IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 12, 2022, accepted May 4, 2022, date of publication May 9, 2022, date of current version May 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3173288

Federated Learning-Based Explainable Anomaly Detection for Industrial Control Systems

TRUONG THU HUONG[1], (Member, IEEE), TA PHUONG BAC[2], KIEU NGAN HA[1], NGUYEN VIET HOANG[1], NGUYEN XUAN HOANG[1], NGUYEN TAI HUNG[1], AND KIM PHUC TRAN[3]

[1]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Hai Ba Trung 100000, Vietnam
[2]School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea
[3]Génie et Matériaux Textiles (GEMTEX), National Higher School of Arts and Textile Industries (ENSAIT), University of Lille, 59000 Lille, France

Corresponding authors: Truong Thu Huong (huong.truongthu@hust.edu.vn) and Nguyen Tai Hung (hung.nguyentai@hust.edu.vn)

This work was supported by the Hanoi University of Science and Technology (HUST) under Project T2021-PC-010.

ABSTRACT We are now witnessing the rapid growth of advanced technologies and their application, leading to Smart Manufacturing (SM). The Internet of Things (IoT) is one of the main technologies used to enable smart factories, which is connecting all industrial assets, including machines and control systems, with the information systems and the business processes. Industrial Control Systems of smart IoT-based factories are one of the top industries attacked by numerous threats, especially unknown and novel attacks. As a result, with the distributed structure of plenty of IoT front-end sensing devices in SM, an effectively distributed anomaly detection (AD) architecture for IoT-based ICSs should: achieve high detection performance, train and learn new data patterns in a fast time scale, and have lightweight to be deployed on resource-constrained edge devices. To date, most solutions for anomaly detection have not fulfilled all of these requirements. In addition, the interpretability of why an instance is predicted to be abnormal is hardly concerned. In this paper, we propose the so-called FedeX architecture to address those challenges. The experiments show that FedeX outperforms 14 other existing anomaly detection solutions on all detection metrics with the liquid storage data set. And with Recall of 1 and F1-score of 0.9857, it also outperforms those solutions on the SWAT data set. FedeX is also proven to be fast in terms of training time of about 7.5 minutes and lightweight in terms of hardware requirement with memory consumption of 14%, allowing us to deploy anomaly detection tasks on top of edge computing infrastructure and in real-time. Besides, FedeX is considered as one of the frameworks at the forefront of interpreting the predicted anomalies by using XAI, which enables experts to make quick decisions and trust the model more.

INDEX TERMS Anomaly detection, ICS, federated learning, XAI, VAE, SVDD.

I. INTRODUCTION

An Industrial Control System (ICS) is an automation system that controls and monitors functionality in industrial processes. Wireless and control devices of ICSs are widely deployed in industrial sectors and critical infrastructures such as power grids, water treatment facilities. As illustrated in Fig.1, a typical ICS comprises multiple control loops connected with human-machine interfaces (HMIs), and remote diagnostics and maintenance functions based on network protocols.

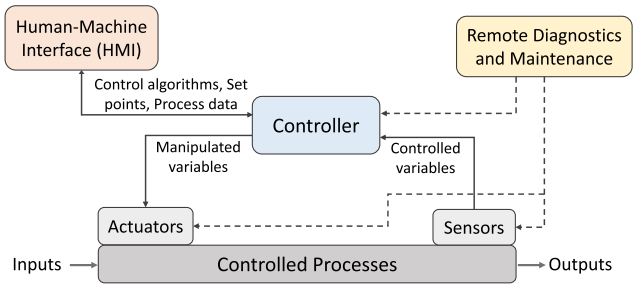


FIGURE 1. Components of a basic ICS.

Nowadays, AI and Bigdata present excellent potential in migrating the manufacturing paradigm to smart manufacturing (SM), as it enables Industrial Internet of Things (IIoT) based systems to operate in real-time and

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

be more precise and efficient [1]. However, the exponential rise of IIoT brings not only enormous benefits but also significant obstacles in terms of developing and deploying secured ICSs [2], [3]. In reality, a contemporary ICS is no longer a stand-alone system but rather linked to the Internet. As a result, if hackers were to acquire control of a network and steal security-critical data, viruses and infections would infiltrate and damage the operating system of a production line. So the effects would be severe and costly. An IIoT-based Industrial Control Systems is currently one of the top industries attacked by various threats. As threats are becoming more complex, it is required to have an anomaly detection (AD) method that can identify attacks quickly and correctly. Meanwhile the method should be enough lightweight to be deployed in IoT devices with limited processing capacity.

