Anomaly Detection Approach Using Adaptive Cumulative Sum Algorithm for Controller Area Network

Habeeb Olufowobi  
Howard University  
habeeb.olufowobi@howard.edu

Uchenna Ezeobi  
Howard University  
uchenna.ezeobi@howard.edu

Eric Muhati  
Howard University  
eric.muhati@howard.edu

Gaylon Robinson  
Howard University  
gaylon.robinson@howard.edu

Clinton Young  
Iowa State University  
cwyoung@iastate.edu

Joseph Zambreno  
Iowa State University  
zambreno@iastate.edu

Gedare Bloom  
Howard University  
gedare.bloom@howard.edu

ABSTRACT

The modern vehicle has transformed from a purely mechanical system to a system that embeds several electronic devices. These devices communicate through the in-vehicle network for enhanced safety and comfort but are vulnerable to cyber-physical risks and attacks. A well-known technique of detecting these attacks and unusual events is by using intrusion detection systems. Anomalies in the network occur at unknown points and produce abrupt changes in the statistical features of the message stream. In this paper, we propose an anomaly-based intrusion detection approach using the cumulative sum (CUSUM) change-point detection algorithm to detect data injection attacks on the controller area network (CAN) bus. We leverage the parameters required for the change-point algorithm to reduce false alarm rate and detection delay. Using real dataset generated from a car in normal operation, we evaluate our detection approach on three different kinds of attack scenarios.

CCS CONCEPTS

• Security and privacy → Intrusion detection systems; Security protocols;

KEYWORDS

CAN, intrusion detection, data injection, sequential methods, change-point detection, CUSUM.

ACM Reference Format:


Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AutoSec ’19, March 27, 2019, Richardson, TX, USA
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6180-4/19/03...
https://doi.org/10.1145/3309171.3309178

1 INTRODUCTION

Increasingly, automobiles are becoming more advanced in terms of the numerous electronic control unit (ECU) embedded in them and their interaction with the outside environment. The automotive system continues to embed several computing devices which have become an essential part of the vehicle architecture. With these ECUs, modern vehicles are more connected to the outside world and external networks through various remote surfaces that proportionally increase to new vehicle features. Similar to computing devices, these ECUs are susceptible to cyber and physical attacks. This susceptibility has opened up the vehicular system to remote and physical attacks. Researchers have demonstrated how these remote surfaces can be exploited to compromise the vehicular networks and control the entire vehicle operations remotely [2, 7, 10]. Koscher et al. [7] were the first to demonstrate and performed practical attacks on vehicles by sniffing the controller area network (CAN) bus messages and reverse engineering of the ECU codes to take control of a range of automotive functions. Hoppe et al. [4] demonstrated possible attacks on the CAN bus and their vulnerabilities while Miller and Valasek [10] demonstrated an attack on the Jeep Cherokee by exploiting the weakness in the WiFi network code generation protocol. Similarly, attacks on the Bluetooth connection of a car has been demonstrated by Checkoway et al. [2]. These attacks produce unexpected changes in the patterns of messages communicated on the bus.

Of most importance is the security of the in-vehicle network that facilitate the communication of ECUs over various networks and protocols. A prominent network is the CAN that provides shared priority-based communication designed to be simple and efficient, but with no security mechanism. Remote attacks have been demonstrated on the network exploiting this security weakness to manipulate, degrade and take over control of a vehicle. Some of the major concerns are the broadcast nature, the lack of message authentication and encryption which presents an opportunity for the adversary to use them as an access point to carry out a large-scale attack on the vehicle.

Detecting changes in statistical properties of a network stream have been broadly studied in different domains, and change-point detection approach has been applied [18, 21]. The key idea of this
detection approach is to model CAN bus messages as a sequence of measurement over time to describe the vehicle behavior. Hence, detecting abrupt changes in the network can be formulated and solved as a change-point detection problem. Due to external and internal events such as fuzzy and DoS attacks on the bus, there can be a significant change in the behavior of the messages broadcast on the CAN bus. Change-point analysis can be used to determine the point or multiple points in time where the changes occurred and their degrees with a sequential approach (average delay) while controlling the false alarm rate [1]. Hence, we implement the adaptive cumulative sum (CUSUM) change-point detection procedure for time series analysis to model CAN bus messages.

