2025 International Wireless Communications and Mobile Computing (IWCMC) | 979-8-3315-0887-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/IWCMC65282.2025.11059481

Open-FARI: An Open-source testbed for Federated Anomaly detection in the Railway Industrial Internet of Things

Alessandra Rizzardi*, Raffaele Della Corte†, Jesús F. Cevallos-Moreno*, Simona De Vivo†, Vittorio Orbinato†, Sabrina Sicari*, Domenico Cotroneo†, Alberto Coen-Porisini*

*Dipartimento di Scienze Teoriche e Applicate - Università degli Studi dell'Insubria
via O. Rossi 9, Varese (VA), 21100, Italy
†Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - Università degli Studi di Napoli Federico II
via Claudio 21, Naples (NA), 80125, Italy

{alessandra.rizzardi, jf.cevallosmoreno, sabrina.sicari, alberto.coenporisini}@uninsubria.it
{raffaele.dellacorte2, vittorio.orbinato, simona.devivo, domenico.cotroneo}@unina.it

***Abstract*—The paper presents Open-FARI, an open-source testbed for evaluating federated learning algorithms for anomaly detection in the railway Industrial Internet of Things domain. Open-FARI uses synthetic data generation modules trained from real train sensor data to generate realistic sensor data of a fleet of trains. Generated data encompass normal and anomalous data, enabling the evaluation of federated learning algorithms for anomaly detection. The paper addresses the lack of testbeds and datasets tailored to the railway domain, which represents an obstacle to research on Machine Learning-driven solutions in this domain.**

***Index Terms*—Anomaly detection, Federated Learning, IIoT, Railway.**

I. Introduction

The last two decades have witnessed an explosion in the adoption of Machine Learning (ML) and Deep Learning (DL) techniques for solving predictive maintenance [1], optimized control [2], and fault detection [3] tasks in a data-centric way. However, most of these scenarios apply to industrial contexts where legal and privacy concerns may restrict the practical availability of training data [4]. Leaving aside the wide plethora of data-efficiency inductive biases [5], the task of learning under low-data regimes has been faced using two main strategies that are not completely different nor alternative: Synthetic Data Generation (SDG) [6] and Federated Learning (FL) [7].

One domain where *generative* models based on ML and DL are being widely used for realistic SDG is the Industrial Internet of Things (IIoT) domain [8]. This paper focuses on one particular IIoT context in which such a restriction holds: the railway domain [9]. More specifically, the focus of this paper is sensor probes from the internals of a train and its main components, being this information susceptible to be exploited by malicious actors [10] and useful for anomaly detection. For this reason, realistic SDG is a key component of many ML pipelines related to the previously mentioned tasks [11]. Unfortunately, there is a lack of SDG instruments tailored to the railway domain. Consequently, the absence of realistic datasets and testbeds represents a high barrier entrance for the research on ML-driven solutions for the railway IIoT domain.

Federated Learning [7] is an ML paradigm that seeks to orchestrate a distributed training regime where local data cannot be shared globally. FL algorithms accelerate the convergence of any ML-based model using a combination of the parameters of every local model trained on private data [12]. The exact combination manner of these parameters varies from algorithm to algorithm, where every algorithm seeks to reach convergence under different constraints related to time, computation, and communication overhead [13].

In the railway context, the FL approach could be potentially used to accelerate the convergence of online fault detection, predictive maintenance, or control-related tasks [14]. This is especially true if considering that railway companies often manage a fleet of trains, with each train characterized by its own peculiarities in terms of event/data generation, anomalies, etc. In this scenario, FL is a key technology allowing a single train to potentially benefit from a global ML model that aggregates discriminative-generative criteria from the whole fleet [15].

This work presents Open-FARI, an **open**-source testbed for evaluating **F**ederated-learning algorithms for **A**nomaly detection in the **R**ailway **I**IoT domain[1]. The testbed uses publish-subscribe middleware and monitoring micro-services to give users a global view of the statistics. The simulated network containing the train fleet, the coordinator, and dashboard nodes can be easily instantiated using the container manager web GUI and a *Hydra*-based run-configuration framework[2].

The main contributions of Open-FARI are:

[1] Open-FARI is available at https://github.com/DIETI-DISTA-IoT/OF
[2] https://hydra.cc/

- To our knowledge, this is the first open-source testbed focused on FL-algorithm benchmarking in the railway IIoT.
- Open-FARI uses synthetic data generation modules trained from the real train (normal and anomalous) IIoT sensor data.
- We propose a case study to evaluate an anomaly detection FL approach based on Multi-Layer Perceptron (MLP) and FedAvg.