From another perspective, recently, IIoT has been designed to make use of edge computing technology to perform computational tasks right at the edge of a network. This avoids offloading intensively computational tasks to a cloud centre as traditional IIoT would deploy. Edge computing can solve a serious drawback in a smart factory such as latency required for transmitting, receiving and processing big data collected from IoT devices. Within that context, Federated Learning (FL) - a distributed machine learning mechanism [4] was found to be a promising scheme for edge computing in a distributed environment.

In addition, as IIoT evolves, the amount of data collected from ICS systems will increase exponentially. As a result, network latency and bandwidth become a barrier when these data are sent to the cloud from distributed edge nodes [5]. With this issue come along privacy issues of sending sensitive data over a transmission channel. Therefore, ensuring the privacy and safety of data in ICSs is also an issue receiving much attention, along with the development of anomaly detection algorithms [6], [7].

Besides, although deploying FL enables distributed deep learning algorithms to work efficiently for anomaly detection in IIoT-based ICSs, anomaly detection techniques can only help detect abnormalities. The output of the Machine Learning-based detection model is difficult to explain or interpret, especially in ICSs where information is often abstract. Interpretability is the degree to which a human can understand the cause of a decision. An explanation denotes the subset of elements in a sample that has the highest impact on predicting a label output of an ML-based detection model. In the domain of cybersecurity analysts, a satisfying explanation would also need a description of ''why'' those attributes are critical. Because of this limitation, persuading experts to accept and use anomaly detection technologies is difficult. Such ML-based model' outputs may contain abnormal cases that the systems analyst was previously unaware of, and an explanation of why an instance is abnormal might boost the analyst's confidence in the algorithm. The higher the interpretability of an ML model is, the more easily administrators can comprehend why certain predictions have been made. Furthermore, explanations might be contradictory, which is valuable and important for explaining anomalies. To overcome this drawback, the concept of eXplainable Artificial Intelligence (XAI) comes into play for ICSs. XAI has been developed to explain predictions from anomaly detection algorithms.

Motivated by these potentials, in this paper, we propose a **Fed**erated learning-based **Ex**plainable Anomaly Detection for Industrial Control Systems - called FedeX as a whole architecture to detect and analyze anomalies in ICSs and to enable detection in a distributed environment with FL. FedeX is a combined design of Variational autoencoder (VAE) as an efficient detection model, Federated Learning as a solution for missing training data, Support vector data description (SVDD) as an automatic threshold determination and, XAI to interpret the black-box learning model.

The benefits of FedeX can be summarized as follows: FedeX is one of the first frameworks that applies XAI to explain anomalies for ICSs in a liquid-storage infrastructure. Thanks to the XAI function, experts could define which real features contribute mainly to the anomaly.

Besides, FedeX enables faster system response capability upon attacks since detection is deployed near anomaly sources. With FL aggregating distributed models into a united global model, FedeX can yield even higher detection performance than the centralized learning manner with Accuracy, Precision of up to 0.99; Recall, F1-score, and AUC of up to 1 at maximum on the SCADA liquid storage infrastructure dataset, as illustrated in Table 2. Additionally, as evaluated in Table 3, FedeX is proven to outperform 14 other anomaly detection reference methods on all detection metrics with the main case study of SCADA liquid storage. For the cross validation case on the SWaT dataset, with Recall of 1 and F1-score of 0.9857, FedeX peforms better those 14 solutions.

Moreover, FedeX is able to retrain its learning model fast enough in every 7.5 minutes, so as to cope with any drift in the normal/abnormal behaviour of data coming from devices (for example, drift caused by device ageing inside a smart factory).

In addition, FedeX is quite lightweight in terms of bandwidth and memory occupation that could be deployable on top of edge devices with limited computing capacity.

The rest of our paper is structured as follows. Section II discovers related and cutting-edge researches in the field of anomaly detection and XAI for ICSs. The FedeX anomaly detection architecture will be detailed in Section III. The evaluation of FedeX's performance in terms of detection capability, system response time, edge computing capability, and anomalies explanation is presented in Section IV. Next, in Section V, we discuss the contributions, practical implications, limitations, and future work of this research. Finally, the conclusions are presented in Section VI.

II. RELATED WORK

A short summary of related work is briefly described in Table.1, highlighting some differences such as ICS contexts,

TABLE 1. Summary of recent works for anomaly detection in Industrial Control Systems (ICS) scenarios.

