Intrusion Detection of In-Vehicle CAN Bus in Intelligent Connected Vehicles Based on Tsetlin Machine*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—With the advancements in intelligent and connected vehicle technology, the Controller Area Network (CAN) bus, responsible for vehicle safety and real-time communication, has emerged as a primary target for cyberattacks, posing a threat to automotive operational safety. The paper proposes a low-power, lightweight, and highly interpretable CAN bus intrusion detection system (IDS) based on the Tsetlin Machine (TM). TM uses logic-based rules for pattern recognition, effectively detecting network intrusion behaviors. The paper introduces the CAN data processing approach and experimental configurations, experimentally validating the detection accuracy of TM under various conditions through experiments. Furthermore, TM is compared with traditional methods such as LSTM and DCNN, demonstrating its significant advantages in inference time and computational efficiency.**

***Index Terms*—intelligent connected vehicle, intrusion detection system, tsetlin machine, controller area network**

I. Introduction

With the rapid development of modern automobile technology, intelligent connected vehicles have become the mainstream of the future. These vehicles achieve smart information exchange between vehicles, roads, people, and the cloud through communication means such as Bluetooth, Wi-Fi, GPS, and cellular networks. While these technologies improve driving safety and convenience, they also pose significant challenges in network and data security. Attackers can remotely control vehicles by sending malicious data . For instance, researchers have successfully remotely hacked a Jeep Cherokee, controlling its steering, braking, and acceleration functions via the internet. Similarly, Tesla vehicles' electronic control units and Controller Area Network (CAN) bus networks have been successfully hacked.

CAN is the core protocol for internal communication in cars. It is equivalent to the neural network of a car,interconnecting diverse control systems within it. CAN adopts the broadcasting mechanism, so all connected components can send and receive control messages. It has high communication efficiency and can ensure the real-time nature of communication, so it is the bottom line of attack protection. Due to the lack of encryption and authentication mechanisms, attackers can send fake data frames by accessing the CAN bus, achieving illegal control of the vehicle. Therefore, the security of the CAN bus is crucial for ensuring the safe operation of intelligent connected vehicles.

Currently, the intrusion detection systems (IDS) for CAN bus are diverse and can be primarily grouped into traditional methodologies and machine learning-driven approaches [3]. In traditional methods, Muter et al. [4] achieved in-vehicle bus intrusion detection by analyzing changes in the entropy values of the on-board network. In the realm of machine learning, Taylor et al. [5] disregarded the dependencies within CAN data sequences, treating them as standalone sequences. They proposed utilizing long short-term memory (LSTM) neural networks to detect abnormalities in the sequential data on automotive CAN bus, achieving a remarkably low false alarm rate. Similarly, Li et al. [6] transformed CAN bus communication data into images, subsequently employing CNNs to extract key features. Through the utilization of LSTM with an attention mechanism, they effectively implemented intrusion detection for CAN bus. Although these methods improve detection accuracy to some extent, they face challenges in

Identify applicable funding agency here. If none, delete this.

real automotive environments, including high computational complexity, real-time performance issues, and excessive power consumption. Moreover, neural network-based methods use black-box operations, posing certain risks to CAN bus security. Therefore, this paper proposes a low-power, lightweight, and highly interpretable Tsetlin Machine (TM) machine learning algorithm to achieve CAN bus intrusion detection for intelligent connected vehicles.

II. Methodology

A. Tsetlin Machine

TM is a machine learning model based on logical rules, using human-understandable rules for pattern recognition and classification. Distinct from traditional neural networks, TM uses clauses to represent and learn rules, making it more interpretable to some extent. Each clause consists of Tsetlin Automaton (TA) and performs logical AND operations to derive classification results.

As shown in Figure 1, the features input to TM need to be binary converted. The original dataset is processed through a booleanization module , which generates booleanized data X and $\bar{X}$. This step also determines the number of Tsetlin Automatons (TAs) within each clause. The booleanized data is then fed into the clause computation module, where it undergoes calculations and processing using Boolean logic operations.The results of each clause are summed and passed through a threshold module to vote on the expected output class Y . During the inference phase, the final output class Y is the classification result of the input data. In the training phase, the feedback module receives the information of the expected output class Y and generates reward or punishment penalty signals. These feedback signals guide the TA within the clauses to adjust their state values, continuously improving the overall system performance.

TM also sets two hyperparameters the s value and the Threshold(T).T is used to regulate the frequency of feedback occurrences and s controls the probability of state transitions within the Tsetlin Automatons (TAs). By strategically adjusting these hyperparameters, the model can gradually optimize its learning process, learning more accurate feature matching and classification rules, enhancing the overall performance of the model.

B. CAN Data Processing

CAN data processing is the foundation of the entire intrusion detection system. The degree of effectiveness and efficiency in handling and transforming CAN data directly influences the accuracy and responsiveness of the detection process. A standard CAN data frame typically consists of multiple fields, including start bit, arbitration field (containing CAN ID), control field, data field, CRC check field, and end field. In intrusion detection, the most critical fields are the CAN ID and data content, as shown in Figure 2. CAN ID serves to identify the type and priority of the data frame, while the data content contains specific control or status information.