The rest of the paper is structured as follows. Section II presents related literature, while Section III exposes details on the synthetic data generated by the train vehicles in Open-FARI. The inner workings of the Open-FARI testbed are presented in Section IV. Section V documents an experimental use-case that shows the benefits of using Open-FARI, while Section VI concludes the work.

II. Related Work

A. Testbeds for Federated Learning

CoLExT [16] is a recent physical testbed that helps test multiple federated learning algorithms over heterogeneous edge networks. Traditional metrics, such as inference accuracy and other metrics focused on resource usage, are collected automatically by CoLExT, which also supports Android-based devices in the client nodes. FedBed [17] offers a simulation-based open-sourced testbed that relies on virtualization and emulation to create realistic edge environments. FedBed allows experimentation with different network scenarios, Quality of Service specifications, and networking delay simulations. Similar recent testbeds for FL are available in [18], [19].

These studies offer a fine-grain monitoring opportunity for FL algorithms on various network scenarios. This focus intersects with the FL-evaluation purpose of Open-FARI, which instead focuses on using FL for anomaly detection in an online realistic Railway IIoT monitoring tool.

B. Realistic Sensor Data Generation

Adversarial settings have largely been used to generate realistic data in IIoT contexts [20], [21]. Some of them use probabilistic inductive biases, such as mixture-density networks [22] or time-series modeling [23]. A recent trend in synthetic data generation is the usage of diffusion models [24], [25], where conditional models can generate tailored outputs. Language models are also widely used for realistic sensor data generation [26]. However, the Open-FARI framework focuses on lightweight and scalable generation of low-dimensional synthetic data. For this reason, the copula-based approach [27] is adopted in Open-FARI to generate normal and anomalous sensor data.

C. Fault Diagnostics in Railway IIoT

The most used techniques for anomaly detection pipelines in the railway domain include support vector machines [28], Kalman filters [29], and DL-based methods [30]. The work in [31] 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. Multiple types of faults are modeled, such as stator, rotor, and bearing faults, among others. Open-FARI focuses on detecting anomalies from a heterogeneous stream of sensor probes, leaving the particular DL architectural choices open. A recent review on fault detection over train sensor data is presented in [32].

D. Federated Anomaly Detection

Federated Anomaly Detection (FAD) combines FL techniques with anomaly detection algorithms to address challenges in distributed and privacy-aware environments. In particular, the FAD approach enables collaborative anomaly detection across distributed nodes without transferring raw data. This paradigm has been explored in domains like IoT networks [33] and industrial infrastructures [34], where algorithms like FedAvg [35] and FedProx [36] have been adapted for anomaly detection in heterogeneous environments, considering local constraints and inter-node variability. The use of FAD in the railway domain is still limited, but it can be useful for predictive maintenance and detecting faults. For example, recent studies use distributed models for anomaly detection in the IoT scenario [37], [38] with differential privacy and homomorphic encryption [39] or deep neural networks [40], demonstrating that global models can identify anomalies that are difficult to spot from a purely local perspective. Our proposed approach fits well into this research area, since it provides a platform that integrates FL with synthetic data generation to realistically simulate railway IIoT scenarios, extending the possibilities to test FAD algorithms under controlled conditions. By proposing this platform, we aim to overcome the challenges related to data sparsity and sensor heterogeneity in the railway domain.

III. Realistic data generation

Open-FARI uses synthetic data to train anomaly detectors through federated learning.

A. Real data

Open-FARI is based on real-world traces from high-speed train internals provided by an Italian railway company. The dataset contains approximately 60,000 diagnostic signals collected over one week of train operations. It includes telemetry from a diverse range of onboard sensors and systems. These include measurements from the braking system (e.g., cylinder pressures for multiple cars), and power supply diagnostics (battery and line voltages, as well as current measurements), among others. The dataset also records train speed, the state of control cabins, and the status of critical subsystems, such as the European Rail Traffic Management System (ERTMS) [41] and other control interfaces. More details on the dataset are available in [42].

B. Pre-processing

Each record in the original real dataset is accompanied by diagnostic messages that provide detailed insights into the train's operational status and performance throughout the observation period. An offline analysis was conducted over these messages to classify events as faults or normal probes.