Work	AD Method	Data	Learning Manner	Runtime Assessment	Hardware Assessment	XAI Integration	Year
[8]	LR, LDA, KNN, CART, NB, and SVM	SWaT	Centralized	No	No	No	2021
[9]	SVM and DNN	SWaT	Centralized	No	No	No	2017
[10]	MLP, CNN, and RNN	SWaT	Centralized	No	No	No	2018
[11]	LSTM	SWaT	Centralized	No	No	No	2020
[12]	LSTM	SWaT	Centralized	No	No	No	2020
[13]	1D-CNN	SWaT	Centralized	Yes	No	No	2018
[14]	Rule-based approach	SWaT	Centralized	Yes	No	No	2020
[15]	CUSUM control chart	SWaT	Centralized	No	No	No	2018
[16]	VAE and gradient-based fingerpriting	UGR16	Centralized	Yes	No	No	2019
[17]	CNN and LSTM	Power demand, Engine, Space shuttle, ECG	Federated	Yes	No	No	2020
[18]	GRU and LSTM	Cyber-security MODBUS ICS dataset	Federated	Yes	No	No	2021
[7]	K-mean and VAE	Gas pipeline, SWaT	Centralized	No	No	No	2019
[19]	DNN and DT	Gas pipeline, SWaT	Centralized	No	No	No	2020
[20]	DNN	NSL-KDD	Centralized	No	No	Yes	2018
[21]	Bi-LSTM	HAI	Centralized	No	No	Yes	2021
Ours	VAE and SVDD	SCADA liquid storage infrastructure	Federated	Yes	Yes	Yes	—

Centralized or Federated learning manners, XAI inclusion. Some important performance metrics such as running time and hardware consumption are also considered.

In more detail, for the non purely time-series data type, we can find a range of anomaly detection methods using different algorithms. Recently, in [14], the author proposed a Logical Analysis of Data (LAD-ADS) solution using a rule-based method to detect anomalous behaviours in ICS systems over the Secure Water Treatment (SWaT) dataset [22], which is, in our opinion, not a purely time-series data scenario. LAD-ADS performs detection by extracting rules from a huge of data in the past. However, rule-based systems are often complex and challenging to manage and determine the cause of detected anomalies. In the same scenario of ICSs, [15] proposed a state-aware anomaly detection method that uses the CUSUM (Cumulative Sum) control chart to the state-dependent detection threshold. The training process is done centrally on a large amount of data. In fact, the data in ICSs are often distributed; so using a centralized solution can cause disadvantages such as latency for sending all raw data to the central cloud and huge computer resource consumption for training. Moreover, in CUSUM [15], no detection performance metric such as Accuracy, Precision, Recall, or F1-score was revealed except the false alarm rate. In [12], the authors presented a methodology called MADICS for Anomaly Detection in ICSs using a semi-supervised anomaly detection paradigm with five main steps. The performance of MADICS in terms of Recall is slightly low over its testing dataset. In addition, this mechanism requires a large amount of data for semi-supervised learning, which faces data privacy issues when transmitting a large amount of raw data for training, resource capacity, and computational resources of the system. In terms of detection performance, FedeX is also proved to outperform the previously proposed solutions MADICS [12], LAD-ADS [14] in the same factory contexts. In [13], a statistical window-based anomaly detection method was adopted by using various deep-neural network architectures, showing effectiveness in detecting the attacks in a SWaT infrastructure. However, the authors also indicated that their work needs to be improved with the interpretability of the outcomes and the behaviour detection of fault ICS components.

For the time-series data type, we have observed various proposed AD solutions. Training in a distributed environment

and the privacy of data was also an issue that needs to be addressed in [16]. From the aspect of using Federated Learning to implement an anomaly detection solution in a distributed ICS system, ensuring high accuracy while protecting data privacy, we can find some researches such as [17], [18]. In [17], the author proposed an FL framework that allows decentralized edge devices to cooperate in training an anomaly detector with an attention mechanism-based convolutional neural network long short-term memory (AMCNN-LSTM) model. Although designing an FL-based approach, the experiments lack insight analysis in the performance of deploying such a learning model in an edge environment (i.e in weak hardware of an edge node). Due to the complexity of AMCNN-LSTM caused by using multi-layer CNN and LSTM, according to our experience, it is hard to feasibly deploy such a learning model on edge devices, much less for an expectation of achieving low computing complexity for running the learning model in minute-time scale and low power consumption. With a similar lack of performance testing on the edge hardware, work [18] proposed an FL-based anomaly detection approach for IoT networks based on the combination between Gated Recurrent Units (GRUs) and Long short term memory toward detecting anomalies with decentralized on-device data. However, the performance of the proposed method is not good enough in the distributed scenario; accuracy in each FL client is just around 90% on average.