In this work, we investigate the performance of the anomaly-based sequential change-point detection using CUSUM algorithm to detect data injection attacks on the CAN bus. The change-point detection monitors and compares the features of the observed message sequence against a predetermined pattern of normal behavior of the bus to detect any significant deviation. We leverage the features of the detection algorithm to reduce the number of false positive and increase the detection accuracy. Also, we examine the performance of the algorithm with different tuning parameters and the effect of attack intensity. We evaluate the effectiveness of our approach using a real-world CAN dataset. The datasets represent different attack scenarios. The main contributions of this paper are in threefold:

1. We develop a sliding window approach to identify sequential patterns of CAN bus logs which are used to characterize the adaptive CUSUM algorithm for detecting message injection attacks in real-time.
2. We use the model to differentiate normal and anomalous messages at varied intervals based on significant changes compared to a reasonably selected threshold value.
3. We prototype and evaluate the performance of our detection algorithm using CAN logs generated from a real vehicle.

1.1 Threat Model

In this paper, we assume that an adversary can perform read and write operation on the CAN bus. A read operation involves eavesdropping and intercepting messages while a write operation involves forging, replaying and transmitting anomalous messages on the bus. An adversary can gain access to the CAN network through physical or remote attack surfaces to target a particular node or compromise the entire network. To evaluate the effectiveness of our detection algorithm, we investigate the following attack scenarios:

1. Data injection attack: An adversary can execute a replay or man-in-the-middle attack by sniffing the legitimate operation of the network. In this attack, a victim ECU message structure is imitated and injected into the bus at random to disrupt the normal working of the network.
2. Denial of service (DoS) attack: In a DoS attack, the CAN bus is flooded with too many messages of high priority keeping the network busy and unavailable to other nodes.
3. Fuzzy attack: In this attack scenario, the adversary injects randomly spoofed messages of different ID. As a result, nodes in the network receive lots of messages that can cause malfunction of the vehicle.

These attacks vectors are connected. An adversary starts by intercepting messages on the bus and reverse engineer them to understand their properties. The decoded messages are then injected into the network to alter the vehicle behavior. With this, an adversary can launch a DoS attack on the network that could paralyze the entire operation of the vehicular networks. The severity of the impact of potential attacks depends on the vehicular component targeted by the adversary, i.e., an attack on the vehicle brake system, steering, and accelerator will have more impact than an attack on the infotainment system.

2 BACKGROUND

2.1 Controller Area Network Protocol Overview

The controller area network (CAN) standard is a serial communication protocol that implements the carrier sense multiple access protocols with collision detection and arbitration on message priority (CSMA/CD+AMP) developed for use in automotive applications. It is principally the dominant communication protocol in the modern automobile, as well as used in industrial automation and embedded control applications. Messages sent in the bus are broadcast to the entire node on the network. CAN efficiently implements static fixed priority non-preemptive scheduling of messages through bus arbitration. Nodes with a lower arbitration ID have higher priority and always wins bus access. When a message wins arbitration and starts transmission, it becomes non-preemptible.

The CAN bus protocol was designed to be a robust communication protocol with sophisticated error detection and handling capabilities. However, CAN has different inherent security vulnerabilities because it implements no security mechanism to protect messages exchanged on the network. These vulnerabilities make it attractive to cyber attackers to easily monitor, intercept and inject malicious messages in the network. An adversarial node can perform impersonation and replay attacks on spoofed messages to disrupt the bus operation and compromise the driver and passenger safety. Also, since the messages sent on the bus are not encrypted, reverse engineering can be used to understand vehicle functions. Besides, arbitration of the message protocol can be exploited through the transmission of high priority messages continuously. An adversary can launch a DoS attack using the message arbitration method by continuously flooding the network with malicious messages that are of high priority.

2.2 Sequential Change-Point Detection

Sequential anomaly detection describes a problem of detecting patterns in an observation at which one or more abrupt changes occur in a data sequence. Analyzing sequences in data is a statistical approach and theory for processing data in which the total number of observations is not fixed but depends somehow on the observed data as they become available. Therefore, the anomaly detection problem can be modeled as a change-point detection problem [19]. This work explores the performance of anomaly detection techniques based on the sequential data model using the change-point approach to characterize the pre-change parameters with unknown post-change parameters. A change-point is an instance in time
where the statistical properties of the data before and after this instance are noticeably different. It represents a transition in the state of the process that generates the data. The requirement for quality control motivates the development of change-point detection [14].

