Learning-assisted Attack Detection and Classification for Controller Area Network

Ren Hu

***Abstract*—Cybersecurity in in-vehicle network, such as controller area network, is vital to the security of vehicle systems and safe-driving of drivers. In this paper, we analyze the vulnerability of controller area network from the perspective of encryption and authentication. Then, we develop learning-based attack detection models based on the hex data from in-vehicle controller area network. The best attack detection model is selected through comparing models learned by logistic regression, random forest, convolutional neural network, and support vector regression. The simulation based on 26985 samples are performed. The simulation results show that all learning models perform well on predicting 0-1 decision problem with 100% accuracy, while in predicting multiple class decision problem, random forest-based model outperforms other three models with 99.99% accuracy.**

***Index Terms*—Cybersecurity, controller area network, machine learning, logistic regression, support vector machine, convolutional neural network, random forest.**

I. Introduction

The widespread applications of in-vehicle networks, such as controller area network (CAN), LIN, MOST, etc., inevitably render the attack events in automotive in-vehicle communication network. The following Fig.1 specifies a series of interfaces of modern cars exposed, which can be potential entries of cyberattack [1].

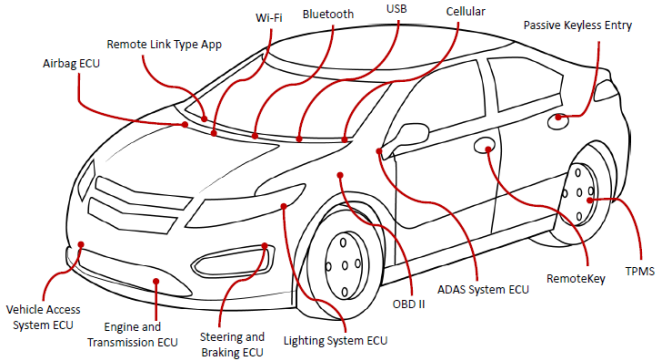

Fig.1 Potential Attack Entries distributed on cars

As the state-of-the-art self-driving techniques are developed and evolved, in the future self-driving cars may gradually substitute the current automotive vehicles. However, the convenience and advancement from the self-driving cars may still suffer from cyberattack events due to using in-vehicle controller area network.

To guarantee the cybersecurity of controller area network in vehicles, it is quite significant to explore and make efforts on the effective defense system and efficient vulnerability detection platform with respect to the controller area network. Many standard protocols have been studied a lot, such as FlexRay, CAN, MOST, etc., applied within in-vehicle communication networks [3]-[7]. Among them, the most common and popular one is the CAN, due to its merits of high reliability, large throughput of data messages, and outstanding error identification. CAN has been widely used in transmitting communication data messages between sensors and micro-controllers in vehicles [2]. Additionally, there are many references on attack detection using machine learning and deep learning [4]-[7]. Therefore, in this paper, we attempt to develop smart detection models to identify and classify the cyberattack events with CAN based on the historical in-vehicle transmission data, through machine learning and deep learning approaches.

R. Hu is with the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: hurenlaker@knights.ucf.edu).

The key of the proposed models is to mining the hex data efficiently and properly.

II. VULNERABILITY ANALYSIS

As CAN allows sensors and controller units share messages in one line in a multi-master communication fashion, the network complexity and connection expenses are greatly reduced. The properties of lightweight and robustness also are coupled with being unencrypted and lack of authentication, resulting in the vulnerabilities of CAN [1].

A. Unencrypted Data

Since from 1980s, CAN has been proposed with lightweight and robust properties. At that time, cyberattack seems to be impossible and adding encryption in CAN might be redundant and make the network communication congested. However, currently, with the popularity of internet-of-things, CAN may be attacked by spoofing, fuzzing, flooding, replaying, etc. We will discuss how to detect attacks and classify them into different groups.

B. Lacking Data Authentication

All controller units and sensors send and receive messages on the CAN bus. By default, CAN cannot prohibit any unauthorized sources from connecting the bus line and releasing malicious data to all sensors and controller units. For instance, attackers can use spoofed, or replayed data to hack into any controller units in the network.

