International Journal of Intelligent Transportation Systems Research
https://doi.org/10.1007/s13177-021-00274-1

Improve Safety and Security of Intelligent Railway Transportation System Based on Balise Using Machine Learning Algorithm and Fuzzy System

Abolfazl Falahati[1] · **Ebrahim Shafiee**[1]

Received: 23 December 2020 / Revised: 25 July 2021 / Accepted: 31 August 2021

Abstract
With the advancement of modern rail transport systems, high-speed railways' safety and reliability is improved enormously due to proper intelligent traffic management systems. The automatic train control and operating system receive the train location beacons and the railway line's essential information through various channels, such as Balise wirelessly. However, this technology is vulnerable to cyber-physical attacks. This article aims to investigate the existing cyber attacks on Balise that can result a physical turmoil. Due to the limitations and constraints of the railway infrastructures, the attacks and failure detection methods are proposed based on machine learning. Also, a fuzzy countermeasure system is developed to improve train safety against known and unknown cyber-attacks. The simulation results show 92% accuracy in the proposed successful attacks detection system. Moreover, a small amount of false-positive and false-negative warnings can be also revealed employing the proposed scheme. The proposed method does not require change railway infrastructure.

Keywords Train transport security · Machin learning · Cyber-attacks · European train control system (ETCS) · Adaptive neural-fuzzy inference system (ANFIS) · Support vector machine (SVM)

1 Introduction

The development of intelligent rail transportation systems has increased the speed and number of modern trains over the rail tracks, so that today, over 4000 billion passenger-kilometers travel across the world during the year 2020 [1]. Consequently, it is vital to ensure that passengers safety and security. Communication-based train control (CBTC), European rail traffic management systems (ERTMS), or other similar systems worldwide have a crucial role in safety, management, signaling, and train control systems. One of the mentioned systems is the European train control system (ETCS) which is responsible for signaling, Automatic train control (ATC), and Automatic train protection (ATP) [2]. The primary objectives for designing the aforementioned systems are standardization and passenger safety, but security is not taken into account. It should be emphasized that the security vulnerabilities could compromise the safety of passengers, which is the main issue of this paper.

According to the ETCS standard, on-board train systems include Vital computer (VC), Balise transmission module (BTM), odometric sensors, Doppler radar, and lineside equipment which includes Balise and Lineside electronic unit (LEU) [3]. The Balise is a beacon transponder installed between the rails. When the train passes over, it communicates with the train BTM via an air-gap interface. BTM is an on-board module with an antenna installed under the train to send and receive a message (telegram) from Balise. The pre-programmed telegram contains Balise information, such as the header ID, train position and geographical location, speed limit, route, movement authority, and linking data (distance to the next Balise or its group). Balise has a mechanism similar to Radio frequency identification (RFID), including two types of fixed and controllable Balise. Fixed Balise is activated by BTM tele-powering that continually responds to the BTM the copies of telegram until the train passes over. Controllable Balise is energized by the LEU and can transmit (up-link) and receive (down-link) dynamic data [4].

✉ Abolfazl Falahati
afalahati@iust.ac.ir

[1] Present Address: Department of Electrical Engineering (DCCS Lab), Iran University of Science and Technology, University St, 1311416846 Tehran, Iran

Published online: 30 September 2021

The ETCS system is designed to increase the safety, speed, and automaticity of train transportation; however, security against cyber-attacks has not been considered [2]. The Balise telegram is sent as plaintext without checking the integrity and timestamps, increasing the potential for possible attacks. Moreover, an attacker can be a dissatisfied team member, or a malicious contractor who can collect, tamper, relay, replay, and block telegram data. As a result, ambiguity in positioning and compromising train safety can lead to passenger life threats, train derailment, or catastrophic collisions [5, 6]. According to the mentioned issues, this study aims to improve the security of the train control system based on artificial intelligence systems.

Until now, many researchers have proposed a scheme to improve the security of the train control system. These schemes can be categorized in a cryptographic method, a challenge-response authentication mechanism, and a localization approach (i.e., employ on-board sensors and equipments).

Guo et al. [7] proposed a method based on the AES encryption and a hash-based Message authentication code (MAC) to check the integrity, authenticity, and confidentiality of the telegram message. Lim et al. [4] presented a lightweight encryption algorithm to protect the telegram's integrity on the Balise side and, on the train side, they also designed a hybrid controller to reduce the impacts of various attacks. Another paper proposed a cryptography solution to improve Balise-BTM communication security which is based on the Deoxys II encryption method [8]. The authors of the aforementioned papers do not explain how to implement their algorithms according to the ETCS standards regarding to Balise processing time and memory limitations.

Many researchers employ the challenge-response authentication mechanism over the Balise communication systems [9, 10]. This mechanism is proposed by the distance-bounding protocol to defeat relay and replay attacks based on the private key and distance measurement to calculate the Round-trip delay (RTD) of sending the challenge and receiving the relevant response [11, 12]. Wu et al. [6] have demonstrated the impact of attacks by simulation, and their proposed solution is to provide a challenge-response authentication mechanism.

One of the schemes that do not require trackside signalling equipment and focus on using on-board equipment of the train is solely the reference [13] that uses data fusion of the Global navigation satellite system (GNSS) and speed sensors to improve safety and security. In references [14, 15], the train's position and speed are estimated online with data fusion of the Odometer sensor and IMUs; the weakness of such methods is the uncertainty and inaccuracy of the sensors.

References [16, 17] analyze cyber-attacks on the CBTC, and [18] has proposed an intrusion detection system to detect attacks in the wireless train network protocols.

The contribution of this paper can be summarized as follows. We analyze Balise security issues to identify vulnerabilities and abnormal behavioral patterns. To detect threat patterns, we employ machine learning algorithms. The features of the attack pattern are extracted from the radio communication characteristics of Balise and the train's sensory data. The proposed method can detect an anomaly in the control system by monitoring the extracted features. By integrating a novel auxiliary fuzzy controller, the destructive effects of the attacks are reduced.

In the rest of the paper, an overview of existing attacks is presented. In Section 3, we present the proposed approach based on machine learning for attack and failure detection. In this algorithm, train sensory data are collected as features. According to observing features, machine learning algorithms such as ANFIS, SVM, and Multi-layer perceptron (MLP) classify standard operations and attacks. Subsection 3.2 presents a fuzzy controller for mitigating the impact of attacks. Section 4 discusses the simulation of the proposed algorithms using a real urban rail configuration as an example. Section 5 presents the simulation results analysis. In Section 6, we make a comparison between the proposed method and other well-known articles methods.

2 Balise Security, Vulnerabilities, and Threats

The cyber-physical attacks on the rail transportation system can aim to endanger safety, passenger injuries (even death), property damage, and economic losses. So, it is essential to know the types of cyber-physical attacks on Balise and their impact. By understanding the strategy of attacks and their effects, we can extract features to provide an intelligent solution to deal with possible attacks. In order to model threats, the following five assumptions are considered:

- An attacker is aware of the theory and details of the train control mechanism's operation and can be mobile [19].
- An attacker can install fake Balise or manipulate the information of legitimate Balise.
- An attacker is able to activate Balise by tele-powering and relay or record its telegram to replay by modification with another time and place [9].
- An attacker is able to eavesdrop or transmit fake and jamming signals by using radio equipment around the railway line.
- An attacker cannot manipulate train on-board devices or physically remove, destruct, or move Balise without being detected [4].

