technologies. The connectedness of vehicles is likely to rise with increases in vehicle-to-vehicle and vehicle-to-internet networks, and autonomous vehicles, increasing the risk of cyber-attack, especially to the CAN [19].

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208223

2.1 CAN Overview

The CAN supports the electronic control units (ECUs) for the vehicle's core systems such as braking, steering, engine management, and transmission. Its protocol was formulated for fast, fault-tolerant data exchange, at a time when vehicles were isolated and security was not an issue. Consequently, it has properties that now seem inherent security weaknesses [8]:

- CAN packets are broadcast to all nodes, so a malicious node could listen to all the traffic, or broadcast to all other nodes.
- CAN has a priority-based arbitration scheme that might allow a malicious node to dominate the bus indefinitely.
- CAN packets have no authenticator field or source identifier, so a malicious node could indistinguishably send packets to the other nodes.

2.2 CAN Vulnerability

The vulnerability of the CAN has been highlighted by staged attacks, for example:

- Sniffing CAN packets via the OBD port, examining, altering and reinjecting them to control the braking and engine [8].
- Compromising the CAN through Bluetooth and cellular networks, CD-based firmware updates, and music files [4, 9].
- Injecting phoney CAN messages to control braking, steering, and acceleration at speed [12].

Proposals to reduce CAN vulnerability fall into two broad camps: i) security improvements such as authentication or access control; and, ii) intrusion detection systems (IDS) that quickly detect an attack so that ameliorative action can be taken. Both approaches are warranted, although costs, production chains, component development speed and safety priorities, form an obstacle to implementing security improvements requiring CAN modification [8].

2.3 CAN Intrusion Detection

Attributes in CAN packet flows that have been explored for patterns that might indicate attacks are: frequency [10], data content change [10], time intervals [16], and clock skews between nodes [5]. The common factor in these studies is that they have sought an anomaly detection approach.

It is likely that a viable CAN IDS would need an ensemble of detection methods, providing a consensus decision. For example, [18] propose an attack in which a compromised ECU broadcasts malicious messages to the Keep Lane Assist system, nudging the car off road. The messages would appear legitimate, with plausible values; and would only be seen to be anomalous in the context of data from other devices. We are therefore exploring a range of anomalies, including anomalies to the data payload. In particular, we are interested in classifiers that compare data from multiple sources and treat intrusion detection as a one-class problem.

3 ONE-CLASS CLASSIFICATION

In one-class classification, training data from normal operation is analysed during learning and a threshold boundary determined, within which reside the normal instances. Subsequent instances that fall beyond this threshold are flagged as anomalies, i.e. an attack in the case of an IDS.

According to [18], one-class classification is well-suited for a CAN IDS because the CAN packet flow is predictable and constant. Also, one-class classification might be preferred where training data instances of anomalous behaviour are difficult to generate or predict, or where the formulation of an anomalous class might be too restrictive to enable generalisation or application across multiple cases not available at training (such as in a biometric security system that needs to detect all candidates who are not the authorised individual) [20]. These are qualities pertinent to CAN intrusions, where the potential variety of the attack modes and the complex lifecycle of the vehicle, suggests it would be unwise to assume knowledge of the entire range of attacks.

4 COMPOUND CLASSIFIER

Methods proposed for one-class classification include: one-class support vector machines and neural networks of various types, probability estimations, decision trees, and nearest neighbour methods [7]. The method explored here is the Compound Classifier devised by Batchelor [1, 2], which is based on Euclidean distance and nearest neighbour analysis. Advantages of the Compound Classifier stated by Batchelor are:

- It lends itself well to visualisation, important for understanding and checking.
- It is suitable for dimensional analysis, where the decision surface can be represented as circles, spheres or hyperspheres.
- It has been implemented in fast hardware, and can be scaled by combining.

The Compound Classifier requires that data is numeric, and normalised to give each dimension parity. Vectors need to offer a good range of discriminatory attributes. In the case of the CAN this might be packet frequency, time between packets, message priority, or data payload values which our study used.