C. Attacks

The research in CAN cybersecurity has attracted a lot attentions due to more and more breach or cyberattack events in in-vehicle networks. Refence [8] carried out and indicated that an attacker can implement the attack through On-Board Diagnostics. By probing the engineering controller unit code in CAN network, hackers can disable the brakes, stop the engine, and control other vehicle a range of functions.

III. ATTACK DETECTION METHODS AND RELATED WORK

There are mainly two clusters of attack detection approaches: anomaly-based and signature-based ones.

A. Anomaly-based Identification Approach

Anomaly-based attack identification methods in general are developed through the historical message data of CAN bus communication. The data should contain the labelled anomaly events and normal ones to represent the system's activities. To construct a good enough detection model, the data must be sufficiently provided. The existing anomaly detection systems (ADS) is used to monitor the system's activities and compare the intruded data with previous normal ones to determine how to label the data. Therefore, the ADS itself should experience large number of normal events to avoid the high false positive rate. The goal of training ADS is to enhance the detection ability with respect to the new unseen attacks.

B. Signature-based Identification Approach

Signature-based attack detection methods apply a pre-defined attack-signatures to identify cyberattack events. This class of detection methods identifies the known attacks with high accuracy. Their disadvantage is that they may not sensitive to unknown attacks or new anomaly activities not defined in the historical records. Hence, updating the historical data record and labels is significant and necessary to maintain their performance of detecting anomaly activities.

IV. METHODOLOGY

In this section, we will introduce machine learning and deep learning methods used in our research on attack detection. More precisely, since the attack detection is in fact a classification problem, some classification methods such as logistic regression (LR), random forest (RF), convolutional neural network (CNN) and support vector machine (SVM) are employed to learn the attack detection models. Each method is illustrated as below.

A. Logistic Regression

Logistic regression is originally proposed to model binary classification decision problems, i.e., binary decisions. This method models the probability of decision events, such as fail or success, dead or alive, win or lose, etc. However,

events with more than two classes can be also modeled by multinomial logistic regression. Logistic regression is a popular model choice to describe the relationship between the categorical response and explanatory variables (predictors). The specific formulation of logistic regression is shown as below:

$$\log(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + .. + \beta_n X_n \quad (1)$$

where $p(x)$ is the estimated probability of interest, and $\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n$ represents the parameters and explanatory variables from the generalized linear regression model structure.

B. Random Forest

Random forest, also called random decision forest, is an ensemble learning method through assembling multiple decision trees to solve classification and regression problems. A classification decision tree is used to predict each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. The misclassification rate is the fraction of the test observations in that region that do not belongs to the most common class. Decision trees are believed to much closer to mirror human's decision-making. However, they generally do not have the same level of predictive accuracy as some of the other classification approaches. That is why ensemble learning is used to assemble decision trees into random forest algorithm.

C. Convolutional Neural Network

Convolutional neural network (CNN) is one of deep learning methods, widely used in computer vision and natural language processing. A common CNN architecture can consist of many layers, such as convolutional layer, pooling layer, ReLU layer, etc., where pooling layer can mitigate the overfitting problem and ReLU layer can introduce nonlinearity to the decision function. Each layer has a set of parameters to learn and tune according to how the performance of model can be improved. The general architecture of CNN is shown in Fig.2 as below.

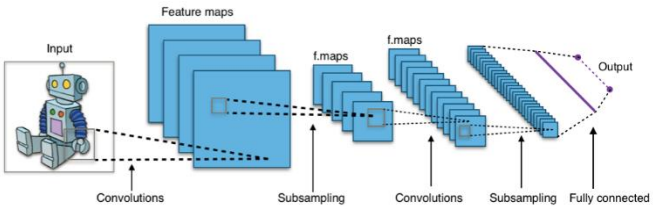

Fig.2 CNN Architecture

D. Support Vector Machine

Support vector machine is also a popular supervised machine learning method, which can be used in both classification and regression problems. Solving the model of SVM itself is a convex optimization problem, which triggers many variants of SVM, such as hard-margin or soft-margin, and nonlinear ones. Its kernel tricks have attracted a lot attentions, efficiently mapping the input data into high-dimensional space. There are linear, polynomial, Gaussian, and sigmoid kernels.

