Digital Twin-Based Federated Transfer Learning for Anomaly Detection in Industrial IoT

Mohammed Ayalew Belay*, Adil Rasheed†, Pierluigi Salvo Rossi*‡
* Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway
† Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
‡ Department of Gas Technology, SINTEF Energy Research, Trondheim, Norway
Emails: mohammed.a.belay@ntnu.no, adil.rasheed@ntnu.no, salvorossi@ieee.org

2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES) | 979-8-3315-0827-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/CIES64955.2025.11007631

***Abstract*—Anomaly detection is increasingly becoming crucial for maintaining the safety, reliability, and efficiency of industrial systems. Recently, with the advent of data-driven decision-making, several statistical and machine-learning methods have been proposed. However, these methods face several challenges, such as dependence on only real sensor datasets, limited labeled data, high false alarm rates, and privacy concerns due to centralized data processing. To address these issues, we propose a digital twin-based anomaly detection approach using federated transfer learning. Digital twins allow effective anomaly detection without requiring extensive real-world failure data. The integration of digital twins with federated learning offers a powerful solution to the challenges of anomaly detection in industrial systems. The proposed method integrates digital twin data for initial training with real physical system data for model refinement. Federated learning enhances this process by maintaining data privacy through the sharing of model updates instead of raw data. The proposed combination improves model generalization, training efficiency, and performance while ensuring data privacy. We perform an extensive analysis using publicly available datasets from real-world digital and physical asset counterparts. The results demonstrate significant improvements in anomaly detection performance, highlighting the effectiveness of integrating digital twins with federated learning.**

***Index Terms*—Anomaly detection, transformer, unsupervised learning, and low-rank approximation**

I. Introduction

ANOMALY detection has gained significant importance in several applications such as industrial monitoring, network security, sensor networks, healthcare, and medical applications. In industrial systems, anomaly detection enables asset monitoring to identify faults, predict failures, and enhance system performance [1], [2]. For cybersecurity applications, anomaly detection algorithms enable network intrusion detection [3], [4]. Anomaly detection plays a crucial role in maintaining the safety, reliability, and efficiency of several industrial systems. As a result, several statistical and machine learning methods have been proposed for anomaly detection in supervised, semi-supervised, and unsupervised learning paradigms. Recently, the adoption of deep neural networks for anomaly detection has achieved remarkable success [5]–[7]. Several architectures have been implemented that include recurrent networks [8], [9], convolutional networks [10], autoencoders [11], generative adversarial networks [10], [12], transformers [13]–[15], and graph neural networks [16]. However, both statistical and deep learning anomaly detection methods face several significant challenges [17]–[19]. However, these methods are limited by several challenges, such as dependence on only real sensor datasets and the limited availability of labeled data, particularly for rare events. This scarcity limits the training of robust models and often leads to high false alarm rates. The lack of diverse data makes it difficult for these approaches to generalize across scenarios and variations, resulting in suboptimal performance in real-world applications. Additionally, these methods frequently rely on centralized data processing, which raises significant privacy and security concerns as sensitive data must be transferred and stored in a central location, making it vulnerable to breaches. Furthermore, they need a centralized high-performance computing infrastructure and are often not well suited for real-time continuous learning.

The recent advances in Industry 4.0, driven by interconnected systems and advanced computing, have enabled the widespread application of digital twins. Digital twins are virtual representations of physical systems that provide real-time monitoring, simulation, and optimization of their real-world counterparts [20]. Currently, digital twins are employed in several industrial applications for improved decision-making and operational efficiency [21], [22]. These digital twins offer performance optimization and predictive maintenance, enabling anomaly prediction and detection, resulting in reduced operational costs and downtime [23]. A digital twin-based anomaly detection approach allows for modeling unusual patterns or anomalies that deviate from the normal operation of a system without collecting expensive real-world failure test data. A digital twin-based anomaly detection model can be integrated with a real-time model of physical assets in a federated learning mechanism to address the aforementioned challenges. Federated learning is a decentralized approach to machine learning that allows models to be trained across multiple devices without transferring the raw data to a central server [24], [25]. This method maintains data privacy and security by only sharing model updates rather than the data itself. The integration of digital twins with federated learning offers a powerful approach to anomaly detection. Digital twins generate vast amounts of data, which can be leveraged to train robust

This work was partially supported by the Research Council of Norway under the project DIGITAL TWIN within the PETROMAKS2 framework (project nr. 318899).

anomaly detection models [26]. Federated learning enhances this process by allowing these models to be trained across multiple digital twins without compromising data privacy. This combination can lead to more accurate and timely anomaly detection while ensuring that sensitive data remains secure [27], [28]. Moreover, applying federated learning in scenarios involving real systems and their digital twins offers several advantages beyond data security. Training models based on data from several sources (different physical assets) enable the resulting global model to generalize better to a wider range of scenarios and variations. Federated learning also reduces the need for centralized high-performance computing infrastructure and reduces the attack surface for potential breaches. It can also support continual learning, which allows models to adapt quickly to new data and emerging trends without waiting for centralized retraining [29].