After the manual labeling process, some events that referred to specific physical components or train cars were excluded from the dataset to keep a more balanced percentage of anomalies concerning normal data. Overarching operational parameters and high-level diagnostics were kept, such as those related to overall speed, general power states, emergency braking, fault aggregation, and global statuses, among others. The final dataset contained 4K records with 50 time-series data, where only 60% of these were normal data.

C. Time-series modelling

Note that any realistic SDG module, including those specifically tailored for the railway IIoT, may benefit from the ability to steer the generative distributions toward context-specific scenarios. In the case of the railway IIoT, for example, different train vehicles might be used on different routes, external weather conditions, passenger densities, speed patterns, and operational schedules, and these context factors may influence the distribution of diagnostic signals in the train.

Open-FARI uses different generation pipelines for normal and anomalous data samples. To reproduce realistic inter-arrival times with a lightweight model, exponential distributions were fitted to each data stream (anomalous and normal), having removed the outliers before fitting the distributions. When using Open-FARI, the frequency parameter for each data stream can be overwritten and/or scaled. Notice that, by setting different inter-arrival times for different vehicles, Open-FARI can be used to evaluate FL algorithms where different local nodes have different training data throughputs.

D. Copula-based data generation

After dividing anomalous from normal data and clustering each type of data, multivariate Gaussian copulas were fitted to the cluster-specific samples of each class using the *Datacebo*'s *Copulas* library[3]. A random-forest classifier was then trained on real data obtaining 87% classification accuracy over test-time real data and 71% over synthetic data. The pre-trained copulas and the corresponding exponential inter-arrival time generators are saved and imported into the container image of vehicles to generate realistic local data during experiments.

IV. The Open-FARI testbed

A. Architecture

To validate federated-learning scenarios, Open-FARI implements a distributed system with five types of nodes, which are schematically represented in Figure 1a and described in the following. Instead, Figure 1b shows the concrete implementation architecture of the testbed.

Vehicle nodes. Vehicle nodes represent each train in the railway network, which is responsible for producing IIoT-sensor probes and training a local anomaly detector module. The configuration file in Open-FARI specifies the number of vehicles to instantiate. For each vehicle, an ID must be provided in the configuration file, and optionally, the default configuration parameters can be overwritten for each vehicle. These parameters refer mostly to the data generation process. Specifically, the inter-arrival times for the anomalies and normal events can be adjusted, varying the default vehicle configuration. Vehicle nodes contain a local neural module trained using the data produced locally by the train. The training hyper-parameters can be set via the configuration file. During training, the vehicle node reports the train loss and detection accuracy to a message broker, which will be useful for monitoring experiments and implementing FL algorithms.

Manager node. The manager node is in charge of instantiating vehicles and managing the training loops for each vehicle. Specifically, this node reads the configuration files to start the requested vehicle nodes, each one with its configuration in terms of data generation. The manager node can start and stop each train's data generation and model training processes. Additionally, the manager node is in charge of initiating and finishing the dashboard node's metric collection processes.

Orchestrator. The orchestrator node collects metrics from the local modules in the vehicles and implements the FL algorithm. More specifically, it implements the Federated Averaging (FedAvg) aggregation function. FedAvg periodically collects model updates from the participating vehicles, averages their weights, and distributes the aggregated model back to the local modules, ensuring collaborative learning while preserving data privacy.

Message Broker. The nodes communicate through the message broker node, which a publish-subscribe middleware implements. Each node creates a custom topic where the training metrics are published. The multiple topic names follow a naming template that allows the orchestrator to dynamically subscribe to every vehicle-specific topic using expression matching. Also, the dashboard node subscribes to all the topics related to vehicle training statistics.

Dashboard. The dashboard node collects metrics from both the vehicle nodes and the orchestrator to provide visual insights to the administrator regarding the vehicle-wise training and the overall convergence process. The metric collection and reporting processes are asynchronous, involving the single local training processes in the vehicles. 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.