V. Case Study

To develop the best attack detection model, we implement the four methods mentioned in last section. The following parts will discuss data description, data preparation, simulation, and result analysis.

A. Data Description

In this paper, the dataset we used is from a competition of car hacking and defense based on in-vehicle network [2]. In total, there are 26985 samples randomly selected in this dataset. The dataset contains 13546 samples of attack events, including four classes of attacks, i.e., flooding, fuzzing, replay, and spoofing. It also has 13439 samples of normal events. The specific sample size information of each class of events are summarized as below in Table.1.

Table.1 Sample size of each class of events

Class	Subclass	sample size	in total
attack	Flooding	4209	13546
	Fuzzing	4341	
	Replay	4008	
	Spoofing	988	
Normal		13439	13439

Each row of samples has six indicators, i.e., Timestamp, Arbitration ID, DLC, Data, Class, and Subclass. The specific description of each indicator

is written as below in Table.2. Note that among these indicators in Table.2, we choose Class pr Subclass as the dependent variable, other indicators are independent variables where Class and Subclass are categorical data represented by strings. In Table.3, a part of samples are shown as examples.

Table.2 Description of each indicator

No.	Description of each indicator
1	Timestamp: the time when the observation is recorded.
2	Arbitration ID: includes 11-bit identifier, 1-bit dominant or recessive where the dominant means data frame and the recessive means remote frame.
3	DLC: shows how many bit at most in data field.
4	Data: hex data with different lengths.
5	Class: Normal or Attack.
6	Subclass: flooding, fuzzing, replay, or spoofing.

Table.3 A part of sample rows

Timestamp	Arbitration_ID	DLC	Data	Class	SubClass
1599043358	251	8	FE 03 17 D3 00 FE 07 80	Normal	Normal
1599043358	2B0	6	C1 FF 00 07 D7 32	Normal	Normal
1599043358	38D	8	00 00 49 00 F0 7F FE 51	Normal	Normal
1599043394	0	8	00 00 00 00 00 00 00 00	Attack	Flooding
1599043394	0	8	00 00 00 00 00 00 00 00	Attack	Flooding
1599043474	130	8	5C 2D 65 BF CE D9 C7 C6	Attack	Fuzzing
1599043474	43	8	27 96 87 6C 38 42 4F 63	Attack	Fuzzing
1599043474	4A4	8	EC 81 07 26 97 3E 22 80	Attack	Fuzzing
1599043449	130	8	F8 7F A4 80 00 00 0E 1C	Attack	Replay
1599043449	140	8	10 80 00 6E 20 00 0E AE	Attack	Replay
1599043449	164	4	00 08 08 43	Attack	Replay
1599043419	553	8	00 00 00 02 01 00 80 00	Attack	Spoofing
1599043419	553	8	00 00 00 02 01 00 80 00	Attack	Spoofing
1599043419	553	8	00 00 00 02 01 00 80 00	Attack	Spoofing

B. Data Preparation

1. Missing Values

Missing values occur when no data is recorded for an observation. A couple of methods have been suggested to deal with missing value issues: (1) ignore the record, (2) fill the missing value manually, (3) use a global constant, (4) replace the missing value with the mean, (5) replace the missing value with the mean of that category, (6) use the most likely value through the help of regression. In this study, we have few observations with missing values.

2. Outliers

An outlier is a point for which the observed value is far from the value predicted by the model. Outliers can arise for a variety of reasons, such as incorrect recording of an observation during data collection. Often outliers decrease the accuracy and efficiency of the models. The detection of outliers of the continuous/ quantitative variables can be done through the determination of the upper and lower limits, which is normally the $\pm$ 3 standard deviation from the mean value of that variable. In our study, several outliers are adjusted to be $\pm$ 3 standard deviation of the mean.