2.1 Threats Overview

In this section, we present a brief description of how Balise attacks are performed and their risks. Based on this information, the impact of such attacks on the train system is recognized. As a result, we can identify the signs of attacks. Afterwards, we contribute an artificial intelligence-based solution for detecting and preventing attacks. So, the aim of this study is to introduce novel measures to improve security against all the threats as follows:

Sniffing Attack When a legitimate BTM of a train energizes a Balise, a second malicious receiver can monitor the plaintext communication and obtain critical information on the telegram or a BTM impersonation sniffed telegram data. This attack is the prelude to other attacks [20].
Jamming Attack When sending and receiving data between BTM-Balise, the attacker can jam the communication by emitting a high-power electromagnetic interference in the Balise operating frequency band. Due to Balise jamming or covering, Balise telegram is not received. This attack is also known as the Balise missing attack [6].
Tampering Attack The Balise is designed with a rewritable memory to allow railway maintenance staff to update the telegram on the Balises. This opens up the possibility of rewriting telegram data which can motivate the malicious adversary to manipulate vital information in telegram. Injecting false data by the adversary causes automatic train control (ATC) to show incorrect reactions and compromise the lives of passengers [4].
Balise Cloning Balise cloning is to make an unauthorized copy of a valid telegram from a legitimate Balise to a new Balise. Cloned Balise can be a genuine Balise or producing a circuit design of the Balise. The Balise cloning threat can cause a significant train transportation disruption [4].
Fake Telegram In this attack, the adversary fabricates an arbitrary telegram to transmit for passing trains BTM by software-defined radio. The ATC system cannot distinguish the received message as a fake telegram. A fake telegram can make the train to stop immediately or to continue at full speed at the same track occupied by another train which can cause a fatal collision [9].
Transmission Extension Attack (TE) In this attack, the adversary can extend the Balise activation time by telepowering or increasing the total number of received telegrams with repeated replaying telegram. Eventually, make a significant train parking (stopping) error [6].
Telegram Relay Attack According to the safety rules, the train must reduce its speed before passing through a bent rail. Therefore, if the attacker relays the valid telegram after the bent rail for the on-board BTM, the train may derail from rails and cause disastrous consequences.
Telegram Replay Attack There are no proper authentication mechanisms for BTM. So, the attacker can activate the Balise, record the valid telegram and replay it at another strategic place and time without any difficulty. As Fig. 1a depicts, this attack creates ambiguity and inconsistency in train position ($\tilde{P}_n$) and disrupts the train movement. Also, this attack is known as Balise displacement [19].

2.2 Multi-Stage Advanced Attacks

If an adversary inserts a significant error in the train navigation system, the probability of detecting and announcing a security breach increases; under this condition, the train is switched to a fail-safe state, so the attacker can launch several combined independent attacks with limited error to create a serious security challenge. Some of these attacks are introduced:

Joint Jamming-Faking Attack According to Fig. 1b, when the train passes over the legitimate Balise, the attacker jams the valid telegram using radio equipment and immediately generates a fake telegram with the correct identification and structure according to the ETCS standard. Furthermore, it sends the fake telegram with high power

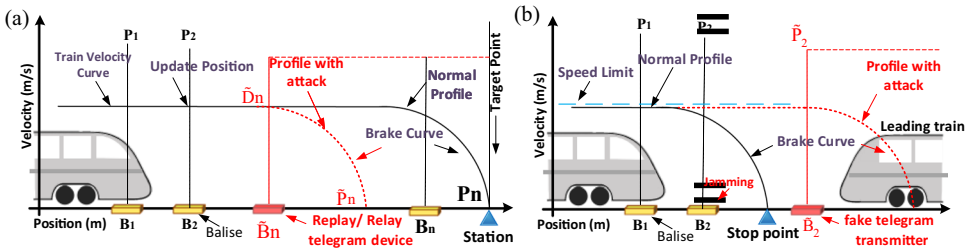


Fig. 1 Attacks on balise, **a** The adversary stops the train before the station in position $\tilde{P}_n$ by replaying/relaying the telegram B_n. **b** The attacker first jams balise B_2 signal, then sends a fake telegram containing wrong rail information that the rail is not occupied and the train continues on the route. Finally, it causes the train to collide with a leading train stopped on the railway. "=" means that telegram information does not receive bypassing trains

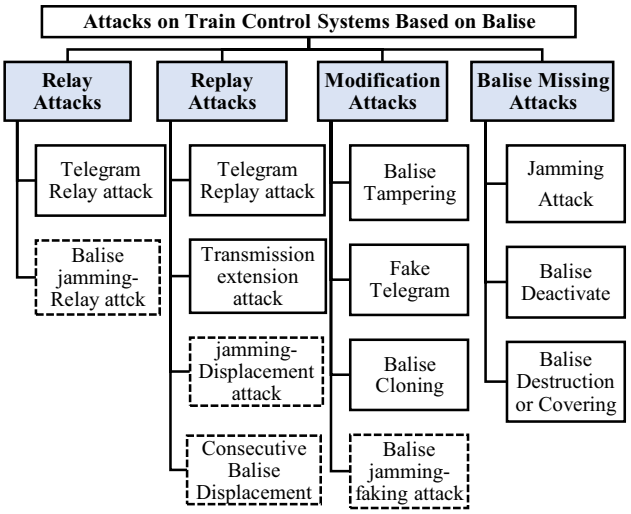


Fig. 2 Flowchart of attacks on Balise. In Multi-Stage attacks (---), the jamming attacks have been used to cover valid Balise

ratio. In this case, the on-board VC receives the fake telegram instead of the original telegram [9].

Balise Jamming-Displacement Attack Because the jamming attack is detected as Balise missing, the attacker exploits security vulnerability. The attacker jams the valid telegram sent by an authorized Balise and then transmits that valid telegram at another location after a specific time has elapsed. In other words, the jamming and the replay attacks occur successively, thus reducing the probability of attack detection. In another scenario, the attacker can first replay the pre-recorded victim's Balise telegram sooner and then prevent the train BTM from receiving the valid telegram when the train passes over the genuine victim's Balise by jamming. Hence, the railway safety of service is reduced [6].

Consecutive Balise Displacement Attack To increase the probability of the attack success, the adversary makes a Balise displacement attack on each Balise of the Balise group with limited error. The on-board VC cannot detect the error based on the information of the Balise data link (the relationship between neighboring Balise). Eventually, these attacks cause a significant error after passing over several Balises, which may be financial loss or fatal accidents depending on the train conditions. Another scenario of this attack is direction reversal. Attackers can replay the telegram of all Balises in the reverse direction to train BTM in the bi-directional railway [7].

2.3 Attacks Classification

In this article, the mentioned attacks are classified based on similarity into four categories, which are shown in the Fig. 2:

3 Proposed Approach

The railways have already been widely spread to large geographical areas of every countries. Upgrading the current railway signalling infrastructure to secure the railways against cyber-physical attacks is costly. Therefore, the proposed method must be applied to the current infrastructure and does not require a significant change in the existing system or the purchase of new equipment. According to the fixed data format and structure of passive Balise as well as low power and processing capabilities, it is far-fetched to be possible to implement a classic challenge-response protocol or a cryptographic algorithm. We must also propose a method that can detect known and unknown attacks by insiders as well as outsiders because Balise documents are available to everyone.

The proposed method is placed in the middleware layer (Virtual computer) and analyzing VC's receiving data. This method monitors the data received from the various sensors and records its observations into an audit log. The audit records are analyzed to detect anomalous operations. When an abnormal operation is detected, the audit records have deviated from the normal train operation or signatures of activity close to some previously introduced attack definition. After an attack detection, an alerting signal is triggered, and the appropriate reciprocal reaction is accomplished. Thereupon, this type of proposed method can call a train-based Intrusion detection system (IDS). The components of the proposed approach are shown in Fig. 3. The implementation procedure is presented in the three following steps:

Stage One When the train is in the range of information exchange with Balise, each sensory data (suitable features) is recorded in an audit recorder. The auditing module then store these extracted features in an audit log. The audit log is a dataset whose columns are the features and whose rows are from each record.

Stage Two The Attack detection module (ADM) is an artificial intelligence system based on machine learning for pattern recognition. This module analyzes the received audit records to classify train operations and warn suspicious behavior or malicious activities.

Stage Three The vital computer decides to perform the procedural tasks according to the alert signals and train conditions. These countermeasures can include: rejecting the telegram, reducing speed until safety is assured, immediately stopping the train (Emergency brake), or other appropriate actions that will be discussed in Section. 3.3.

