electronics

MDPI

Article

A Novel Intrusion Detection Model Using a Fusion of Network and Device States for Communication-Based Train Control Systems

Yajie Song *, Bing Bu and Li Zhu

State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China; bbu@bjtu.edu.cn (B.B.); lizhu@bjtu.edu.cn (L.Z.)

* Correspondence: 17120267@bjtu.edu.cn; Tel.: +86-1352-076-6670

Received: 17 December 2019; Accepted: 16 January 2020; Published: 18 January 2020

Abstract: Security is crucial in cyber-physical systems (CPS). As a typical CPS, the communication-based train control (CBTC) system is facing increasingly serious cyber-attacks. Intrusion detection systems (IDSs) are vital to protect the system against cyber-attacks. The traditional IDS cannot distinguish between cyber-attacks and system faults. Furthermore, the design of the traditional IDS does not take the principles of CBTC systems into consideration. When deployed, it cannot effectively detect cyber-attacks against CBTC systems. In this paper, we propose a novel intrusion detection method that considers both the status of the networks and those of the equipment to identify if the abnormality is caused by cyber-attacks or by system faults. The proposed method is verified on a hardware-in-the-loop simulation platform of CBTC systems. Simulation results indicate that the proposed method has achieved 97.64% true positive rate, which can significantly improve the security protection level of CBTC systems.

Keywords: CPS; CBTC; cybersecurity; IDS

1. Introduction

Urban rail transit plays an important role in addressing the issues of traffic congestion and environmental pollution. Communication-based train control (CBTC) is an automated train control system using communication technologies to ensure the safe operation of rail vehicles [1]. CBTC can improve the utilization of railway infrastructure and help to provide better services to passengers.

CBTC systems are typical cyber-physical systems (CPS) which bridge the computing and communication of the cyber world with the control of the physical world [2]. The extensive application of computer, communication and control technologies in CBTC has greatly improved the automation level of the system but exposed it to the threats of cyber-attack at the same time [3].

Over the last years, cyber security incidents in the rail transit field are increased. For instance, about 60 computers of the metro system in Seoul were infected by malware, which caused data and information leaks in March of 2014 [4]. In November 2016, the hackers hijacked the San Francisco metro-rail system by attacking the station fire systems [5]. The attack caused 2112 computers to be damaged. In 2018, an unprecedented distributed denial of service attacks interfered with Danish state rail operators. Passengers across the country were prevented from buying tickets on Sunday [6]. Since CBTC systems are safety-critical, cyber-attacks could result in emergency brake and affect the travel time of passengers seriously.

Intrusion detection systems (IDSs) are deployed to detect cyber-attacks against the system. It monitors the system for abnormal activities and policy violations to take appropriate actions immediately [7]. Building an IDS for CBTC is an efficient way to increase the security protection

level. As a result, a complete and deep analysis of the intrusion detection issue is important for CBTC systems.

Although lots of researches have been carried out on IDSs of CPS, few studies are suitable for CBTC systems. The features of CBTC have not been considered in the design of traditional IDSs. In addition, attacks are likely to trigger the fault-safety mechanism of CBTC systems. They may be misjudged as random faults, such as faults of equipment and failures of communication. As faults may have the same impact as cyber-attacks do on CBTC systems [8], traditional IDSs cannot distinguish between faults and cyber-attacks. They will lead to false negatives or false positives, as well as reduce the performance of the IDS.

This paper proposes a novel intrusion detection method for CBTC systems. A detection model that integrates the status of networks with the states of devices is set up to get comprehensive information on CBTC systems. A hidden Markov model (HMM) is used to fuse the information from different models to make decisions on the results of detection. The main contributions of this paper can be listed as follows:

1. Unlike the traditional IDS, which focus on the anomaly analysis of packets, the throughput of the network and the characteristics of the data packets in the CBTC systems are jointly considered in the detection model based on the status of networks.
2. A detection model based on the status of devices is adopted to take the fault-safe principle of CBTC systems into consideration.
3. A HMM classifier is designed that synthesizes the anomalies detected by different models. Experimental results show that the proposed IDS can differentiate cyber-attacks from random system faults.

The remaining part of the paper is organized in the following way. Section 2 introduces the related intrusion detection technologies in CPS and CBTC. A brief introduction of CBTC systems is given in Section 3. The impact of cyber-attacks on CBTC systems is analyzed. In Section 4, the architecture, the different detection models and the HMM classifier are proposed. The proposed IDS method is applied and tested on a hardware-in-the-loop simulation platform of CBTC systems in Section 5. In Section 6, the performance of the proposed IDS is discussed in detail. Last but not least, the conclusions and future work are given in Section 7.

2. Related Works

Currently, intrusion detection is an essential technology of security protection. It can be classified into signature based detection and anomaly based detection [9]. As signature based IDSs rely on fixed signatures to detect attacks, it is unable to identify unknown cyber attacks. The anomaly based methods identify attacks by detecting a deviation from normal behavior. As the anomaly based IDS can be used to detect "zero day" attacks that have not been disclosed before, it has attracted the attention of more and more researchers [10,11]. The anomaly based methods in CPS can be categorized into associated rule mining, statistical-based, and machine learning algorithms [12].

The rule-based IDS checks whether events occur together to detect attacks. It digs out the relations among the attributes of the data set to identify if the system is under cyber attack [13]. Yang designs an IDS for supervisory control and data acquisition (SCADA) systems using the associated rule mining algorithm. The IDS can realize the in-depth analysis of the protocol and deep inspection of packets [14]. As traditional IDSs cannot monitor physical behaviors, Koyena adopts association rule mining to process the data of sensors and actuators to detect more attacks [15]. Besides, most rule mining algorithms can not deal with continuous attributes. Li proposes a fuzzy if-then rules mining method to handle the vague and imprecise among data. The novel algorithm can significantly reduce false alarms in medical CPS [16].

Rare events are regarded as anomalies in the statistical based intrusion detection method [17]. Both parametric and non-parametric techniques have been applied to design statistical models for

anomaly detection. While parametric techniques estimate the parameters from the given data [18], such systems may generate incorrect results in non-stationary systems. To overcome the problem, non-parametric techniques are used [19], which can provide accurate notification of abnormal activities and detect DoS attacks without delay [20]. However, the detection rate is low when the anomaly traffic intensity is lower than 5% of the background traffic. Manikopoulos introduces a multi-window statistical method using statistical modeling and neural network classification to achieve high detection rate along with a low misclassification rate [21].

Machine learning algorithms are widely used in IDSs [11], such as decision trees, neural networks, support vector machines, clustering, and so on. Among the algorithms, decision tree is easily comprehensible and requires little data preprocessing. Sindhu uses it to construct a lightweight IDS that can discover specific attacks with a true positives rate of 98.4% [22]. When the data set becomes larger, the decision tree grows deeper and broader, and it is much more challenging to extract rules. Thus, random forest is used instead to process the vast amounts of data [23]. However, the traditional decision tree may have a low detection rate on highly imbalanced data. Jahromi combines a deep unsupervised learning approach with the decision tree for effective detection [24].

Although extensive research has been carried out on intrusion detection of CPS, few single studies exist which are suitable for CBTC systems. Melaragno proposes a signature-based rail radio intrusion detection system (RRIDS) to detect command replay, guessing, and message corruption attacks [25]. RRIDS detects intrusion by modeling each type of attack, which relies on fixed signatures and requires frequent database updates. As CBTC systems are continuously running and widely distributed in space, frequent updates may be unsuitable for CBTC systems. Zhang studies on the data tampering attacks on trains and proposes an intrusion detection method based on the running status of the train through Kalman filter and χ^2 detector [26]. However, the method can only detect data tampering attacks, which may not be effective against other anomalies. Gao proposes an improved Adaboost multi-classification IDS based on the n-gram model [27]. Experiments show that the IDSs can effectively detect attacks on the train-ground communication subsystem.

Analysis of the related works shows that the existing methods mostly focus on a certain attack or subsystem. They can not provide security protection for an entire CBTC system. Therefore, the intrusion detection mechanism should adopt a combination of different techniques to achieve good performance.

3. CBTC Systems and Cyber-Attacks

In this section, an overview of CBTC systems is firstly presented, followed by the impacts of cyber-attacks on CBTC systems.

3.1. An Introduction of CBTC Systems

As shown in Figure 1, a CBTC system consists of cyber networks and physical processes. A typical CBTC system is comprised of wayside equipment, on-board equipment, and data communication systems (DCS). The wayside equipment, including automatic train supervision (ATS), zone controller (ZC), computer interlocking (CI), and data storage unit (DSU), is connected through the wired backbone network. Through the wireless network, the wayside equipment communicates with the onboard equipment, which is called the vehicle onboard controller (VOBC), including automatic train protection (ATP), automatic train operation (ATO), and mobile station (MS). Due to the high reliability and safety requirements of CBTC systems, redundant and fault-tolerant equipment is adopted [28]. Meanwhile, redundant networks and dedicated safe communication protocols are deployed for data transmission.