In several other studies [7], [19], the authors developed and investigated attack detection solutions in ICS cyberspaces. In [19], the authors proposed an attack detection model that uses a Deep Neural Network and a Decision Tree classifier to identify cyber-attacks in the ICS context with an F1-Score of 93.83% with the ICS gas pipeline dataset, which is higher than other algorithms such as SVM, LSTM, Naïve Bayes, Decision Tree (DT), DNN, Random Forest (RF). Study [7] used the semi-supervised techniques by leveraging K-means and Convolutional Autoencoder to protect the ICS system from cyberattack. Like in [19], the experiments of the proposed methods were performed with the gas pipeline dataset and the water storage tank dataset. However, the anomaly detection performance of the proposed method still needs to be improved. In contrast to our study, these studies only focused on evaluating the performance of detection algorithms and did not consider other important metrics when being implemented in the edge environments of an ICS such as detection time and power/memory/bandwidth consumption. In addition, we have found a variety of classic machine learning algorithms for anomaly detection purposes based on the SWaT data. Accordingly, Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-nearest neighbours (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), and Classificationand Regression Tree (CART) are reported by [8]; one-class Support Vector Machines (SVM) and Deep Neural Networks (DNN) are reported by [9]. Similarly, Multi-layer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) are reported by [10]; Long Short-term Memory (LSTM) is reported by [11]. Thus, to have an comprehensive overview of the existing methods, in this paper, we will evaluate our solution and the those reference solutions in terms of detection performance.

Although the studies described above solve challenges surrounding cyber-attack detection in ICSs, all of them have not concerned the interpretability of the model's detected results up to now. As stated in [20], the interpretability of an anomaly detection model is almost as crucial as the prediction accuracy of the model. In the field of explaining the detection outcomes (XAI - Explainable AI), Kasun *et al.* in [20] used a method named Layer-wise Relevance Propagation (LRP) to calculate the input features relevance to explain the trained Deep Neural Network model with DoS attacks detection task. The evaluation is conducted with a subset of NSL-KDD Dataset – an old network intrusion detection dataset released in 1999. Even though the combination of solutions to solve the black-box problem of DNN helps domain experts intuitively access the insight of the DNN algorithms, classification accuracy improvement is required when producing predictions in the test set. Very recently, the authors in [21] have proposed to use XAI to interpret anomaly detection outcomes of the multiple Bi-LSTM learning models in an ICS ecosystem. The scope of the ICS is the smart factory of steam-turbine power generation and pumped-storage hydropower generation. This paper can be considered as the forefront of interpreting anomaly detection in the ICS ecosystem.

III. FEDERATED LEARNING-BASED EXPLAINABLE ANOMALY DETECTION FOR ICS - FEDEX

A. FEDEX OVERVIEW

In this paper, an architecture using FL for anomaly detection is proposed for ICSs named FedeX, standing for **Fed**erated Learning-based **Ex**plainable Anomaly Detection. As Fig.2 shows, ICSs in smart factories can be organized in various zones (i.e. Zone 1, Zone 2, Zone 3…), and each of which is monitored by a local unit (i.e. Edge 1, Edge 2…) to detect anomalies.

Those local monitoring units run an anomaly detection function based on their own incoming local data. In fact, the detection task can be carried out at the edges as long as the task requires a reasonable amount of computing capacity suitable the edge hardware. Such distributed monitoring makes the whole detection process more responsive because the detection process is close to the attack sources. Furthermore, this solution reduces workloads offloading up on the central cloud server as traditional centralized computing architectures do.

As illustrated in Fig.2, the FedeX operation consists of 6 main steps, as follows: Step ①: The edge device uses the sensing data collected from nodes within a zone