4.1 Decision Making by Compound Classifier

Distance between instances can be used as an indication of their similarity. The Euclidean distance D between any two vectors $U = (U_1, \ldots, U_q)$ and $V = (V_1, \ldots, V_q)$ can be calculated as:

$$D(U, V) = \sqrt{\sum_{j=1}^{q} (U_j - V_j)^2} \tag{1}$$

Although distance-based nearest neighbour analysis might be used for deciding whether an instance falls into either of two classes, it can not be used in its pure form for one-class determination, and might be prone to over-fitting, where the classifier follows the training set boundary too tightly, losing the capacity to generalise to the population. Also, there are considerable overheads in searching the full dataset when making each prediction. Therefore the Compound Classifier reduces the decision surface into a set of circles (for a vector having two attributes), spheres (three attributes) or hyperspheres (four or more attributes), which between them cover the instances in the training set[1]. Decision making compares the coordinates of the target instance against the centre coordinates and radius of each sphere. If the target instance falls within any

[1]The term "sphere" is used in the rest of this paper for convenience. Clearly the shape will depend on the dimensionality of the vector.

sphere, it is classed as normal. If it falls outside all spheres, it is classed as an anomaly. Thus, the decision of the classifier is given by:

$$M = sign\{max_{i=1\ldots N}[F_i - D_2(B_i, X)]\} \quad (2)$$

Here N is the size of the set of spheres, F_i is the radius of the ith sphere and B_i is the centre of the ith sphere, $D_2()$ is the Euclidean distance as defined in (1), X is the target vector we wish to classify. X is determined as a normal class member if M is positive or zero, and an anomaly if M is negative.

4.2 Training the Compound Classifier

Training the Compound Classifier uses the above equations, and involves adjusting the size and location of each sphere. The training set is processed for a few iterations, with the training data shuffled between each to reduce the risk of systematic biases within the time sequence of the data. During training, for each instance X:

i) If M is negative, the nearest sphere is moved towards X and is made larger. All other spheres are made smaller.
ii) If M is positive, X lies within one or more spheres. Those spheres are moved towards X and made smaller. All other spheres are made smaller [2].

The values for the sphere movement and the size-change are kept small to enhance the accuracy of the classifier, and are suggested in [1, 2], together with the desired numbers of training examples, training iterations, and initial spheres. Batchelor also devised processes for adding and pruning spheres to give efficient coverage of the training data set. As more spheres are added, the classification space could become messy, so methods are proposed to remove spheres nested inside other spheres, and divide large spheres that reduce the classifiers compactness and discernibility.

At the start of training, the initial sphere centres could use the location of any candidate instances. However, randomly assigned centres might not be optimal starting points. An alternative is to choose a subset of instances that are furthest apart, ensuring the initial spheres are spread throughout the data space. However, this might favour lone outliers, which again might be suboptimal. Therefore, following analyses conducted by [13], the Compound Classifier adopts "Maximindist", which combines nearest neighbour measurements with probability density function estimates, to locate the initial sphere centres close to cluster centres.

5 EXPERIMENT AND RESULTS

A Compound Classifier following specifications in [1, 2] was built using Python on a standard laptop. Python holds the advantage of being implementable on small single-board computers, such as the Raspberry Pi, which could be used for testing in situ. Initial analysis used three data fields captured from the CAN log of a journey lasting a few minutes. Data values were averaged over 0.1 second slots, enabling the data fields to be combined into an array with three columns, one per field, and nearly 6000 records, one per 0.1 second. The fields comprised a small subset of those identified within the CAN log (28 packet-types, each with up to eight data fields), but were chosen because their data clearly represented sensor data, i.e the values showed steady change and had a large range of values. The data was taken from packets having three separate CAN IDs, so probably came from three different ECUs. Manufactures do not

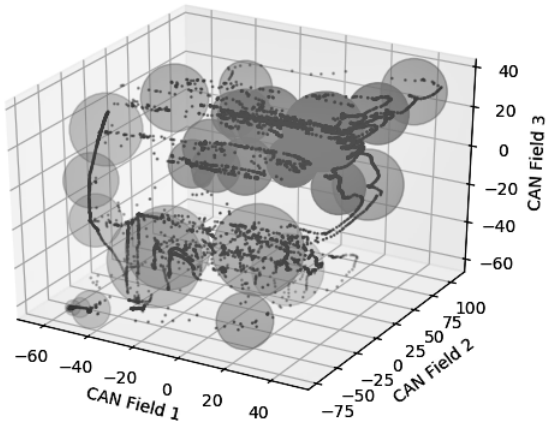

Figure 1: Compound Classifier on completion of training sufficient to enclose 95% of instances. During training spheres have grown and moved, and new spheres added. The training data is shown as dark points.

publish their CAN specifications, including the meaning of the data, how values are derived, and the broadcast triggers. However, matching the values in the fields to driving events suggested they were connected to the speed of the car, throttle position, or some related aspect of engine output.

