Journal of Network and Systems Management (2026) 34:52
https://doi.org/10.1007/s10922-025-10026-1

# SereWay: Toward SEcurity and REliability Benchmarking for the RailWAY IIoT

**Alessandra Rizzardi[1] · Raffaele Della Corte[2] · Jesús F. Cevallos M.[1] · Simona De Vivo[2] · Sabrina Sicari[2] · Domenico Cotroneo[1] · Alberto Coen-Porisini[2]**

Received: 10 April 2025 / Revised: 12 November 2025 / Accepted: 23 December 2025


**Abstract**
This paper presents SereWay, an open-source testbed for benchmarking the detection of security and reliability-related anomalies in the context of the railway Industrial Internet of Things domain. The proposal leverages lightweight virtualization and synthetic data generation modules trained from real train sensor data to create both normal and faulty behavior of the system, while Distributed Denial of Service attacks are generated through a BotNet simulation framework. The envisioned solution aims to facilitate the benchmarking of Machine Learning-based security and reliability solutions for the railway domain, and it is built on top of Open-FARI, an existing railway testbed for anomaly detection. The proposal has been used in the context of an experimental campaign for the evaluation of a Federated Learning-based system aiming at detecting both security and reliability anomalies. The obtained results highlight the potentialities of the proposed framework to emulate realistic railway scenarios and support the benchmarking of Machine Learning-based security and reliability instruments in this context.

**Keywords** Industrial IoT · Railway · Network security · Network reliability · European rail traffic management system

## 1 Introduction

The European Rail Traffic Management System (ERTMS) is a significant railway system standardization initiative by the European Commission emphasizing interoperability and cybersecurity [1]. The overarching objectives of ERTMS are multifaceted, aiming to achieve a more competitive and efficient European railway system. These objectives include opening and restructuring the rail market, increasing com-

Extended author information available on the last page of the article

Published online: 18 February 2026

petitiveness among rail companies, developing interoperable infrastructure, improving efficiency in infrastructure use and safety, and ensuring fair consumer prices [2]. According to the EU's 797/2016 directive [3] in the context of ERTMS, 'interoperability' means the *ability of a rail system to allow the safe and uninterrupted movement of trains, accomplishing the required performance levels*. In the context of the system's security, such a definition may be commonly associated with *resiliency*, and research efforts devoted to preserving this feature are especially important considering the EU commitment to start upgrading the standard railway communication systems in the context of the ERTMS standardization [4, 5]. In this respect, security and reliability benchmarking play a key role in pursuing system resiliency. It allows for highlighting anomalies that may potentially compromise the system's functioning and/or evaluating the system's mechanisms to identify them, which represent the fundamentals of building a resilient system. This is especially true in critical contexts like railways, where malfunctioning may lead to severe consequences.

Machine Learning (ML) and Deep Learning (DL) techniques have been largely used in industrial contexts for both reliability- and security-related tasks, such as predictive maintenance [6], optimized control [7], fault detection [8], and intrusion detection [9]. However, their evaluation requires training and testing data tailored to the specific domain of application, which may represent a problem in many industrial contexts where both legal and privacy concerns may restrict the practical availability of data [10]. Realistic Synthetic Data Generation (SDG) [11, 12] represents one of the potential solutions to face the problem of learning under low-data regimes, which has also been used in the Industrial Internet of Things (IIoT) domain [13]. Unfortunately, there is a lack of SDG instruments tailored to the railway domain. Consequently, the absence of realistic datasets and testbeds represents a significant barrier to the entrance into research on ML-driven solutions for both reliability and security in the railway IIoT domain.

A first step in this direction was made with the Open-FARI framework [14], which is an open-source Federated Learning (FL) [15] testbed centered on anomaly detection for railway IIoT. Open-FARI leverages synthetic data generation modules trained from real train (normal and anomalous) sensor data to generate realistic railway scenarios. It supports the resiliency objectives in the ERTMS Blueprint from a reliability point of view by enabling the identification of potential malfunctions or deviations from normal operational parameters, thereby contributing to a safer railway environment. However, beyond reliability, more causally oriented research focuses on cyber-intrusions that can be crucial to maintaining the overall safety of the railway environment [16]. Preventing, detecting, and mitigating cyber-security breaches is part of the cybersecurity side of ERTMS and, therefore, is an objective of utmost importance in the landscape of the European standardization of high-speed train management [17].

This paper presents SEREWAY, an open-source testbed for benchmarking the detection and mitigation of anomalies and cyber intrusions within a simulated ERTMS environment, based on lightweight virtualization (i.e., containers). The proposal augments the reliability functionalities of Open-FARI by improving its synthetic data generation and integrating a BotNet simulation framework that enables cybersecurity-oriented benchmarking associated with Distributed Denial of Service (DDoS) attack detection and mitigation. This work extends the Open-FARI platform for feder-

ated learning experiments over the railway IIoT to focus on the specific interplay of automatic security and reliability management. The main novelties introduced by SereWay are:

- Improved synthetic diagnostic data generation for a more heterogeneous and fine-grained simulation of a fleet of high-speed trains;
- Integration of the security perspective implementing attack vectors over Openfari to emulate DDoS attacks in the railway IIoT context;
- Integration of a health-status monitoring framework for the simulated railway network;
- An online attack detection module to test attack identification and mitigation using high-level informative features.

Thanks to the aforementioned contributions, SereWay emerges as an open-source testbed designed to evaluate security and reliability within railway IIoT systems, bridging two areas that have been treated separately so far. Unlike prior frameworks, exclusively focused on functional validation or on isolated aspects of network security, the SereWay framework provides an integrated view that enables the assessment of both system dependability and cyber-resilience under realistic operating conditions. Hence, SereWay contributes to filling the current gap of reproducible tools for studying how learning-based detectors perform in distributed railway infrastructures, characterized by constrained connectivity and heterogeneous data sources.

We show the usage of SereWay to conduct an experimental campaign for the evaluation of a FL-based system, which aims at detecting anomalies from both security and reliability perspectives. The obtained results highlight the potentialities of the proposal to emulate realistic railway scenarios to facilitate the benchmarking of ML-based security and reliability instruments for the railway domain.

The rest of the paper is organized as follows. Section 2 explores related work, while Sect. 3 provides some background concepts. Section 4 presents the SereWay testbed, its components, and its working mechanism. Section 5 introduces a case study of intrusion detection and reliability monitoring made in the presented testbed. Section 6 discusses the potential development criteria for further standardization of the ERTMS ecosystem, along with the validity analysis. Section 7 provides concluding remarks and future work directions.

## 2 Related Work

This Section presents the current state-of-the-art concerning existing testbeds for the railway domain and FL, from both a security and reliability perspective. Moreover, a discussion about data generation, fault diagnostics, and security benchmarking solutions in the railway IIoT context is also provided.

**Testbeds for the Railway IIoT.** Mera et al. [18] simulate the Communication Based Train Control (CBTC) system to accelerate the test and validation process of new functionalities and modules to such a system. Instead, the testbed in Kim et al. [19] focuses on the mitigation of Man-in-the-Middle (MIDM) attacks over the CBTC.

A similar approach is used by Xu et al. [20]. The work in [21] focused instead on railway-level crossings and created a distributed and hardware-compatible platform that monitors key evaluation metrics for simulation. This work unites various modules such as the *TrainDirector* simulator in [22] and the *Command & Control Wind Tunnel* framework in [23]. A follow-up work in [24] focused on reliability aspects under a heterogeneous cyber-attack scenario. Similarly, the *SecureRails* [25] software is an extension of the *OpenRails* open-source project [26] that focuses on evaluating the effects of cyber-attacks over the outcome of railway system operations.

**Testbeds for Federated Learning.**

CoLExT [27] is a physical testbed that helps test multiple federated learning algorithms over heterogeneous edge networks. FedBed [28] provides a simulation-based, open-source testbed that leverages virtualization and emulation, enabling experimentation with various network scenarios, QoS specifications, and networking delay simulations. Instead, Popovic et al. [29] created the Python-centric testbed for FL, and a later evolution in [30] extended it to physically distributed domains. Techtile [31] focuses instead on benchmarking information-rich edge computing scenarios. Finally, [32] concentrates on the heterogeneity of network protocols and FL algorithms. SereWay builds upon the Open-fari testbed [14], which evaluates federated learning algorithms for anomaly detection over a simulated railway IIoT. Such architectural choices are aligned with recent secure IIoT frameworks leveraging edge intelligence and federated learning for scalable anomaly detection in industrial contexts [33, 34].

**Realistic Sensor Data Generation.**

Adversarial settings have generated realistic data in IIoT contexts [35, 36]. Some of them use probabilistic inductive biases, such as mixture-density networks [37] or time-series modeling [38]. A recent trend in synthetic data generation is the use of diffusion models [39, 40], which can generate tailored outputs through conditional models. Similar works using DL for synthetic IIoT data generation are available in [37, 41, 42]. Recent non-parametric data generation studies are available in [43, 44]. Our previous work in [14] employs a lightweight copula-based approach [45] to generate synthetic data, which is also adopted in SereWay.