3. data transformation

Since there may have patterns and strings in dataset, transforming them into proper formats are better for the application in model fitting. For instance, the time stamp indicator has very close maximum and minimum values. Hence, normalized this column of data by subtracting the minimum value of time stamp can reduce the influence of

large order of magnitude. The arbitration ID is represented by hex data with the length between 1 to 3, shown as string data. Converting it into decimal type will make the string changed to be integer, which is easier to handle in fitting models. Data indicator is also treated as strings. Split the column of Data indicator into 8 substrings at most, then only select the four rightmost substrings to create four new columns of indicators (d1, d2, d3, and d4), replacing the original long string of Data indicator. Finally, we convert the categorical indicators Class and Subclass into dummy variables with integer labels. The summary of converting each indicator is summarized in Table.4.

Table.4 Transformation of each indicator

No.	each indicator
1	Timestamp: normalized by subtracting the minimum value of time stamp.
2	Arbitration ID: converted to decimal values.
3	Data: split it into 8 substrings at most, keep the last four substrings (d1~d4), and convert them into decimal values from hex format.
4	Class: converted to 0 and 1.
5	Subclass: converted to 0, 1, 2, 3, 4.

4. Variable Selection

Variable selection is an approach for excluding irrelevant variables from a fitted model. The stepwise regression is the most commonly used method for selecting the variables to be included in the learning models because it was found to produce the most parsimonious model. The default entry and stay significance level is 0.05. The selection criterion is the Akaike's Information Criterion (AIC). This means that the final model selected has the lowest value of AIC statistic. Through the "backward" and "both" selection methods, we obtain the same variable selection result with 6 variables, excluding DLC indicator.

5. data splitting

In our simulation, we randomly split the data into training and test dataset with 67% and 33%, respectively.

C. Simulation and Analysis

The simulation is performed using Class or Subclass as the response (dependent) variable. Other independent variables include: Timestamp, Arbitration ID, new created four columns.

Confusion matrix. Fig.3 and Fig.4 visualize the confusion matrices for binary and multiple-class decision problems through logistic regression.