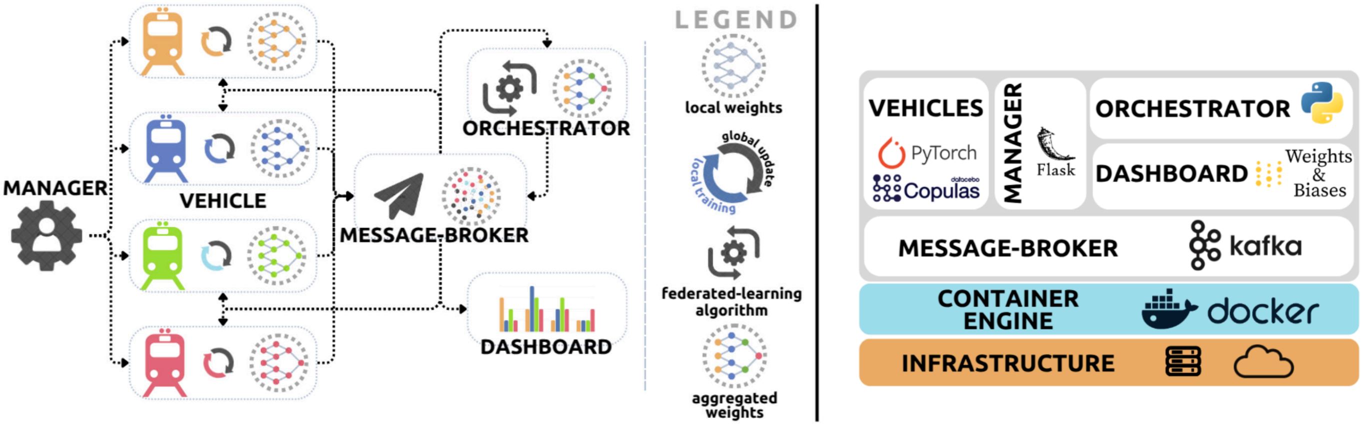
B. Implementation Details

The architecture and software stack of Open-FARI are shown in Figure 1b. To reduce the computational overhead of the simulations, Open-FARI uses process virtualization. Namely, the nodes in Figure 1a are implemented using containers. The container engine of Open-FARI is *Docker*[4], which is the *de-facto* standard for containerization. More specifically, the nodes of Open-FARI are implemented as follows:

- Each vehicle node is implemented by two running containers, i.e., a vehicle producer and a vehicle consumer.

[3] https://pypi.org/project/copulas/

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



(a) Modular view of Open-FARI.

(b) Architectural view of Open-FARI.

Figure 1: Modular (left) and architectural (right) view of Open-FARI. Vehicle nodes represent trains and corresponding anomaly detection systems trained using local data. The FL orchestrator uses extensible built-in functions to aggregate weights and send them back to vehicles. These nodes are implemented using *Docker* containers. The manager node is a *Flask* application in the host that instantiates, trains and starts training. A message broker is implemented using *Apache Kafka* for inter-node communication. The dashboard runs on a cloud-based instance of *Weights and Biases* and helps visualize the convergence status for each node and the aggregated model.

The producer emits normal and anomalous sensor probes from the train while the consumer trains the local anomaly detection module.

- The orchestrator container is in charge of collecting metrics and weights, and sending them to a *Weights and Biases*[5] dashboard in the cloud.
- The communication middleware is implemented by a container-based *Apache-Kafka*[6] cluster.
- The container manager at the top of Open-FARI runs in the host system and is implemented using the *Flask*[7] web application framework.

V. Experiments

A. Objective

Our experiments aim to demonstrate the possibility to use Open-FARI for evaluating FL algorithms in the railway context. With this aim, Open-FARI is used to compare a FL approach against local training in identifying anomalies in the IIoT data generated by vehicles.

B. Implementation Details

The experiments have been conducted using a Linux machine equipped with 9 GB of RAM and 2.6 GHz CPU, running Ubuntu 24.04 and implementing the following components.

Containerized Nodes: Every vehicle is encapsulated in a Docker container for isolation and better resource management. There will be two major types of containers: one for the node that generates data to simulate the environment of a certain vehicle, and another container for nodes running training processes to enhance machine-learning models.

Communication Middleware: We used Apache Kafka to communicate with different nodes. This messaging platform provides smooth and reliable data flow, allowing real-time interaction between different containers.

Federated Learning Orchestrator: The orchestrator manages the FL approach. Specifically, it combines model weights using the FedAvg method. We used TensorFlow Federated to coordinate the decentralized training sessions happening across multiple nodes.

Dashboard: We monitor the training process by using W&B-based dashboard. It provides an overview of the training progress as well as performance metrics to track how our model is evolving and allows developers and researchers to make changes when necessary.