**Fault Diagnostics in Railway IIoT**.

Anomaly detection pipelines in the railway domain use support vector machines [46–48], Kalman filters [49], optimization [50, 51], methods based on other feature analysis-based [52–54], and Deep Learning (DL) pipelines [53, 55–57]. The work in [58] proposes a fault-detection system for training sensor data that combines cloud-based pre-training with online learning strategies for local adaptation of detectors at the edge. These studies detect faults in components such as stators, rotors, and bearings. Our work clusters the synthetic sensor probes in [14] to create a controllable multi-modal sensor data stream. Recent reviews on fault detection over train sensor data are presented in [59, 60].

**Security Benchmarking in Railway IIoT.**

Despite the increasing digitalization of railway systems, the cybersecurity of railway IIoT remains significantly less explored than fault tolerance and operational reliability. Kour et al. [61] underscores the heightened vulnerability of contemporary rail infrastructures to cyber threats, highlighting the need for comprehensive

cybersecurity frameworks. However, the current research is predominantly conceptual, offering few practical implementations, particularly those that employ Artificial Intelligence (AI)-based methods for threat detection and mitigation. Recent studies in network and system management have proposed promising frameworks that could be adapted to the railway context. For instance, Jullian et al. [62] present a distributed deep learning-based framework for cyber-attack detection in IoT networks, demonstrating high effectiveness in decentralized scenarios. Froehlich et al. [63] propose a secure IIoT gateway architecture based on Trusted Execution Environments (TEEs), which enables robust edge-level intrusion mitigation. Moreover, Bovenzi et al. [34] investigate class-incremental learning in traffic classification using deep neural networks, which is relevant for evolving IIoT environments such as those found in train networks. Similarly, Gioacchini et al. [33] propose a method for transferring traffic embeddings across heterogeneous networks, a technique that could be used to adapt intrusion detection models across trains operating on diverse infrastructures.

Research on benchmarking methodologies for assessing cybersecurity performance in railway IIoT environments is lacking. In the IIoT landscape, the work in [64] suggests a security testbed that checks cyber defenses one step at a time. Alsaedi et al. [65] and Ferrag et al. [66] created realistic, diverse datasets called TON_IoT and Edge-IIoTset, respectively. These datasets are designed to support research on intrusion detection. In [9], DDoShield-IoT is presented, an AI-powered framework designed to detect and mitigate DDoS attacks in smart environments, showing the potential of DL for real-time protection. However, these solutions are not tailored to the railway domain and do not account for its unique operational and safety constraints. Additionally, Axon et al. [67] point out that there are cybersecurity gaps in IIoT, especially in areas like benchmarking, validation, and risk control. This highlights the need to create specific tools for this field. One possible approach is to adapt the safety benchmarking framework introduced by Blumenfeld et al. [68] for railways, extending it to include cybersecurity aspects such as intrusion resilience, threat response efficiency, and adversarial robustness.

Considering the state-of-the-art, it can be noted that there is a lack of railway-specific testbeds allowing the benchmarking of security and reliability ML-driven solutions, which reflect real-world operational and threat scenarios in railway IIoT environments.

## 3 Background

Herein, the main features of the European Rail Traffic Management System, along with the generation of railway diagnostic data, are presented to better specify the context in which the approach proposed in this paper has been developed.

### 3.1 European Rail Traffic Management System

The ERTMS aims to revitalize the railway sector by establishing a single, interoperable signaling and speed control system across Europe [69]. This standardization aims to replace the multitude of current national train control systems, thereby reducing

purchasing and maintenance costs, increasing train speeds and infrastructure capacity, and enhancing safety in rail transport. As per the fourth version of the ERTMS, the framework comprises three primary elements: the European Train Control System (ETCS), the Global System for Mobile Communications – Railway (GSM-R), and the Automatic Train Operation module (ATO) [70].

The ETCS is the core system of the ERTMS blueprint, functioning as both a train protection system (Automatic Train Protection - ATP) and a cabin signaling system. Its primary function is to ensure safe train operation by continuously supervising the train's speed and movement authority based on trackside signaling information. The system intervenes by applying emergency braking if the train exceeds permitted limits or encroaches upon unauthorized areas. Instead, the GSM-R is a dedicated radio communication system for railways, providing a secure and reliable channel for both voice communication between train drivers and signalers, as well as data communication for ETCS. The ATO module is the third key component of ERTMS, designed to automate train operation, including traction and braking.

### 3.2 Railway Diagnostic Data

The data used in this paper consists of diagnostic logs obtained from a collaboration with an Italian railway company. The diagnostic logs contain approximately 60,000 diagnostic sensor and event signals collected over one week of high-speed train operations, which report key aspects of an ERTMS/ETCS-controlled train environment. The ERTMS-compliant trains continuously exchange data (such as braking commands, communication status, and power metrics) to ensure safe operations. The data mirror these dynamics by including event descriptors (e.g., inter-component communication failures or unexpected traction cuts) alongside numerical measurements (voltages, currents, etc.), resembling the signals transmitted between onboard and trackside systems.

The sensor probes generated by the nodes in SereWay are a specialized version of our previous work on realistic train data generation [14]. More specifically, synthetic numerical train data is generated in each node concerning the braking system (e.g., cylinder pressures for multiple cars), power supply diagnostics (battery, line voltages, and current), and other vehicle-wise operational parameters such as speed, general power states, emergency braking, fault aggregation, status probes from the ERTMS [70] status, among others.

To better simulate faults or potential safety-critical issues in the ERTMS/ETCS system, the events in our real traces were manually classified, leveraging domain knowledge to separate records related to normal operations from those regarding faulty conditions. Some of the most relevant events identified in the dataset are described in Table 1. Classifying events into normal and anomalous permits implementing a high-level behavioral emulation of the ETCS.[1] More details on the original dataset's synthetic generators are available in [14, 71].

[1] Further research could involve a more detailed mapping of the dataset's specific signals and event codes to the technical specifications of ERTMS to gain an even deeper understanding of their interactions.

**Table 1** Some of the most representative events reproduced in our simulation

| Normal Event | Description |
|---|---|
| `Manual braking command active` | This indicates a braking command initiated by the driver, a standard operational procedure |
| `Signalling system requires service braking` | The signaling system is requesting maximum service braking, a core function of ATP |
| `Standby state activated` | The train is not actively running but is powered on, typically waiting at a station or in a depot |
| `DC pantograph raised on car X` | This is normal for the trains analyzed in this work, which are assumed to operate on a DC power line |
| `RCB/MCB opening command received` | The driver requests to open the circuit breakers (Rapid vs. Main Circuit Breaker). A normal operational procedure |
| Anomalous Event | Description |
| `Pressure below threshold in car X` | A low pressure indicates a potential issue with the compressed air system in that specific car |
| `No closed Circuit Breakers found` | If Circuit Breakers are not closed, it suggests a problem with the train's power distribution system |
| `Communication Error with CCU in car X` | Communication losses with Control and Command Units (CCU) can impede the proper functioning of the TCMS |

## 4 Proposal

This section starts with an explanation of the task related to synthetic data generation, followed by a presentation of the system's architecture and the corresponding implementation details.

### 4.1 Synthetic Data Generation

SereWay uses the same real ETCS traces of Open-fari but refines the synthetic data generation pipeline to account for a more heterogeneous and fine-grained simulation of a fleet of high-speed trains. The main idea is to enhance the ability to address the complexity of real-world railway operations under ERTMS. For example, different trains may be used on various routes, external weather conditions, passenger densities, speed patterns, and operational schedules, which may influence the distribution of diagnostic signals in each simulated train.

To mimic such heterogeneity, after separating normal and anomalous signals in the original diagnostic logs, the traces were clustered using density-based techniques over the corresponding UMAP (Uniform Manifold Approximation and Projection) manifold approximations [72]. A suitable number of clusters was found for each trace, maximizing the Silhouette score [73], which evaluates clustering quality by measuring cluster separability and consistency. Finally, multivariate Gaussian copulas [74] were fitted to each cluster in both normal and anomalous traces, and the parameters of the trained copulas were saved for sampling purposes. More specifi-

cally, given a multivariate dataset of real train-sensor data, $\mathrm{X} \in \mathbb{R}^{n,d}$, SereWay uses the *Data-cebo*'s *Copulas* library to perform parametric estimation of the marginal distribution of each sensor stream. These estimated marginals are then transformed to uniform variables using the Probability Integral Transform (PIT), and subsequently mapped to latent Gaussian variables via the inverse standard normal CDF, to obtain a dataset of multivariate Gaussian latent samples, as shown in Eq. 1:

$$x_{ij} \xrightarrow{\text{PIT}} u_{ij} = \hat{F}_j(x_{ij}) \xrightarrow{\text{Gaussianization}} z_{ij} = \Phi^{-1}(u_{ij}) \tag{1}$$

where $x_{ij}$ is the original data entry (row *i*, variable *j*), $\hat{F}_j$ is the estimated marginal Cumulative Distribution Function (CDF) of variable *j*, $\Phi^{-1}$ is the inverse standard normal CDF, $u_{ij} \in [0, 1]$ are uniform values, $z_{ij} \in \mathbb{R}$ are the latent Gaussian scores.