In sequential change-point detection, the goal is to detect as quickly as possible the point in time a change occurs in a statistical model of data and flag an alert signifying the change while reducing the false alarm rate. When an attack is detected at time $t$, the time series shows a statistical change around or at a time greater than $t$. For a quick response, the sequential hypothesis testing is often used when an attack occurs which saves memory and computation time. Thus, we consider the CUSUM statistics which is the basis of the change-point detection procedure. With a very light computation load, CUSUM uses the features of sequential and non-parametric tests to detect attacks in a time series data and is asymptotically optimal for a wide range of change-point detection problems when the time series are independent identically distributed (i.i.d.) with a parametric model [21].

### 3 CUSUM FOR CAN ANOMALY DETECTION

CUSUM algorithm was first proposed by Page [15] and is based on hypothesis testing developed for i.i.d. random variables. The CUSUM algorithm is a sequential detection technique useful for detecting irregular patterns that cause changes in an observation. To detect changes in the distribution, CUSUM periodically computes two sums, the upper control limit and the lower control limit, which represents the cumulative deviation between the expected value and the observed value. This detection rule is a comparison of the cumulative sum with an adaptive threshold which is not only updated online but also keeps a total memory of the useful information contained in the past observations. An essential feature of this algorithm is in determining and defining the regular pattern of the dataset. Deviations relative to this pattern are classified as anomalies when the upper or lower control limit exceeds a certain threshold. Using a sliding window approach, CUSUM can detect small shifts in statistical parameters (e.g., mean) relative to the regular pattern. The output of the algorithm is the potential list of anomalies along with the corresponding plot of the time series and its anomalies. This detection algorithm is a cost-effective and straightforward approach that can be adapted to different vehicles. A computing module (dongle) running the CUSUM algorithm can be connected to the vehicle OBD-II port and act as a monitoring node on the CAN bus.

#### 3.1 Adaptive CUSUM Algorithm

Adaptive CUSUM is proposed to solve the problem of unknown parameters that vary over time. The combination of the process of detecting change and parameter estimation is a practice considered to give better performance [18]. The idea is to estimate the parameters in a continuous form with the CUSUM test starting immediately regardless of the estimation accuracy. Since more sample estimation could lead to more accurate estimation, the estimation process continues while performing detection. Therefore, we model the messages transmitted in the CAN bus using change-point detection procedure. A change can be modeled using two hypotheses, $\theta_0$ and $\theta_1$ with thresholds 0 and $h$. The first hypothesis represents the statistical distribution of CAN message stream before the change while the second represent the distribution after the change. The essential steps in this algorithm are on how to decide between $\theta_0$ and $\theta_1$ and how to estimate the time of change efficiently from the measured sample of the message instances. These steps are called the detection and estimation steps respectively. We follow an online approach to develop the CUSUM algorithm as described in [3]. The framework of the adaptive CUSUM algorithm used to model messages transmitted in the CAN bus is described below.

Let $M = \{M_1, M_2, \ldots, M_n\}$ be a random set of messages observed sequentially, and are independent and identically distributed on the CAN bus network. Message $M$ represents a data frame on the CAN bus and each of the messages are released sequentially. Message $M$ is said to be “in-control” at first and each $M_i$ follows a probability density function (PDF), $p(M_i, \theta)$ depending on the deterministic parameter $\theta$. These parameters are assumed to be known mean $\mu$ and variance $\sigma^2$. This messages may contain a change that occur abruptly at time $t_c$ called the out-of-control that is modeled by an instant modification to the value of $\theta$. Therefore, $\theta = \theta_0$ before $t_c$, pre-change, and $\theta = \theta_1$ after that, post-change. When a change occurs, an alarm should be signaled as soon as possible for a proper action to be taken with few false positives. In the detection step, the problem is to decide between two possible hypotheses $H_0$ and $H_1$ from observed messages $M$. The instantaneous log-likelihood ratio test is used to decide between the hypothesis i.e., test for signaling a change, which is given by:

\[
S_i = \ln \left( \frac{p(M_i, \theta_1)}{p(M_i, \theta_0)} \right) \tag{1}
\]

and the cumulative sum from 0 to $n$ is:

\[
S_n = \sum_{i=0}^{n} S_i \tag{2}
\]