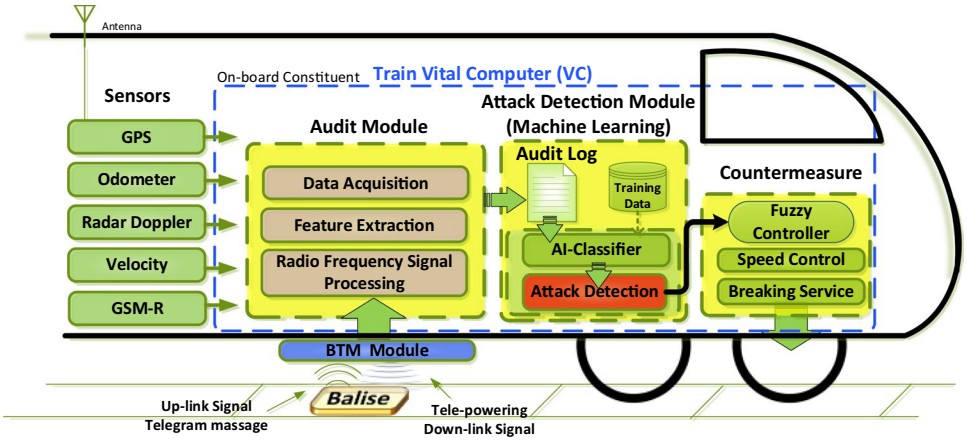


Fig. 3 Intelligent attack detection and countermeasure algorithm and operation details

3.1 Auditing Module

This module's task is to receive the sensory data and extract the appropriate features to detect an attack, and then these features are stored in the audit record. In order to collect appropriate features from the raw data, we have extracted seven numerical features that have a different pattern during the malicious activities and are evidence of abnormal behavior. Ultimately, this makes the artificial intelligence algorithms perform well. Since the attacker uses the air gap between Balise and BTM to carry out attacks, the features can be extracted from the Balise physical layer parameters and Balise information content. By measuring flowing communication link characteristics in a normal communication, any deviation from normal values can be considered as malicious behavior. The details of each feature, how they cause distinguishing between normal and abnormal behaviors, are described in the following:

(1) *Balise Transmitting Power* The attacker propagates its fake signal from another location with a particular radio transmitter. When the train passes over the fixed Balise, the Balise is excited and communicates with the tele-powering at the 4.233 MHz frequency (f_c). Most of the attacks mentioned show that the faraway adversary transmits a fake signal to the train's BTM independent of the train's tele-powering, which does not correspond to the calculated amount of receiving power. The Friis's formula can be used to calculate the power received from the Balise in the free space:

$$P_{Rx.BTM} = \frac{P_{Tx.balise} G_{Tx.balise} G_{Rx.BTM} \psi^2}{(4\pi)^2 r^2} \tag{1}$$

In (1), $P_{Tx.Balise}$, $G_{Tx.Balise,}$ and $G_{Rx.BTM}$ are the transmit power of Balise, the Balise transmitter antenna gain, and BTM receiver antenna gain, respectively. r is the transmitter–receiver distance, and ψ is the wavelength.

According to the conditions of near field communication and considering the electromagnetic coupling, the relation (1) can be written as:

$$P_{Rx.BTM} = \frac{P_{Tx.balise} G_{Tx.balise} G_{Rx.BTM}}{4} \left(\frac{1}{(kr)^2} - \frac{1}{(kr)^4} + \frac{1}{(kr)^6}\right) \tag{2}$$

where, $k = 2\pi/\psi$. At near field communication mode, power rolls off as powers higher than inverse square, typically inverse fourth ($1/r^4$) or higher. The value calculated from this equation is a good approximation of the measured values [21]. The histogram of the Received signal strength (RSS) of the valid Balise ($P_{Rx.BTM}$) in the up-link is shown in Fig. 4a. In order for the train to receive fake telegram correctly, the attacker sends the fake telegram with greater power. Any noticeable deviation in the expected mean (μ) and variance (σ) values of the calculated received power probability density function can indicate an anomaly or an attack. Deviation from determined μ and σ are then selected as the input feature of the machine learning algorithm.

(2) *Balise Signal-to-Noise Ratio* The Signal-to-noise ratio (SNR) is defined as the ratio of the Balise signal level to the noise level. The SNR value of the attacker signal and the valid signal are different because the noise level of an attacker transmitter and the level of the fake signal are different.; thus, we can detect all the attacks using the difference between the mean of normal SNR and the measured value. It is in metric form as:

$$SNR = \frac{P_{Signal}}{P_{Noise}} \rightarrow SNR\,(dB) = 10 \log \left(\frac{P_{signal}}{P_{noise}}\right) \tag{3}$$

where P is the average power in the signal bandwidth. In Fig. 4b, the solid line shows the amount of normal SNR changes when a train is passing over the Balise, and the dash-line reveals the amount of SNR changes when

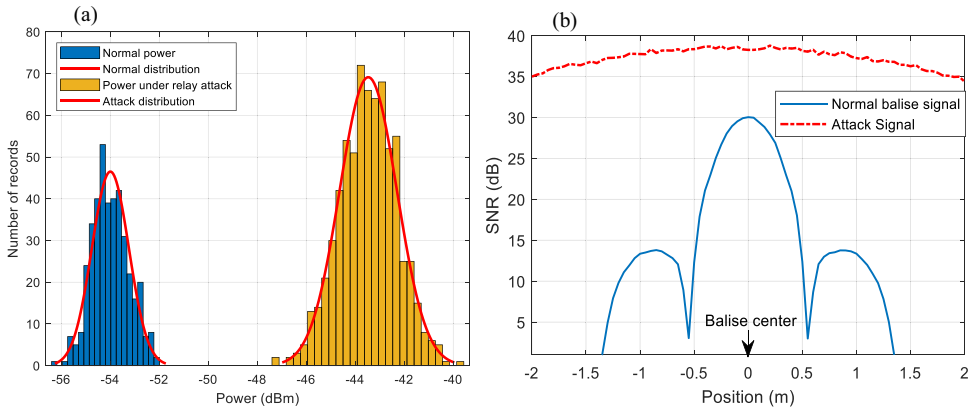


Fig. 4 **a** Histogram for the recived power probability density function of simulated received telegrams in the up-link channel with normal (without any attack) and attacked scenarios, **b** Signal to noise ratio of the received telegram in the two-scenario simulations

a replay or relay attack occurs. A jamming attack on the Balise signal is known to cause the SNR drop sharply to less than 1.

(3) *Telegram Bit Error Rate* The number of altered received bits divided by the total number of the telegram bits is known as the Bit error rate (*BER*). This altered bit can be related to interference, noise, jamming, jitter, or intentional bit change. These changes all indicate abnormalities. Thus, the average BER transmitting ($\overline{BER}$) is considered as a feature, which is obtained from:

$$\overline{BER} = \frac{\sum_{i=1}^{N_{telegram}} BER_i}{N_{telegram}} \tag{4}$$

$$BER_i = \frac{Number\ of\ altered\ bits}{Telegram\ data\ Lenght} \tag{5}$$

where $N_{Telegram}$ is the number of received telegrams. The correctness of telegram data bits is checked with the Cyclic redundancy check (CRC) algorithm [3].

(4) *Balise Communication Frequency (ΔF)* According to ERTMS specification documents, Balise-BTM communication takes place with Frequency shift keying (FSK) modulation. The up-link signal power spectrum is shown in Fig. 5. For a logical bit '0' transmission, a narrowband continuous wave is emitted at a frequency of f_0=3.951 MHz in seven periods, and for a logical bit '1' transmission, eight periods of continuous-wave f_1=4.516 MHz is emitted [3].

A shift in the frequency (ΔF) of the receiving signal may occur due to inaccuracy of an attacker's local oscillator or due to the fact that the attacker transmitter is moving. This shift in frequency signifies anomaly that could

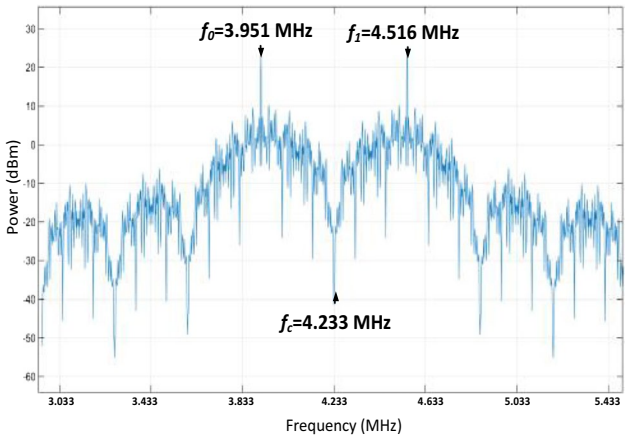


Fig. 5 The spectrum of Balise signal in up-link

indicate an attack. By monitoring the communication frequency, an unauthorized telegram transmitter can be recognized as:

$$\Delta F = \left|f_0 - \hat{f}_0\right| + \left|f_c - \hat{f}_c\right| + \left|f_1 - \hat{f}_1\right| \tag{6}$$

In (6), $\hat{f}$ is the frequency of receiving signal.