II. Related Work and Contributions

Digital twin-based anomaly detection is a significant area of research due to its potential for enhanced monitoring and maintenance of industrial systems. Benedictis et al. [30] introduce a conceptual architecture for industrial Internet of Things (IIoT) anomaly detection based on digital twins and autonomic computing paradigms. Gupta et al. [31] introduce a Hierarchical Federated Learning (HFL)-based anomaly detection model for Vehicular Internet of Things (V-IoT), integrating both digital twin and HFL approaches. This model focuses on autonomous vehicles and intelligent transportation systems, involving connected vehicles that communicate with various sensors and devices. Kamalakannan et al. [32] propose DTFL-DF, a digital twin-based federated learning approach, to mitigate fire accidents in the mining industry. They introduce a modified random forest method called Federated Decision Tree, customized for a federated fog environment, to improve fire prediction with low latency. Chatterjee et al. [33] propose blockchain-enabled federated learning (BFL) for credit card fraud detection, merging digital twins with federated learning (FL) to create a dynamic approach for identifying known and emerging fraud patterns effectively. Gupta et al. [28] propose a digital twin-based hierarchical federated anomaly detection approach for smart health applications. They develop a federated time-distributed Long Short-Term Memory (LSTM) model to enhance the anomaly detection process.

Although several digital twin-based federated anomaly detection frameworks have been introduced, several challenges still need to be addressed. One significant challenge is communication overhead. Federated learning inherently requires frequent communication between devices and the central server, leading to increased latency and bandwidth consumption. Additionally, many existing frameworks are purely digital twin-based and do not integrate digital twin models with their physical counterparts effectively. This lack of integration can lead to discrepancies between the virtual and real-world systems. Furthermore, there is a notable absence of methodologies for seamlessly integrating digital twin models with real-world data models. This integration is crucial for ensuring that digital twins accurately reflect the physical systems they represent, thereby enhancing the reliability and accuracy of anomaly detection processes.

To address these challenges, we propose a digital twin-based anomaly detection approach using federated transfer learning. In the proposed algorithm, we initialize a global model using a digital twin-based model trained on digital twin data. Local models for each asset are then built using real physical system data. During each federated learning round, a subset of clients updates their local models, which are aggregated to refine the global model. By leveraging both digital twin data for initial training and real system data for fine-tuning through federated learning, the proposed approach ensures a robust and well-generalized global model. This method leverages both data sources, significantly improving training efficiency and model performance while maintaining data privacy.

Specifically, our contribution can be summarized as follows:

- We propose a digital twin-based anomaly detection method using federated transfer learning that improves model generalization, allowing continuous learning and real-time model updates.
- The proposed approach avoids the need for centralized high-performance computing and addresses data privacy and security issues.
- We performed an experimental analysis using publicly available datasets from real-world digital and physical asset counterparts.

The rest of the paper is organized as follows: Section III describes the proposed method; Section IV presents the experimental setup and the datasets; while Section V illustrates the performance analysis and related discussion; finally, conclusions and future research directions are given in Section VI.

III. The Proposed Method

In this study, we proposed a novel digital twin and federated learning-based anomaly detection (DTFed) framework. The method utilizes a federated transfer learning framework to leverage both digital twin data and real system data for training a robust neural network model. The framework involves several steps, including digital-twin model training, digital-twin transfer learning, local model training, model training, and federated averaging for model aggregation across physical assets, as shown in figure 1. The DTFed algorithm is designed to facilitate federated learning, where multiple assets, each possessing their own local data, collaboratively train a shared global model. This collaborative process aims to preserve data privacy and minimize communication costs by leveraging local computations. The DTFed framework enhances the global model through the local data of assets, with an initial pre-training phase using digital twin data. Let $\mathcal{D}_{\text{phys}}$ and $\mathcal{D}_{\text{twin}}$ denote the datasets from the real system and digital twin, respectively. Let $\mathcal{D}_{\text{phys}}$ and $\mathcal{D}_{\text{twin}}$ represent the datasets from the real system and digital twin, respectively. The process begins with training the digital twin-based model. The digital twin data is used to minimize the binary cross-entropy loss