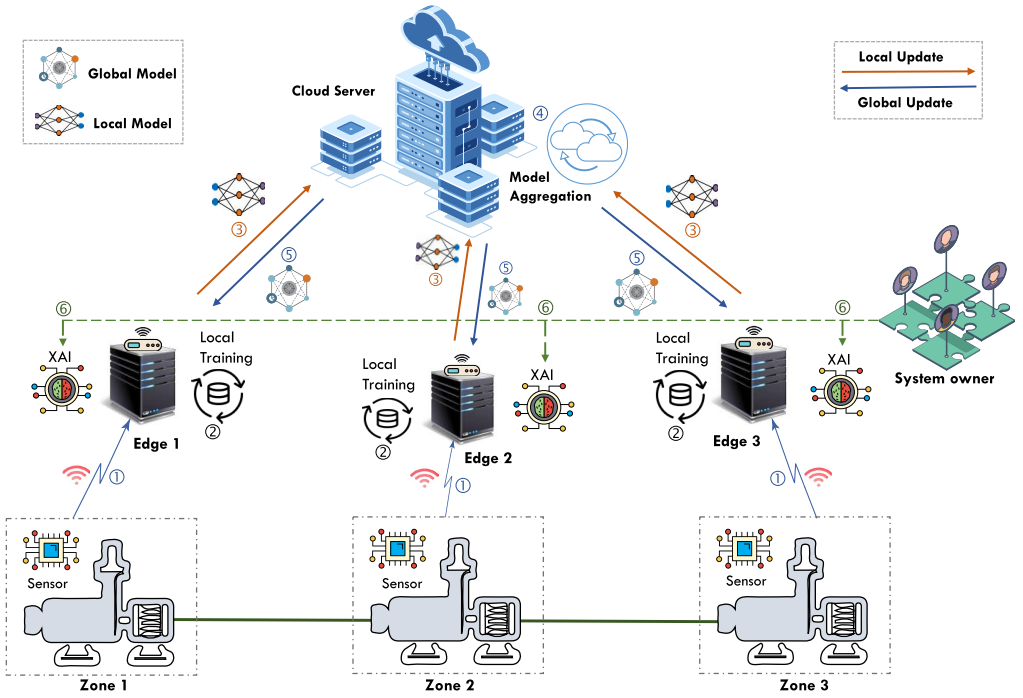


FIGURE 2. Overall FedeX architecture deployed in ICS.

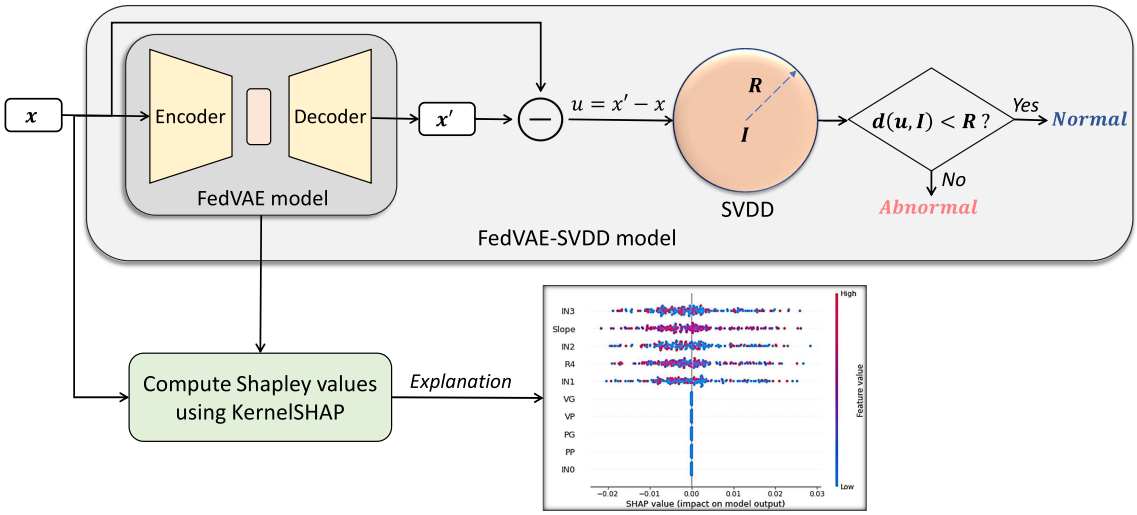


FIGURE 3. The operation of anomaly detection and explanation at each zone.

as a local dataset. Step ②: The edge device performs the local model (i.e., VAE model) and the mechanism for determining a threshold at the last communication round (i.e., FedVAE-SVDD model) training on the local dataset. Step ③: The edge device uploads the weight matrix to the cloud aggregator. Step ④: The cloud aggregator obtains a new global model by aggregating the weights sent by the edge device. Step ⑤: The cloud aggregator sends the new global model to each edge device. The steps above are repeated until the global model achieves optimal convergence. This ideal global model can be used by decentralized devices to conduct anomaly detection tasks. Step ⑥: Periodically, the XAI-SHAP model will be run to interpret and verify the anomaly detection model; and identify the anomaly-causing elements in ICSs.