In a CBTC system, the position and speed of the foregoing train are transmitted to ZC through the wireless network. After receiving the information from the foregoing train and the information of the safe route from CI, ZC generates and forwards the limitation of the movement authority (LMA), a location on the line that the train cannot travel cross, to the following train. The VOBC controls

the train to run below the protective curve, which is calculated based on the LMA and the status of the train.

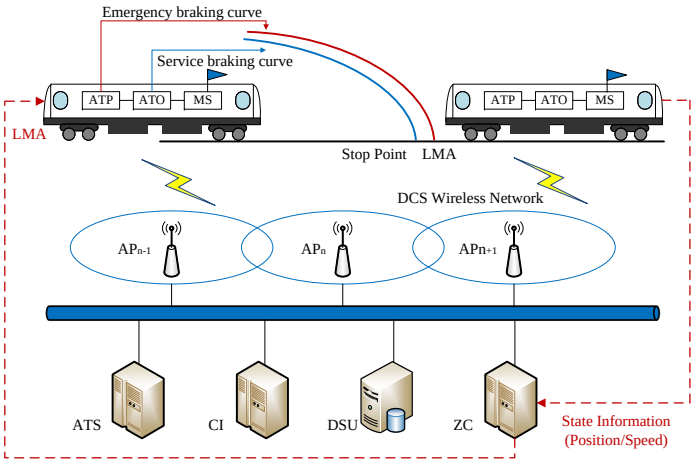


Figure 1. A communication-based train control (CBTC) system.

3.2. Impacts of Cyber-Attacks on CBTC Systems

The traditional rail system is a track-based train control (TBTC) system, which uses track circuits to transmit information [29]. As TBTC is designed physically isolated from external networks, issues of cybersecurity are not considered. With the increasing passenger volume of urban rail transit, CBTC systems are widely deployed all over the world. Commercial off-the-shelf (COTS) products are extensively used in CBTC systems, including general computers, commercial operating systems, standard communication protocols, etc. COTS improves the automation level of the system, shortens the headways between trains, enhances the capacity of the urban rail transit, however, introduces the risk of cyber-attacks at the same time. In this paper, we consider cyber-attacks that have serious impacts on CBTC, including denial of service (DoS) and data integrity attacks (DIA) [30].

When the CBTC systems operate normally, the ATP of the following train calculates the protective position/speed curve based on the received LMA. As shown in Figure 2, the LMA of the following train at time t is $L_m(t)$, the safe position of the tail of the foregoing train. The ATO of the following train calculates a service braking curve under the ATP curve and controls the train to run under the service braking curve. In this paper, we consider the cyber-attacks interfere with the operation of trains. To achieve this goal, the cyber-attacks impair the availability or the integrity of the LMAs directly or indirectly.

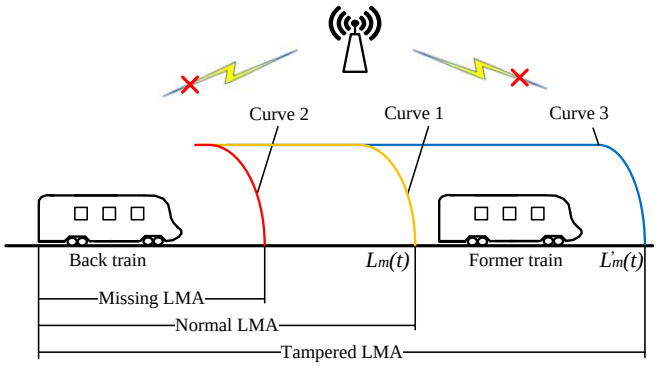


Figure 2. Schematic diagram of normal train operation.

DoS attack reduces the availability of a train's LMA. When a train cannot receive LMA, it uses the latest received LMA to calculate the ATP and the ATO curves. If the train unable to receive LMAs

continuously and the interruption time exceeds a specified threshold, it applies an emergency brake to ensure safety. As depicted in Figure 2, if the following train does not receive $L_m(t)$, it uses the latest received LMA, $L_m(t-1)$, to generate the ATP curve, C_2. As C_2 is closer to the following train than C_1, the speed of the train may be lower down unnecessarily, the efficiency of the train's operation is decreased.

If an attacker has prior knowledge of CBTC systems, he may launch a DIA attack that tampers with the LMA of a train directly or indirectly to cause more damage. There are three possible consequences. If the tampered LMA violates the communication protocol or is unreasonable in logic, it is perceived and discarded by the train. In this case, the DIA attack has the same impact on the train's operation as the DoS attack. If the tampered LMA is behind the real LMA and is used to calculate the ATP and ATO curves, the DIA attack may impair the efficiency of the train's operation. If the tempered LMA is in front of the real LMA and passes the inspection of the train, the DIA attack may lead to an accident. The train may crash into barriers after crossing the real LMA. As shown in Figure 2, if $L_m(t)$ is tampered into $L'_m(t)$, which is in front of the foregoing train. The following train runs under the ATP curve C3, it may collide with the foregoing train.

This paper presents an IDS to detect specific CBTC attacks including DoS and DIA. The main technical challenges and proposed solutions are summarized in Table 1.

Table 1. Technical challenges and proposed solutions.

Issues	Challenges	Solutions
DoS detection	Small changes in traffic flow are difficult to detect.	• Using the sequential analysis technique
DIA detection	Existing detection model can hardly identify DIA attacks.	• In-depth analysis of CBTC protocols. • Checking the authenticity of LMAs
Distinguishing attacks from faults	Faults may have the same impact as attacks do.	• Adding a detection model based on device states. • Designing an HMM classifier.

4. The Intrusion Detection Model Using a Fusion of Network and Device States

In this section, the framework of the proposed IDS is presented firstly. Then the detection models based on the network status and the device states are described in detail, respectively. Lastly, an HMM classifier is adopted to distinguish between random faults and cyber attacks.

As depicted in Figure 3, the intrusion detection process of the IDS is divided into two phases. One is the anomaly detection phase. The other is the result classification phase. The anomaly detection phase includes two models which are the network detection model and the device detection model. The network detection model analyzes the data throughput and the content of packets to detect the anomalies of networks. The device detection model detects abnormalities of devices based on the tasks and resource usage of hosts in CBTC systems. In the classification phase, the anomalies of network and devices are fused through the HMM model to distinguish between random faults and cyber-attacks.

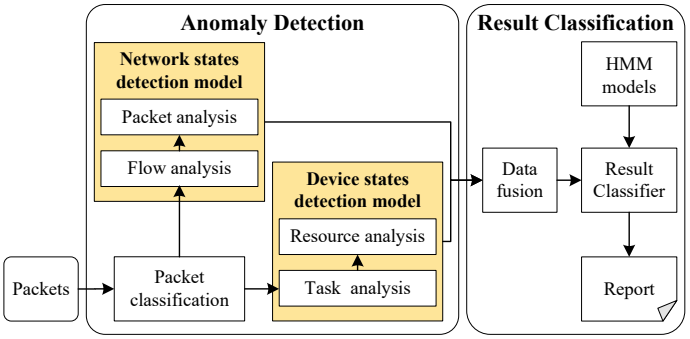


Figure 3. A novel intrusion detection system (IDS) for CBTC systems.

4.1. The Network Detection Model

As shown in Figure 4, the network detection model includes a throughput analysis module and a packet analysis module. Before being used to detect anomalies, the model is trained through historical packets to model the normal behavior of a CBTC system in throughput and transmitted packets. A DoS attack may hinder the normal operation of an IDS through a very high data throughput as lots of packets consume excessive resources of the IDS. To avoid the above situation, a threshold of throughput is predefined. The throughput analysis module delivers packets to the packet analysis module only if the throughput is below the threshold. Otherwise, the throughput analysis module outputs detection results directly.

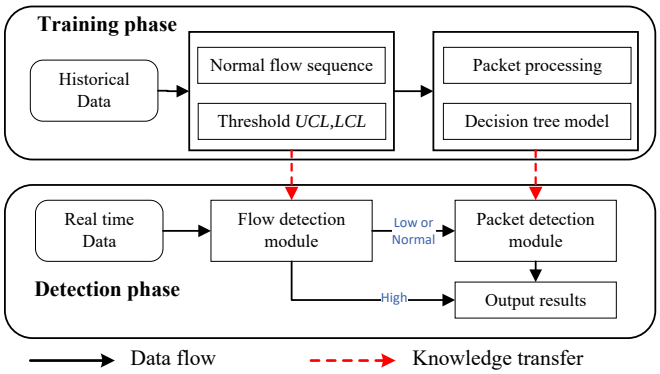


Figure 4. The anomaly detection model based on network states.

Due to the periodic communication between different equipment, the data throughput in a CBTC system is stable. A successful DoS attack on a CBTC system leads to abrupt changes in throughput. Consequently, a sudden change in the statistics parameters can be observed. The data throughput detection is equivalent to the problem of change point detection [31,32]. The exponentially weighted moving average (EWMA) control chart, a sequential analysis technique, is typically used for change point detection. EWMA is an efficient statistical method in detecting small shifts, which is superior to other control charts. EWMA can detect small changes more easily and quickly as it combines the current and historical data [33,34]. In this paper, EWMA is used to identify sudden changes in data throughput of a CBTC system.