C. Experimental Setup

We conducted a 140-minute-long experiment, simulating three vehicles symbolically named *Bob*, *Claude*, and *Angela*. Each vehicle is equipped with its own anomaly detection system based on a MLP neural network using ReLU activations, dropout regularization, and a sigmoid output layer for binary classification. The three vehicles produce distinct datasets, with each dataset having a different event distribution. In this way, each vehicle can generate all the events but with different occurrences, giving sense to the FL approach application. The idea here is to simulate three different kinds of vehicle, each one generating different event (both normal and anomalous) distribution, and leverage FL to allow the model on a vehicle to detect anomalies seen on the other vehicles.

As detailed in Section III, we generate the datasets using a copula-based method to ensure realistic correlations between variables, including a mix of normal and anomalous data, with an imbalance that mimics real-world conditions.

The experiments involved training the local models in two distinct scenarios:

1) *Local-only Training*: Each vehicle trains its model solely on its local dataset without any external collaboration, focusing entirely on its data.

[5] https://wandb.ai/
[6] https://kafka.apache.org/
[7] https://flask.palletsprojects.com/en/latest/

Table I: Local-only and FL pre-trained Model's Anomaly Detection Performance Comparison.

Metric (avg)	No FL Approach	FL Approach
F1-score	0.67	0.71
Accuracy	0.69	0.71
Precision	0.74	0.71
Recall	0.60	0.72

2) *Federated Learning*: Models are trained locally on each vehicle's data, with weights aggregated across vehicles using the FedAvg algorithm.

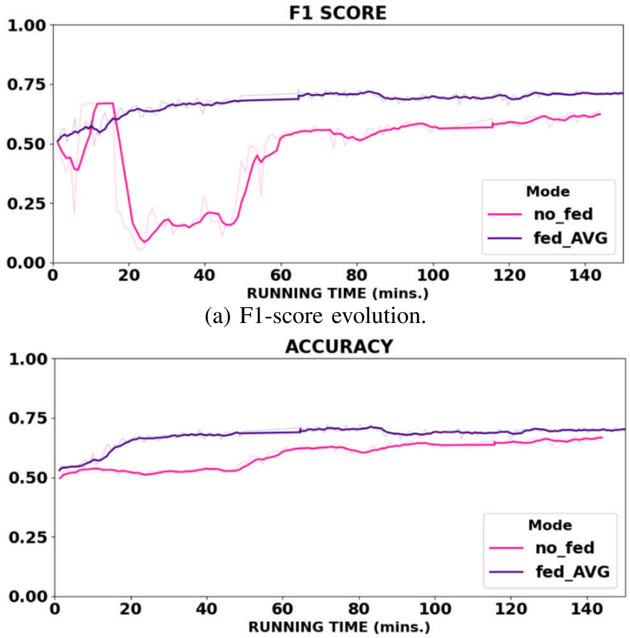
Later, each pre-trained model was evaluated on an independent dataset, with a different event distribution from those generated by the vehicles during pre-training. This evaluation involves the most common anomaly detection accuracy metrics, i.e., precision, recall, F1-score, and accuracy.

D. Results

Table I reports the average of the considered metrics, considering all the vehicles. The results suggest an overall improvement in performance when using the FL approach to train models. The F1-score increase indicates a better balance between precision and recall in anomaly detection. Accuracy slightly improved from 0.69 to 0.71, suggesting that FL contributes to a more reliable classification. The slight Precision decrease may imply a rise in false positives, though this reduction is minimal and could be offset by the significant improvement in Recall, which increased notably from 0.60 to 0.72, highlighting that the FL model is more effective in detecting anomalies, even if at the cost of a few additional false positives. This demonstrates the capability of FL to enhance generalization across diverse datasets.

Figure 2 illustrates an example that further highlights the points discussed earlier. In detail, Figure 2 presents the F1-score and Accuracy progression for one of the vehicles, i.e., *Angela*, comparing the performance of the FL and Local-Only trained MLP. We specifically chose to present the F1-score chart (along with Accuracy) since this metric provides a comprehensive overview, balancing both precision and recall, making it the most informative metric among those evaluated. As it can be seen in Figure 2, the MLP trained using the FL approach (green line, i.e., *fed_AVG*) achieves superior overall performance compared to the locally trained model (gray line, i.e., *no_fed*), and maintains a significantly more stable F1-score. In contrast, the locally trained model exhibits substantial variability in terms of F1-score. Notably, during the initial minutes, its performance is particularly low. This happens because, at that time, it is processing data patterns that occurred frequently in other vehicles, but infrequently in its dataset, making them more challenging to detect. This highlights the advantages of Federated Learning in leveraging shared knowledge across distributed nodes in the railway context, ultimately enhancing model generalization and robustness.