From these datasets, a correlation matrix $\hat{\Sigma}$ is estimated using Maximum Likelihood Estimation (MLE):

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^\top, \quad \bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i, \quad z_i = \begin{bmatrix} \Phi^{-1}(\hat{F}_1(x_{i1})) \\ \vdots \\ \Phi^{-1}(\hat{F}_d(x_{id})) \end{bmatrix} \tag{2}$$

And, finally, $\hat{\Sigma}$ is saved and used at inference time to generate synthetic multivariate train-sensor samples $\hat{\mathrm{x}} \in \mathbb{R}^d$ that follow the same estimated marginal distributions and, at the same time, a fitted Gaussian pair-wise Covariance:

$$z \sim \mathcal{N}(0, \hat{\Sigma}), \quad u_j = \Phi(z_j), \quad \hat{x}_j = \hat{F}_j^{-1}(u_j), \quad \text{for } j = 1, \ldots, d \tag{3}$$

Notice that our sampling method is conditioned on clusters of similar signals, both for anomalous and normal events. The real clusters' cardinalities were normalized to resemble the real distribution of clusters, and a multinomial distribution was created using these cardinalities. The multinomial cluster distribution is a categorical distribution that the train signal emulator periodically samples to determine the Copula to use for the next event generation. The specific cluster generation log-probabilities used in our experiments are reported in Figure 1. Finally, the time lapse between

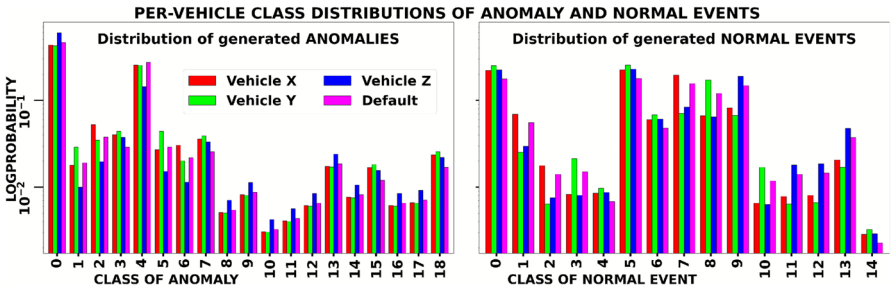


**Fig. 1** Class-specific data distributions for synthetic data generation used in our experiments

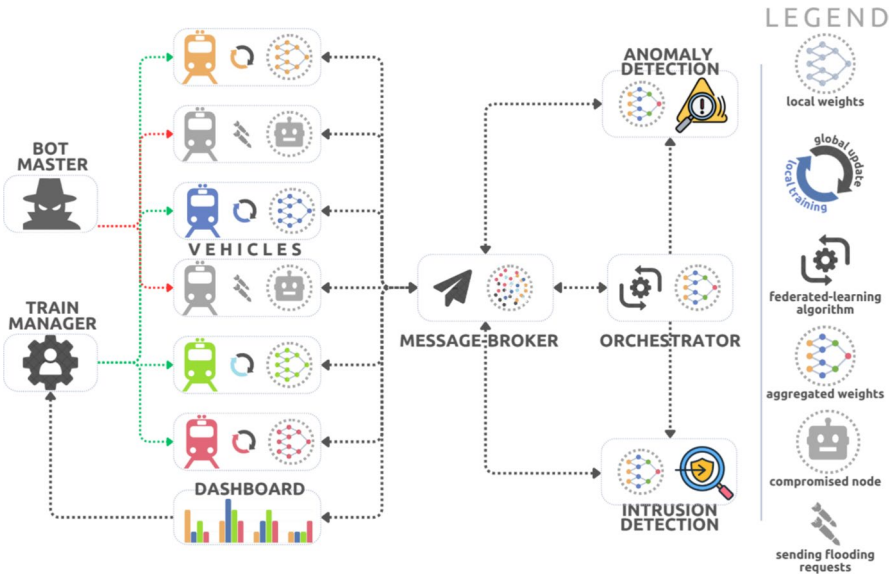


**Fig. 2** Modular view of the SereWay testbed. A Train Manager node controls the vehicles in the simulated railway network, while a Bot Master can infect various nodes to conduct a DDoS attack in the network. A middleware-based message broker permits the training of global anomaly detection and intrusion detection modules using a federated learning algorithm implemented by the orchestrator. Instead, the dashboard node provides comprehensive monitoring of the overall network health status to the Train Manager

one event generation and the other follows the realistic inter-arrival time distribution distilled from the real traces. Specifically, log-normal distributions were fitted to the normal and anomalous event inter-arrival times according to the real traces. The corresponding parameters were saved and used during the simulation. By doing such modeling, SereWay allows varying the cluster distributions of each train using a configuration file, allowing the creation of the desired scenario for the benchmark.

### 4.2 Architecture

The SereWay testbed contains eight types of nodes. The first five are already available in the original version of the testbed, and they have been extended here to implement the new data generation process described earlier, as well as to implement an online learning approach[2] (discussed later). Instead, the latter three nodes extend the testbed with the security-related dimension. All the nodes in the SereWay testbed are depicted in Figure 2 and described in the following.

**Message Broker.** The sensor probes in each vehicle, the health-status probes, the training metrics, and other information, such as global module weights, are communicated between nodes through the message broker. This broker uses a publish-subscribe middleware, and the multiple topic names ("sensor probes" and "health status") follow a naming template that enables the orchestrator to subscribe to every vehicle-

[2] The solution implemented in Open-fari allows the evaluation of only offline learning scenarios.

specific topic using expression matching. Also, the dashboard node subscribes to all the topics related to vehicle training statistics. Note that the message broker mirrors the communication infrastructure within ERTMS, which needs reliable and timely data exchange between trackside equipment and onboard units.

**Vehicle.** Vehicle nodes simulate the functionality of an ETCS onboard unit, i.e., they represent single trains in the railway network, responsible for producing sensor probes and training local anomaly detector modules. Each vehicle is trained with data sampled from a specific multinomial distribution where each category corresponds to a different cluster of sensor probes. Additionally, vehicle nodes report their health status and training metrics to the message broker.

**Train Manager.** The manager node is responsible for instantiating vehicles and managing their training loops. The train manager also starts and stops each vehicle's data generation and training processes. Furthermore, the manager node initiates and completes the metric collection and federated learning processes. Note that the Train Manager's role in instantiating and managing vehicles, their training loops, and the corresponding data generation processes can be associated with the functions of a Traffic Management System (TMS) in a real railway network: the TMS is responsible for scheduling, dispatching, and monitoring trains, and the Train Manager performs similar orchestration tasks within our simulated environment.

**Federated Learning Orchestrator.** The Federated Learning Orchestrator collects the local training metrics and models' weights from the message broker. It is responsible for aggregating these weights using an FL algorithm and redistributing the aggregated weights to the participating nodes to enhance the convergence of the anomaly detection modules. FL can enable a single train to potentially benefit from a global model that aggregates discriminative-generative criteria from the whole fleet [75].

**Dashboard.** The dashboard node collects and presents metrics from both the vehicle nodes and the orchestrator, providing visual insights to the administrator regarding the vehicle-wise and global convergence processes. The metric collection and reporting processes are asynchronous, involving the single local training processes in the vehicles and the centralized processes in the management nodes (i.e., the orchestrator and the security manager). By imposing such a temporal decoupling, the dashboard can report metrics from multiple sets of nodes that either join or leave the experiment at different moments. By providing visual insights into a vehicle and global convergence processes, the dashboard makes the Driver Machine Interface (DMI) function in the train cab and the monitoring systems used in railway control centres.

**Bot Master** The Bot Master is an external node that is intended to have access to a predefined set of nodes in the railway network and issue flooding commands to these nodes. The idea is to emulate a scenario where an external attacker takes advantage of vulnerable nodes in the railway network to launch a DDoS attack. The flooding target can be any node in the network, including the FL orchestrator and Broker nodes, which are the most critical ones. In addition, attacks with different intensities can be performed, with a frequency ranging from 1Hz to 10kHz.

**Bot Nodes** The Bot Nodes are a specific set of vehicle nodes equipped, for simulation purposes, with backdoor access to the Bot Master. They represent the vulnerable

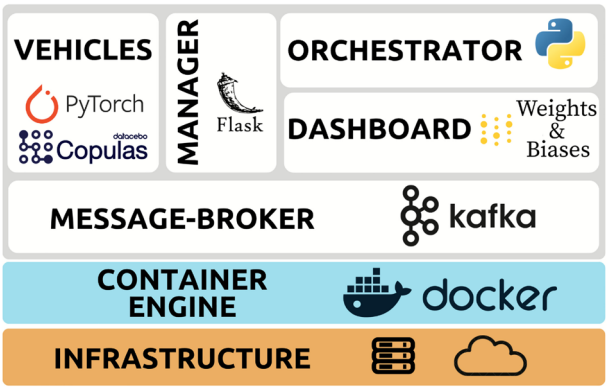


**Fig. 3** Technological stack for the current implementation of SereWay

nodes of the railway network, which obey the flooding commands of the Bot Master, generating flooding UDP attacks.