Taking the data throughput between VOBC and ZC as an example, the predicted throughput of EWMA is calculated as [35]

$$z(i) = \lambda \cdot x(i) + (1 - \lambda) \cdot z(i - 1), 0 < \lambda \leq 1, \quad (1)$$

where $x(i)$ is the throughput at time i. λ is the smoothing factor indicating the sensitivity of $z(i)$ to the observed $x(i)$. The mean and variance of $z(i)$ can be expressed as

$$\mu_z = \mu_x, \tag{2}$$

$$\sigma_z^2 = \sigma_x^2 \cdot \frac{\lambda}{2-\lambda}, \tag{3}$$

where μ_x and σ_x are the mean and variance of $x(i)$, which can be estimated from historical data in the training phase. μ_z and σ_z are the mean and variance of $z(i)$, respectively. A change point is detected if $z(i)$ is outside the interval, $[D_z, U_z]$. It has

$$D_z = \mu_z - L \cdot \sigma_z, \tag{4}$$

$$U_z = \mu_z + L \cdot \sigma_z, \tag{5}$$

where L is a coefficient that effects the results.

However, due to the communication of CBTC systems is periodic, the observed data of throughput have strong autocorrelation. The traditional EWMA is not suitable for highly autocorrelated data [36]. To solve this problem, the error between $x(i)$ and $z(i-1)$ is defined and used to detect change point, which is defined as

$$e(i) = x(i) - z(i-1). \tag{6}$$

The variance of $e(i)$ can be rewritten as

$$\sigma_e^2 = \alpha \cdot e(i)^2 + (1-\alpha) \cdot \sigma_e^2(i-1), \quad 0 < \alpha \leq 1, \tag{7}$$

where α is a coefficient that affects the sensitivity of upper and lower limits of the interval to $e(i)$, which is identified in the fourth chapter. Accordingly, the lower and upper limits of $e(i)$ for change point detection are

$$D_e = -L \cdot \sigma_e, \tag{8}$$

$$U_e = L \cdot \sigma_e, \tag{9}$$

In the EWMA control chart, the values of L have an important influence on the performance of the detection. When L becomes larger, the threshold of the EWMA control chart will become higher, which may cause more attacks to be missed. However, if L is small, it may cause more false alarms. In the detection phase, an anomaly is discovered if $z(i)$ falls outside the D or U.

As the types of packets in CBTC are much less than those in a general network [37], a decision tree is adopted in the proposed IDS. A decision tree is one of the most popular and useful machine learning algorithms mainly used for classification. It uses a tree-like structure in which each internal node denotes a test on an attribute, each branch represents the output of the test, each leaf node corresponds to a class label. The merits of the decision tree include high classification accuracy and simple implementation. The best-known method to build a decision tree automatically is the ID3 algorithm [38]. Information gain, I, is defined to choose the attribute for each internal node to classify data.

The entropy of D and D_v can be calculated as follows

$$E(D) = -\sum_{k \in C} \frac{|D_k|}{|D|} log \frac{|D_k|}{|D|}, \tag{10}$$

$$E(D_v) = -\sum_{k \in C} \frac{|D_{vk}|}{|D_v|} log \frac{|D_{vk}|}{|D_v|}, \tag{11}$$

where D_k and D_{vk} are the subset of D and D_v, respectively. All the samples of D_k and D_{vk} belong to the kth category. $|D_{vk}|$ and $|D_v|$ are the number of samples in D_{vk} and D_v, respectively.

The information gain of using property a to classify D is defined as

$$I(D,a) = E(D) - \sum_{a_v \in A} \frac{|D_v|}{|D|} E(D_v). \tag{12}$$

The attribute with the highest information gain is chosen as the root node. Then I is computed on the other attributes to select a branch node until all the remaining samples belong to the same class. The root node, branch nodes, and leaf nodes make up a decision tree.

In the proposed IDS, the following attributes of the data are chosen for detecting anomalies in CBTC systems.

$$AN = \{sMAC, sIP, sPort, dMAC, dIP, dPort, Len, P, M\} \tag{13}$$

where the first six parameters represent the MAC, IP, port of the source and the MAC, IP, port of the destination, respectively. *Len* indicates the length of a packet. P represents the protocol type, including TCP, UDP, and ICMP. M is the position of the train.

As a CBTC system adopts the specified protocol, the packet length varies in a predictable range. It is found through analyzing a typical CBTC system that the normal value of *Len* is between 0 and 400 or in the range of 800 to 900.

A vicious cyber attacker may try to threaten the safety of a train through tampering with the LMA of the train to create a great sensation. The most predictable DIA method is adding an offset to the real LMA. To detect the anomaly caused by DIA, attribute M is adopted in the decision tree to check if the position change of the foregoing train conforms to the kinematic equation.

The position change of the foregoing train is equivalent to the difference between two consecutive LMAs, which should be less than

$$S_{max} = v_t T + \frac{1}{2} a_m T^2, \tag{14}$$

where S_{max} is the maximum position change of the foregoing train during one communication period of T. v_t is the train speed.

If the difference between two consecutive LMAs is larger than S_{max}, the attribute M is defined as abnormal.

4.2. The Detection Model Based on Device States

To satisfy the high requirement on reliability and safety, different redundant architectures are adopted in a CBTC system [39], such as hot-standby, two-out-of-three, and double 2-vote-2. Taking the hot-standby structure shown in Figure 5 which is adopted in ATS as an example, if the processing unit A is not working properly, the switching unit automatically switches to the processing unit B. The redundant device maintains the normal operation of the system even when the other unit is out of order due to random faults or cyber-attacks. It can be seen that the redundant architectures in CBTC which are adopted for safety can also protect against security risks to a certain extent. On the one hand, since the cyber-attack has not caused any communication abnormal behavior at this stage, the network detection model can not detect it yet. On the other hand, random faults and cyber-attacks may cause the same communication abnormal behaviors, the network detection model along can not identify the causes of the anomalies. To solve the above problem, the states of devices are analyzed to distinguish the anomalies caused by random faults from those introduced by cyber-attacks.

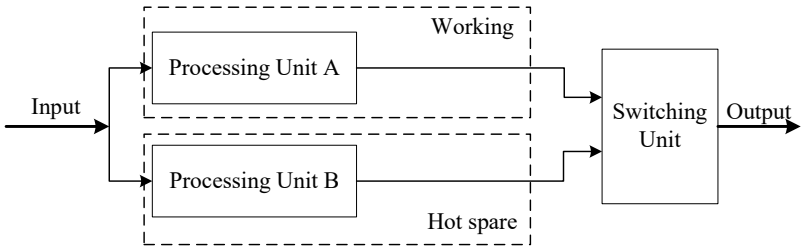


Figure 5. The hot-standby structure.

As the trains of a CBTC system run in fixed headways, the tasks and load of the key subsystems are stable. The hosts of the subsystems have stable resource utilization, including CPU, memory, disk, and network. Association rule mining is a rule-based machine learning method to discover the potential relations among variables in large databases [40,41]. In the device detection model, the associated rule mining method is used to check the resource usage of hosts. A set of all the items is defined as follows

$$I = \left\{ S_t, S_{ip}, R_c, R_m, R_d, R_n \right\}, \tag{15}$$

where S_t is the task running on the host, S_{ip} is the IP address of the host, R_c, R_m, R_d, and R_n are the usage of the CPU, memory, disk, and network of the host, respectively.

Based on the item set, the status of the concerned hosts are gathered to form the database of CBTC, which is a collection of transactions. Each transaction is a non-empty subset of I, such as

$$T_j = \{ZC1, 192.168.1.2, 11\%, 2\%, 0\%, 0.1\%\}, \quad T_j \subseteq I, \tag{16}$$

where $ZC1$ represents a task running on the host with an IP address of 192.168.1.2. Its occupancy of CPU, memory, disk, and network are 11%, 2%, 0%, 0.1%, respectively.

Then an association rule is defined as

$$X \to Y(s, c), \quad X, Y \subseteq I, X \cap Y = \emptyset, \tag{17}$$

where X is the antecedent, Y is the consequent, s is the support of the rule that indicates the percentage of the transactions that contain both X and Y. c is the confidence of the rule that represents the ratio of the number of transactions containing both X and Y compared to the number of transactions containing only X.

The Apriori algorithm is a classical association rule mining algorithm working in two steps. It finds frequent itemsets firstly and then generates association rules. However, Apriori can not handle continuous attributes such as CPU usage. For example, to discretize the CPU usage, the value range of R_{cpu} is divided equally into ten intervals. The R_{cpu} is discretized as follows:

$$\bar{R}_{cpu} = n, \quad R_l^n < R_{cpu} < R_u^n, \tag{18}$$

where $\bar{R}_{cpu}$ is the discretized R_{cpu}. R_l^n and R_u^n are the lower and upper limits of the nth interval, respectively.