Fig. 1 includes the normalised values for the data fields plotted against each other. They show a complex correlation, with a mixture of clusters and thin, twisting, filaments. This might make them challenging for a classifier, including the Compound Classifier which has been used in applications such as colour recognition, where instances tend to describe a simpler, more globular, clustering.

The spheres in Fig. 1 are the classifier on completion of training sufficient to enclose 95% of training instances. Training took a few minutes, with the classifier osculating in the 90% region for many of the iterations. The 95% level seems to be the maximum that the classifier in its original form can reach with this data set. We curtailed attempts to achieve higher levels when the classifier showed no improvement after running for many iterations.

As can be seen in Fig. 1, a few clusters remain outside spheres. Nearly 5% of training data points remained outside of the classifier and would incorrectly be classified as anomalies, hence potential attacks. We tested the classifier against data from another journey, and obtained a similar classification rate. CAN packets are broadcast at a high rate, so this would correspond to many false positives per second. A CAN IDS would clearly have to achieve false positive rates far lower than this. Changes to the training algorithm (such as ensuring added sphere locates are outside the existing classifier surface), or adjustments to the training parameters, might improve the classifier. This would require empirical research.

Testing the ability of a classifier to detect attack data presents challenges. For example, one type of attack demonstrated on the CAN involves injecting packets with fuzzed data; that is, data values that are random [9]. This attack might be done to confuse the system

or to test the system responses. Mimicking the values that might be used by a naive attacker, we separately fuzzed each field in a test sample using randomly generated values relevant to the bit-size capacity of the field. The classifier was poor at detecting this attack, managing to classify only 65%, 52% and 45% of attack records depending on which field was fuzzed. Fig. 1 shows large voids in some of the spheres which would contribute to misclassification. We repeated this with training and test data from another journey, but results were similar. These results seem poor, although a fuzzing attack will inevitably produce some records that fall within the cluster of normal data. Thus a perfect detection rate is elusive. That said, comparing different types of classifier against the same data set would determine which showed the most discerning decision surface.

6 CONCLUSION AND FURTHER WORK

CAN intrusion detection warrants being treated as a one-class problem. The long life of vehicles, their divergent maintenance scenarios, the diversity of currently envisaged attacks, and newness of the field, suggest unforeseen attack modes will emerge. Studies have identified CAN attributes that might indicate an attack, including time intervals between packets and number of packets broadcast in unit time. More challenging is the detection of manipulated packet payload, since this can present values that might be legitimate in other driving contexts. The Compound Classifier could offer utility by detecting anomalies in one ECU sensor-reading when compared against others. Its qualities include: producing a decision structure that lends itself to visual inspection; the ability to generalise, and the reduction of the normal data set into a smaller number of spheres enabling rapid classification.

In our initial tests, a Compound Classifier determined boundaries around a class comprising three CAN data fields that had a complex correlation pattern. However, the class boundaries still left many instances incorrectly classified. We have yet to explore whether improvements could be made by adjusting the training parameters or algorithm. So far, we have tested the classifier with only three dimensions of sensor readings, representing only a small part of the data fields broadcast within the CAN. Attacks might involve any of these, so an IDS would need to be configured accordingly.

Methods, such as one-class SVMs, are worth considering in comparison, and benefit from broader empirical testing, as well as pre-built packages in tools such as Python and R. A CAN IDS is likely to need an ensemble of detection methods, each optimal for a specific task. We will therefore be comparing a range of methods.

We have only tested the classifier on a single type of attack — fuzzing, in which the CAN packet is flooded with random data. Attacks that involve plausible data (e.g. mis-broadcasting the angle of the wheels for a lane assist system) might be even harder to detect since the attack data is likely to reside closer to the heart of the normal class. Also problematical is our current reliance on simulating an attack by manipulating previously captured output logs, which presents a simplistic rendition. This problem is common to other studies in this field, since cyber-attacking a actual car is costly and dangerous. Here, developments in simulation test rigs, such as those proposed by [6], are promising, and we are looking at how to validate them and integrate them into our research.