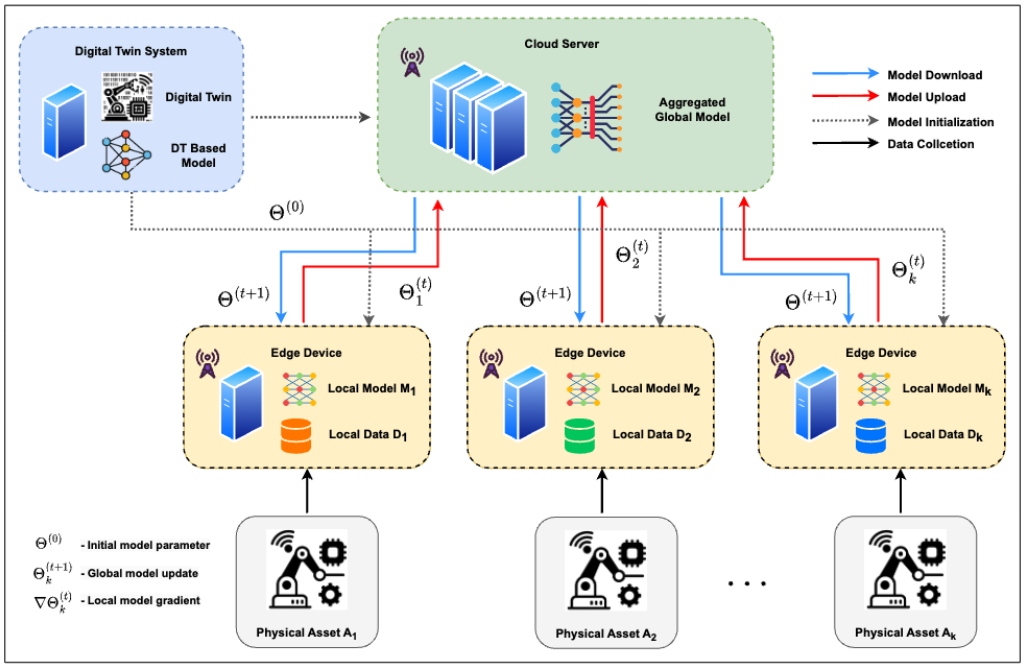

Fig. 1. Digital twin-based federated learning.

function ℓ_{BCE} through mini-batch stochastic gradient descent as follows:

$$\boldsymbol{\Theta}^{(0)} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{|\mathcal{D}_{\text{twin}}|} \sum_{i=1}^{|\mathcal{D}_{\text{twin}}|} \ell_{\text{BCE}}(\boldsymbol{\Theta}; \mathcal{D}_{\text{twin}}) \tag{1}$$

where $\boldsymbol{\Theta}^{(0)}$ represent the parameters of the digital twin model and $|\mathcal{D}_{\text{twin}}|$ denotes the number of samples in the digital twin data. This initialization leverages synthetic data generated to simulate the real system, providing a robust starting point for the global model. We then proceed through a federated transfer learning mechanism where K physical assets collaboratively learn global model parameters. Each physical asset k has a local dataset $\mathcal{D}_k \subseteq \mathcal{D}_{\text{phys}}$. The objective of the DTLFed is to train a global model with parameter $\boldsymbol{\Theta}$ by collaboratively optimizing the objective function $J(\boldsymbol{\Theta})$ as:

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \frac{1}{\sum_{k=1}^{K} n_k} \sum_{k=1}^{K} \sum_{l=1}^{n_k} \ell(\boldsymbol{\Theta}; d_{k,j}) \tag{2}$$

Where $\boldsymbol{\Theta}^*$ is optimized parameters, $d_{k,j}$ denotes the j-th data sample from the local physical dataset $\mathcal{D}_k$ of the k-th asset. The algorithm proceeds through a series of rounds $(t = 1, 2, \ldots)$ and in each round, the server selects a subset of assets to participate in. The number of clients m is determined by the fraction C of the total number of assets K, ensuring at least one client asset is selected:

$$m = \max(C \cdot K, 1) \tag{3}$$

The set of selected clients S_t is chosen randomly. Each selected client $k \in S_t$ initializes its local model with the current global model parameters $\boldsymbol{\Theta}^{(t)}$ and performs local training for E epochs using its local data. Each asset client splits its local dataset $\mathcal{D}_k$ into batches of size B, and for each batch b, performs a local gradient descent step. On the t-th communication round, the updated parameter of the model, $\boldsymbol{\Theta}^{(t)}$, is sent to each asset from the server. Each asset k updates the local model parameter using:

$$\boldsymbol{\Theta}_k^{(t+1)} = \boldsymbol{\Theta}^{(t)} - \eta_k^{(t)} \cdot \frac{1}{n_k^{(t)}} \nabla_{\boldsymbol{\Theta}} \sum_{d_{k,j} \in \mathcal{D}_k^{(t)}} \ell(\boldsymbol{\Theta}^{(t)}; d_{k,j}) \tag{4}$$