To satisfy to the input requirements of TM , CAN data needs to be converted into a suitable format. This work adopts a binary representation for CAN image data, as illustrated in Figure 2, to facilitate seamless processing and analysis by the algorithm. The standard CAN ID is 11 bits, and this work uses 12 bits of binary representation, reserving one bit for extension or verification. Data content, typically 8 bytes in size, is directly encoded into 64 bits of binary. Each individual CAN data frame is converted into 76-bit binary data. Subsequently, these 76-bit binary strings are stacked chronologically, forming the CAN image data that is input into TM . This chronological ordering is crucial as it preserves the temporal relationship between the CAN data frames, allowing TM to analyze patterns and detect attacks .This paper selected the Car-Hacking Dataset and trained data for four attack types: DoS, Fuzzy, Gear, and RPM. Normal data and attack data are divided according to data flags, with normal data marked as 'R' and attack data marked as 'T'. In CAN image, the mere presence of a 'T' flag indicates that the corresponding segment represents attack data.

III. Experiment and Result

A. Experiment Environment

This paper chooses the Intel(R) Xeon(R) Platinum 8255C CPU as the experimental platform. During the experiments, the hyperparameters of TM were kept at s=2.5, T=15,with a total of 10 training rounds.A confusion matrix was employed to provide a visual summary of the performance of TM . In addition to the confusion matrix, accuracy was further utilized to observe the performance of TM under different conditions. Throughout the experiment, the ratio of training data to testing data was consistently maintained at 8:2.

B. Impact of CAN Image Size on CAN Bus Detection

CAN data is a series of time-series data, with certain time dependencies between each CAN data frame. For example, during vehicle braking, CAN data of multiple related components (such as taillights, brake system, etc.) will change simultaneously.An excessively large or small image size may fail to adequately capture the characteristics of the time series, leading to a decrease in detection accuracy.To fully utilize this time dependency, this experiment adjusts the number of input CAN data to adjust the CAN image size, evaluating the impact of different sizes on detection accuracy.

As shown in Figure 3, when the input size increases from 10 to 35, the detection accuracy gradually improves. When the input size exceeds 35, the detection accuracy begins to decline. Therefore, the optimal input size is defined as 35 in this experiment , and it can be changed flexibly depending on the vehicle environment. At this size, the system ensures high detection accuracy while avoiding computational complexity and resource consumption issues caused by excessive data. The data distribution of different attack types in the system at the optimal input size is shown in Table 1.

TABLE I
CAN DATA TYPE AND SIZE

Attack type	Normal	Dos	Fuzzy	gear	RPM
Attack image	57455	6173	7265	10842	11801

C. Impact of Clauses on CAN Bus Detection

Clauses are similar to neurons in neural networks, affecting TM performance. The number of clauses is a critical factor affecting detection performance. When the number of clauses is low, the model tends to be simplistic, potentially failing to capture complex attack patterns, leading to lower detection accuracy. By increasing the number of clauses, the model complexity rises, enabling it to capture more intricate patterns and consequently improving detection accuracy. As shown in Figure 4, detection accuracy fluctuates slightly when clause numbers exceed 200. Considering overall accuracy, time, and power consumption, 200 clauses are selected for subsequent operations. The confusion matrix results obtained from the experiment, as shown in Figure 5, indicate that while there are some instances of false positives and false negatives, the values along the main diagonal demonstrate that the proposed model in this paper is able to distinguish between normal and abnormal samples quite well.

D. Comparison with Traditional Methods

To verifythe effectiveness of TM in CAN bus intrusion detection, this work compares it with two traditional algorithms: Long Short-Term Memory (LSTM) and Deep Convolutional Neural Networks (DCNN). DCNN consists of two convolution layers, each followed by a batch normalization layer, ReLU activation function, and pooling layer. Then there are two fully connected layers with a Dropout layer in between. LSTM consists of five LSTM layers, each with 128 hidden units corresponding to Dropout. Table 2 shows the performance comparison of the three algorithms in terms of accuracy, parameter count, and inference time.

TABLE II
COMPARISON OF TM ALGORITHM PERFORMANCE

Name	Accuracy (%)	Parameters	Inference Time (s)
LSTM	99.82	1,957,509	63.96
DCNN	99.92	5,002,821	78.46
TM	99.74	4,520,000	**19.13**

As shown in Table 2, although TM has slightly lower accuracy than other algorithms, TM demonstrates significantly faster inference times, even with a larger number of parameters.As shown in Figure 5.5, the parameters (clauses) in TM are discrete and each clause operates independently from the others, which is fundamentally different from the complex interconnections between neurons in neural networks. This characteristic of TM contributes to its higher computational efficiency.

E. CAN Bus Detection in Embedded Systems

To demonstrate the low latency and low power consumption of TM), we deployed TM on an embedded system, specifically the Raspberry Pi 4B. Given the resource constraints of embedded devices, this experiment bypasses the training process and directly loads a pre-trained model for inference.

For this experiment, we selected 10,000 test data points to verify the performance of TM in terms of time and power consumption.

TABLE III
XXXX

	CAN Image Conversion	TM Inference
Time	xxx	xxx
Power	xxx	xxx

Based on the data presented in Table 3, TM exhibits exceptional performance in terms of both power consumption and processing time at the edge. This makes TM highly suitable for deployment in resource-constrained edge computing environments, effectively enabling real-time detection of CAN bus intrusion.

IV. CONCLUSION

The paper proposes a CAN bus intrusion detection method based on TM. Experimental results show that TM is capable of achieving high detection accuracy while significantly reducing computational complexity and resource consumption, demonstrating its potential application in in-vehicle CAN bus systems.

Compared to traditional neural network methods, TM not only exhibits faster detection speeds but also offers greater interpretability, providing an effective solution for the security of intelligent connected vehicles. Future work can further optimize the TM algorithm and explore its application in larger-scale in-vehicle bus systems.