In addition, Apriori may suffer from heavy computational load in mining association rules. Since different subsystem implements variable functions, they have unique rules of their own. The computational load is closely related to the size of the database [42]. Thus, each subsystem can manipulate its own database and mine association rules individually. Furthermore, as the running processes in the concerned hosts are determined in a CBTC system, transactions with illegal processes can be directly classified as anomalies. Assuming that s and c are 60% and 70%, respectively, some associated rules obtained through the Apriori algorithm are listed in Table 2.

Table 2. Some detection rules of the device detection model.

X		Y				s	c
S_t	S_{ip}	R_c	R_m	R_d	R_n		
ZC1	192.168.1.2	1	1	1	1	81	76
DSU	192.168.7.1	0	—	1	1	77	90
ATS	192.168.10.11	0	—	—	2	69	71
CI	—	1	0	0	0	71	74

4.3. The HMM Classifier Distinguishing Faults and Attacks

Due to the fail-safe mechanism of CBTC systems, both random faults and cyber-attacks may lead to anomalies. If anomalies are caused by failures, the broken devices should be repaired or replaced. However, if abnormalities are intrigued by cyber-attacks, not only the equipment should be restored, but also defensive measures should be taken to prevent similar incidents. Adopting an anomaly detection model based on network status or device states alone, the IDS cannot identify the causes of anomalies, not to mention advising administrators to take appropriate measures.

HMM is a statistical method to characterize observation samples arranged in discrete time series, which can predict the hidden states through observations [43]. As investigated by Hindy, HMM can meet the requirements of network detection, such as high detection rate, online learning ability, and high stability [44]. Besides that, HMM requires little time to train the detection models.

In the proposed IDS, the results of different detection models are observable, while the state of the system is invisible. An HMM classifier can fuse the results of different models to differentiate cyber-attacks from failures and improve the performance of detection effectively.

In an HMM, Q is a set of possible hidden states. V is a set of possible observations.

$$Q = \{q_1, q_2, ..., q_N\}, \tag{19}$$

$$V = \{v_1, v_2, ..., v_M\}, \tag{20}$$

where N is the number of hidden states and M is the number of observations.

In this paper, the hidden states are normal, fault, and attack, $N = 3$. The most typical structure of a CBTC system is hot-standby, where two redundant devices are connected to two physically independent networks, respectively. Based on the structure, the observation is designed as shown in Figure 6. An observation includes data obtained from the two networks, indicated by Network A and Network B. The data get from network A includes F_A, P_A, H_A, and T_A, which are the analysis results of throughput, packet content, running process, and host state, respectively. The possible values of F_A are "0," "1," and "2," indicate "normal," "low," and "high," respectively. The values of P_A, H_A, and T_A are "0" and "1," represent "normal" and "abnormal," respectively. The data get from network B are the same as those collected from network A. The M of the HMM classifier is 576.

An observation is given as an example

$$v_m = \{1, 0, 0, 0, 0, 0, 0, 0\}, \quad 1 \leq m \leq M,$$

where the throughput is lower than the predefined threshold, the other results are normal.

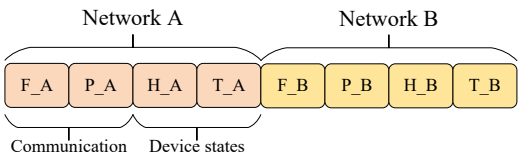


Figure 6. The format of the input data.

The following sequences are defined:

$$\begin{array}{ll} S = \{s_1, s_2, ..., s_T\}, & \forall t \in [1, T], \quad s_t \in Q, \\ O = \{o_1, o_2, ..., o_T\}, & \forall s \in [1, T], \quad o_s \in V, \end{array} \tag{21}$$

where I and Q are the sequence of hidden system states and the sequence of observations, respectively. i_t is the hidden system state at time t, o_s is the observation at time s.

Two assumptions are embodied in the HMM classifier. One is the Markov assumption on the probabilities of the sequence of system states. It is assumed that the probability of a system state depends only on the previous state.

The state transition probability is defined as

$$a_{ij} = P\left(s_{t+1} = j \mid s_t = i\right), \quad i, j \in Q, \tag{22}$$

where a_{ij} represents the probability of moving from state i to state j.

The transition probability matrix is composed of all the a_{ij}. It has

$$A = \left[a_{ij}\right], \quad A \subseteq R^{N \times N}. \tag{23}$$

Another assumption is the probability of an observation depends only on the hidden system state that produced the observation. The observation probability is defined as

$$b_j(k) = P\left(o_t = k \mid s_t = j\right), \quad j \in Q, k \in V. \tag{24}$$

Then the observation probability matrix is

$$B = [b_j(k)], \quad B \subseteq R^{N \times M}. \tag{25}$$

Besides A and B, an initial probability distribution of the system state is defines as

$$\Pi = [P\left(s_1 = q_i\right)], \quad q_i \in Q, \Pi \subseteq R^{1 \times N}. \tag{26}$$

The HMM classifier is specified by A, B, and Π, which is expressed as

$$\lambda = (A, B, \Pi). \tag{27}$$

The classification scheme of the proposed IDS is shown in Figure 7. In the offline phase, the historical data, a sequence of observations, is used to learn the parameters of the HMM classifier. The Baum–Welch (BW) algorithm, which is also known as a special case of the expectation-maximization algorithm [45], is adopted to train the A and B matrices.

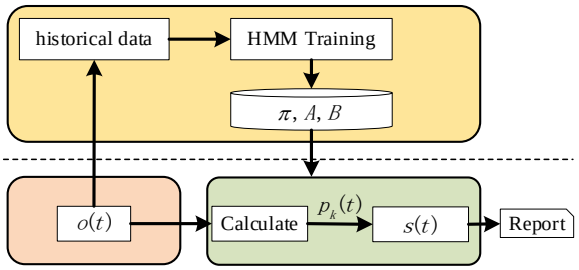


Figure 7. A hidden Markov model (HMM)-based anomaly classification model.

The forward probability which represents the probability of a state given the sequence of pre-observations is expressed as

$$\alpha_t(i) = P\left(o_1, o_2, ..., o_t, s_t = i \mid \lambda\right), \quad i \in Q, \tag{28}$$

where $\alpha_t(i)$ is the forward probability of state i at time t.

The forward probability is calculated as

$$\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} b_i(o_t), \quad i \in Q. \tag{29}$$

The backward probability which indicates the probability of a state given the sequence of post-observations is represented as

$$\beta_t(i) = P\left(o_{t+1}, o_{t+2}, ..., o_T \mid s_t = i, \lambda\right), \quad i \in Q, \tag{30}$$

where $\beta_t(i)$ is the backward probability of state i at time t.

The backward probability is computed as

$$\beta_t(i) = \sum_{j=q_1}^{q_N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad i \in Q. \tag{31}$$

The probability of a system state can be rewritten as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{j=q_i}^{q_N} \alpha_t(j)\beta_t(j)}, \tag{32}$$

where $\gamma_t(t)$ is the probability of system state i at time t.

Given the observation sequence and the HMM, the probability of being in state i at time t and state j at time $t+1$ is defined as $\xi_t(i,j)$. It has

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}{\sum\limits_{k=q_1}^{q_N} \sum\limits_{l=q_1}^{q_N} \alpha_t(k) a_{kl} b_l\left(o_{t+1}\right) \beta_{t+1}(l)}, \tag{33}$$

The state transition probability can be estimated as

$$\hat{a}_{ij} = \frac{\sum\limits_{t=1}^{T-1} \xi_t(i,j)}{\sum\limits_{t=1}^{T-1} \sum\limits_{k=q_1}^{q_N} \xi_t(i,k)}, \tag{34}$$

where $\hat{a}_{ij}$ is the estimation of a_{ij}.

The observation probability can be estimated as

$$\hat{b}_j(v_k) = \frac{\sum\limits_{t=1, o_t=v_k}^{T} \gamma_t(j)}{\sum\limits_{t=1}^{T} \gamma_t(j)} \tag{35}$$

To train the HMM classifier of the proposed IDS, a sequence of historical observations and the set of possible hidden system states are input to the BW algorithm. It is assumed that the initial

hidden system state is "normal," $P(q_1 = 0) = 1$. The A and B matrices are initialized randomly at the beginning of the iterations. The convergence conditions of the algorithm are set as follows:

$$\log_{10} P(O \mid \lambda) \leq 10^{-6},$$
$$\|A_n\| - \|A_{n-1}\| \leq 10^{-6},$$
$$\|B_n\| - \|B_{n-1}\| \leq 10^{-6},$$

where $\|A\|$ and $\|B\|$ are the 1-norm of the matrices A and B, respectively.