Where $\eta_k^{(t)}$ denotes the local learning rate, $\mathcal{D}_k^{(t)} \subseteq \mathcal{D}_k$ is a randomly selected batch of data to perform a local gradient update on the t-th asset. After completing the local updates, each client k returns its updated model $\boldsymbol{\Theta}_k^{(t+1)}$ to the server. The server aggregates the model by averaging the parameters [24] from the selected clients to obtain the new global model parameters, $\boldsymbol{\Theta}^{(t+1)}$, as:

$$\boldsymbol{\Theta}^{(t+1)} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\Theta}_k^{(t+1)}. \tag{5}$$

This iterative process continues with the server sending the updated global model to each asset. DTFed improves training efficiency and model performance by integrating digital twin data for initial training and real system data for fine-tuning through federated learning. It leverages the strengths of both data sources, ensuring robust and generalized model performance while preserving data privacy and minimizing communication overhead. The DTFed procedure is summarized in algorithm 1.

Algorithm 1: DTFed Algorithm

Input: $\mathcal{D}_{\text{twin}}$, $\mathcal{D}_{\text{real}}$, K, C, mini-batch size B, number of local epochs E, learning rate η

Output: Trained global model parameters θ

Initialize global model parameters θ_0 by pre-training on digital twin data $\mathcal{D}_{\text{twin}}$ (Eq. 1);
for *each round* $t = 1, 2, \ldots, T$ **do**
  Determine the number of assets m (Eq. 3);
  Randomly select a set of assets S_t from the K clients;
  for *each asset* $k \in S_t$ ***in parallel*** **do**
    Initialize local model parameters $\theta_{k,t}$ with current global model parameters θ_t;
    for *each local epoch* $e = 1$ *to* E **do**
      Split local dataset $\mathcal{D}_k$ into mini-batches of size B;
      for *each mini-batch* b **do**
        Perform a local gradient descent step for each asset (Eq. 4);
      end for
    end for
    Send updated model parameters $\theta_{k,t+1}$ to the server;
  end for
  Update global model parameters θ_{t+1} by averaging the client models (Eq. 5);
end for

IV. Experimental Setup

A. Datasets

We performed our experiments using datasets from an Industry 4.0 production line system and its digital twin under cyberattack [34]. The real system in our experiment comprises four production stations: pick and place, assembly, muscle compressing, and sorting. These modular stations are controlled by Siemens programmable logic controllers (PLCs). The data was gathered from both the real manufacturing system and its digital twin when subjected to a denial of service (DoS) attack that creates delays in system response and increases the cycle time. Data was recorded during both normal operation and under attack conditions, and label information is provided.

The dataset contains 58 features that represent sensor readings, actuator statuses, timing information, throughput time, and PLC parameters for the four stations. The processes covered include pick and place, loading, air pressing, controlling panels, sorting, and other related activities.

B. Implementation Details

In our implementation, we apply the normalization preprocessing step to real system data and digital twin data. We employ the PyTorch deep learning framework for building a simple neural network model for both the digital twin and physical asset local models. The architecture consists of an input layer matching the feature size of the training data, an 8-neuron hidden layer with ReLU activation, and a single neuron output layer. The model is trained initially on digital twin data using a mini-batch size of 32, a learning rate of 0.001, and for 10 epochs to establish the global model initial parameters. Each client trains a similar model locally using the proposed DTFed algorithm, which selects a fraction of clients (C) per round, with each client performing local updates over a specified number of epochs (E) and mini-batch size (B).