**Security Manager** The Security Manager node implements an ML-based discriminator that receives health status metrics from the broker and identifies compromised nodes in the railway network. Based on its inferences, the Security Manager also requests attack mitigation actions to the Train Manager node, which are supposed to stop flooding actions on the infected nodes. Mitigation, in this case, is simulated by stopping the attack thread. Conceptually, this can be seen as the restarting of the compromised train system or subsystem. However, in the current code implementation, this happens without any "extra" latency. Each train, when started, exposes a microservice that acts as a backdoor for the Bot Master. Through an HTTP request, the Bot Master can initiate an attack from a train. Similarly, when the Security Manager detects that a node is under attack, it uses the corresponding node's microservice to shut down the attack.

### 4.3 Implementation Details

SereWay uses process virtualization to reduce the computational overhead of the simulations. Namely, the nodes in Figure 2 are implemented using lightweight virtualization, i.e., containers. As shown in Figure 3, which sketches an overview of the technological stack over which SereWay is implemented, the container engine in SereWay is *Docker*,[3] and each container is given a separate portion of system resources to enhance the realism of the simulation. The SereWay testbed can be deployed automatically using the *DockerCompose* file, which sets up the container network, downloads, configures, and builds the third-party images. The nodes are implemented as follows:

- The communication middleware, i.e., the **Message Broker**, is implemented by

[3] https://www.docker.com/

a container-based *Apache-Kafka*[4] cluster composed of a Zookeeper node and a Kafka node. The broker holds the active topic list, subscribers' offsets, and message backups. The Zookeeper node ensures that the single Kafka broker functions correctly and maintains its state[5] The current version of the testbed uses the official Kafka and Zookeper Images from Confluent Inc., versions 7.2.15. These are mounted as Services by the DockerCompose file inside a bridge network for isolated container communication.

- The **Train Manager** runs in the host system and is implemented using the *Flask*[6] web application framework. Also, the **Bot Master** runs on the host and is accessible through a separate panel in the former web application. The Train Manager creates, starts, and stops vehicles, their training cycles, and health monitoring threads, as specified by the parameters in the testbed configuration files. To achieve this, the manager utilizes the *Hydra* library and the Docker Python SDK. The Bot Master instead uses an attack microservice implemented on the vehicles using *FastAPI* to issue attacks.
- Each **Vehicle** node is implemented by two running containers, i.e., a vehicle producer and a vehicle consumer. The producer emits normal and anomalous sensor probes from the train alongside the health status probes, while the consumer trains the local anomaly detection module. The producers use the *Psutil*[7] library to implement a cyclic thread that sends CPU, RAM, and network-related metrics. Consumers use PyTorch neural modules for anomaly detection-related inferences and training. The architecture of these modules is also read by the manager node from the configuration files and sent to the consumers through the Docker SDK when the user requests vehicle creation through the manager GUI.
- The **FL Orchestrator** and the **Security Manager** are also Docker containers, which run the ML-based reliability and security solutions under test, i.e., the target of the benchmark. The Security Manager subscribes to the health-probe-related topics through the Broker, instantiates the neural modules for online intrusion-detection inference, training, and evaluation; and implements data rebalancing techniques through resampling to accelerate convergence. The FL Orchestrator instantiates a copy of the anomaly-detection neural modules in the vehicles for aggregation purposes: rather than training, it uses FL algorithms to combine the optimized weights, which are sent to it through the Broker, and sends the combined weights back to the vehicles. The aggregation algorithm and period are also settable through the configuration file.
- The **Dashboard** node is a container that registers all the relevant topics in the Kafka middleware and pushes back the received messages to a cloud-hosted *Weights and Biases* (W&B)[8] dashboard utilizing the corresponding API. Among many others, the training and evaluation performance metrics are sent to the

[4] https://kafka.apache.org/

[5] In large-scale scenarios, more Kafka nodes can be added. In this case, ZooKeeper will play a more critical role in managing and coordinating the cluster.

[6] https://flask.palletsprojects.com/en/latest/

[7] https://pypi.org/project/psutil/

[8] https://wandb.ai/

W&B dashboard during each experiment. Also, the running distributions of the clustered synthetic data generated by each vehicle are visualized in the dashboard to better assess the effect of data imbalance locally and after the aggregation rounds.[9]

Each train in the simulated network is equipped with a **high-level monitoring** daemon that periodically reports resource usage metrics for the corresponding container, e.g., CPU and memory usage, latency, and inbound/outbound network throughput. These indicators are collected by the Security Manager and serve as a basis for **attack detection**, identifying vehicles' healthy/compromised status. Indeed, DDoS attacks (which are emulated in our testbed) are expected to affect the resource usage of the target nodes. Note that the sensor probes, health status updates, and training metrics flowing through the broker represent the data exchanged between onboard systems (like ETCS and potentially ATO) and ground control in an ERTMS environment.

## 5 Case Study

Modern railway infrastructure uses advanced communication systems to ensure safe and efficient operations. The evolution toward increasingly connected systems, including interconnected rail vehicles, IIoT devices, and distributed control networks, introduced new vulnerabilities. In this context, sensor-based predictive maintenance reduced unexpected failures, but it also made rail systems dependent on operational data processing. Moreover, the increased attack surface has made rail networks a target for cyber threats, including DDoS attacks, which aim to compromise the availability of critical services.

In this perspective, this section analyzes the SereWay testbed via a case study that resembles a train network composed of six trains, all infected by the BotNets' backdoor. The aim here is to run a benchmark and provide an overview of the potentialities of SereWay at evaluating ML-based solutions for addressing the challenges of operational failure detection and cyber threat protection.

### 5.1 System Description

SereWay has been deployed on six rail vehicles, with each vehicle equipped with an operational sensor network that collects critical parameters, including vibration, temperature, power consumption, and health status metrics. As per the SereWay architecture, these parameters are analyzed by a security and reliability system consisting of two main components: (i) the centralized *Security Manager*, which is responsible for evaluating security threats across all rail vehicles, and (ii) the *FL-based anomaly detector* dedicated to operational anomaly detection and diagnostics.

The Security Manager collects statistics from all nodes and determines whether a specific node is under attack, while the FL-based detector operates independently

[9] An example series of experiments and the corresponding tracked metrics can be seen at https://wandb.ai/jfcevallos/SEREWAY

to identify anomalies related to operational malfunctions. It is important to note that the deployed system follows a fully online approach, avoiding static datasets and enabling dynamic adaptation of the model to varying operating conditions. This type of learning renders it incremental and utilizes ongoing update mechanisms that allow new anomaly and intrusion detection without service interruption.

The *Bot Manager* has been configured to emulate a botnet that checks for benign nodes within the network at configurable intervals, set to a default of 8 s. Upon identifying these nodes, the Bot Manager instructs them to flood the messaging broker, i.e., Zookeeper. However, the target broker and flooding frequency can be dynamically configured via a configuration file.

Based on this setup, we conducted experiments using diverse attack strategies, including low-frequency Sybil attacks and varying flooding intensities, enabling us to assess the system's robustness across different threat levels. Specifically, we considered 5 different levels of intensity, i.e., 1Hz, 10Hz, 100Hz, 1000Hz, and 10000Hz, which are executed during 20-minute runs of SereWay.

### 5.2 Solutions Under Test

The goal of SereWay is to provide an easy-to-use benchmarking testbed for functional anomaly detection and ML-based cyber threat detection models in the railway IIoT domain. In this case study, the benchmark subjects are represented by the solutions implemented in the Security Manager and the FL-based anomaly detector. The Security Manager continuously monitors the network state, employing several critical metrics, such as network bandwidth (both inbound and outbound), network latency (round-trip time, RTT), and hardware resource utilization (i.e., CPU and memory). The FL-based detector leverages decentralized training sessions happening across multiple nodes. Each vehicle has its own local model instance, which processes the collected diagnostic data, computes updates to the anomaly detection model, and shares only the learned parameters with the central server. Then, the server aggregates updates from multiple vehicles to improve the generalization of the global model for anomaly detection. Overall, the detection process under test consists of the following steps:

1. **Immediate signal processing**: The local anomaly detector collects data to test any neural model without going through traditional preprocessing or bulk normalization procedures.
2. **Inference and online classification**: The neural networks classify the events online, counting on two independent supervisory signals: one for operational anomalies/diagnostics and one for each train's compromised/healthy status.
3. **Federated aggregation of updates**: each train vehicle is equipped with a local neural module that is also trained and can periodically transmit weight changes to the central Orchestrator server, which updates the global anomaly detection model (using the FedAvg method [76]) and redistributes the new parameters to the local detectors.
4. **Security Manager Evaluation**: The centralized Security Manager continuously monitors statistics from all nodes and evaluates whether a vehicle is under attack.

Its decisions are evaluated through a reward function that quantifies the effectiveness of intrusion detection.

In the considered case study, we leveraged MLP models for both intrusion and anomaly detection. Therefore, we have a two-stream MLP module, where each stream performs a binary classification. The first classifier identifies operational anomalies, distinguishing between normal conditions and mechanical or electronic malfunctions. The second classifier detects security threats by analyzing specific sensor probes on rail vehicles. In this context, the online approach allows the system to react immediately to events, continuously updating the model to improve detection accuracy and reduce false positives.