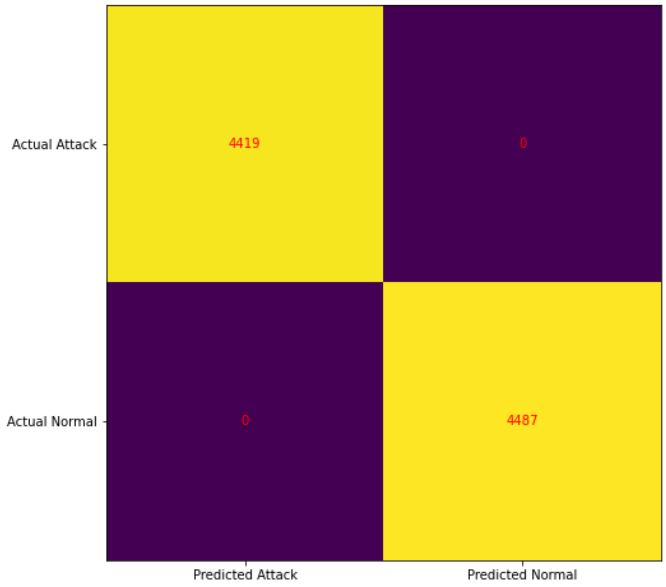

Fig.3 Confusion matrix using Class as the response

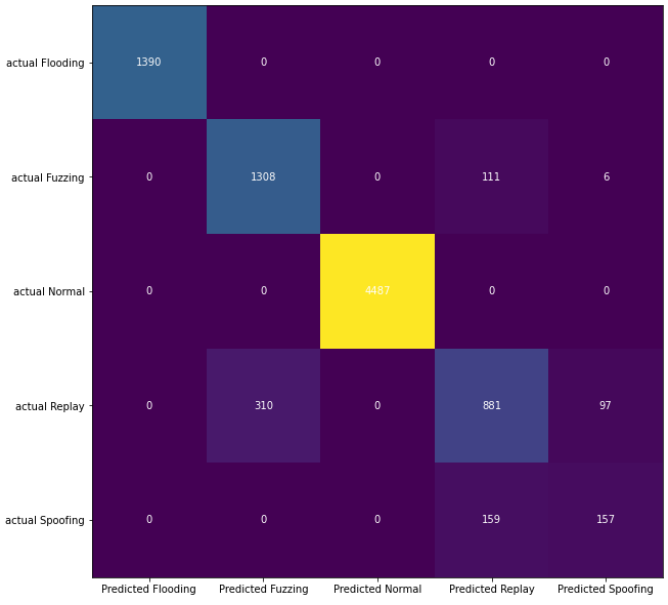

Fig.4 Confusion matrix using Subclass as the response

Model Performance. To compare the performance of four models developed through logistic regression, random forest, CNN, and SVM, their predictive accuracy data are summarized in Table.5 shown as below.

Table.5 Compare the performance of methods

	Class accuracy		Subclass accuracy	
	training	**test**	**training**	**test**
logistic	1	1	0.9233	0.9212
random forest	1	1	0.99998	0.99996
CNN	1	1	0.1613	0.1600
SVM	1	1	0.98456	0.98394

Form Table.5 we can observe that:

In predicting 2-class problem using Class as the response variable, all four methods perform perfectly with 100% accuracy.

In predicting 5-class problem using Subclass as the response variable, random forest works best with 99.99% accuracy and CNN has the worst performance.

Feature importance. To see which indicators are more important for the attack detection model, the feature importance values of indicators are computed from random forest: Timestamp (0.4425), Arbitration ID (0.5563), d1(3.2e-04), d2(2.44e-05), d3(4.46e-05), d4(8.36e-04). In general, the larger the feature importance value, the more important the indicator (feature).

VI. CONCLUSION

Timestamp and Arbitration ID are two most significant features, implying that the attacks occur at certain patterns of time interval and arbitration ID. The data message sent themselves seem to be not as important as the occurrence time and arbitration ID of the data message.

VII. REFERENCE

[1] Young, Clinton, et al. "Survey of automotive controller area network intrusion detection systems." IEEE Design & Test 36.6 (2019): 48-55.

[2] Kang, Hyunjae, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee, and Huy Kang Kim. "Car Hacking and Defense Competition on In-Vehicle Network." In Workshop on Automotive and Autonomous Vehicle Security (AutoSec), vol. 2021, p. 25. 2021.

[3] Boudguiga, Aymen, Witold Klaudel, Antoine Boulanger, and Pascal Chiron. "A simple intrusion detection method for controller area network." In 2016 IEEE International Conference on Communications (ICC), pp. 1-7. IEEE, 2016.

[4] Chockalingam, Valliappa, Ian Larson, Daniel Lin, and Spencer Nofzinger. "Detecting attacks on the can protocol with machine learning." Annu. EECS 588, no. 7 (2016).

[5] Ansari, Mohammad Raashid, Shucheng Yu, and Qiaoyan Yu. "IntelliCAN: Attack-resilient controller area network (CAN) for secure automobiles." In 2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), pp. 233-236, 2015.

[6] Lokman, Siti-Farhana, Abu Talib Othman, and Muhammad-Husaini Abu-Bakar. "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review." EURASIP Journal on Wireless Communications and Networking 2019, no. 1 (2019): 1-17.

[7] Minawi, Omar, Jason Whelan, Abdulaziz Almehmadi, and Khalil El-Khatib. "Machine Learning-Based Intrusion Detection System for Controller Area Networks." In Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, pp. 41-47. 2020.

[8] Infosec In the City, “Overview of SINCON car security kampung,” https://www.infosec-city.com/post/sin20-ctf-car-security, accessed on: Jan. 11, 2021.