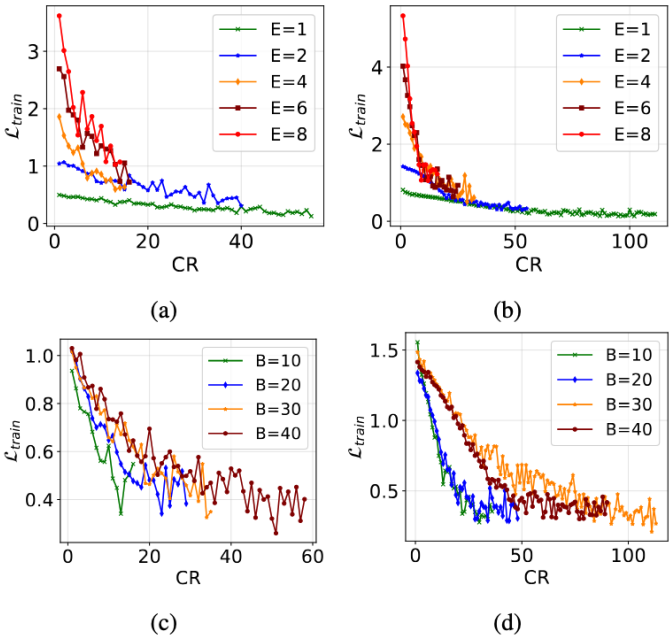

Fig. 2. Convergence analysis of federated learning models. (a) and (c) are digital Twin-Based models, and (b) and (d) are without digital twin.

C. Evaluation Metrics

The global model performance is evaluated on the test set using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve across different scenarios. These metrics are based on the correct detection of anomalies (true positives, TP), false alarms (false positives, FP), correct identification of normal samples (true negatives, TN), and missed anomalies (false negatives, FN). The true positive rate (TPR) and false positive rate (FPR) are defined as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}\,, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}\,, \tag{6}$$

The model accuracy (A) is calculated as:

$$\text{A} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}$$

while precision (P), recall (R), and F1-score (F_1) are defined as:

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}}\,, \quad \text{R} = \text{TPR}\,, \quad F_1 = \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}}\,, \tag{8}$$

The ROC curve, which plots TPR vs. FPR, is used to evaluate the model's performance across different thresholds.

V. Results and Discussions

A. Convergence Analysis

We compared the convergence performance of the proposed digital twin-based anomaly detection model against a non-digital twin-based model. As illustrated in Figure 2, the convergence was evaluated by plotting communication rounds (CR) versus training loss for different local epochs (E) and batch sizes (B) with a fixed client fraction ($C = 0.1$). The total communication rounds were determined based on a target accuracy of 95%. Figures 2a and 2b demonstrate that increasing E accelerated convergence for both models, with the digital twin-based model consistently requiring fewer communication rounds. For instance, with $E = 4$, the digital twin-based model converged in 15 rounds, while the non-digital twin-based model required 32 rounds. Similarly, experiments with $E = 2$ and varying B (Figures 2c and 2d) showed faster convergence for the digital twin-based model. Smaller batch sizes led to quicker convergence for both methods. For $B = 10$, the digital twin-based model achieved target accuracy in 16 rounds, compared to 41 rounds for the non-digital twin model. Overall, the digital twin-based method demonstrated superior convergence, reducing the number of communication rounds needed to achieve high accuracy.

B. Scalability Analysis

We analyze the effect of varying client fractions C on the number of communication rounds needed to achieve 95% accuracy. The experiments were conducted with $K = 20$ clients, $E = 1$ local epochs, and different batch sizes B. As shown in Figures 3a and 3b, the digital twin-based model consistently converged faster than the non-digital twin-based model. For instance, with $C = 0.1$ and $B = 10$, the digital twin-based model required 35 rounds, while the non-digital twin model needed 74 rounds.

Generally, more clients led to more communication rounds for convergence, but increasing C sometimes reduced this number. Smaller batch sizes facilitated faster convergence for both models. For example, with $C = 0.2$, the digital twin-based model required 54 rounds with $B = 10$ and 144 rounds with $B = 30$, compared to the non-digital twin-based model's 58 and 141 rounds, respectively. The digital twin-based model showed greater efficiency, especially with smaller batch sizes and higher client fractions. With $C = 0.3$ and $B = 10$, it converged in 48 rounds, compared to 82 rounds for the non-digital twin model.

C. Local Computation on anomaly detection Performance

We evaluate the impact of local epoch and mini-batch on the anomaly detection performance. We first consider fixed local epochs $E = 1$ with different local mini-batch sizes B The experiments were conducted with a client fraction $C = 0.1$ to achieve an accuracy of 95% on the test data. Figures 4a and 4b present an anomaly detection accuracy with communication round for a fixed local epoch. The results show that smaller batch sizes ($B = 10$ and $B = 20$) require fewer communication rounds to achieve the target anomaly detection

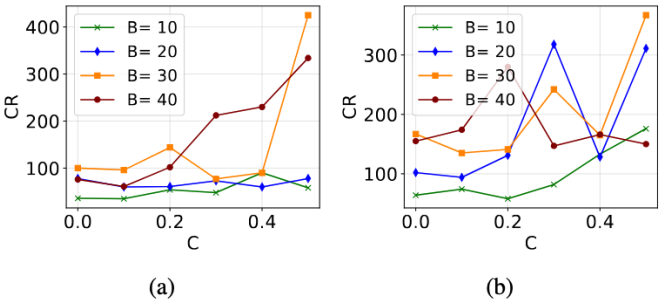

Fig. 3. Client fractions vs. the number of communication rounds required to achieve the target anomaly detection accuracy of 95%. (a) Digital Twin-Based Model. (b) Without Digital Twin.