REFERENCES

[1] Bruce G Batchelor. 1978. Classification and Data Analysis in Vector Spaces. In *Pattern Recognition: Ideas in Practice*, Bruce G Batchelor (Ed.). Springer US, Chapter 4, 65–116. https://doi.org/10.1007/978-1-4613-4154-3_4

[2] Bruce G Batchelor. 2012. Colour Recognition. In *Machine Vision Handbook*, Bruce G Batchelor (Ed.). Springer-Verlag, London, Chapter 16, 665–694. https://doi.org/10.1007/978-1-84996-169-1

[3] Dhruba Bhattacharyya and Jugal Kalita. 2013. *Network Anomaly Detection*. CRC Press. 336 pages. https://doi.org/10.1201/b15088

[4] 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

[5] Kyong-Tak Cho and Kang G. Shin. 2016. Error Handling of In-vehicle Networks Makes Them Vulnerable. In *23rd ACM Conference on Computer and Communications Security*. https://doi.org/10.1145/2976749.2978302

[6] Daniel S. Fowler, Madeline Cheah, Siraj Ahmed Shaikh, and Jeremy Bryans. 2017. Towards a Testbed for Automotive Cybersecurity. *Proceedings - 10th IEEE International Conference on Software Testing, Verification and Validation, ICST 2017* (2017), 540–541. https://doi.org/10.1109/ICST.2017.62

[7] Shehroz S. Khan and Michael G. Madden. 2010. A survey of recent trends in one class classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6206 LNAI (2010), 188–197. https://doi.org/10.1007/978-3-642-17080-5_21

[8] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon Mccoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. 2010. Experimental Security Analysis of a Modern Automobile. In *IEEE Symposium on Security and Privacy*. Oakland, CA, 1–16. https://doi.org/10.1109/SP.2010.34

[9] Hyeryun Lee, Kyunghee Choi, Kihyun Chung, Jaein Kim, and Kangbin Yim. 2015. Fuzzing CAN packets into automobiles. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA* 2015-April (2015), 817–821. https://doi.org/10.1109/AINA.2015.274

[10] Leandros A Maglaras. 2015. A Novel Distributed Intrusion Detection System for Vehicular Ad Hoc Networks. *International Journal of Advanced Computer Science and Applications* 6, 4 (2015), 1–6. https://doi.org/10.14569/IJACSA.2015.060414

[11] Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. 2017. Car Hacking Identification through Fuzzy Logic Algorithms. In *IEEE International Conference on Fuzzy Systems*. Naples.

[12] Charlie Miller and Chris Valasek. 2016. *CAN Message Injection: OG Dynamite Edition*. Technical Report. http://illmatics.com/canmessageinjection.pdf

[13] M.M. Naim, J.P. Chan, and A.M. Huneiti. 1994. Quantifying the learning curve of a vision inspection system. In *IEE Conference Publication*. 3–5.

[14] Lee Pike, Jamey Sharp, Mark Tullsen, Patrick C Hickey, and James Bielman. 2015. Securing the Automobile: a Comprehensive Approach. *Embedded Security in Cars (ESCAR) Conference* May (2015). http://www.galois.com/{~}leepike/pike-car-security.pdf

[15] Christoph Schmittner, Zhendong Ma, Carolina Reyes, Oliver Dillinger, and Peter Puschner. 2016. Using SAE J3061 for Automotive Security Requirement Engineering. In *Computer Safety, Reliability, and Security: Proceedings SAFECOMP 2016 Workshops, ASSURE, DECSoS, SASSUR, and TIPS, Trondheim, Norway, September 20, 2016,*, A. Skavhaug, J. Guiochet, E. Schoitsch, and F. Bitsch (Eds.). Springer International Publishing, 157–170. https://doi.org/10.1007/978-3-319-45480-1_13 arXiv:0809.4107

[16] Hyun Min Song, Ha Rang Kim, and Huy Kang Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. *International Conference on Information Networking* 2016-March (2016), 63–68. https://doi.org/10.1109/ICOIN.2016.7427089

[17] Ivan Studnia, Eric Alata, Vincent Nicomette, Mohamed KaËĘ, Ivan Studnia, Eric Alata, Vincent Nicomette, Mohamed KaËĘ, and Youssef Laarouchi A. 2015. A language-based intrusion detection approach for automotive embedded network. In *21st IEEE Pacific Rim International Symposium on Dependable Computing*, IEEE (Ed.). Zhangjiajie, China.

[18] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. 2016. Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2016), 130–139. https://doi.org/10.1109/DSAA.2016.20

[19] The Institution of Engineering and Technology (IET) and The Knowledge Transfer Network (KTN). 2015. *Automotive Cyber Security: An IET/KTN Thought Leadership Review of risk perspectives for connected vehicles*. Technical Report. https://www.theiet.org/sectors/transport/topics/autonomous-vehicles/articles/auto-cs.cfm?

[20] Ian H Witten, Eibe Frank, and Mark A Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Morgan Kaufmann Publishers.