The decision function $G_n$ and the change time estimate $\hat{t}_c$ are:

\[
G_n = S_n - \min_{1 \leq t_c \leq n} S_{t_c-1} \tag{3}
\]

\[
\hat{t}_c = \min_{1 \leq t_c \leq n} S_{t_c-1} \tag{4}
\]

Equations 2, 3, and 4 gives the direct form of the CUSUM algorithm. For the real-time detection of change, the equations are rewritten in a recursive form and are given by:

\[
S_n = S_{n-1} + S_n \tag{5}
\]

The decision function $G_n$ compared to a positive threshold is given by:

\[
G_n = (G_{n-1} + S_n)^+, \tag{6}
\]

where $[a]^+ = \sup(a, 0)$. Once the abrupt change has been detected, equation 4 can be used to estimate the change time $t_c$ from the measured samples $M_1, M_2, \ldots, M_n$ efficiently. Thus, the sample size $M_i$, the reference value $k$ which determines the level of past memory held by the CUSUM statistics and the varying decision limits $h$ are the tuning parameters required for operating an adaptive CUSUM.
### Table 1: Overview of the dataset

<table>
<thead>
<tr>
<th>Type of Attack</th>
<th>Total Messages</th>
<th>Normal Messages</th>
<th>Injected Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoS Attack</td>
<td>3,665,771</td>
<td>3,078,250</td>
<td>587,521</td>
</tr>
<tr>
<td>Fuzzy Attack</td>
<td>3,838,860</td>
<td>3,347,013</td>
<td>481,847</td>
</tr>
<tr>
<td>Spoofing the drive gear</td>
<td>4,443,142</td>
<td>3,845,890</td>
<td>597,252</td>
</tr>
<tr>
<td>Spoofing the RPM gauge</td>
<td>4,621,702</td>
<td>3,966,805</td>
<td>654,897</td>
</tr>
</tbody>
</table>

### 3.2 Detection Approach

A significant feature of the proposed detection approach is the rate at which message instances are released and transmitted in the CAN bus. In normal operation, each message instance has a regular frequency or interval. When a message injection attack occurs on the bus, this rate or interval will change significantly as the ECUs under the attack will also be transmitting their message. Thus, the rate of messages on the bus is increased more than double the average rate. To characterize the message frequency, we use a window-based technique to extract a fixed length of overlapping windows from the attack-free dataset. The frequency of each message instance $M_i$ in the unique window $\omega_i$ is maintained and they form the training set.

The process steps we use in detecting abrupt changes in each window follows as described in Section 3.1. We compute the PDF $p(M_i)$ by calculating the $\mu$ and $\sigma^2$ of each sample $M_i$ of the dataset using the maintained frequency of message instances in $\omega_i$. We then calculate the CUSUM, $S_0$ as described in equation 5 by calculating the instantaneous log-likelihood ratio $S_i$ given by:

$$S_i = \frac{\mu_{M_i} - \mu_{M_0}}{\sigma_M^2} \left( M_i - \frac{\mu_{M_0} + \mu_{M_0}}{2} \right).$$

When an abnormal event is detected, and there is a shift in the process mean, the algorithm terminates and signals an alarm. The algorithm considers at least five average run length (ARL) before the alarm signals for the out-of-control ARL that is measured in a steady state. The steady-state ARL values are based on the delayed shifts in our chosen parameters.

### 4 EXPERIMENTAL VALIDATION

An evaluation was conducted using real CAN dataset available for research purposes. This dataset contains the normal vehicle operation and four different types of message injection attacks to disrupt the operation of the car. These attacks include DOS, fuzzy, and spoofing of the gear and vehicle RPM. The recorded datasets are logged through the OBD-II port of a real vehicle with complete knowledge of the ground truth of the normal and injected messages.