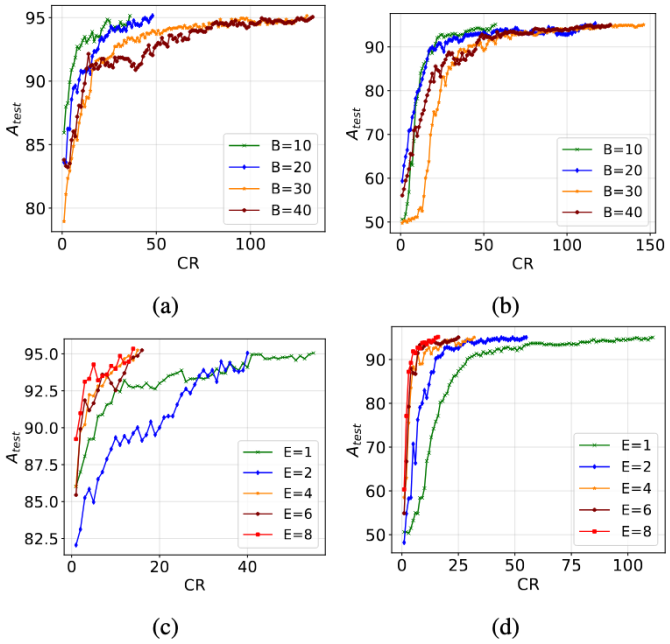

Fig. 4. Accuracy vs. communication rounds with fixed asset fraction ($C = 0.1$). (a) and (c) are digital Twin-Based models, and (b) and (d) are without digital twin.

accuracy. As the batch size increases, the number of communication rounds required for convergence also increases. In addition, the digital twin-based model reduces the number of communication rounds for the target anomaly detection accuracy, which underscores the advantage of using digital twin data for initial training.

We further evaluate the impact of varying the number of local epochs E while keeping the local mini-batch size fixed at $B = 20$. The results show that increasing the number of local epochs E generally leads to an increase in detection accuracy for both models. However, the digital twin-based model consistently required fewer communication rounds across all configurations of local epochs compared to the non-digital twin-based model. A similar trend is observed for detection precision and recall metrics, as shown in figure 5.

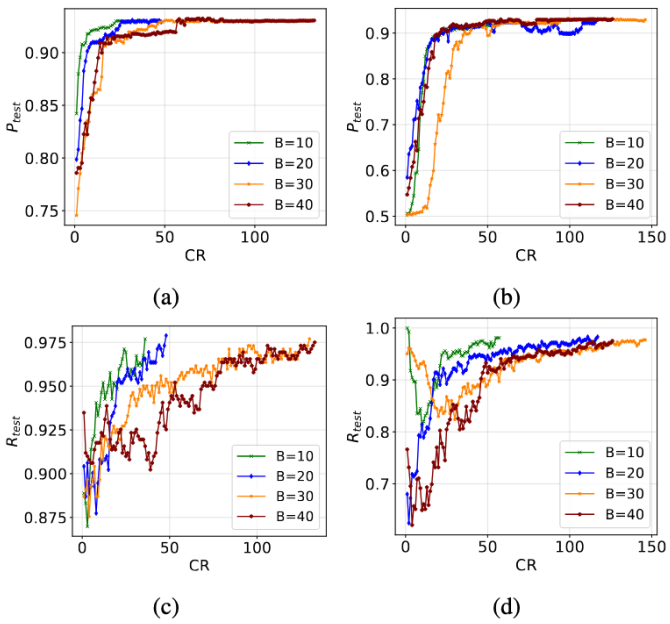

Fig. 5. Precision and recall vs. communication rounds with fixed asset fraction ($C = 0.1$). (a) and (c) are digital Twin-Based models, and (b) and (d) are without digital twin.