It should be noted that if the overall results in Table I show a slight improvement of FL over local training, the experiments mainly aim at demonstrating that Open-FARI can easily enable the evaluation of FL-based anomaly detection approaches in a realistic railway testbed.



(a) F1-score evolution.

(b) Accuracy evolution.

Figure 2: Vehicle performance metrics comparing FL and Local-Only approaches.

VI. Conclusion

The paper presented an open-source testbed, named Open-FARI, that enables the evaluation of federated learning algorithms for anomaly detection in the railway IIoT domain. Open-FARI generates realistic synthetic data through modules trained from sensor data of real-world high-speed trains. Generated data encompass both normal and anomalous data, enabling the evaluation of federated learning algorithms for anomaly detection. Open-FARI allows running simulations with several trains, emulating a fleet of trains, enabling the evaluation of FL anomaly detection approaches in the railway context. Future work will be devoted to using the platform with online real-world traces and evaluating its performance on a broader range of anomaly detection tasks in larger railway environments.

Acknowledgment

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.

References

[1] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Applied Intelligence*, vol. 52, no. 10, pp. 10 934–10 964, 2022.

[2] S. Latif *et al.*, "Deep learning for the industrial internet of things (iiot): A comprehensive survey of techniques, implementation frameworks, potential applications, and future directions," *Sensors*, vol. 21, no. 22, p. 7518, 2021.

[3] J. Yu and Y. Zhang, “Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review,” *Neural Computing and Applications*, vol. 35, no. 1, pp. 211–252, 2023.

[4] V. Demertzi, S. Demertzis, and K. Demertzis, “An overview of privacy dimensions on the industrial internet of things (iiot),” *Algorithms*, vol. 16, no. 8, p. 378, 2023.

[5] S. Segert, “Maximum Entropy, Symmetry, and the Relational Bottleneck: Unraveling the Impact of Inductive Biases on Systematic Reasoning,” Ph.D. dissertation, Princeton University, 2024.

[6] Y. Lu *et al.*, “Machine learning for synthetic data generation: a review,” *arXiv preprint arXiv:2302.04062*, 2023.

[7] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, “A survey on federated learning: challenges and applications,” *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513–535, 2023.

[8] Y.-T. Chen, C.-Y. Hsu, C.-M. Yu, M. Barhamgi, and C. Perera, “On the private data synthesis through deep generative models for data scarcity of industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 551–560, 2021.

[9] J. Feng, J. Xu, W. Liao, and Y. Liu, “Review on the traction system sensor technology of a rail transit train,” *Sensors*, vol. 17, no. 6, p. 1356, 2017.

[10] Z. Yu, Z. Kaplan, Q. Yan, and N. Zhang, “Security and privacy in the emerging cyber-physical world: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1879–1919, 2021.

[11] K. T. Chui *et al.*, “A survey of internet of things and cyber-physical systems: standards, algorithms, applications, security, challenges, and future directions,” *Information*, vol. 14, no. 7, p. 388, 2023.

[12] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy *et al.*, “A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications,” *Journal of Big Data*, vol. 10, no. 1, p. 46, 2023.

[13] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, “Model aggregation techniques in federated learning: A comprehensive survey,” *Future Generation Computer Systems*, vol. 150, pp. 272–293, 2024.

[14] J. Du, J. Cheng, Y. You, T. Wang, N. Qin, and D. Huang, “Federated transfer learning for fault diagnosis of high-speed train bogie with data security and training optimization,” in *2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*. IEEE, 2023, pp. 1–6.

[15] Z. Zhang, H. Jiang, H. Zhao, and Y. Li, “Federated learning-based edge computing for automatic train operation in communication-based train control systems,” *The Journal of Supercomputing*, pp. 1–19, 2024.

[16] J. Božič, A. R. Faustino, B. Radovič, M. Canini, and V. Pejović, “Where is the testbed for my federated learning research?” *arXiv preprint arXiv:2407.14154*, 2024.

[17] M. Symeonides, F. Nikolaidis, D. Trihinas, G. Pallis, M. D. Dikaiakos, and A. Bilas, “Fedbed: Benchmarking federated learning over virtualized edge testbeds,” in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, 2023, pp. 1–10.