(5) *The Number of Received Telegrams ($N_{Telegram}$)* According to SUBSET-036 Balise specifications [3], the coverage zone of the Balise is activated with tele-powering ($D_{tele\_powering}$) is 1.1 to 1.4 m. By considering the train speed (V_{Train}), the approximate time of the Balise-BTM communicating connection can be calculated as:

$$T_{Com} = \frac{D_{tele\_powering}}{V_{Train}} \tag{7}$$

There are two standard forms of the short telegram (341 bits) for the high-speed train and the long telegram (1023 bits). According to the data transfer rate (B_{Rate}=564.48

kbit/s), the number of received copies of the telegram ($N_{Telegram}$) can be calculated from the following equations, and a criterion can be considered to detect the anomaly:

$$T_{Telegram} = \frac{L_{Telegram}}{B_{Rate}} = \frac{1023\,(Bit)}{564.48 \times 10^3\,(Bit/s)} = 1.812\ \ ms \tag{8}$$

$$N_{Telegram} = \frac{T_{Com}}{T_{telegram}} \tag{9}$$

where $L_{Telegram}$ is the telegram length. In most mentioned attacks, the adversary keeps sending copies of the fake telegram, regardless of the train's speed, which is inconsistent with the number of valid telegrams that BTM receives each time the train passes over the genuine Balise. Also, in the transmission extension attack, the attacker replays more valid telegrams to obscure the train's position. If the measured number of received telegrams significantly drifts from the calculated value, a suspicious action is detected.

(6) *Inconsistency in the Balise Position (ΔD)* After decoding the telegram information of a Balise, its distance to the next Balise group is registered in the linking data packet. Besides, the distance between the pre-installed Balise is already known [19]. According to the following relation, the actual travelled distance ($D_{travelled}$) is obtained by train internal sensory data and distance to the next Balise (D_{Next}) decoding from telegram information. The difference between $D_{travelled}$ and D_{Next} is calculated as a convenient feature ($\Delta D_{=}|D_{travelled}-D_{Next}|$) for attack detection.

$$D_{travelled} = \frac{2\pi \times N_{counter}}{N_{tacho}} \times R_{Wheel} \tag{10}$$

where $N_{counter}$ and N_{tacho} are the number of train tachometer pulse output in travel time and the number of pulses per wheel revolution, respectively. R_{wheel} is the radius of the train wheel.

(7) *Position Difference obtained from GNSS and Balise (ΔP)* One of the features that can detect a fake telegram is the difference between the decoded telegram position and the position obtained from the satellite positioning system. If the feature values have a significant drift from the statistic mean, it is a sign of an anomaly.

There are many methods to drive ΔP, such as Euclidean distance, Manhattan Distance or Minkowski Distance, here we employ the two former methods as follows:

Each balise has a unique identification number, and its geographical position along the track is known by VC. Given the geographical position of each installed balise and obtaining the train's location through the global navigation satellite system (GNSS), when the train passes over the balise, we can calculate ΔP from:

$$\Delta P = \sqrt{(Long_{Balise} - Long_{Train})^2 + (Lat_{Balise} - Lat_{Train})^2} \tag{11}$$

where $Long_{Balise}$ and Lat_{Balise} determine the longitude and latitude of balise position, respectively. $Long_{Train}$ and Lat_{Ttrain} are the longitude and latitude of train position obtained from GNSS [6].

It must be noted that, according to Telegram Data Structure standards, Telegram data packet 79 contains the geographic position in the payload data structure of balise [3, 8]. Besides, the longitudinal position of the train along the track is being derived from the telegram, which are being used as absolute position references. As a result, by calculating the longitudinal position of the train through the GNSS, we can obtain ΔP employing a simple difference Distance method as:

$$\Delta P = \left|Position_{Balise} - Position_{Train}\right| \tag{12}$$

where *Position* is the longitudinal position of the balise or train.

3.2 Attack Detection Module

The seven features extracted by the auditing module are forwarded to the Attack detection module (ADM). Hence, the ADM consists of a machine learning algorithm for pattern recognition. The output of this module is classified as the standard or abnormal operation.

It is challenging to model normal and abnormal behavior patterns and determine decision-making rules in ambiguous and nonlinear issues. The use of human skills and experience alone in this issue reduces the accuracy and quality of results. Thus, machine learning and human skill are used simultaneously to detect attacks. Expert human knowledge and skills are used to extract characteristics and label activities. Furthermore, the machine is trained based on labelled data and makes decisions for previously unseen conditions.

In this paper, to solve attack detection problems, three different and well-known machine learning algorithms called MLP, SVM, and ANFIS are employed. These algorithms can categorize normal logs from abnormal ones after the training process. At the end, the results are compared with each other.

Artificial neural networks (ANNs) are suitable for supervised classification and pattern recognition. Its function is

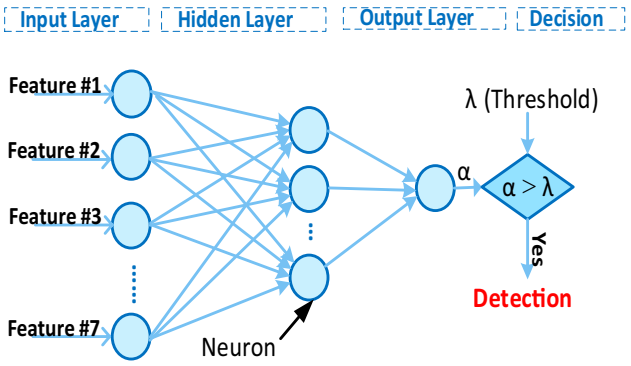


Fig. 6 The proposed artificial neural network structure (MLP) for attack detection

inspired by the human brain, which has the least error in modelling the output by observing an event's characteristics as input and determining the type of output. The least modelling error during the training process is done by determining the optimal weight of the links and the neurons' bias (Fig. 6). Each neuron is a nonlinear mathematical transfer function of given weighted inputs [22].

3.2.1 Machine Learning Algorithms

MLP Algorithm The proposed MLP network can identify the attacks by classifying the operating data. MLP structure consists of seven neurons in the input layer (features), five neurons in the hidden layer, and one neuron in the output layer (Fig. 6). The value of output neuron (α) changes between zero and one, with zero indicating normal behavior and one indicating exactly an attack occurrence. The sigmoid function defines the output of each neuron. In Fig. 6, λ is a constant number in the range of 0 to 1 that determines the attack detection threshold.

SVM Algorithm In the proposed method; the SVM algorithm is a classifier with a linear kernel function. In this algorithm, the input is a features vector in the coordination space. Each axis represents the value of a feature. Based on the training data, the SVM finds an optimal separating hyperplane in the coordinate space with maximum margins than the two sets of normal and abnormal data. In other words, the optimal hyperplane has the maximum distance from the data in the margin between the two categories. The optimal hyperplane is obtained by the Quadratic Programming (QP) method, which is a well-known method for solving constrained optimization problems [22].

ANFIS The attack detection module must be able to classify data with uncertainty. For this reason, an ANFIS is employed to detect any attack. Identification of member components in fuzzy rules, extraction, and optimization of the structure of fuzzy inference rules are done using learning concepts in neural networks. This tuning operation allows fuzzy systems to learn their structure from a set of data. These attractive features of ANFIS have led to its success and application in various scientific fields [23]. Figure 7 shows the ANFIS structure consisting of five layers that are directly linked to each other. The first layer determines the degree of membership of seven features as input to each fuzzy set, which is called fuzzification. In the second layer, the activation rate of each rule is specified (w_i). The output of the normalized third layer ($\overline{w_i}$) is the output of the previous layer. The third layer takes the normalized values and calculates the non-fuzzy values from:

$$\overline{w}_i F_i = \overline{w}_i(c_0 + x_1 c_1 + \ldots + x_7 c_7) \tag{13}$$

where c_i is a parameter obtained during the training process, and x_i is the value of the *ith* feature. In the last layer, the final overall output is $\alpha = \sum \overline{w}_i F_i$.