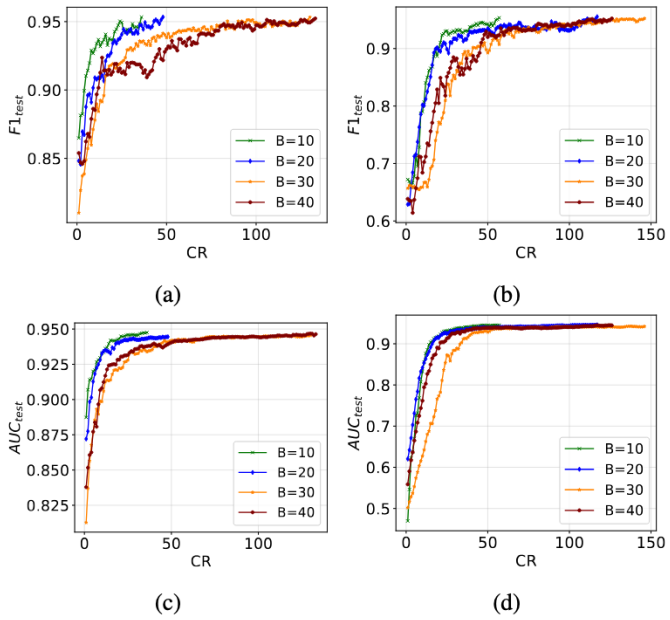

Fig. 6. F1-score and AUC vs. communication rounds. (a) and (c) are digital Twin-Based models, and (b) and (d) are without digital twin.

D. Global Model anomaly detection Performance

We analyze the global model performance for both digital twin-based and non-digital twin-based models based on various performance metrics. The experiments were conducted with a fixed client fraction ($C = 0.1$), local epochs ($E = 1$), and local mini-batch size ($B = 20$). Tables I summarize the results for different numbers of communication rounds. The results show that the digital twin-based model consistently

TABLE I
GLOBAL MODEL PERFORMANCE ANALYSIS WITH FIXED $E = 1$ AND $B = 20$

CR	Digital twin based			Without Digital twin		
	A	F1	AUC	A	F1	AUC
1	86.0330	0.8693	0.8793	50.6305	0.6722	0.4433
5	89.2338	0.8938	0.9129	53.2493	0.6761	0.5800
10	92.6285	0.9278	0.9269	60.6208	0.6818	0.6972
20	93.0165	0.9317	0.9355	82.2502	0.8208	0.8704
30	93.3075	0.9346	0.9437	90.3977	0.9070	0.9185
40	94.0834	0.9425	0.9469	91.3676	0.9156	0.9300
50	94.8594	0.9503	0.9481	92.6285	0.9288	0.9363

improves across all metrics as the number of communication rounds increases. The non-digital twin-based model also shows improvements with increasing communication rounds but lags behind the digital twin-based model in all performance metrics. Figure 6 presents detection F1-score and AUC for the global model for different local batches. The result shows that small batch sizes reduce the number of communication rounds to achieve a high F1-score and AUC. In general, the inclusion of digital twin data in the initial training phase significantly enhances the federated learning process, leading to faster convergence and higher overall performance.

VI. CONCLUSIONS AND FUTURE WORK

In this study, we proposed communication-efficient anomaly detection in digital twins using digital twin-based federated transfer learning. The objective was to reduce the communication overhead challenges inherent in federated learning environments by utilizing digital twin. The experimental results showed that our approach consistently outperformed the traditional non-digital twin FedAvg model across metrics such as accuracy, F1-score, and AUC, while requiring fewer communication rounds, demonstrating its efficiency. The use of digital twin data for initial training was key in improving model performance and reducing communication costs. Future work includes incorporating unsupervised and semi-supervised learning, enhancing privacy through techniques like homomorphic encryption, and exploring non-identical data sources. Additionally, integrating digital twin-based knowledge distillation and extending DTFed to fields like healthcare and smart cities can further validate its scalability and versatility.

ACKNOWLEDGMENT

This work was partially supported by the Research Council of Norway under the project DIGITAL TWIN within the PETROMAKS2 framework (project nr. 318899).

REFERENCES

[1] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic, "Big-data-driven anomaly detection in industry (4.0): An approach and a case study," *IEEE International Conference on Big Data*, pp. 1647–1652, 2016.

[2] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[3] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 303–336, 6 2014.

[4] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 6 2009.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[6] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 54, no. 2, 6 2021. [Online]. Available: https://doi.org/10.1145/3439950

[7] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. Salvo Rossi, "Unsupervised Anomaly Detection for IoT-Based Multivariate Time Series: Existing Solutions, Performance Analysis and Future Directions," *Sensors*, vol. 23, no. 5, p. 2844, 6 2023.

[8] Y. Su, R. Liu, Y. Zhao, W. Sun, C. Niu, and D. Pei, "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network," *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837, 7 2019.