[18] M. Popovic, M. Popovic, I. Kastelan, M. Djukic, I. Basicevic, and P. Vasiljevic, “Micropython testbed for federated learning algorithms,” in *32nd Telecommunications Forum (TELFOR)*. IEEE, 2024, pp. 1–4.

[19] G. Callebaut, J. Van Mulders, G. Ottoy, D. Delabie, B. Cox, N. Stevens, and L. Van der Perre, “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)*. IEEE, 2022, pp. 417–422.

[20] E. C. Fidelis, F. Reway, H. Ribeiro, P. L. Campos, W. Huber, C. Icking, L. A. Faria, and T. Schön, “Generation of realistic synthetic raw radar data for automated driving applications using generative adversarial networks,” *arXiv preprint arXiv:2308.02632*, 2023.

[21] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, “Surfelgan: Synthesizing realistic sensor data for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 118–11 127.

[22] M. Alzantot, S. Chakraborty, and M. Srivastava, “Sensegen: A deep learning architecture for synthetic sensor data generation,” in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2017, pp. 188–193.

[23] E. Brophy, Z. Wang, Q. She, and T. Ward, “Generative adversarial networks in time series: A systematic literature review,” *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–31, 2023.

[24] Y. Yang *et al.*, “A survey on diffusion models for time series and spatio-temporal data,” *arXiv preprint arXiv:2404.18886*, 2024.

[25] X. Wang *et al.*, “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*, 2023, pp. 2409–2418.

[26] D. Nguyen, S. Gupta, K. Do, T. Nguyen, and S. Venkatesh, “Generating realistic tabular data with large language models,” *arXiv preprint arXiv:2410.21717*, 2024.

[27] J. P. Restrepo, J. C. Rivera, H. Laniado, P. Osorio, and O. A. Becerra, “Nonparametric generation of synthetic data using copulas,” *Electronics*, vol. 12, no. 7, p. 1601, 2023.

[28] J. Liu, Y.-F. Li, and E. Zio, “A svm framework for fault detection of the braking system in a high speed train,” *Mechanical Systems and Signal Processing*, vol. 87, pp. 401–409, 2017.

[29] S. Zoljic-Beglerovic, G. Stettinger, B. Luber, and M. Horn, “Railway suspension system fault diagnosis using cubature kalman filter techniques,” *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 1330–1335, 2018.

[30] H. Hu, B. Tang, X. Gong, W. Wei, and H. Wang, “Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2106–2116, 2017.

[31] K. Zhang, W. Huang, X. Hou, J. Xu, R. Su, and H. Xu, “A fault diagnosis and visualization method for high-speed train based on edge and cloud collaboration,” *Applied Sciences*, vol. 11, no. 3, p. 1251, 2021.

[32] S. Xie, H. Tan, C. Yang, and H. Yan, “A review of fault diagnosis methods for key systems of the high-speed train,” *Applied Sciences*, vol. 13, no. 8, p. 4790, 2023.

[33] R. C. Geyer, T. Klein, and M. Nabi, “Differentially private federated learning: A client level perspective,” *arXiv preprint arXiv:1712.07557*, 2017.

[34] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, “A secure federated transfer learning framework,” *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[35] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[36] T. Li *et al.*, “Federated optimization in heterogeneous networks,” *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[37] A. Ghasempour, “Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges,” *Inventions*, vol. 4, no. 1, p. 22, 2019.

[38] IBM, “Internet of Things (IoT) – Topics — IBM,” 2023, accessed: 2023-10-25. [Online]. Available: https://www.ibm.com/think/topics/internet-of-things

[39] S. Truex *et al.*, “A hybrid approach to privacy-preserving federated learning,” in *Proceedings of the 12th ACM workshop on artificial intelligence and security*, 2019, pp. 1–11.

[40] S. Zhou and G. Y. Li, “Federated learning via inexact admm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 9699–9708, 2023.

[41] J. Młyńczak, A. Toruń, and L. Bester, “European rail traffic management system (ertms),” in *Intelligent Transportation Systems–Problems and Perspectives*. Springer, 2015, pp. 217–242.

[42] A. Rizzardi, R. Della Corte, V. Orbinato, J. Cevallos, S. De Vivo, S. Sicari, D. Cotroneo, A. Coen-Porisini *et al.*, “Raiired: 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)*. IEEE, 2024, pp. 671–674.