3.3 Attack Countermeasures Module

In the previous section, the detection module's output (α) is mapped between zero and one. It then is entered into the Attack countermeasure module (ACM) input. ACM is a fuzzy controller system to mitigate the impact of attacks and increases safety. Depending on the inputs, the ACM can stop the train immediately (emergency brake), brake safely, slow down, or continue the route. Figure 8 depicts the ACM structure. The first input, α is mapped to three Gaussian combination membership function (normal, suspect, and attack). The second input, β is the ratio of obstacle distance ($D_{Obstacle}$) to train's speed, which is obtained by dividing the Doppler value in meters by the on-board speedometer (m/s). Also, $\beta = \frac{D_{\text{Obstacle}}}{V_{Trian}}$ is mapped to three Gaussian combination membership functions (Low, Medium, and High). This part is called fuzzification, where the membership value of each crisp input value to each fuzzy set is calculated [23].

When all the values of the introduced properties change within the normal range, the output of the attack detection module (α) is close to zero. If some features' value is

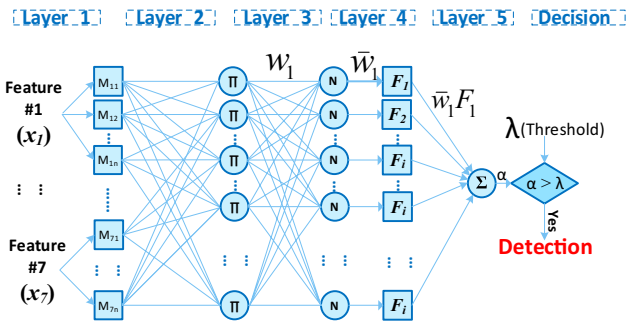


Fig. 7 The architect ANFIS attack detection module

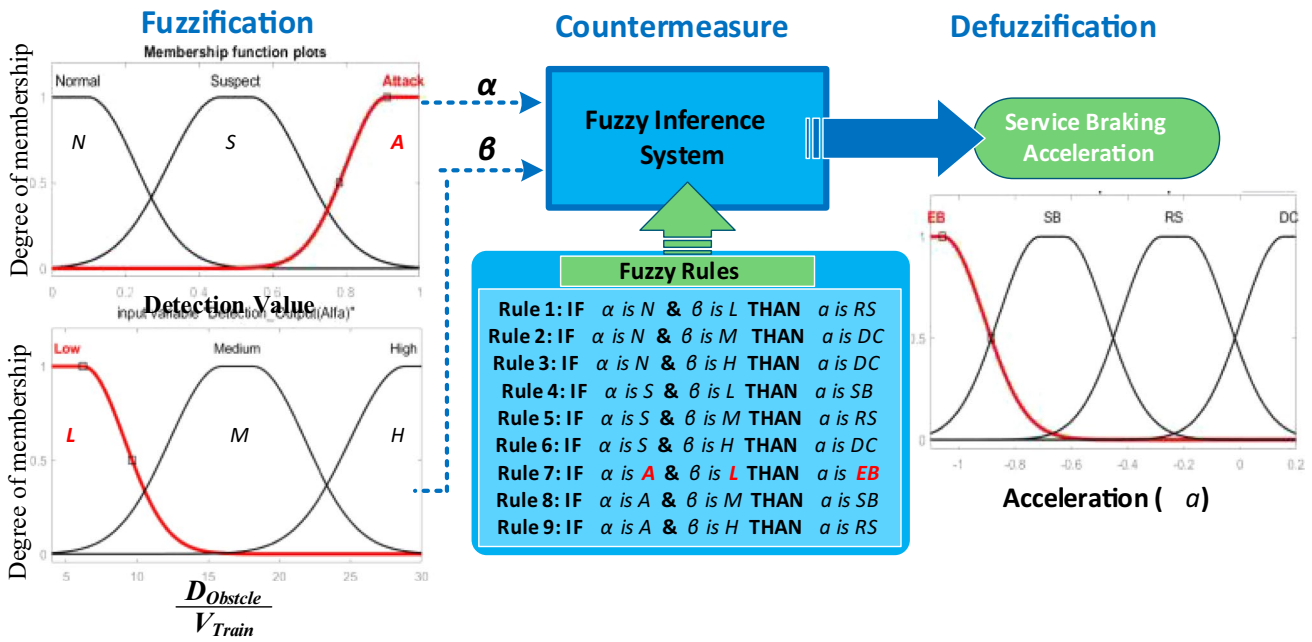


Fig. 8 Conservative fuzzy logic controller for increasing safety and security

Table 1 Fundamentals of fuzzy logic control

		Attack detection output (α)		
		Normal	Suspect	Attack
$\beta = \frac{D_{Obstacle}}{V_{Trian}}$	Low	(1) **RS**	(4) **SB**	(7) **EB**
	Medium	(2) **DC**	(5) **RS**	(8) **SB**
	High	(3) **DC**	(6) **DC**	(9) **RS**

RS: Reduce speed, ***DC***: Do not care, ***SB:*** Safe breaking, (#) Rule number, ***EB***: Emergency braking

slightly higher or lower than the normal value, the ADM output is close to 0.5, and the suspect system mode is declared. In this case, the received telegram is rejected, and the fuzzy inference system reacts according to the table of rules (Table 1 and Fig. 8). Each entry in the table defines a rule. For clarification purposes, e.g., rule 7: If the characteristics are far from their normal values, and there is a possibility of collision, the alarm is triggered, and the control center is notified. Then, the VC executes the consequent part according to the fuzzy antecedent part, i.e., Emergency braking (EB). In rule 8, VC calculates the safe braking (SB) distance curve and takes action.

Figure 9 shows the control surface of the confrontation system, the two axes of which are the inputs. Another axis is shown relative to the system's final output, which is acceleration. Due to its nature, this fuzzy mitigates the

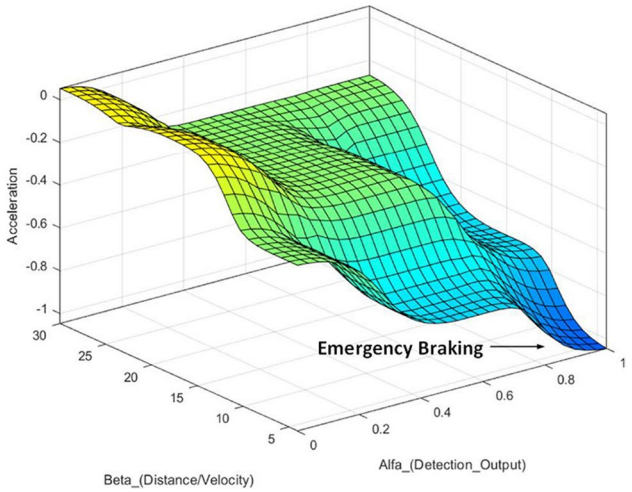


Fig. 9 Resulting fuzzy control surface obtained by plotting the inferred control action

impact of the attacks, increasing safety and security, as well as passenger comfortations.

4 Simulation

In this section, the proposed method is tested by high-fidelity simulation using a real urban rail configuration as a case study. The case study is a real rapid transit urban rail line with a length of 19.5 km with 22 stations.

When the train passes over the Balises, each sensory data and system mode are recorded in an audit record for normal data collection. In fact, simulated attacks has been recoded for this proposed studies because performing real attacks can endanger passengers and a train for collecting data. The train's trajectory is simulated based on the actual position of the Balise, the stations' position, the dynamic train movement, and signalling data. Each record data contains eight elements. The first seven elements are the introduced features in subsection 3.1, and the 8th element determines its status, i.e., normal or type of attack.