The algorithm converges after 152 iterations. The trained parameters of the HMM classifier are as follows:

$$A = \begin{bmatrix} 0.7365 & 0.0848 & 0.1787 \\ 0.9130 & 0.0290 & 0.0580 \\ 0.9342 & 0.0066 & 0.0592 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0.0290 & 0.1449 & \cdots & 0.2754 \\ 0.0987 & 0.0066 & \cdots & 0 \end{bmatrix}$$

Given the assumed Π, the trained matrices A and B, the HMM classifier is determined and used to classify the detection results of different models in the online phase. Through using (29), (31), and (32), the probability of the most possible hidden system state, given a sequence of observations, can be calculated as

$$i_t^* = arg \max_{i \in Q} [\gamma_t(i)], \quad t = 1, 2, \cdots, T. \tag{36}$$

Finally, the proposed IDS outputs i_t^* as the detection result.

5. Experimental Data Collection

In this section, an experimental environment is constructed to evaluate the proposed IDS. Therefore, the platform of Beijing Subway Line No. 7 is introduced, where attack scenarios are designed to collect experimental data.

5.1. Semi-Physical Simulation Platform of CBTC

The proposed method is verified on the hardware-in-the-loop simulation platform of CBTC systems. As shown in Figure 8, there are two networks, automatic train supervision (ATS), and automatic train control (ATC), which are connected by the gateway. In the platform, ZC and VOBC are real devices, while the other devices such as CI, DSU, and the gateway are simulated by software on different computers. Additionally, CBTC systems support degraded modes to ensure high availability. We can simulate different operation modes and collect all kinds of data on the platform.

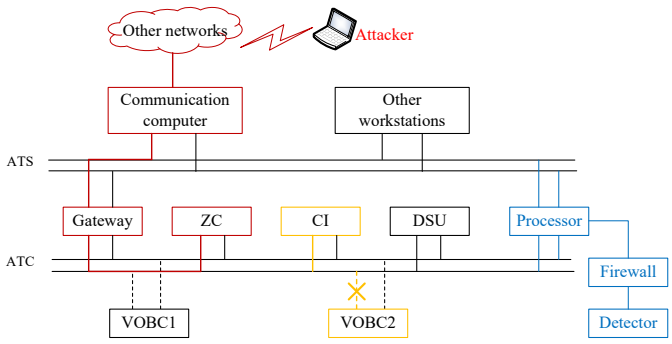


Figure 8. A typical attack path and major fault injection scenarios.

The operational modes can be classified as CBTC mode and intermittent ATP mode. In CBTC mode, information is transmitted continuously to realize automatic train protection. While in intermittent ATP mode, MA is only updated at discrete locations along the track. CBTC mode provides accurate closed-loop control of trains through continuous, bidirectional, and high-capacity communication between trains and wayside equipment. As the LMA is calculated based on the front train position, CBTC mode is a moving block signaling system. However, if continuous communication is interrupted, the system will convert to intermittent ATP mode, where the LMA will be transformed through beacons.

The proposed IDS is also implemented on this platform. As shown in Figure 8, the processor collects all packets and information from the CBTC platform. The detector is responsible for anomaly detection. As IDSs should not introduce new threats to CBTC, a firewall is set between the processor and the detector. Attacks can not be carried out through intrusion detection devices.

5.2. Experimental Scenarios

As shown in Figure 8, a typical attack path is highlighted in red and the major injection locations are highlighted in yellow. CBTC is connected with other systems, such as passenger information system (PIS), through the communication computer in the ATS network. As the security protection of other systems may be weak, attackers may first capture the communication computer through other networks, and then launch an attack on the ATS. To connect to the ATC network, they need to attack the gateway next. Finally, they will directly attack equipment such as ZC and seriously affect the train operation.

There is a wide variety of attacks in IT systems. However, most of the common attacks can not be achieved in CBTC because CBTC systems are not connected to the Internet directly. In this paper, we only select attack scenarios that may occur in CBTC systems, including DoS and DIA. Among them, DoS includes vulnerability triggering and resource exhaustion. As COTS products are widely used in CBTC systems, various buffer overflow vulnerabilities also exist. The exhaustion of resources is mainly caused by flood attacks, such as Smurf, synchronize sequence numbers (SYN) flood and so on. Additionally, LMAs directly determine where the trains can travel to. Therefore, DIA scenarios in the experiments are launched against the LMAs.

As our IDS can distinguish between faults and attacks, fault injection scenarios are also designed. Equipment faults and communication failure are selected as they mostly occur in CBTC systems.

When implementing experiments, the difference between faults and attacks is also considered. The goal of an attacker is usually to affect the operation of the train. As a result, the target choice and duration of attacks are purposeful while the faults are random.

The way of attack emulation and fault injection are introduced in detail next.

- Buffer overflow

A buffer overflow occurs during program execution when a fixed-size buffer has had too much data copied into it [46]. Buffer overflow attacks can take place in the process of using a stack during program execution. It can overwrite data into adjacent memory locations and affect the behavior of the software. Since most applications in CBTC are developed in C, a publicly-available suite is applied in this paper to identify buffer overflows and help launch attacks.

- Smurf

Smurf is a type of DoS attack which floods a victim network via spoofed broadcast ping messages [47]. Currently, Windows operating systems have adopted strategies to avoid this attack. However, this vulnerability may still exist in other operating systems, such as VxWorks in ZC. We simulate attackers sending ICMP echo request packets to the broadcast address and forging the source address to be the IP address of ZC. Then significant traffic will be generated on the ZC subsystem, which will cause ZC to be down.

- SYN flood

When the SYN flood attack occurs, all open ports may be saturated with requests and none are available for legitimate users to connect to. In this paper, the targets of SYN flood are communication computer, gateway, ZC, CI, and DSU respectively.

- Tamper attack

The tamper attack mainly affects the location information or the LMA transformed between the trains and ZC. We design attack nodes that can modify the train position or LMA before the packets are sent to the final nodes in the experiments. Therefore, the targets of tamper attacks are ZC or VOBC.

- Replay attack

The replay attack is also most likely to occur in the communication between ZC and VOBC, where the impact of the attack is greatest. In this paper, the attacker is simulated to eavesdrop and repeatedly send LMAs to disturb the normal operation of the trains.

- Equipment faults

In this paper, the target of equipment fault injection is chosen randomly. As shown in Figure 8, we take CI as an example to show how faults are injected. In the first case, an application error is simulated by shutting down related tasks running on CI. In the second case, unexpected device faults are considered and simulated by shutting down the CI host.

- Communication faults

Wire communications are used between wayside devices, while VOBC communicates with wayside devices wirelessly. Therefore, the probability of faults between VOBC and the ATC network is much greater. As shown in Figure 8, communication faults are injected between VOBC and the wayside equipment, including simulating packet loss and increasing transmission delay.

5.3. Experimental Data

The collected data includes traffic flow, packets, process lists and resource utilization of each process. The normal operation of CBTC systems is simulated on the platform, and the data is marked as "normal". The data collected during the attack is labeled as "attack". Similarly, the data is marked as "fault" if it is obtained during the fault injection.

True positives rate (TPR) and false positives rate (FPR) are selected to measure the performance of the IDS. TPR is the proportion of anomalous instances classified as correct ones over the total number of anomalous instances, while FPR is the proportion of normal instances classified as anomalous ones over the total number of normal instances. In this paper, we define the number of non-attack records which are detected true as non-attack $\rightarrow$ non-attack (TN). Similarly, we get non-attack $\rightarrow$ attack (FP), attack $\rightarrow$ attack (TP), and attack $\rightarrow$ non-attack (FN).

$$TPR = \frac{TP}{TP + FN} \tag{37}$$

$$FPR = \frac{FP}{FP + TN} \tag{38}$$

To get the appropriate data size, we verify the impact of dataset size on detection performance. Figure 9 shows the performance of the IDS with 20, 40, 60, 80, 100, 120, 140, and 160 experiment times, respectively. From 0 to 80 times, detection performance improves as the dataset gets larger. Compared with 80 times, the performance of 100 times improves less. In addition, TPR and FPR are

basically stable after 100 times. Therefore, to save computing resources and reduce model training time, the dataset contains 100 experiments in this paper.

Finally, all of the data is summarized as shown in Table 3, including 560,977 packets and 31,649 state messages. The whole data set will be divided into training set and test set. The former is used for model training and the latter for performance evaluation. If the training set is too small, the training model is not accurate enough. On the contrary, if the test set is too small, the performance evaluation is not comprehensive. In general, 80% of the data is randomly selected as a training set and the remaining data is a test set.

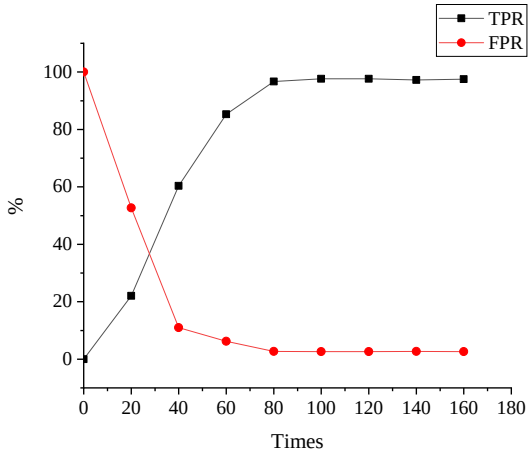


Figure 9. Detection performance of different numbers of experiments.