[9] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection," *arXiv.org preprint*, 7 2016. [Online]. Available: https://arxiv.org/abs/1607.00148v2

[10] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2118–2132, 2 2021.

[11] C. Zhou and R. C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders," *Knowledge Discovery and Data Mining*, vol. Part F129685, pp. 665–674, 8 2017.

[12] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate Time-series Anomaly Detection via Graph Attention Network," *Industrial Conference on Data Mining*, vol. 2020-November, pp. 841–850, 11 2020.

[13] M. A. Belay, A. Rasheed, and P. Salvo Rossi, "MTAD: Multiobjective Transformer Network for Unsupervised Multisensor Anomaly Detection," *IEEE Sensors Journal*, vol. 24, no. 12, pp. 20 254–20 265, 6 2024.

[14] Y. He and J. Zhao, "Temporal Convolutional Networks for Anomaly Detection in Time Series," *Journal of Physics: Conference Series*, vol. 1213, no. 4, p. 042050, 6 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1213/4/042050 https://iopscience.iop.org/article/10.1088/1742-6596/1213/4/042050/meta

[15] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data," *VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 1 2022. [Online]. Available: http://arxiv.org/abs/2201.07284

[16] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A Comprehensive Survey on Graph Anomaly Detection with Deep Learning," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.

[17] M. A. Belay, A. Rasheed, and P. S. Rossi, "Multivariate Time Series Anomaly Detection via Low-Rank and Sparse Decomposition," *IEEE Sensors Journal*, vol. 24, no. 21, pp. 34 942–34 952, 2024.

[18] M. A. Belay, A. Rasheed, and P. Salvo Rossi, "Self-Supervised Modular Architecture for Multi-Sensor Anomaly Detection and Localization," *IEEE Conference on Artificial Intelligence (CAI)*, pp. 1278–1283, 6 2024.

[19] M. A. Belay, A. Rasheed, and P. S. Rossi, "Sparse Non-Linear Vector Autoregressive Networks for Multivariate Time Series Anomaly Detection," *IEEE Signal Processing Letters*, pp. 1–5, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10806816/

[20] E. H. Glaessgen and D. S. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2012.

[21] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[22] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 4 2019.

[23] F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," *IEEE Access*, vol. 5, pp. 20 418–20 427, 9 2017.

[24] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *International Conference on Artificial Intelligence and Statistics*, 2 2017. [Online]. Available: http://arxiv.org/abs/1602.05629

[25] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 5 2020.

[26] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning," *IEEE Access*, vol. 7, pp. 19 990–19 999, 2019.

[27] S. He, C. Du, and M. S. Hossain, "6G-Enabled Consumer Electronics Device Intrusion Detection With Federated Meta-Learning and Digital Twins in a Meta-Verse Environment," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3111–3119, 2 2024.

[28] D. Gupta, O. Kayode, S. Bhatt, M. Gupta, and A. S. Tosun, "Hierarchical Federated Learning based Anomaly Detection using Digital Twins for Smart Healthcare," *IEEE International Conference on Collaboration and Internet Computing*, pp. 16–25, 2021.

[29] L. Tang, M. Wen, Z. Shan, L. Li, Q. Liu, and Q. Chen, "Digital Twin-Enabled Efficient Federated Learning for Collision Warning in Intelligent Driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2573–2585, 3 2024.

[30] A. De Benedictis, F. Flammini, N. Mazzocca, A. Somma, and F. Vitale, "Digital Twins for Anomaly Detection in the Industrial Internet of Things: Conceptual Architecture and Proof-of-Concept," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11 553–11 563, 12 2023.

[31] D. Gupta, S. S. Moni, and A. S. Tosun, "Integration of Digital Twin and Federated Learning for Securing Vehicular Internet of Things," *2023 Research in Adaptive and Convergent Systems RACS 2023*, 8 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3599957.3606250

[32] U. Kamalakannan, R. Sriramulu, and P. Ramamurthi, "DTFL-DF: Digital twin architecture powered by federated learning decision forest to mitigate fire accidents in mining industry," *Systems Engineering*, vol. 27, no. 5, pp. 869–885, 2024. [Online]. Available: https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/sys.21755

[33] P. Chatterjee, D. Das, and D. B. Rawat, "Digital twin for credit card fraud detection: opportunities, challenges, and fraud detection advancements," *Future Generation Computer Systems*, vol. 158, pp. 410–426, 9 2024.

[34] L. Shi, "A Industry 4.0 production line system and its Digital Twin under cyber attack," *IEEE DataPort*, 2023. [Online]. Available: https://dx.doi.org/10.21227/8fn7-gz90