### 5.3 Results

We validated the proposed framework's ability to simulate a real-world railway environment and to benchmark security and reliability solutions by conducting the diverse tests described earlier, which are based on real operational scenarios and include the considered attack simulations.

To assess the performance of the detection system, we employed the most common evaluation metrics in the anomaly detection context, namely Accuracy, Precision, Recall, and F1-Score. In addition, the performance of the Security Manager has also been evaluated with a rewarding mechanism, which assigns a highly negative reward to the SM whenever a false-negative inference is made (i.e., when the SM fails to detect an infected train), a low negative reward each time the SM signals a false-positive, and a low positive reward is given to the system each time an attack is correctly detected. In contrast, no reward is given when the SM makes true-negative inferences. Additionally, we measured the Average Mitigation Latency, which is the time required by the system to detect and defensively react to an attack. By doing so, these abstract rewards simulate the health of the whole TMS, including its anomaly and intrusion detection functions.

The results of our online intrusion detection tests are depicted in Figure 4, which depicts 5-steps running averages for the obtained Accuracy, Precision, Recall, and F1-score, evaluated over a 20-minute run for each flood packet frequency. The results

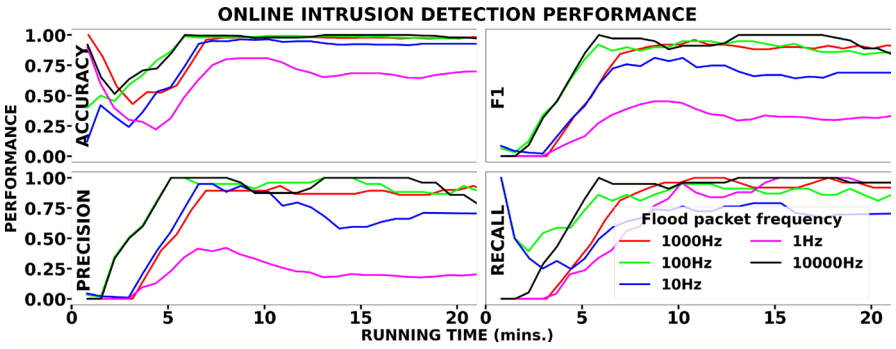


**Fig. 4** DDoS attack detection performance

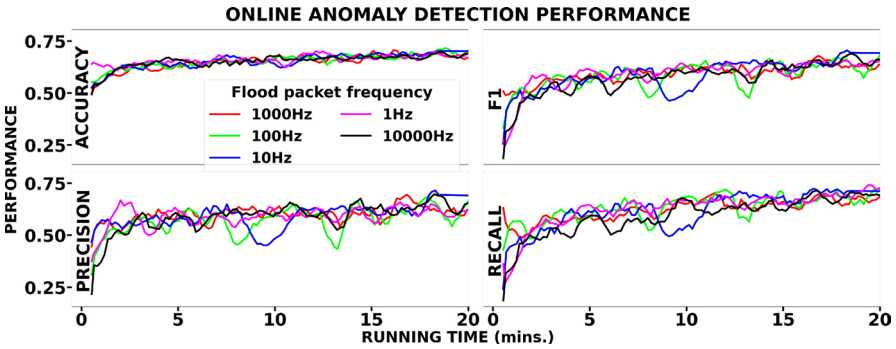


**Fig. 5** Anomaly detection performance

show promising performance, with all metrics showing improvement over time, as expected, thanks to online learning. Specifically, for higher flooding frequencies (1000Hz and 10000Hz), the Accuracy and F1-score rapidly approached and maintained values close to 1.0, suggesting high effectiveness in identifying intrusions under intense attack conditions. Precision also reached near-perfect levels for these frequencies, minimizing false positives, which is critical for security systems. Recall was similarly high, indicating a strong ability to detect most intrusion attempts. For lower flooding frequencies (1Hz, 10Hz, and 100Hz), the metrics showed a more gradual increase but still stabilized at high levels (typically above 0.75 for Accuracy and F1-Score), demonstrating the system's capability to detect intrusions even at lower intensities, albeit potentially with a slightly longer initial learning phase.

In contrast, the online classification of synthetic functional anomalies generated by the train nodes showed a different performance profile, as it can be inferred from Figure 5. While the system demonstrated the ability to classify anomalies online, the performance metrics (i.e., Accuracy, F1-score, Precision, and Recall) evaluated over a 20-minute run for each flooding frequency, generally stabilized at lower levels compared to intrusion detection. Accuracy typically ranged between 0.6 and 0.7, and the F1-score between 0.5 and 0.7 after an initial learning phase. Precision and Recall showed similar trends, indicating a moderate ability to correctly classify anomalies without excessive false positives or negatives. Notably, the performance in functional anomaly detection appeared less sensitive to the flood packet frequency compared to intrusion detection, with the metrics for different frequencies clustering more closely after the initial learning period. The lower overall performance in functional anomaly detection could be attributed to the inherent complexity of distinguishing subtle functional deviations from normal operational behavior or the characteristics of the synthetic data used for evaluation.

In addition to these evaluations, SereWay also allows to analyze the Security Manager detection performance based on the rewarding mechanism described earlier. Figure 6 shows a 20-step running average of both the intrusion detection rewards and mitigation latency.[10] It can be noted that the Security Manager initially faced chal-

[10] Note that in the current implementation of SereWay the mitigation actions take place with no extra latency with respect to detection; therefore, the plots in Figure 6 (left) can be seen as detection latency.

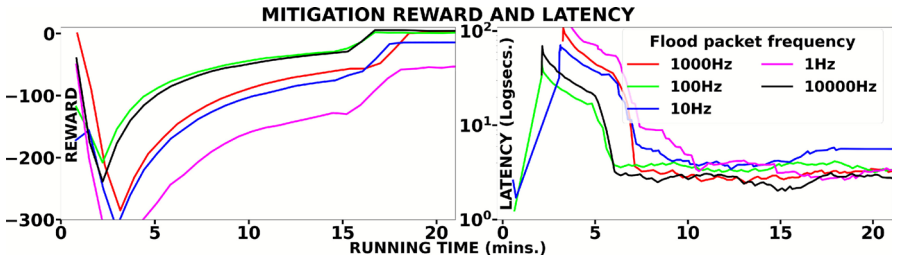


**Fig. 6** Intrusion detection performance based on reward mechanism (left) and attack mitigation latency (right)

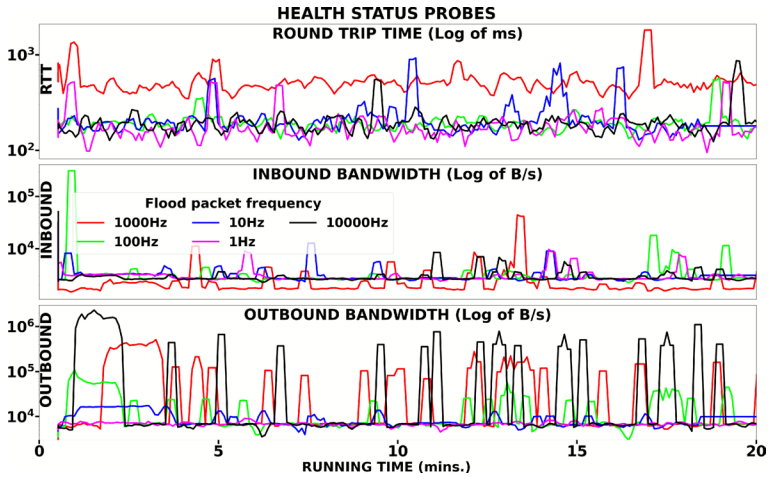


**Fig. 7** Probes of the round trip time (up), inbound (middle), and outbound (down) bandwidth for a single vehicle during simulations

lenges in accurately detecting attacks. Indeed, we notice an initial significant decline in the mitigation reward (Figure 6 - left), which dipped to a low of approximately -300. However, as it learns online, it quickly adapts, improving its performance and gradually bringing the reward back up to stabilize near zero. This sub-optimality obeys the current challenging rewarding scheme: false negatives incur a -10 penalty, false positives -4, true negatives get 0, and true positives earn +2. Considering that attack events are less frequent than benign situations. Meanwhile, the mitigation latency (Figure 6 - right) follows a different pattern. Initially, it spikes as the system takes longer to respond, but then it drops rapidly and stabilizes. This shows that the SM not only improves at detecting attacks, but it consequently learns to contain them more quickly. The log-scale view makes it clear that, regardless of the attack intensity, the system consistently brings mitigation times under control, proving its adaptability and efficiency in handling different levels of threats.

Finally, Figure 7 gives a view of the main health status probes collected from vehicles during the online learning experiments regarding anomaly and intrusion detection. As for the other figures, each line in Figure 7 corresponds to a different

configuration of flooding attack in terms of the frequency of packets that the flooding bot sends to the victim, where a 3-step running average is plotted for each metric. As can be seen from the upper plot in this figure, higher frequency attack scenarios imply higher network latency, measured in terms of round-trip time. Inbound bandwidth (middle plot) is instead unaffected by the flooding commands, which are UDP-based. Instead, outbound bandwidth (lower plot) is clearly a discriminative feature of the node being under attack. Regarding the outbound bandwidth, notice how initial attacks last longer, while later attacks are more rapidly mitigated as the Security Manager learns to discriminate and mitigate them. Finally, notice also how different higher flooding packet frequencies correspond to higher attack peaks in this plot.