Table 3. Quantity of the experimental data.

Type	Category	Times	Packets	State
Normal	CBTC mode	30	128,742	13,721
	Intermittent ATP mode	5	14,263	2188
Attack (DoS)	Buffer overflow	10	13,527	3418
	Smurf	10	71,572	4113
	SYN flood	5	154,759	1478
Attack (DIA)	Tamper	10	38,531	1726
	Replay	10	82,861	1266
Fault	Equipment	10	46,447	1981
	Communication	10	10,275	1758

6. Results and Discussion

6.1. Parameter Settings

It should be noted that when the results are calculated, both "normal" and "fault" instances are used as "non-attack" ones. To implement the proposed IDS, the parameters of each model need to be determined.

Due to the EWMA control chart used in the flow statistics, L, λ, and α may all have impacts on the detection performance. According to [48], L is assigned a value of 1.96. When λ and α take different values, the changes of TPR and FPR are shown in Table 4. When $\lambda = 0.01$ and $\alpha = 0.001$, EWMA has the best performance.

Table 4. Detection results under different combinations of parameter values.

λ	α	TPR (%)	FPR (%)
0.1	0.01	76.65	8.19
	0.001	84.55	14.19
	0.0001	81.81	8.12
0.05	0.01	82.41	7.21
	0.001	94.49	7.03
	0.0001	93.39	5.74
0.01	0.01	90.39	6.27
	0.001	94.86	3.21
	0.0001	93.91	3.82

In the detection model based on device states, the support s and the confidence c of association rule mining also have a great impact on the performance. The larger these two parameters are, the fewer frequent itemsets are mined. Thus, fewer association rules are generated and the FPR may be higher. However, if s and c are too small, lots of redundant rules may be generated. Mining these rules will consume a large number of computing resources. Therefore, we set s equal to 0.6 and c equal to 0.8, just as the experts suggest.

6.2. Experiment Results

In this section, we compare the detection performance of the proposed IDS with other approaches. As multiple methods are applied from multiple perspectives in the IDS, it is difficult to compare with other intrusion detection approaches directly. Therefore, we firstly compare the performance of a single detection model with the entire IDS in Experiment 1. Then the network states anomaly detection (NAD) model, the device states anomaly detection (DAD) model and the HMM classifier are compared with other approaches, respectively. As shown in Figure 10, the test dataset is used to generate the following results.

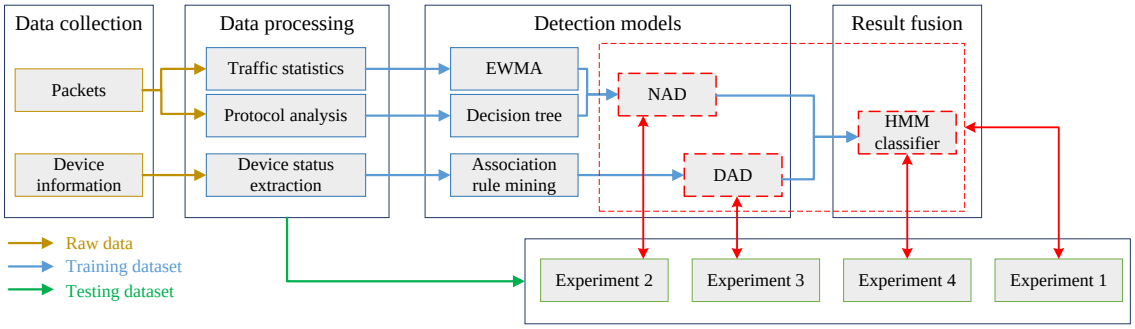


Figure 10. Overall flow of the processing steps for the data and results.

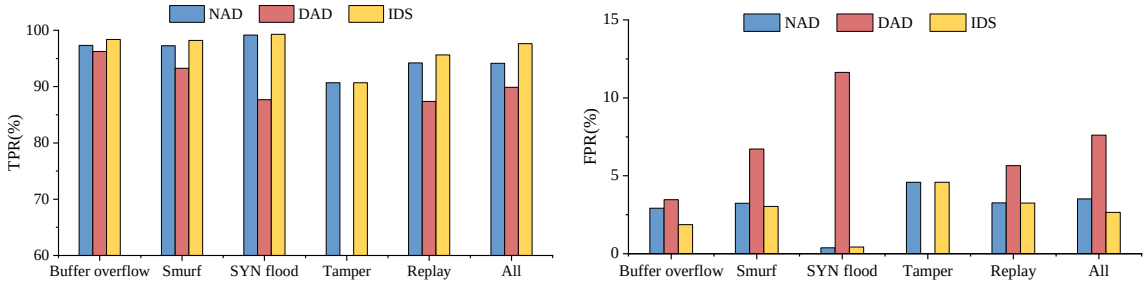
- Experiment 1

To prove the proposed IDS has better performance than the NAD model or the DAD model alone, we calculate their detection results as shown in Table 5. The data in Table 5 is plotted as Figure 11 to display the results more intuitively. As the NAD or DAD model can only obtain information from one aspect, the TPR of a single model is lower than the entire IDS. Generally speaking, the HMM classifier can process different information on NAD and DAD to obtain a lower FPR. However, in the case of SYN flood attacks, FPR reaches 11.64% in DAD. Due to many false alarms generated in DAD, the HMM classifier may also generate more false positives. As a result, the FPR of the IDS is 0.05% higher than NAD. For the entire dataset, the TPR is increased by 3.51% and 7.76% after applying the

fusion of the detection models. At the meantime, the *FPR* is reduced by 0.86% and 4.95%. In summary, the proposed method has better performance than a single detection model.

Table 5. Performance of different models.

Anomaly	TPR %			FPR %		
	NAD	DAD	IDS	NAD	DAD	IDS
Buffer overflow	97.31	96.24	98.37	2.92	3.47	1.87
Smurf	97.25	93.27	98.20	3.24	6.72	3.04
SYN flood	99.16	87.68	99.29	0.38	11.64	0.43
Tamper	90.67	–	90.67	4.58	–	4.58
Replay	94.22	87.35	95.64	3.27	5.65	3.25
All	94.13	89.88	97.64	3.52	7.61	2.66



(**a**) True Positives Rate. (**b**) False Positives Rate.

Figure 11. Detection results of different models.

- Experiment 2

Table 6 gives some different methods, application scenarios, and results of other published IDS. Yang and Liu adopt a statistical-based method to analyze the network traffic [49,50]. Akbar applies a supervised classifier to detect attacks in Voice over Internet Protocol (VoIP) networks [51]. Using a single detection method, their TPR is lower than that of the combined methods. Some works combine statistical methods with machine learning algorithms to detect attacks in different systems [52–56]. Among them, Verba does not illustrate the detailed detection performance [52]. According to TPR, Valdes and Amini are effective in detecting DoS attacks [53,54]. However, they do not take the DIA scenarios into account. Only Goh gives the detection results of DIA, while it does not analyze the more common DoS attacks [56]. The NAD model adopts both the statistical method and decision tree. It can detect DoS and DIA at the same time. What's more, the NAD model is designed according to the characteristics of CBTC systems. It has good detection performance, where TPR is 98.86% for DoS and 92.95% for DIA.

Table 6. Comparison of the network states anomaly detection (NAD) model with other methods.

IDS	Methods	Application Scenarios	DoS		DIA	
			TPR (%)	FPR (%)	TPR (%)	FPR (%)
[49]	Cumulative sum	Wireless network	93.27	4.32	–	–
[50]	Statistical change-point detection	Real-world network	90.61	5.64	–	–
[51]	Supervised classifiers	VoIP network	82.17	0.05	–	–
[52]	Traffic flow analysis and packet detection	SCADA	–	–	–	–
[53]	Flow-based and pattern-based	Process Control System	99.9	–	–	–
[54]	Statistical preprocessing and Neural Network	Local networks	97.42	1.99	–	–
[55]	Statistical feature vectors and Neural Network	KDD99	94	0.2	–	–
[56]	Recurrent Neural network and Cumulative Sum	Water treatment plant	–	–	90	4
NAD	The statistical method and Decision Tree	CBTC	98.86	1.92	92.95	3.84

- Experiment 3

As mentioned before, TPR and FPR are selected to measure the performance of the IDS, where TPR represents the detection rate. When we compare the performance of the DAD model with other methods, the receiver operating characteristic (ROC) curve is applied in this paper. An ROC curve can evaluate the tradeoff between TPR and FPR. By carrying out several tests using different s and c of the association rule, the ROC curve can be plotted as shown in Figure 12. Since the states of the CBTC subsystems are generally stable, the association rule mining algorithm has good detection performance to detect the abnormal device states. As shown in Figure 12, the DAD model has a higher detection rate under the same FPR compared with the length decreasing support (LDS) Apriori [57] and the hybrid IDS [58]. When the FPR is higher than 15%, the detection rate of the Fuzzy IDS is higher than that of the DAD model [59]. However, the FPR of the Fuzzy IDS is too high for CBTC systems. A large number of false alarms may be generated in the case of a high FPR. Then the trains may perform emergency braking and the efficiency of CBTC systems will be reduced. When FPR is 7.61%, the detection rate of the DAD model is 89.88%, which is higher than LDS by 13.34%, fuzzy IDS by 61.88%, and mixed IDS by 38.91%.