The DOS attack dataset contains attacks where the most dominant message with ID 0000 is injected every 0.3 milliseconds while the fuzzy dataset contains attacks where random message IDs are injected every 0.5 milliseconds to meddle with the vehicle operations. Other datasets are spoofing the drive gear and the rpm where their respective IDs are injected every 1 millisecond. Table 1 shows an overview of the overall number of messages in the dataset.

To measure the performance of our algorithm, we used the ARL function. The ARL function is the expected number of samples

$$E[k] = \sum_{x=0}^{\infty} x \cdot p(x)$$

where $p(x)$ is the probability of observing $x$ samples before an alarm is triggered. The ARL function takes two values: $ARL = E[k(x)]$, where $E[k(x)]$ is the expected number of samples before a false alarm, and $ARL = \theta$, the in-control ARL value is the expected number of samples before a shift in the mean is detected. A specific value is required for $ARL$ while we aim to minimize $ARL$ value over a range of process shifts. We also evaluate the performance by calculating the true positive rate (TPR) and the false positive rate (FPR) after measuring the true negative, true positive, false positive, and false negative.

### 4.1 Experimental Setup

The behavior of messages in the CAN bus can be learned by examining the average number of message instances and intervals between the subsequent message of the same ID. Our goal is to obtain the optimal parameters by learning these features.

We run our detection algorithm on the attack-free dataset to achieve the lowest possible false positives based on the selected interval, threshold, and window size. We learned the number of message instances in 0.355 seconds window with a usual choice of $k = 0.5$ and $h = 3$ as the CUSUM value is never greater than 3 as shown in Figure 1. These parameter values are chosen based on the performance of the algorithm on the attack-free dataset such that the algorithm reaches desired performance in respect to the mean time between false alarms $ARL_0$ and mean detection delay $ARL_1$.

As observed in the Figure 1, the CUSUM algorithm was run for 14 seconds for the attack-free dataset for different IDs with multiple instances. The graph stays in-control, and there is no presence of a change in the mean or false alarm as $G_n$ is calculated. It is also common to set the value of $k$ at (0.5 to 1) of the standard deviation $\sigma$. Therefore, we varied the value of $k$ using the standard...
We conducted three different experiments with the parameters obtained from the attack-free dataset to evaluate the adaptive CUSUM algorithm. As in the case of the attack-free dataset, we assume that the first five windows of the attack datasets do not contain anomalous messages instances and they form the training set used in estimating the parameters. In our datasets, these windows do not contain any attack data. We expect the mean of the message instances to change at an unknown time. We set the detection threshold \( h = 3 \) to detect attacks very quickly with low false alarm rate. When an attack is detected, the decision function grows continuously after the change, and an alarm is signaled when it is greater than the threshold.

For spoofing the gear and the RPM dataset, we identified the injected IDs and plotted the message instances against time in seconds in the samples and CUSUM graphs corresponding to both IDs. Figure 3 shows the plots for the attack-free dataset instances and the one-sided CUSUM chart for gear and rpm IDs. Visual inspection reveals that there is no alarm signal as \( G_n < h \).

The corresponding Figure 4 shows the same plots with the CUSUM signaling an alarm when the values of \( G_n > h \). As shown in the figure, the algorithm analyzed the incoming messages to calculate the CUSUM parameters and started detection when it reached a steady state. If \( G_n \approx 1 \) is greater than \( h = 3 \), the CUSUM algorithm alert that change has occurred and an alarm is signaled before the algorithm terminates. Similar plots were obtained with spoofing the RPM gauge dataset and the DOS attack dataset. The figures show the successful detection of the injected IDs with a very short delay. By manual analysis of both datasets, we observe that the first set of injections for the spoofing gear dataset was around time \( t \approx 1.2682 \) while our detection algorithm signals the alarm at \( t \approx 1.30 \). This implies that the detection delay for the gear data injection is \( n_d \approx 0.032 \). Similarly, manual inspection of the RPM

4.2 Experimental Results

We conducted three different experiments with the parameters obtained from the attack-free dataset to evaluate the adaptive CUSUM algorithm. As in the case of the attack-free dataset, we assume that the first five windows of the attack datasets do not contain anomalous messages instances and they form the training set used in estimating the parameters. In our datasets, these windows do not contain any attack data. We expect the mean of the message instances to change at an unknown time. We set the detection threshold \( h = 3 \) to detect attacks very quickly with low false alarm rate. When an attack is detected, the decision function grows continuously after the change, and an alarm is signaled when it is greater than the threshold.