The normal sensory data of the train was collected from the studied transport route for the normal dataset. In order to provide the attacks dataset, it is necessary to analyze the theory of attack strategy so that we can determine the amount of change in each detection feature based on the wireless channel theory and the attack scenario. Therefore, the logical manipulated normal record data can be a good representation of the real attack data. Figure 10 illustrates the train trajectory tracks in normal transit mode as well as the performance of the introduced simulated attacks on the case study [24, 25]. For further clarification, we provide the following examples:

In the transmission extension attack, an adversary has extended Balise activation by prolonging the duration of telepowering or by replaying more copies of the original telegram to create *"e"* meters error in actual Balise position (P). As a result, this makes a significant train parking error. In order to create a record of this attack with the erroneous position ($\tilde{P}$) in the dataset, we increase the number of telegrams received in the normal record based on m:

$$e = \tilde{P} - P \approx \frac{L_{Telegram}.V_{Train}(m+1)}{2B_{Rate}} - \frac{\delta}{2}$$
$$0 \leq \delta < \frac{L_{Telegram}V_{Train}}{B_{Rate}} \tag{14}$$

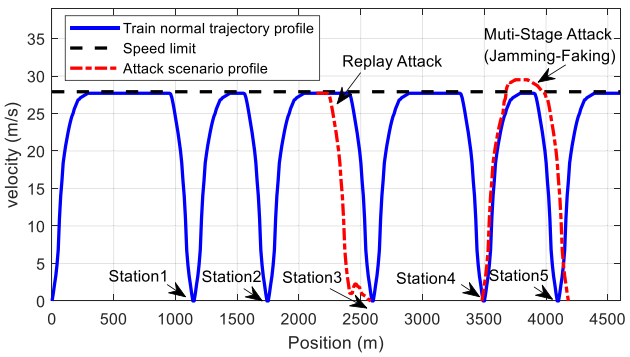


Fig. 10 Simulation of train trajectory due to the stations' position, installed Balises, speed limit, and other configurations

$$m = \frac{e.2B_{Rate}}{L_{Telegram}.V_{Train}} - 1 \quad if \quad \delta = 0 \tag{15}$$

where $L_{Telegram}$ and B_{Rate} are the lengths of the telegram and telegram transmission bitrate, respectively. In a situation where the train passes over the Balise at a constant speed V_{Train}., if the train computer receive m extra telegram massage, e meter positioning errors can occur. This attack may also have symptoms such as replay and relay attacks, affecting the $P_{Rx.BTM}$, SNR, $\overline{BER}$, ΔF, ΔD and ΔP features [5]. The effect of such attack on each feature can be calculated from (1) to (12).

In another example, we describe how a joint jamming-faking attack affects the features of an audit record (Fig. 10). According to the attack strategy described in subsection 2.2 (Fig. 1b), the adversary interrupts the signal of a valid Balise, causing a sharp decrease in signal-to-noise ratio ($SNR < 1$). Also, the telegram message is not received correctly, and bit error rates are increasing ($\overline{BER} > 10^{-4}$) [26]. Immediately afterwards, the adversary sends a fake telegram with greater power toward the BTM, which causes the mean (μ) and standard deviation (σ) of the density of received signal power ($P_{Rx.BTM}$) to change. The adversary sends multiple copies of fake telegrams to make sure the BTM receives the fake telegram. Due to this, the number of telegrams received by BTM ($N_{telegram}$) exceeds the normal value calculated from (9). Furthermore, there is an inconsistency in the position extracted from fake telegram data with the location obtained from satellite positioning (GPS) and telemetry systems. Consequently, the ΔP and ΔD features become tens of meters higher than the normal range.

After analyzing the sensory data records in the audit log, the data set are preprocessed. We applied various preprocessing methods to the features, such as data normalization and multidimensional scaling. As a result, the data are both weighted and scaled, so one feature does not have a greater effect on ML algorithms than another. In addition, by identifying valuable features that have more entropy, we separated them from redundant and irrelevant features, and outlier attributes, as well as noisy data, are removed. This preprocessing has improved performance and generalized model and reduced overfitting and misleading data. The dataset reduces to 86 unique records. Because of this, the classifiers will not have any bias towards more frequent records. It should be noted that, the number of abnormal records is produced on the same scale so that the algorithm is appropriately trained by the same number of normal and abnormal data in the database.

Monte Carlo Cross-Validation (CV) is a method used to assess the performance of the proposed machine learning algorithms on unseen data. This method is also known as a repeated random sub-sampling CV. This method creates multiple splits of the datasets into training, validation,

and test data with a particular share. In other words, about 70% of the database, i.e., 60 records of the normal dataset and in the same way, 60 records of the abnormal dataset, are randomly selected for training, and the remaining 26 records from each category are used for validation and test. Throughout the CV procedure, the number of iterations continues until the fixed number is reached. Consequently, the algorithm can perform reliably with a limited set of training data in general.

In the attack detection algorithm based on MLP, the Levenberg–Marquardt optimization method has been used to update weight and bias values and, as a result, to minimize the classification error.

This paper uses a hybrid approach for training ANFIS. The hybrid learning procedure combines gradient descent and least-squares estimators. Then, the attack detection fuzzy membership functions are automatically calculated by subtractive clustering. Furthermore, training iterations are repeated until the classification error is minimized.

In the SVM algorithm, the empirical risk minimization is defined as an objective function, and the parameters are adjusted and optimized accordingly [27].

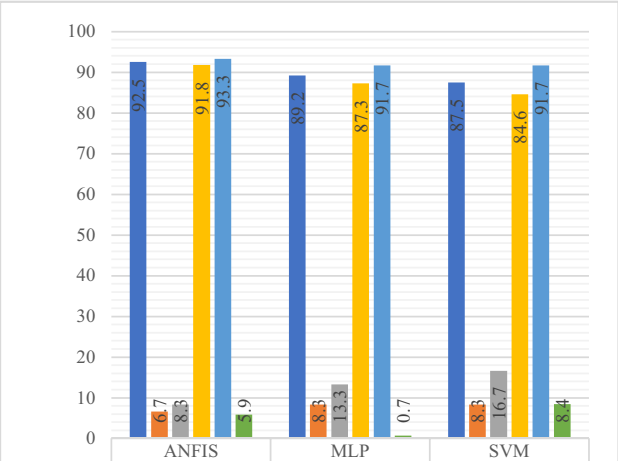
The test dataset contains several known pre-existing attacks that were not present in the training dataset to validate the proposed algorithm. With this technique, the generalizability of the algorithm is well tested to detect unknown attacks. Training with pre-existing attacks may increase the detection rate.

5 Result Analysis

It is clear that for an efficient algorithm, the False-Negative (FN) and False-Positive (FP) rates must be the lowest, while the True Negative (TN) and True Positive (TP) rates must be the highest. Therefore, to measure an algorithm performance, criteria described in Table 2 are employed.

Table. 3 shows the results of the comparison of each proposed method in terms of the evaluation criteria. ANFIS system has the highest detection percentage, and then SVM and MLP are in the next ranks, respectively. Nonetheless, MLP generalizability can categorize new attack patterns with other known attack patterns that have the same distinguishing features.

Table 3 Analysis performance of the algorithms ($\lambda = 0.72$)



	ANFIS	MLP	SVM
ACC	92.5000	89.1667	87.5000
FNR	6.6670	8.3334	8.3334
FPR	8.3340	13.3330	16.6667
PPV	91.8032	87.3015	84.6153
TPR	93.3334	91.6670	91.6670
Td	5.9096	0.6710	8.4288

A fair criterion for evaluating the algorithm's computational complexity is the detection delay (*Td*). The *Td* is the length of time taken for the algorithm's output to be determined after receiving input. To obtain the *Td*, we can measure the propagation delay factor in milliseconds. Propagation delay is the length of time taken for the input signals to reach the output of the attack detection algorithm.

Table 3 provides the *Td* values for the proposed algorithms. These values are not a determinant and critical factor in comparison because the detection times order for all the algorithms are milliseconds. The MLP approach has a minor propagation delay. This result shows the simplicity of

Table 2 Algorithms analysis criteria

Criteria	Formula	Description
Accuracy (ACC)	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$	This criterion shows the number of correct predictions to the total number of predictions
False-negative rate (FNR) (Miss rate)	$FNR = \frac{FN}{TP+FN}$	This criterion shows the number of attacks known as normal to the total number of attacks
False-positive rate (FPR) (False alarm)	$FPR = \frac{FP}{TN+FP}$	The ratio between the standard operations detected as an attack and the total number of actual negative events
Positive predictive value (PPV) (Precision)	$PPV = \frac{TP}{TP+FP}$	The ratio of the number of attacks correctly detected to the total number of true and false detections
True positive rate (TPR) (Sensitivity)	$\mathrm{TPR} = \frac{TP}{TP+FN}$	This criterion shows the number of attacks that have been correctly detected

implementing the MLP structure. Furthermore, *Td* shows higher complexity of the SVM relative to other algorithms.

An additional factor to consider when comparing the computational cost is their order of computational complexity [26, 28]. In Table 4, we listed the computational complexities of the algorithms. The first term is the computational complexity related to the training process, and the second term is related to the testing process. N_{tr}, N_{Test}, and N_{MF} are the number of training instances, testing instances and membership function, respectively. d is the number of features in the data set, I_t is the number of iterations, and A_c is the architect complexity of methods. MLP architect complexity can be obtained from the following relation:

$$A_c = (i_P \times h + h \times o_p)\Theta + (i_p + h + o_p)\Theta \tag{16}$$

where i_p, h, and o_P are the number of functions for input layer, hidden layer, and output layer of structures. The Θ is computational complexity of mathematical functions.