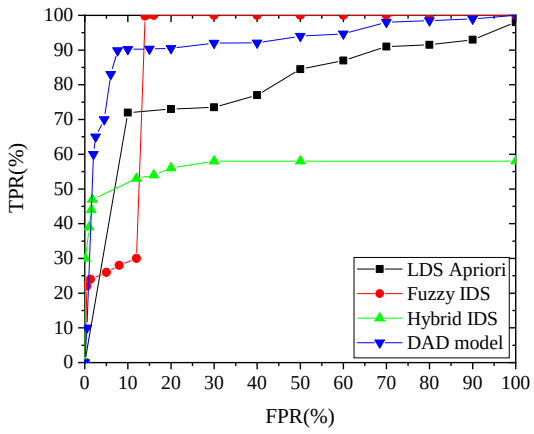


Figure 12. Receiver operating characteristic (ROC) curves of different detection systems.

- Experiment 4

Finally, several commonly used classification algorithms in the field of IDS are selected to compare with the HMM classifier, including naive Bayes, neural networks (NN), and support vector machines (SVM). They are applied to classify the same data with the HMM classifier. The detection results are shown in Figure 13. As naive Bayes, NN and SVM may misjudge faults as attacks, they have higher *FPR* than that of the proposed IDS. As shown in Figure 13, the *FPR* of naive Bayes, NN, and SVM are 19.21%, 4.26%, and 9.77%, which significantly dropped to 2.66% using the HMM classifier. It proves that the HMM classifier can distinguish between faults and attacks effectively.

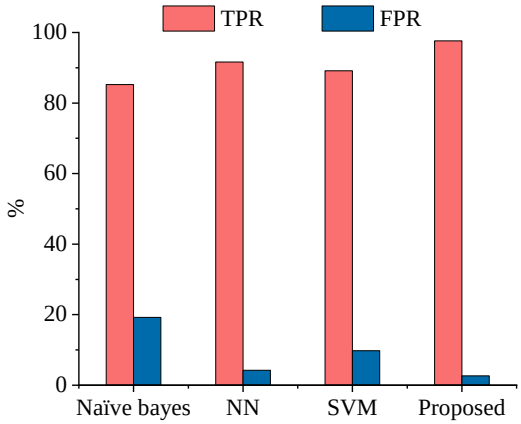
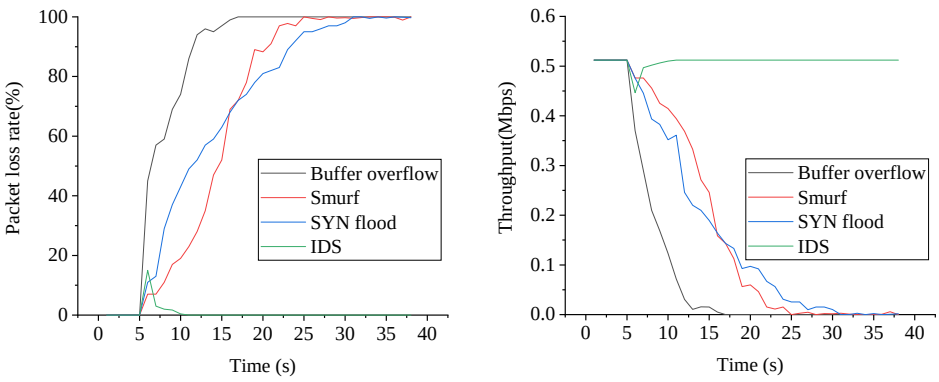


Figure 13. Performance comparison of various classifiers with the proposed approach.

The proposed IDS can also improve the packet loss rate and throughput performance of the system after an attack. Taking the communication between VOBC and the wayside equipment as an example, we repeat experiments with different attacks, whose results are shown in Figure 14. DIA attacks only tamper with the information and do not change the number of packets. These attacks have almost no impact on the packet loss rate or throughput. Therefore, only DoS attacks are simulated. We attack the system at the 5th second. Among the curves, the blue one indicates the case of joining the IDS. The IDS can detect attacks and notify administrators to take defensive measures. The results show that attacks have a great impact on the communication performance of CBTC. The proposed IDS can not only detect attacks but also promptly give alarms. Thus, the system can recover quickly after the communication is abnormal. In summary, the proposed IDS can effectively prevent attacks from causing more serious impacts on CBTC.



(**a**) Packet loss rate. (**b**) Throughput.

Figure 14. Different impacts on system performance of attacks and IDSs.

7. Conclusions

In this paper, a novel intrusion detection method for CBTC based on network and device states is designed. The impact of cyber-attacks on CBTC is analyzed and different detection models are proposed according to the principles of CBTC systems. An HMM classifier is adopted to differentiate cyber-attacks from random system faults. Through limited experimentation, we concluded that the proposed IDS could effectively detect attacks in CBTC systems, where the TPR approached 97.64% while bounding the FPR to below 2.66%.

Future improvements integrated into the proposed IDS would have the ability to use multiple data sources such as fault identification and running status of the train. Additionally, we noticed that the detection rate of data tampering attacks was lower than the other attacks during experiments. More detection patterns are needed to improve the performance of the IDS in the future.

Author Contributions: Conceptualization, Y.S. and B.B.; methodology, Y.S.; validation, Y.S., B.B., and L.Z.; resources, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Innovation Fund Project of Beijing Traffic Control Technology under Grant 9907006507, Project under Grant I19L00090, Beijing Laboratory of Urban Rail Transit, NSFC Project 61973026.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Farooq, J.; Soler, J. Radio communication for communications-based train control (CBTC): A tutorial and survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1377–1402. [CrossRef]
2. Won, Y.; Yu, B.; Park, J.; Park, I.H.; Jeong, H.; Baik, J.; Kang, K.; Lee, I.; Son, S.H.; Park, K.J.; et al. An Attack-Resilient CPS Architecture for Hierarchical Control: A Case Study on Train Control Systems. *Computer* **2018**, *51*, 46–55. [CrossRef]
3. Soderi, S.; Hämäläinen, M.; Iinatti, J. *Cybersecurity considerations for Communication Based Train Control*; Alstom Signalling Solutions: Florence, Italy, 2016.
4. Boo, H.W. An Assessment of North Korean Cyber Threats. *J. East Asian Aff.* **2017**, *31*, 97–117.
5. Chopra, S.S.; Dillon, T.; Bilec, M.M.; Khanna, V. A network-based framework for assessing infrastructure resilience: A case study of the London metro system. *J. R. Soc. Interface* **2016**, *13*, 20160113. [CrossRef] [PubMed]
6. Kour, R.; Aljumaili, M.; Karim, R.; Tretten, P. eMaintenance in railways: Issues and challenges in cybersecurity. *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit* **2019**, *233*. [CrossRef]
7. Kabir, M.F.; Hartmann, S. Cyber security challenges: An efficient intrusion detection system design. In Proceedings of the 2018 International Young Engineers Forum (YEF-ECE), Costa da Caparica, Portugal, 4 May 2018; pp. 19–24.
8. Basile, C.; Gupta, M.; Kalbarczyk, Z.; Iyer, R.K. An approach for detecting and distinguishing errors versus attacks in sensor networks. In Proceedings of the International Conference on Dependable Systems and Networks (DSN'06), Philadelphia, PA, USA, 25–28 June 2006; pp. 473–484.
9. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
10. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
11. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
12. Han, S.; Xie, M.; Chen, H.H.; Ling, Y. Intrusion detection in cyber-physical systems: Techniques and challenges. *IEEE Syst. J.* **2014**, *8*, 1052–1062.
13. Moustafa, N.; Slay, J. A hybrid feature selection for network intrusion detection systems: Central points. *arXiv* **2017**, arXiv:1707.05505.