For spoofing the gear and the RPM dataset, we identified the injected IDs and plotted the message instances against time in seconds in the samples and CUSUM graphs corresponding to both IDs. Figure 3 shows the plots for the attack-free dataset instances and the one-sided CUSUM chart for gear and rpm IDs. Visual inspection reveals that there is no alarm signal as \( G_n < h \).

The corresponding Figure 4 shows the same plots with the CUSUM signaling an alarm when the values of \( G_n > h \). As shown in the figure, the algorithm analyzed the incoming messages to calculate the CUSUM parameters and started detection when it reached a steady state. If \( G_n \approx 1 \) is greater than \( h = 3 \), the CUSUM algorithm alert that change has occurred and an alarm is signaled before the algorithm terminates. Similar plots were obtained with spoofing the RPM gauge dataset and the DOS attack dataset. The figures show the successful detection of the injected IDs with a very short delay. By manual analysis of both datasets, we observe that the first set of injections for the spoofing gear dataset was around time \( t \approx 1.2682 \) while our detection algorithm signals the alarm at \( t \approx 1.30 \). This implies that the detection delay for the gear data injection is \( n_d \approx 0.032 \). Similarly, manual inspection of the RPM

5 RELATED WORK

Anomaly-based intrusion detection system has been applied to traditional network-based systems to detect anomalous behaviors in the network. As most messages in the CAN bus tend to be periodic, detection approach using message inter-arrival time, entropy, and frequency have been proposed to analyze and detect anomalous behavior in the network [11–13, 16].

Taylor et al. [20] developed an anomaly detector by learning to predict the next data word originating from each sender in the bus using long-short-term memory (LSTM) recurrent neural network
for CAN bus anomaly detection. In [6] a deep learning based IDS is proposed to monitor CAN packet broadcast on the network by capturing the underlying statistical features of data and use them to detect attacks. Also, [9] proposed a fuzzy algorithm to distinguish between legitimate CAN messages generated by human driver and the injected messages generated by an attacker. Our approach differs from prior work because we focus on detecting anomalies by identifying changes in the statistical properties of the observed messages on the CAN bus based on the hypothesis testing.

Change-point detection has been applied to several systems including wireless network protocols, application, and CPS [5, 8, 17]. Tang et al. [17] used the non-parametric CUSUM test to find abrupt changes in a process without any a priori statistical knowledge and detected the real-time backoff misbehavior problem in IEEE 802.11 based wireless networks. Huang et al. [5] proposed the use of adaptive CUSUM algorithm for defending against false data injection attacks in smart grid networks using a Markov-chain-based analytical model. Similarly, [8] applied the CUSUM test as a collaborative quickest detection model to identify changes in distributed ad hoc networks. However, to the best of our knowledge, this paper presents the first application of the change-point detection approach to identify anomalous behavior in CAN.

6 CONCLUSION

In this paper, we present an anomaly intrusion detection system to identify message injection attacks on CAN bus. The proposed approach is based on change-point detection techniques using adaptive CUSUM algorithm to detect statistical changes and intrusions in CAN bus message stream. We utilized the instance of messages in a sample window and carefully chosen tuning parameters to identify differences in the statistical properties and detect irregular patterns of the messages. Analytical results have shown that the proposed detection algorithm can efficiently detect data injection attacks with low detection delay. Through our experiment, we showed that when the required parameters are carefully selected, there is high detection accuracy with low false alarm rate. Future work will compare the performance of the adaptive CUSUM algorithm with other anomaly detection approaches and analyze the algorithm performance under different attack scenarios and intensities. Additionally, we aim to find other interesting characteristics of the messages that can be used for hypothesis testing.

ACKNOWLEDGMENTS

This material is based upon work supported by Northrop Grumman and the NSF under Grants CNS-1646317 and CNS-1645987.

REFERENCES


Figure 5: ROC curve for varying thresholds for different window sizes using fuzzy attack dataset.