The output variable (α) of the attack detection module varies between 0 and 1. By determining the threshold level (λ) of attack detection, a trade-off can be established between False Alarm and Miss Rate (Fig. 11a). In the case studies, if the output of the detection module exceeds the threshold level (λ = 0.72), the attack is detected, and the train is switched to fail-safe mode. With this value of λ, the ANFIS algorithm has the lowest false alarm and miss rate.

Another method of evaluating the performance of binary classification is the Receiver operating characteristic (ROC). In the ROC diagram, both of these criteria, Sensitivity and False Positive Rate, are plotted due to the change in the detection threshold level (λ) as a curve. According to Fig. 11b, the algorithms with ROC curves close to the top-left show better performance. ANFIS has the best classification performance in the ROC plot. The reliable metric to rank the proposed algorithm is Area Under the ROC Curve (AUC). Since ANFIS has the highest AUC ($AUC_{ANFIS} = 0.942$), the classifier has the lowest error rate and the highest sensitivity. MLP and SVM are ranked second and third, respectively ($AUC_{MLP} = 0.906$, $AUC_{SVM} = 0.887$).

6 Comparison

Several methods are described for attack detection and countermeasure in the literature in the introduction section and Table 5. In the following, we compare and report the technical challenges of the proposed method and related work.

The contribution of references [8, 19], and [7] are based on cryptography. Classic symmetric cipher and authentication, AES and MAC are not compatible with ETCS standards. Indeed, due to the limited processing time and low memory capacity of the passive device (fix Balise), it is impossible to implement advanced cryptography. Furthermore, lightweight encryption/decryption methods do not have high security in data integrity, confidentiality, and authentication. It is notable that, the secret key cannot be securely stored and protected in the Balise memory [3, 8].

Furthermore, these methods are vulnerable to relay attacks such as Balise displacement. For example, in the scheme of checking the integrity of telegram data using HMAC [7], an attacker could be sniffing and recording an

Table 4 Comparison of the computational complexity

Method	Computational complexity	Total
MLP	$O(dN_{Tr} I_t A_c) + O(dN_{Test} A_c)$	27,544,846
SVM	$O(dN_{Tr}^2\Theta + N_{Tr}^3 I_t\Theta) + O(dN_{Test}\Theta)$	346,006,112
ANFIS	$O(dN_{Tr} I_t A_c + dN_{Tr} I_t N_{MF}\Theta) + O(dN_{Test} N_{MF} A_c)$	242,592,000

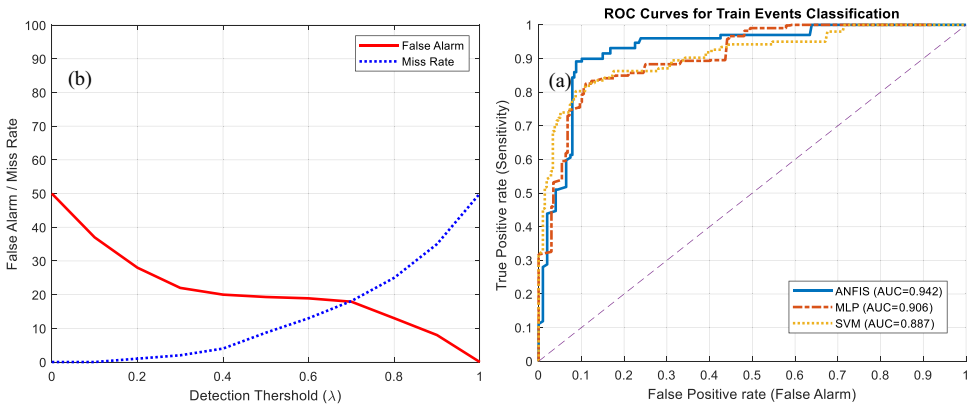


Fig. 11 **a** The trade-off between Miss Rate and False Alarm according to λ, **b** Receiver Operating Characteristic (ROC) diagram

Table 5 Compare the approaches proposed in different articles

Works	Attacks covered								proposed approach	Limitations/Constraints	Advantages
	Jamming	Replay	Relay	Tampering	Cloning	Faking	Advance	Unknown			
[7]				○ X	○ X	○ X			AES and HMAC cryptographic	Processing time and capacity of memory	Telegram data integrity
[13]		○	○	○	○	○			GNSS and velocity sensor	Availability and accuracy	Independent mechanism
[9] [6]	○ X	○ X	○ X	○ X		○ X			Challenge-response authentication	Implantation constraints and clock synchronization	Integrity and Confidentiality telegram
[4]	○	○ X	○ X	○ X	○ X	○ X	○ X	○ X	Lightweight cryptographic and hybrid controller	Implantation Challenges and keys distribution	No additional hardware or sensors required
[15]		○	○	○	○	○	○		Odometer and IMU sensors	Uncertainty and inaccuracy	Independent mechanism
[19]		○ X							Cryptographic random fountains	The latency of communication and Implantation constraints	Telegrams containing random signals
[8]		○ X	○ X	○ X	○ X	○			Deoxys II cryptographic mechanism	Design constraints and costly initialization	Integrity and confidentiality of telegram
[29]	○ X								Frequency hopping	Incompatible with the infrastructure	Jamming mitigation
[10]	○		○ X	○ X				○ X	Hybrid brake controller	Defect threat model	Software-only countermeasure
[30]				○ X	○ X	○ X			Fast telegram decoding algorithm	Changes in hardware and infrastructure	Reduce latency and complexity
This paper	○	○ X	○ X	○ X	○ X	○ X	○ X	○ X	Machine learning and fuzzy system	Require Training Data set	Intelligent attack Detection and fuzzy countermeasure

○ - Attack detection

X - Attack countermeasure

encrypted message with a valid authentication code and then can replay it elsewhere for the train in the same rail track. Alternatively, in the reference [19], to prevent telegram replay, nonce and index are used to communicate between Balise-BTM. In this case, the attacker can relay the train nonce far away from the Balise location and retransmit the train's Balise response [31].

The proposed method is based on artificial intelligence, a machine learning software program to detect and countermeasure anomalies. This software can be installed on the train's vital computer. The proposed method does not require extra equipment or change railway infrastructure. The proposed algorithm requires the normal sensory data along the rail route for training the detection algorithm only in the initial configuration phase.

Mentioned methods based on the challenge-response mechanism can increase security against relays and replay attacks but can only be employed on controllable Balises connected to the LEU because the controllable Balises support both up-link and down-link communications.

In this type of solution, there are challenges to key management and key sharing. Furthermore, the MAC cannot be computed fast on passive devices and in noisy environments. This is because the MAC calculation increases the latency of communication, and changing any of the received nonce bits due to noise can cause changes to the entire MAC code.

It should be emphasized that the ERTMS/ETCS SUBSET-036 standard does not support clock synchronization signalling, which is performed by the challenge-response mechanism by many papers, but this mechanism is vulnerable to some attacks, such as distance attack, mafia attack, and terrorist attack [32] while with machine learning method employed here no clock synchronization is required.

Technical challenge methods that use internal train equipment (such as speed sensors, GNSS, IMU) for safety and security applications solely can be referred to as the sensors' uncertainty and inaccuracy. This disadvantage is especially true when train lines are close together. Besides, how is it possible to locate the train with a satellite positioning system when the train is underground or in a tunnel? Similarly, satellite positioning systems are also vulnerable to attacks, such as spoofing and jamming.

7 Conclusions

In this study, we reviewed the attacks and demonstrated the importance of cybersecurity and the vulnerability of Balise-based train control systems. The proposed method is a software program installed on the train computer to detect and countermeasure various attacks on the Balise. Thence, our concept is not required to add extra equipment or change the railway's infrastructure. The proposed Machine learning (ML) algorithm detects failure behavior using classification by collecting raw train data and extracting the distinguishing features. In the ANFIS algorithm, 92% accuracy and 91% precision of detection are achieved by recognizing the signs of the threats from the standard operating patterns. The proposed algorithms can detect attacks and failure operations. Accordingly, the proposed method has the limitation of requiring training data to detect threats. In order to train the ML algorithm, normal training data were collected from sensors in normal travel mode, and abnormal data were obtained from the simulation of an actual attack scenario. Finally, a fuzzy controller takes the necessary steps to mitigate the impact of attacks and increase safety.