It should be noted that the presented results and their discussion mainly aim at demonstrating that SereWay can easily enable the benchmarking of ML-based anomaly and intrusion detection approaches in a realistic railway testbed. Further, the discussed metrics are just an example of metrics that can be easily analyzed through the dashboard available in SereWay.

## 6 Discussion

### 6.1 Threats to Validity

As with any study proposing a new approach, concerns may arise regarding the validity and generalizability of the proposal and its results. We discuss them, based on the aspects of validity listed in [77].

*Construct Validity*. Our study is built around the need for SDG instruments tailored to the railway domain, which poses a significant barrier to entry for research on ML-driven solutions for both reliability and security in the railway IIoT domain. Therefore, we proposed SereWay, an open-source testbed for benchmarking the detection and mitigation of anomalies and cyber intrusions within a simulated European Rail Traffic Management System. Our implementation leverages consolidated technologies, like Docker containers, Kafka message broker, Flask web application, and Weight and Biases dashboard. Moreover, our synthetic data generation approach relies on real-world diagnostic data generated over one week of high-speed train operations. In contrast, our attack generation module, i.e., the Bot Master, addresses realistic attacks, such as DDoS attacks, which are relevant in the railway IIoT domain. The analysis is based on a case study aiming at showing the potentialities of the proposal to emulate realistic railway scenarios to facilitate the benchmarking of ML-based security and reliability instruments for the railway domain. Therefore, the case study conducts an experimental campaign using SereWay to evaluate an FL-based system that aims to detect anomalies from both security and reliability perspectives. FL is an example of ML-based solutions that can be evaluated in the railway scenario by using our proposal. We selected it for our case study, since FL can be useful for a train fleet, allowing a single train to potentially benefit from a global model aggregating discriminative-generative criteria from the whole fleet.

*Internal and Conclusion Validity*. Findings have been inferred through various experiments that analyze both the reliability and security detection features of the

solution under test. We replicated the experiments under various configurations of packet frequency to emulate different scenarios. We evaluated various metrics; moreover, the analysis has been supplemented by manual investigations of data generated by the proposal (e.g., analyzing the system probes used for attack detection). Overall, this mitigates internal validity threats and provides a reasonable level of confidence in the conclusions.

*External Validity*. Our proposal should be easily applicable to other similar ML-based approaches for detecting anomalies and intrusion detection. Given the widespread adoption of these technologies, practitioners can easily deploy the solutions to test within our proposal and perform their evaluation (from a reliability and security perspective) in a simulated European Rail Traffic Management System. It can be observed that the current implementation does not combine data from both the reliability and security aspects, but rather analyzes them separately. As future work, we will investigate considering both types of data sources simultaneously to enable practitioners to utilize the proposal for evaluating solutions that aim to perform anomaly and intrusion detection in an integrated manner. Finally, the details provided can reasonably support the replication of our study by other researchers and practitioners. Most notably, we made SereWay available to the community.

### 6.2 Contributions for the ERTMS Ecosystem

Based on the modelization and implementation of security and reliability benchmarking, some insights obtained in this study also highlight some potential future contributions that can be useful for the ERTMS ecosystem:

- While our current implementation of the communication between nodes does not emulate the specifics of the GSM-R, our framework could be positioned as a testbed for future integration with FRMCS (Future Railway Mobile Communication System), the successor to GSM-R, which will be vital for the digitalization of rail transport [78].
- While not a direct component of current ERTMS, our Federated Learning Orchestrator showcases a potential future direction for enhancing the resilience and security of railway systems. By aggregating knowledge from individual trains without centralizing all data, SereWay addresses privacy concerns and creates a more robust anomaly detection system that can learn from the collective experience of the fleet. This is highly relevant to the cybersecurity aspects of ERTMS, where distributed intelligence can be beneficial for detecting and responding to threats.
- Introducing a Bot Master and Bot Nodes addresses the growing concern of cybersecurity threats targeting critical infrastructure like railway systems operating under ERTMS. These nodes simulate realistic attack vectors, such as DDoS attacks, which could potentially disrupt train operations, communication, and safety systems within the ERTMS framework.

## 7 Conclusions

This work presented SereWay, a testbed for benchmarking the detection and mitigation of anomalies and cyber intrusions within a simulated ERTMS environment. SereWay uses real-world train sensor data and focuses on the interplay between security and reliability through Federated Learning. Online learning cyber-defensive benchmarking is the main contribution by which SereWay positions itself as a more complete benchmarking instrument for research compared to recent literature. It aims to contribute to railway safety and security in the context of current European standards and their upcoming transformations.

Further extensions of our SereWay platform can impact two main directions: increasing attack heterogeneity and introducing adversarial attack capabilities. Regarding the first direction, beyond DDoS attacks, other lower-frequency attack vectors may target real-world IIoT networks. New attacking nodes, that emulate scanning, Man-in-the-Middle, Command&Control, and other Malware and attacks, could be implemented in SereWay. The second direction targets adversarial threats: as defensive systems increasingly rely on machine learning to detect and mitigate intrusions, attackers are likewise turning to statistical-learning techniques to design Sybil and other adaptive attacks that intentionally evade detectors. To support research on such dynamics, AI-driven attack modules can be added to SereWay in order to generate adaptive, adversarial benchmarks for evaluating and hardening defensive models.

**Acknowledgements** This work was supported in part by the SERENA-IIoT project, which has been funded by EU - NGEU, Mission 4 Component 1, CUP J53D23007090006, under the PRIN 2022 (MUR -*Ministero dell'Università e della Ricerca*) program (project code 2022CN4EBH), and by the project SERICS (project code PE00000014), under the NRRP MUR program funded by the EU - NGEU.

**Author Contributions** AR: Conceptualisation, manuscript writing, manuscript reviewing, fund acquisition. RDC: Conceptualisation, manuscript writing, manuscript reviewing, fund acquisition. JFCM: Conceptualisation, manuscript writing, code implementation. SDV: Conceptualisation, manuscript writing, code implementation. SS: Conceptualisation, manuscript reviewing. DC: Conceptualisation, manuscript reviewing. ACP: Conceptualisation, manuscript reviewing.

**Funding** This work was supported in part by the SERENA-IIoT project, which has been funded by EU - NGEU, Mission 4 Component 1, CUP J53D23007090006, under the PRIN 2022 (MUR -Ministero dell'Università e della Ricerca) program (project code 2022CN4EBH), and by the project SERICS (project code PE00000014), under the NRRP MUR program funded by the EU - NGEU.

**Data Availability** The real data used to train the generative models used in this work is proprietary, based on a collaboration with an Italian Railway company, and not publicly available. On the other hand, our optimized copulas for synthetic data generation are available in the project's repository.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

## References

1. Młyńczak, J., Toruń, A., Bester, L.: European rail traffic management system (ertms). In: Intelligent Transportation Systems–Problems and Perspectives, pp. 217–242 (2015)
2. Long Li, Li, L.: An introduction to the european rail traffic management system. In: International Conference of Logistics Engineering and Management, pp. 488–492 (2012)
3. Parlement Européen et Conseil de l'Union Européenne: Directive (UE) 2016/797 du Parlement Européen et du Conseil du 11 mai 2016 relative à l'interopérabilité du système ferroviaire au sein de l'Union européenne (refonte). Texte présentant de l'intérêt pour l'EEE (2016). https://eur-lex.europa.eu/legal-content/FR/TXT/PDF/?uri=CELEX:32016L0797&rid=1
4. Hu, J., Hu, J., Liu, G., Liu, G., Li, Y., Li, Y., Ma, Z., Ma, Z., Wang, W., Wang, W., Liang, C., Liang, C., Yu, F.R., Yu, F., Fan, P., Fan, P.: Off-network communications for future railway mobile communication systems: challenges and opportunities. IEEE Communications Magazine, pp. 1–7 (2022)
5. González-Plaza, A., González-Plaza, A., Cantarero, R.G., Cantarero, R.G., Banda, R.B.A., Banda, R.B.A., Briso-Rodriguez, C., Briso-Rodríguez, C.: 5G based on MNOs for critical railway signalling services: future railway mobile communication system. Appl. Sci. **12**(18), 9003–9003 (2022)
6. Serradilla, O., Zugasti, E., Rodriguez, J., Zurutuza, U.: Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. Appl. Intell. **52**(10), 10934–10964 (2022)
7. Latif, S.: Deep learning for the industrial internet of things (iiot): a comprehensive survey of techniques, implementation frameworks, potential applications, and future directions. Sensors **21**(22), 7518 (2021)
8. Yu, J., Zhang, Y.: Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. Neural Comput. Appl. **35**(1), 211–252 (2023)
9. De Vivo, S., Obaidat, I., Dai, D., Liguori, P.: Ddoshield-iot: A testbed for simulating and lightweight detection of iot botnet ddos attacks. In: 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 1–8 (2024). IEEE
10. Demertzi, V., Demertzis, S., Demertzis, K.: An overview of privacy dimensions on the industrial internet of things (iiot). Algorithms **16**(8), 378 (2023)
11. Lu, Y., et al.: Machine learning for synthetic data generation: a review. arXiv preprint arXiv:2302.04062 (2023)
12. Chui, K.T.: A survey of internet of things and cyber-physical systems: standards, algorithms, applications, security, challenges, and future directions. Information **14**(7), 388 (2023)
13. Chen, Y.-T., Hsu, C.-Y., Yu, C.-M., Barhamgi, M., Perera, C.: On the private data synthesis through deep generative models for data scarcity of industrial internet of things. IEEE Trans. Ind. Inf. **19**(1), 551–560 (2021)
14. Rizzardi, A., Della Corte, R., Cevallos M., J.F., De Vivo, S., Orbinato, V., Sicari, S., Cotroneo, D., Coen-Porisini, A.: Open-FARI: An open-source testbed for federated anomaly detection in the railway IIoT. In: To Appear in Proc. of International Wireless Communications & Mobile Computing Conference (IWCMC 2025)
15. Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W.: A survey on federated learning: challenges and applications. Int. J. Mach. Learn. Cybern. **14**(2), 513–535 (2023)
16. Wang, Z., Wang, Z., Liu, X., Liu, X.: Cyber security of railway cyber-physical system (CPS) – a risk management methodology. Commun. Transp. Res. **2**, 100078–100078 (2022)
17. Kour, R., Patwardhan, A., Thaduri, A., Karim, R.: A review on cybersecurity in railways. In: Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit (2022)
18. CBTC test simulation bench: J. M. Mera, Mera, J.M., I. Gómez-Rey, I. Gómez-Rey, I. Gómez-Rey, Gómez-Rey, I., Gómez-Rey, I., Everardo Rodrigo, Rodrigo, E., E. Rodrigo, E. Rodrigo, Rodrigo, E. WIT Transa. Built Environ. **114**, 485–495 (2010)