More specifically, in the detection process, there are two modules that work together at each zone: the local anomaly detection model - FedVAE-SVDD and the explanation module, as shown in Fig.3. Any instance x will be fed to the FedVAE module and reconstructed as output x'. Then the loss

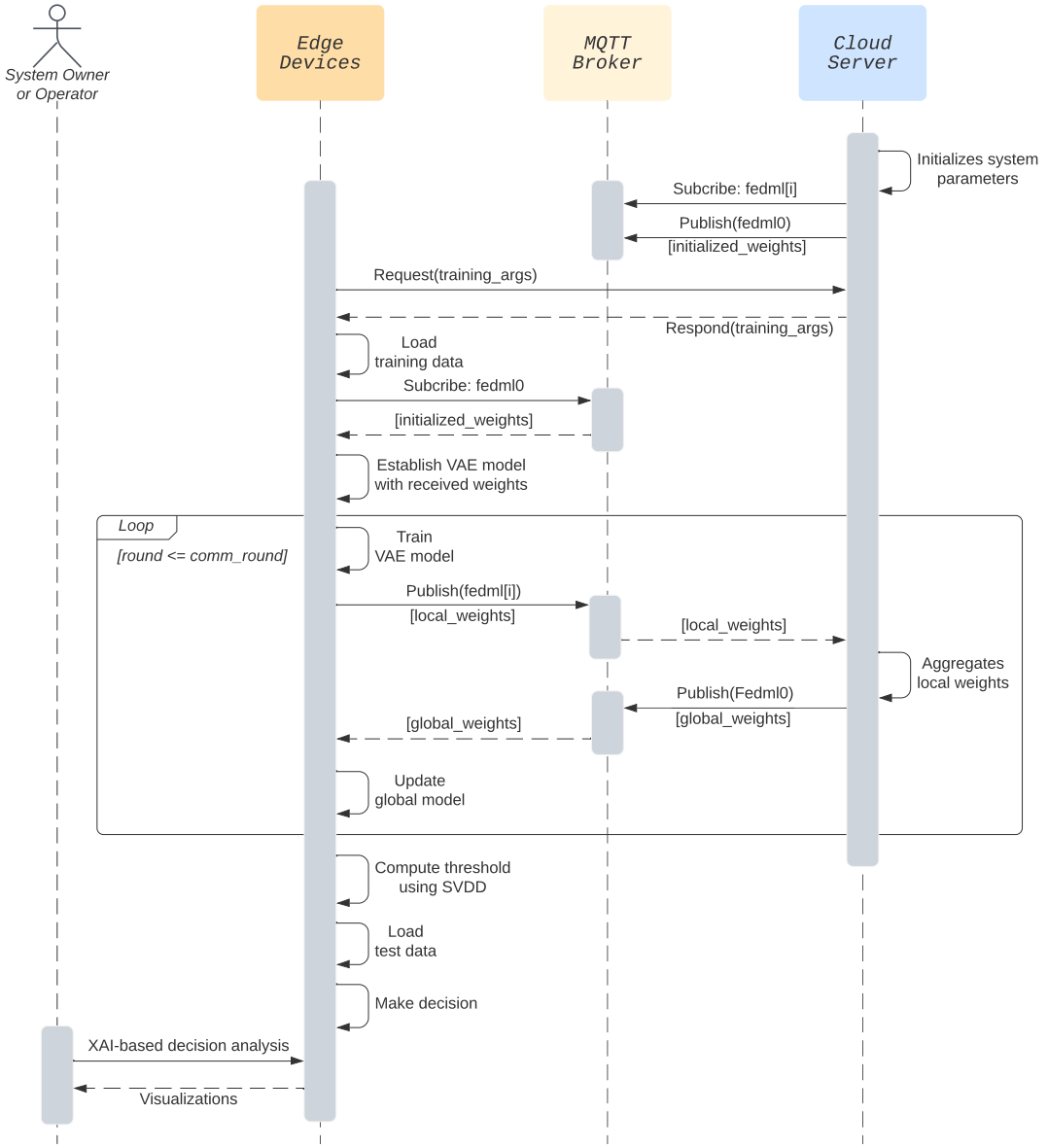


FIGURE 4. Flow chart of the FedeX operation.

value vectors $u = (x' - x)$ will be passed to a trained SVDD model to calculate the distance between it with the centre I of the hypersphere defined in SVDD - $d(u, I)$. Accordingly, based on the comparison between $d(u, I)$ and radius R, the instance x is predicted as an anomaly or normal. The detection results are also explained by the XAI module to define which factor could be the most potential factor of the predicted abnormal instance.