References

1. International Union of Railways: Railway statistics synopsis 2020 edition. 33, 2 (2020)
2. Gheth, W., Rabie, K.M., Adebisi, B., Ijaz, M., Harris, G.: Communication systems of high-speed railway: a survey. Trans. Emerg. Telecommun. Technol. **32**, e4189 (2021). https://doi.org/10.1002/ett.4189
3. Lundberg, P., Prieels, P.: Test specification for Eurobalise FFFIS. Eur. Railw. Agency. 1–341 (2015)
4. Lim, H.W., Temple, W.G., Tran, B.A.N., Chen, B., Kalbarczyk, Z., Zhou, J.: Data integrity threats and countermeasures in railway spot transmission systems. ACM Trans. Cyber-Physical Syst. **4**, 1–26 (2019). https://doi.org/10.1145/3300179
5. Rannjbar, V., Olsson, N.O.E.: Towards mobile and intelligent railway transport: a review of recent ERTMS related research. In: Computers in Railways XIV. pp. 65–73 (2020)
6. Wu, Y., Wei, Z., Weng, J., Deng, R.H.: Position manipulation attacks to Balise-based train automatic stop control. IEEE Trans. Veh. Technol. **67**, 5287–5301 (2018). https://doi.org/10.1109/TVT.2018.2802444
7. Guo, H., Wei Sim, J.Z., Veeravalli, B., Lu, J.: Protecting train Balise telegram data integrity. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). pp. 806–811. IEEE (2018)
8. Ronald, L.: Security improvement on Balise-based train control system. Int. J. Adv. Res. Sci. Eng. Technol. **6**, 9829–9836 (2019)
9. Wu, Y., Weng, J., Tang, Z., Li, X., Deng, R.H.: Vulnerabilities, attacks, and countermeasures in Balise-based train control systems. IEEE Trans. Intell. Transp. Syst. **18**, 814–823 (2017). https://doi.org/10.1109/TITS.2016.2590579
10. Temple, W.G., Tran, B.A.N., Chen, B., Kalbarczyk, Z., Sanders, W.H.: On train automatic stop control using balises: attacks and a software-only countermeasure. Proc. IEEE Pacific Rim Int. Symp. Dependable Comput. PRDC. 274–283 (2017). https://doi.org/10.1109/PRDC.2017.52
11. Falahati, A., Jannati, H.: Distance bounding-based RFID binding proof protocol to protect inpatient medication safety against relay attack. Int. J. Ad Hoc Ubiquitous Comput. **22**, 71–83 (2016). https://doi.org/10.1504/ijahuc.2016.077199
12. Jannati, H., Falahati, A.: An RFID search protocol secured against relay attack based on distance bounding approach. Wirel. Pers. Commun. **85**, 711–726 (2015). https://doi.org/10.1007/s11277-015-2804-5

13. Lauer, M., Stein, D.: A train localization algorithm for train protection systems of the future. IEEE Trans. Intell. Transp. Syst. **16**, 970–979 (2015). https://doi.org/10.1109/TITS.2014.2345498
14. Xu, Z., Wang, W., Sun, Y.: Performance degradation monitoring for onboard speed sensors of trains. IEEE Trans. Intell. Transp. Syst. **13**, 1287–1297 (2012). https://doi.org/10.1109/tits.2012.2188629
15. Malvezzi, M., Vettori, G., Allotta, B., Pugi, L., Ridolfi, A., Cuppini, F., Salotti, F.: Train position and speed estimation by integration of odometers and IMUs. 9th World Congr. Railw. Res. 22–26 (2011)
16. Systems, T.C.: A novel intrusion detection model using a fusion of network and device states for communication-based train control systems. (2020). https://doi.org/10.3390/electronics9010181
17. Chi, M., Bu, B., Wang, H., Lv, Y., Yi, S., Yang, X., Li, J.: Multichannel man-in-the-middle attack against communication-based train control systems: attack implementation and impact. In: Lecture Notes in Electrical Engineering. pp. 129–139 (2020)
18. Gao, B., Bu, B.: A novel intrusion detection method in train-ground communication system. IEEE Access **7**, 178726–178743 (2019). https://doi.org/10.1109/ACCESS.2019.2958198
19. Harshan, J., Chang, S.-Y., Kang, S., Hu, Y.-C.: Securing balise-based train control systems using cryptographic random fountains. In: 2017 IEEE Conference on Communications and Network Security (CNS). pp. 405–410. IEEE (2017)
20. Thamilarasu, G., Sridhar, R.: Intrusion detection in RFID systems. In: MILCOM 2008 - 2008 IEEE Military Communications Conference. pp. 1–7. IEEE (2008)
21. Torchio, R., Cirimele, V., Alotto, P., Freschi, F.: Modelling of road–embedded transmitting coils for wireless power transfer. Comput. Electr. Eng. **88**, 106850 (2020). https://doi.org/10.1016/j.compeleceng.2020.106850
22. Hosseini, S., Zade, B.M.H.: New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. Comput. Networks **173**, 107168 (2020). https://doi.org/10.1016/j.comnet.2020.107168
23. Libor Hadacek, Lenka Sivakova, Radovan Sousek, Mikael Zeegers: Assessment of security risks in railway transport using the fuzzy logical deduction method. Commun. - Sci. Lett. Univ. Zilina. 22, 79–87 (2020). https://doi.org/10.26552/com.C.2020.2.79-87
24. P Sangameswar Raju, Nikhath, F.J., Farook, S., Raju, P.S., Nikhath, F.J., Farook, S.: Automatic train operation and control using MATLAB. Int. J. Electr. Electron. Eng. Telecommun. 2, 148–155 (2013)
25. de Mooij, N.. A.: How to optimize train performance during delays with the use of intelligent automation, (2018)
26. Wardoyo, R., Afifa, L.N.: Computing the time complexity of ANFIS algorithm. Int. J. Adv. Res. Comput. Eng. Technol. **7**, 132–135 (2018)
27. Canedo, E.D., Mendes, B.C.: Software requirements classification using machine learning algorithms. Entropy. 22, (2020). https://doi.org/10.3390/E22091057
28. Orponen, P.: Computational complexity of neural networks: a survey. Nord. J. Comput. **8**, 99–117 (1995)
29. Lakshminarayana, S., Karachiwala, J.S., Chang, S.-Y., Revadigar, G., Kumar, S.L.S., Yau, D.K.Y., Hu, Y.-C.: Signal jamming attacks against communication-based train control. In: Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 160–171. ACM, New York, NY, USA (2018)
30. Moon, S., Park, S., Lee, J.-H., Lee, Y.: Rapid Balise telegram decoder with modified LFSR architecture for train protection systems. IEEE Trans. Circuits Syst. II Express Briefs. 7747, 1–1 (2018) https://doi.org/10.1109/TCSII.2018.2844731
31. Falahati, A., Jannati, H.: All-or-nothing approach to protect a distance bounding protocol against terrorist fraud attack for low-cost devices. Electron. Commer. Res. **15**, 75–95 (2015). https://doi.org/10.1007/s10660-014-9167-y
32. Jannati, H., Falahati, A.: Analysis of false-reject probability in distance bounding protocols with mixed challenges over RFID noisy communication channel. Inf. Process. Lett. **115**, 623–629 (2015). https://doi.org/10.1016/j.ipl.2015.02.012

Abolfazl Falahati received his B.Sc. (Hons) in electronics engineering from the University of Warwick (United Kingdom) in 1982. He was granted M.Sc. and Ph.D. degrees in Digital Communication Systems from Loughborough University of Technology (UK) in 1987 and 1991, respectively, after 2 years of work experience for Datatec Company in the UK. During 1991–1995, he was a research fellow at Rutherford-Appleton Laboratory (Oxford-United Kingdom). From Sept. 1995, he has been at Iran University of Science and Technology where, he is currently an associate professor in Cryptography, Security, Channel Coding and Digital Signal Processing. Dr. Falahati is a Chartered Engineer, a Full Member of the IET and IEEE Communication Society.

Ebrahim Shafiee received his M.Sc. degree in secure communications engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2017. He is a Ph.D. candidate at Iran University of Science and Technology, Tehran, Iran, from 2017 to date. His main research interests include security in wireless communication systems and cryptography. He especially interests in the field of artificial intelligence and machine learning algorithms. Email: Shafiee_e@elec.iust.acir