19. Kim, S., Won, Y., Park, I.-H., Eun, Y., Park, K.-J.: Cyber-physical vulnerability analysis of communication-based train control. IEEE Internet Things J. **6**(4), 6353–6362 (2019)
20. Xu, J., Chen, L., Gao, W., Zhao, M.: CBTC simulation platform design and study. J. Comput. Chem. **3**(9), 61–67 (2015)
21. Neema, H., Potteiger, B., Koutsoukos, X., Tang, C., Stouffer, K.: Metrics-driven evaluation of cybersecurity for critical railway infrastructure. In: 2018 Resilience week (RWS), pp. 155–161 (2018)
22. Caprino, G.: Train director version 3.9.10. https://www.backerstreet.com/traindir/en/trdireng.php. Accessed on 2023-05-08 (2023)
23. Neema, H., Nine, H., Hemingway, G., Sztipanovits, J., Karsai, G.: Rapid synthesis of multi-model simulations for computational experiments in c2 (2009). Technical report
24. Neema, H., Koutsoukos, X., Potteiger, B., Tang, C., Stouffer, K.: Simulation testbed for railway infrastructure security and resilience evaluation. In: Proceedings of the 7th symposium on hot topics in the science of security, pp. 1–8. ACM, Lawrence Kansas (2020)
25. Teo, Z.-T., Tran, B. A. N., Lakshminarayana, S., Temple, W. G., Chen, B., Tan, R., Yau, D. K. Y.: SecureRails: Towards an open simulation platform for analyzing cyber-physical attacks in railways. In: 2016 IEEE Region 10 Conference (TENCON), pp. 95–98 (2016). ISSN: 2159-3450
26. Open Rails. https://www.openrails.org/. Accessed on 17 March 2025 (2025)
27. Božič, J., Faustino, A.R., Radovič, B., Canini, M., Pejović, V.: Where is the testbed for my federated learning research? arXiv preprint arXiv:2407.14154 (2024)
28. Symeonides, M., Nikolaidis, F., Trihinas, D., Pallis, G., Dikaiakos, M.D., Bilas, A.: Fedbed: Benchmarking federated learning over virtualized edge testbeds. In: Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing, pp. 1–10 (2023)
29. Popovic, M., Popovic, M., Kastelan, I., Djukic, M., Ghilezan, S.: A simple python testbed for federated learning algorithms. In: 2023 Zooming Innovation in Consumer Technologies Conference (ZINC), pp. 148–153 (2023)
30. Popovic, M., Popovic, M., Kastelan, I., Djukic, M., Basicevic, I., Vasiljevic, P.: Micropython testbed for federated learning algorithms. In: 32nd Telecommunications Forum (TELFOR), pp. 1–4 (2024). IEEE
31. Callebaut, G., Van Mulders, J., Ottoy, G., Delabie, D., Cox, B., Stevens, N., Perre, L.: Techtile–open 6g r &d testbed for communication, positioning, sensing, wpt and federated learning. In: 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), pp. 417–422 (2022). IEEE
32. Li, B., Su, N., Ying, C., Wang, F.: Plato: An open-source research framework for production federated learning. In: Proceedings of the ACM Turing Award Celebration Conference - China 2023. ACM TURC '23, pp. 1–2. Association for Computing Machinery, New York, NY, USA (2023)
33. Gioacchini, L., Paoloni, A., Rinzivillo, S.: Cross-network embeddings transfer for traffic analysis. IEEE Trans. Netw. Serv. Manag. **21**(3), 204–215 (2024). https://doi.org/10.1109/TNSM.2023.3329442
34. Bovenzi, G., Palmieri, F., Ficco, M., Castiglione, A.: Benchmarking class incremental learning in deep learning traffic classification. IEEE Trans. Netw. Serv. Manag. **21**(1), 82–95 (2024). https://doi.org/10.1109/TNSM.2023.3287430
35. Fidelis, E.C., Reway, F., Ribeiro, H., Campos, P.L., Huber, W., Icking, C., Faria, L.A., Schön, T.: Generation of realistic synthetic raw radar data for automated driving applications using generative adversarial networks. arXiv preprint arXiv:2308.02632 (2023)
36. Yang, Z., Chai, Y., Anguelov, D., Zhou, Y., Sun, P., Erhan, D., Rafferty, S., Kretzschmar, H.: Surfelgan: Synthesizing realistic sensor data for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11118–11127 (2020)
37. Alzantot, M., Chakraborty, S., Srivastava, M.: Sensegen: A deep learning architecture for synthetic sensor data generation. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 188–193 (2017). IEEE
38. Brophy, E., Wang, Z., She, Q., Ward, T.: Generative adversarial networks in time series: A systematic literature review. ACM Comput. Surv. **55**(10), 1–31 (2023)
39. Yang, Y., et al.: A survey on diffusion models for time series and spatio-temporal data. arXiv preprint arXiv:2404.18886 (2024)
40. Wang, X., : An observed value consistent diffusion model for imputing missing values in multivariate time series. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2409–2418 (2023)
41. Romanelli, F., Martinelli, F.: Synthetic sensor data generation exploiting deep learning techniques and multimodal information. IEEE Sens. Lett. **7**(7), 1–4 (2023)