To describe how entities in the FedeX architecture interact with each other in sequence more comprehensively, the procedure from the training process to the testing process is expressed in Fig.4. In the operation of FedeX, the MQTT Broker as a bridge is used for exchanging information between the edge devices and the cloud server.

In the following sections, we will elaborate on our FedeX architecture, including three main phases. The first phase called FedVAE describes the deployment of local models-VAEs based on federated learning, presented in Sections III-B and III-C. The second phase called FedVAE-SVDD implements the mechanism of dynamic threshold - SVDD on FedVAE, described in Section III-D. So, the FedVAE-SVDD model is completed after the first and second phases. Eventually, in Section III-E, FedeX with an explanation module is accomplished by integrating Explainable-Artificial-Intelligent (XAI).

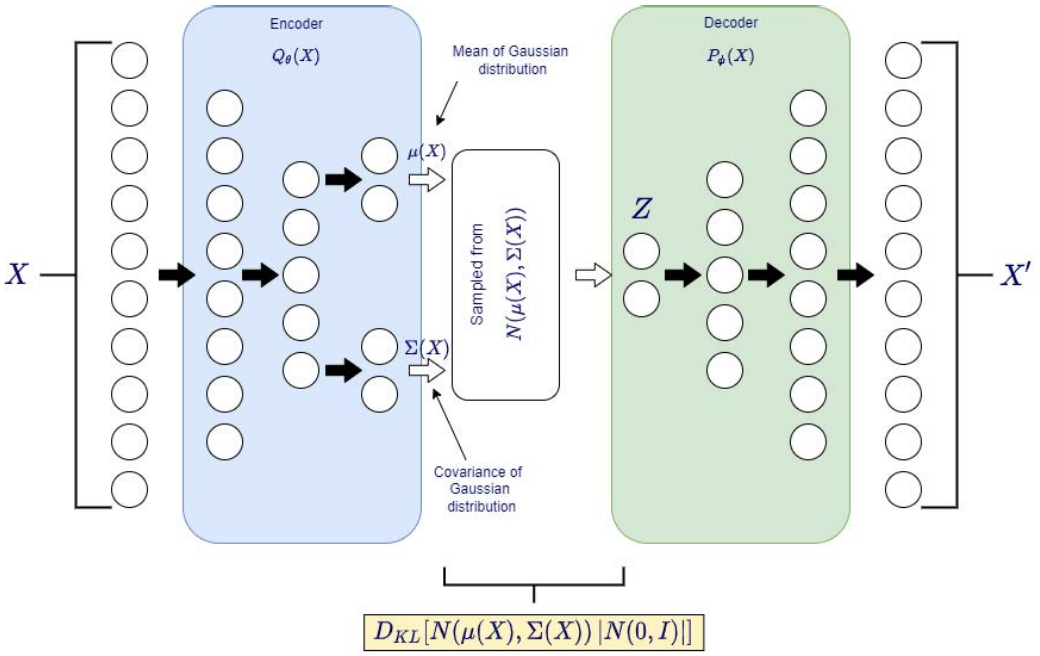


FIGURE 5. VAE Structure.

B. LOCAL TRAINING MODEL ON EDGE - VAE

Since the detection module is implemented on Edge hardware, the overall design is supposed to be lightweight, while still ensuring the detection accuracy requirement. In our design, we try our best to reduce the computing complexity of the algorithms. As elaborated in Figure. 2, the AI-based detection learning model is deployed locally at each single Edge. Then local model updating of each local edge will be done through global updating, powered by Federated Learning.

In this research, we propose to utilize VAE for anomaly detection purposes. VAE is a tuned Autoencoder architecture to run on top of the edge device for efficient anomaly detection. The benefit of VAE is the ability to minimize over-fitting by ensuring that features from its latent space are good enough for data generation. A basic idea here is, if a model was only trained on normal data, then when being encountered with anomalous data, the inability to reconstruct data or, more precisely, the range of the reconstruction error that it entails, can signal the presence of anomalous data.

In fact, many different VAE architectures have been proposed, with different types of layers such as Dense, LSTM, and CNN. We design the VAE encoder and decoder with only two fully connected hidden layers each, because this approach aims to achieve the model's simplicity and lightweight. This lightweight VAE can be trained on top of edge devices with limited hardware resources, while lowering communication costs caused by sending learning models to the cloud in the Federated Learning environment and providing sufficient detection performance. We also emphasize the real-time training guarantee for this model which will be illustrated in Section IV.