14. Yang, Y.; McLaughlin, K.; Littler, T.; Sezer, S.; Wang, H. Rule-based intrusion detection system for SCADA networks. In Proceedings of the 2nd IET Renewable Power Generation Conference (RPG 2013), Beijing, China, 9–11 September 2013.
15. Pal, K.; Adepu, S.; Goh, J. Effectiveness of association rules mining for invariants generation in cyber-physical systems. In Proceedings of the 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, 12–14 January 2017; pp. 124–127.
16. Li, W.; Meng, W.; Su, C.; Kwok, L.F. Towards false alarm reduction using fuzzy if-Then rules for medical cyber physical systems. *IEEE Access* **2018**, *6*, 6530–6539. [CrossRef]
17. Mitchell, R.; Chen, R. Effect of intrusion detection and response on reliability of cyber physical systems. *IEEE Trans. Reliab.* **2013**, *62*, 199–210. [CrossRef]
18. Kuznetsov, A.; Smirnov, A.; Danilenko, D.; Berezovsky, A. The statistical analysis of a network traffic for the intrusion detection and prevention systems. *Telecommun. Radio Eng.* **2015**, *74*, 61–78. [CrossRef]
19. Shitharth, S. An enhanced optimization based algorithm for intrusion detection in SCADA network. *Comput. Secur.* **2017**, *70*, 16–26.
20. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network anomaly detection: Methods, systems and tools. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 303–336. [CrossRef]
21. Jyothi, V.; Wang, X.; Addepalli, S.K.; Karri, R. Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect ddos attacks. In Proceedings of the 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), Kolkata, India, 4–8 January 2016; pp. 587–588.
22. Sindhu, S.S.S.; Geetha, S.; Kannan, A. Decision tree based light weight intrusion detection using a wrapper approach. *Expert Syst. Appl.* **2012**, *39*, 129–141. [CrossRef]
23. Resende, P.A.A.; Drummond, A.C. A survey of random forest based methods for intrusion detection systems. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 48. [CrossRef]
24. Jahromi, A.N.; Sakhnini, J.; Karimpour, H.; Dehghantanha, A. A deep unsupervised representation learning approach for effective cyber-physical attack detection and identification on highly imbalanced data. In Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, Markham, ON, Canada, 4–6 November 2019; IBM Corp.: Riverton, NJ, USA, 2019; pp. 14–23.
25. Melaragno, A.; Bandara, K.R.D.S.; Fewell, A.; Wijesekera, D. Rail Radio Intrusion Detection System (RRIDS) for Communication Based Train Control (CBTC). In Proceedings of the 2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT), Birmingham, UK, 23–25 August 2016.
26. Zhang, W.; Bu, B.; Wang, H. An Intrusion Detection Method of Data Tampering Attack in Communication-Based Train Control System. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 345–350.
27. Gao, B.; Bu, B. A Novel Intrusion Detection Method in Train-Ground Communication System. *IEEE Access* **2019**, *7*, 178726–178743. [CrossRef]
28. Zhu, L.; Yu, F.R.; Ning, B.; Tang, T. Cross-layer handoff design in MIMO-enabled WLANs for communication-based train control (CBTC) systems. *IEEE J. Sel. Areas Commun.* **2012**, *30*, 719–728. [CrossRef]
29. Bu, B.; Yu, F.R.; Tang, T. Performance improved methods for communication-based train control systems with random packet drops. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1179–1192. [CrossRef]
30. Sedjelmaci, H.; Guenab, F.; Boudguiga, A.; Petiot, Y. Cooperative Security Framework for CBTC Network. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
31. Tartakovsky, A.G.; Polunchenko, A.S.; Sokolov, G. Efficient computer network anomaly detection by changepoint detection methods. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 4–11. [CrossRef]
32. Machaka, P.; Bagula, A.; Nelwamondo, F. Using exponentially weighted moving average algorithm to defend against DDoS attacks. In Proceedings of the 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), Stellenbosch, South Africa, 30 November–2 December 2016; pp. 1–6.
33. Kwon, Y.; Kim, H.K.; Lim, Y.H.; Lim, J.I. A behavior-based intrusion detection technique for smart grid infrastructure. In Proceedings of the 2015 IEEE Eindhoven PowerTech, Eindhoven, The Netherlands, 29 June–2 July 2015; pp. 1–6.

34. Silva, A.; Pontes, E.; Zhou, F.; Guelf, A.; Kofuji, S. PRBS/EWMA based model for predicting burst attacks (Brute Froce, DoS) in computer networks. In Proceedings of the Ninth International Conference on Digital Information Management (ICDIM 2014), Phitsanulok, Thailand, 29 September–1 October 2014; pp. 194–200.
35. Tailor, K.S. Some Special Types of Statistical Quality Control Charts for Variables under the Assumption of Moderateness. Ph.D. Thesis, SPB English Medium College of Commerce, Surat, India, 2015.
36. Umer, M.F.; Sher, M.; Bi, Y. Flow-based intrusion detection: Techniques and challenges. *Comput. Secur.* **2017**, *70*, 238–254. [CrossRef]
37. Agrawal, S.; Agrawal, J. Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.* **2015**, *60*, 708–713. [CrossRef]
38. Salzberg, S.L. *C4. 5: Programs for Machine Learning by j. ross Quinlan*; Morgan Kaufmann Publishers, Inc.: San Mateo, CA, USA, 1993.
39. Zhao, W.T.; Cao, G.N.; Chen, X.X.; Ling, Z.J.; Nie, Z.P.; Tang, W.B.; Teng, G.D.; Hu, L.Q. Design of the reliable collection and control scheme of the relay node status of the train carborne control system. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2016; Volume 7, p. 04016.
40. Tang, L.A.; Han, J.; Jiang, G. Mining sensor data in cyber-physical systems. *Tsinghua Sci. Technol.* **2014**, *19*, 225–234. [CrossRef]
41. Zhang, L.; Chen, Y.; Liao, S. Study on Intrusion Detection Based on Data Mining. In Proceedings of the 2018 International Conference on Engineering Simulation and Intelligent Control (ESAIC), Changsha, China, 10–11 August 2018; pp. 323–325.
42. Hidayanto, B.C.; Muhammad, R.F.; Kusumawardani, R.P.; Syafaat, A. Network intrusion detection systems analysis using frequent item set mining algorithm FP-max and apriori. *Procedia Comput. Sci.* **2017**, *124*, 751–758. [CrossRef]
43. Hurley, T.; Perdomo, J.E.; Perez-Pons, A. HMM-Based Intrusion Detection System for Software Defined Networking. In Proceedings of the IEEE International Conference on Machine Learning & Applications, Anaheim, CA, USA, 18–20 December 2016.
44. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.; Bellekens, X. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. *arXiv* **2018**, arXiv:1806.03517.
45. Li, Z.; Fang, H.; Huang, M. Diversified learning for continuous hidden Markov models with application to fault diagnosis. *Expert Syst. Appl.* **2015**, *42*, 9165–9173. [CrossRef]
46. Alouneh, S.; Bsoul, H.; Kharbutli, M. A software tool to protect executable files from buffer overflow attacks. *Int. J. Internet Technol. Secur. Trans.* **2016**, *6*, 133–166. [CrossRef]
47. Gunnam, G.R.; Kumar, S. Do ICMP Security Attacks Have Same Impact on Servers? *J. Inf. Secur.* **2017**, *8*, 274–283. [CrossRef]
48. Giraldo, J.; Urbina, D.; Cardenas, A.; Valente, J.; Faisal, M.; Ruths, J.; Tippenhauer, N.O.; Sandberg, H.; Candell, R. A survey of physics-based attack detection in cyber-physical systems. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 76. [CrossRef]
49. Yang, B.; Wang, X. Research on multi-class CUSUM algorithm for anomaly detection of WSN. In Proceedings of the International Conference on Intelligent Computation Technology & Automation, Changsha, China, 11–12 May 2010.
50. Liu, S.; Yamada, M.; Collier, N.; Sugiyama, M. Change-point detection in time-series data by relative density-ratio estimation. *Neural Netw.* **2013**, *43*, 72–83. [CrossRef] [PubMed]
51. Akbar, M.A.; Farooq, M. Application of evolutionary algorithms in detection of SIP based flooding attacks. In Proceedings of the Conference on Genetic & Evolutionary Computation, Montreal, QC, Canada, 8–12 July 2009.
52. Verba, J.; Milvich, M. Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS). In Proceedings of the IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, 12–13 May 2008.
53. Valdes, A.; Cheung, S. Communication Pattern Anomaly Detection in Process Control Systems. In Proceedings of the IEEE Conference on Technologies for Homeland Security, Boston, MA, USA, 11–12 May 2009.
54. Amini, M.; Jalili, R.; Shahriari, H.R. RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks. *Comput. Secur.* **2006**, *25*, 459–468. [CrossRef]

55. Sui, S.; Li, L.; Manikopoulo, C.N. Flow-based Statistical Aggregation Schemes for Network Anomaly Detection. In Proceedings of the IEEE International Conference on Networking, Ft. Lauderdale, FL, USA, 23–25 April 2006.
56. Goh, J.; Adepu, S.; Tan, M.; Zi, S.L. Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In Proceedings of the IEEE International Symposium on High Assurance Systems Engineering, Singapore, 12–14 January 2017.
57. Li, L.; Yang, D.Z.; Shen, F.C. A novel rule-based Intrusion Detection System using data mining. In Proceedings of the IEEE International Conference on Computer Science & Information Technology, Chengdu, China, 9–11 July 2011.
58. Kai, H.; Cai, M.; Chen, Y.; Qin, M. Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes. *IEEE Trans. Dependable Secur. Comput.* **2007**, *4*, 41–55.
59. Tajbakhsh, A.; Rahmati, M.; Mirzaei, A. Intrusion detection using fuzzy association rules. *Appl. Soft Comput.* **2009**, *9*, 462–469. [CrossRef]