42. Alabdulwahab, S., Kim, Y.-T., Son, Y.: Privacy-preserving synthetic data generation method for IoT-sensor network IDS using CTGAN. Sensors (Basel, Switzerland) **24**(22), 7389 (2024)
43. Savran, E., Karpat, F.: Synthetic data generation using copula model and driving behavior analysis. Ain Shams Eng. J., 103060 (2024)
44. Silva, D., Leonhardt, S., Antink, C.H.: Copula-based data augmentation on a deep learning architecture for cardiac sensor fusion. IEEE J. Biomed. Health Inform. **25**(7), 2521–2532 (2020)
45. Restrepo, J.P., Rivera, J.C., Laniado, H., Osorio, P., Becerra, O.A.: Nonparametric generation of synthetic data using copulas. Electronics **12**(7), 1601 (2023)
46. Liu, J., Li, Y.-F., Zio, E.: A svm framework for fault detection of the braking system in a high speed train. Mech. Syst. Signal Process. **87**, 401–409 (2017)
47. Liu, J., Hu, Y., Yang, S.: A SVM-based framework for fault detection in high-speed trains. Measurement **172**, 108779 (2021)
48. Liu, J., Liu, J., Zio, E., Zio, E.: KNN-FSVM for fault detection in high-speed trains. In: International Conference on Prognostics and Health Management, pp. 1–7 (2018)
49. Zoljic-Beglerovic, S., Stettinger, G., Luber, B., Horn, M.: Railway suspension system fault diagnosis using cubature Kalman filter techniques. IFAC-PapersOnLine **51**(24), 1330–1335 (2018)
50. Qin, L., Yang, G., He, W.: Generalized Shannon entropy sparse wavelet packet transform for fault detection of traction motor bearings in high-speed trains. Structural Health Monitoring (2024)
51. Guo, L., Li, R., Jiang, B.: Fault detection and diagnosis using statistic feature and improved broad learning for traction systems in high-speed trains. IEEE Trans. Artif. Intell. **4**(4), 679–688 (2023)
52. Cheng, C., Sun, X., Shao, J., Chen, H., Chen, S.: Just-in-time learning-aided nonlinear fault detection for traction systems of high-speed trains. Int. J. Control Autom. Syst. **21**(9), 2797–2809 (2023)
53. Chen, H., Chen, H., Jiang, B., Jiang, B., Jiang, B., Jiang, B., Jiang, B., Jiang, B., Lu, N., Lu, N., Mao, Z., Mao, Z.: Deep PCA based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains. IEEE Trans. Veh. Technol. **67**(6), 4819–4830 (2018)
54. Sun, S., Zhang, S., Wang, W.: A new monitoring technology for bearing fault detection in high-speed trains. Sensors (Basel, Switzerland) **23**(14), 6392 (2023)
55. Wang, S., Ju, Y., Xie, P., Cheng, C.: Fault detection using generalized autoencoder with neighborhood restriction for electrical drive systems of high-speed trains. Control Eng. Pract. (2024)
56. Hu, H., Tang, B., Gong, X., Wei, W., Wang, H.: Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks. IEEE Trans. Ind. Inf. **13**(4), 2106–2116 (2017)
57. Cheng, C., Xuedong Li, P., Xie, X.Y.: Transfer-learning-aided fault detection for traction drive systems of high-speed trains. IEEE Trans. Artif. Intell. **4**(4), 689–697 (2023)
58. Zhang, K., Huang, W., Hou, X., Xu, J., Su, R., Xu, H.: A fault diagnosis and visualization method for high-speed train based on edge and cloud collaboration. Appl. Sci. **11**(3), 1251 (2021)
59. Xie, S., Tan, H., Yang, C., Yan, H.: A review of fault diagnosis methods for key systems of the high-speed train. Appl. Sci. **13**(8), 4790 (2023)
60. Chen, H., Chen, H., Jiang, B., Jiang, B., Jiang, B., Jiang, B.: A review of fault detection and diagnosis for the traction system in high-speed trains. IEEE Trans. Intell. Transp. Syst. **21**(2), 450–465 (2020)
61. Kour, R., Patwardhan, A., Thaduri, A., Karim, R.: A review on cybersecurity in railways. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit **237**(1), 3–20 (2023)
62. Jullian, O., Otero, B., Rodriguez, E.: Deep-learning based detection for cyber-attacks in iot networks: a distributed attack detection framework. J. Netw. Syst. Manage. **31**(3), 1–22 (2023). https://doi.org/10.1007/s10922-023-09722-7
63. Fr"ohlich, A.A., Bogo, F., Oliveira, R.: A secure iiot gateway architecture based on trusted execution environments. J. Netw. Syst. Manag. **31**(4), 1–17 (2023). https://doi.org/10.1007/s10922-023-09723-6
64. Al-Hawawreh, M., Sitnikova, E.: Developing a security testbed for industrial internet of things. IEEE Internet Things J. **8**(7), 5558–5573 (2020)
65. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: Ton_iot telemetry dataset: a new generation dataset of iot and iiot for data-driven intrusion detection systems. IEEE Access **8**, 165130–165150 (2020)
66. Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H.: Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. IEEE Access **10**, 40281–40306 (2022)
67. Axon, L., Fletcher, K., Scott, A.S., Stolz, M., Hannigan, R., Kaafarani, A.E., Goldsmith, M., Creese, S.: Emerging cybersecurity capability gaps in the industrial internet of things: overview and research agenda. Digital Threats: Res. Pract. **3**(4), 1–27 (2022)

68. Blumenfeld, M., Lin, C.-Y., Jack, A., Abdurrahman, U.T., Gerstein, T., Barkan, C.P.: Towards measuring national railways' safety through a benchmarking framework of transparency and published data. Saf. Sci. **164**, 106188 (2023)
69. Simoni, B.: ERTMS: the European rail traffic management system. *Accessed on* 2026/01/02 16:16:45. https://voie-libre.com/en/ertms-european-rail-traffic-management-system/
70. European Rail Traffic Management System (ERTMS). ERTMS Organization. *Accessed on* 2026/01/02 16:16:45. https://www.era.europa.eu/domains/infrastructure/european-rail-traffic-management-system-ertms_en
71. Rizzardi, A., Della Corte, R., Orbinato, V., Cevallos M., J.F., De Vivo, S., Sicari, S., Cotroneo, D., Coen-Porisini, A., : Railred: a node-red-based framework for modeling train control management systems. In: 2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 671–674 (2024). IEEE
72. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv. arXiv:1802.03426 (2020)
73. Shahapure, K.R., Nicholas, C.: Cluster quality analysis using silhouette score. 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), 747–748 (2020)
74. Jaworski, P., Durante, F., Härdle, W.K., Rychlik, T. (eds.): Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009. Lecture Notes in Statistics, vol. 198. Springer, Berlin, Heidelberg (2010)
75. Zhang, Z., Jiang, H., Zhao, H., Li, Y.: Federated learning-based edge computing for automatic train operation in communication-based train control systems. J. Supercomput., 1–19 (2024)
76. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR
77. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., et al.: Experimentation in software engineering **236** (2012)
78. Srock, A., Arhelger, T.: ERTMS Conference,: | Valenciennes W. Malfait & J, Hernandez Fernandez (2024)



**Alessandra Rizzardi** is an Associate Professor at University of Insubria (Varese), where she received BS/MS degree in Computer Science 110/110 cum laude in 2011 and 2013, respectively. In 2016 she got Ph.D. in Computer Science and Computational Mathematics at the same university. Her research activity is focused on cyber-security, privacy, and intrusion detection in networked and Internet of Things systems. She is member of Jounal of Big Data, ETT, ITL, and Sensors editorial board. She is IEEE member. She has been the author of 40+ scientific papers which have been published in international journals and conference proceedings.

**Raffaele Della Corte** received the B.S. and M.S. degrees in computer engineering and Ph.D. degree from the Federico II University of Naples, Italy in 2009, 2012, and 2016, respectively. He is currently an Assistant Professor (tenure track) with the Department of Electrical Engineering and Information Technologies, Federico II University of Naples. His research interests include data-driven failure analysis, on-line monitoring of software systems, and security. He serves as Reviewer in several dependabilityconferences and workshops, and he is involved in industrial projects developing techniques for the analysis and monitoring of critical systems.

**Jesús F. Cevallos M.** received a Ph.D. in Computer Science Engineering from Sapienza University (Rome) in 2022. He now covers a post-doc researcher position at the University of Insubria (Varese). His main research interests are industrial/ethical applications of Deep Learning with a special focus on Natural Language Processing and Neural Algorithmic Reasoning.

**Simona De Vivo** is a research fellow at the Department of Electrical Engineering and Information Technology at University of Naples Federico II. Her research activity focuses on cybersecurity for networked and Internet of Things systems, with particular attention to intrusion and anomaly detection techniques and the analysis of software and network behaviour. Her work is grounded in experimental evaluation and data analysis applied to real-world security scenarios.

**Sabrina Sicari** is a Full Professor at University of Insubria (Varese). She received the M.Sc. degree in Electronic Engineering, 110/110 cum laude, from the University of Catania, in 2002, where in 2006, she got Ph.D. in Computer and Telecommunications Engineering, followed by Prof. Aurelio La Corte. She is a COMNET, IEEE IoT, ETT, and ITL editorial board member. Her research concerns security, privacy, and trust in WSN, WMSN, IoT, and distributed systems. She is an IEEE senior member.

**Domenico Cotroneo** prior to joining UNC Charlotte, was a professor at the University of Naples Federico II. He is an IEEE Senior Member and an active member of the IFIP WG 10.4 on Dependable Computing and Fault Tolerance. His research focuses on Software Reliability and Security for large-scale infrastructures. He plays a leading role in the community, serving on the Steering Committees for the premier conferences in the field: DSN and ISSRE. He have mentored more than 15 PhD students and co-authored 200+ papers in top-tier international journals and conferences.

**Alberto Coen Porisini** received the Dr. Eng. degree and Ph.D. in Computer Engineering from Politecnico di Milano in 1987 and 1992. He has been a Full Professor of Software Engineering at Università degli Studi dell'Insubria since 2001, Dean of the School of Science from 2006 and Dean from 2012 to 2018. His research regards specification/design of real-time systems, privacy models, and WSN.

## Authors and Affiliations

**Alessandra Rizzardi[1] · Raffaele Della Corte[2] · Jesús F. Cevallos M.[1] · Simona De Vivo[2] · Sabrina Sicari[2] · Domenico Cotroneo[1] · Alberto Coen-Porisini[2]**

✉ Alessandra Rizzardi
alessandra.rizzardi@uninsubria.it

Raffaele Della Corte
raffaele.dellacorte2@unina.it

Jesús F. Cevallos M.
jf.cevallosmoreno@uninsubria.it

Simona De Vivo
simona.devivo@unina.it

Sabrina Sicari
sabrina.sicari@uninsubria.it

Domenico Cotroneo
domenico.cotroneo@unina.it

Alberto Coen-Porisini
alberto.coenporisini@uninsubria.it

[1] Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Via O. Rossi 9, 21100 Varese, VA, Italy

[2] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Naples, NA, Italy