For the background, an autoencoder (AE) is a symmetrical-unsupervised neural network, slightly different from other network architectures in that: The VAE network uses the input itself as the ground truth. It consists of 3 main parts: encoder, latent representation, and decoder. Usually, the centre hidden layer has fewer nodes than the input and output layer (a "bottleneck"). Thus, the VAE network learns to compress the input to the bottleneck layer and then from which subsequently restores the input. This middle layer thus becomes the "latent representation" of the input, retaining most information about the input using fewer features. The part of the network before this layer becomes the encoder, and the part after becomes the decoder.

A variational autoencoder (VAE) is a combination of the AE with the Variational Bayesian method. But instead of generating a representation in the hidden space for a data point in the original space, the underlying principle behind VAE is to find a probability distribution for that data point.

The VAE structure is illustrated in Fig.5. Given an input dataset $X = \{x_1, x_2, \ldots, x_n\}$ characterized by an unknown probability distribution $q(X)$. The aim is to approximate this true distribution $q(X)$ by using a parametrized distribution $Q_\theta(X)$ with the parameter θ. Let Z be a random vector jointly distributed with X, which represents a latent encoding of X. Since the computation cost of $Q_\theta(X)$ is high, it is needed to introduce function $P_\phi(Z|X)$, with ϕ defined as the set of real values that parametrize P, to speed up the calculus. This function is to approximate the posterior distribution $Q_\theta(Z|X)$.

For VAEs, the objective is to jointly find a pair of optimal model weights θ^* and ϕ^* through a backpropagation process intended to minimize a differentiable loss function $\mathcal{L}_{(\theta,\phi)}$.

$$(\theta^*, \phi^*) = argmin_{\theta,\phi}\mathcal{L}_{(\theta,\phi)}(X) \quad (1)$$

$\mathcal{L}_{(\theta,\phi)}$ is the evidence lower bound (ELBO) loss function is defined as follows:

$$\mathcal{L}_{(\theta,\phi)} = -log(Q_\theta(X)) + D_{KL}(P_\phi(Z|X)\,||\,Q_\theta(Z|X)) \quad (2)$$

where:

$Q_\theta(X)$: The probability distribution to characterize the input dataset X

$P_\phi(Z|X)$: Function to approximate the posterior distribution $Q_\theta(Z|X)$

$D_{KL}(P_\phi(Z|X)\,||\,Q_\theta(Z|X))$: Distance between two distributions $Q_\theta(X)$ and $P_\phi(Z|X)$, calculated by using the Kullback-Leibler divergence function as follows:

$$D_{KL}(P_\phi(Z|X)\,||\,Q_\theta(Z|X)) = \int P_\phi(Z|X) log \frac{P_\phi(Z|X)}{Q_\theta(Z|X)} dZ \quad (3)$$

Because of the stochastic sampling in the latent space, we cannot directly do the backpropagation process through the neural network. Therefore, the parameterization trick should be used to overcome this problem. Using this parameterization method, the distance in formula (2) can be rewritten as follows:

$$D_{KL}[N(\mu(X), \Sigma(X))\|N(0,1)] = \frac{1}{2}\sum_k \Big(\exp(\Sigma(X)) + \mu^2(X) - 1 - \Sigma(X)\Big) \quad (4)$$

where

k is the dimension of the Gaussian distribution

$\Sigma(X)$: Covariance of the Gaussian distribution

$\mu(X)$: Mean of the Gaussian distribution

After the VAE learning model converges, X' is the output of VAE. In other words, X' is the reconstructed input of input X.

C. FEDERATED-LEARNING FRAMEWORK - FEDVAE

We expand the anomaly detection model to a Federated Learning (FL)-based framework to solve the problem of missing training data at each edge device when deep learning models often need large amounts of data to train. With the FL technique, the central cloud can federate information with different characteristics from various zones to improve the detection performance of the overall network without the need of having knowledge of original raw data.

The Federated-Learning based VAE model (or called FedVAE) is described in Algorithm 1. In FedVAE, each edge device performs the training and detection process with local data from each manufacturing area, and an Edge device only sends information of the weight matrix of the trained model to the cloud server, rather than sending the entire raw data, as a traditional cloud-based training system would do. Although the cloud has the storage and computing power to manage the volume of data generated in manufacturing, the computationally intensive operations and vast data storage hosted in cloud servers may cause a delay. Because this delay is caused by the time required to send, transfer, and process massive amounts of data from IoT devices at production sites. This is a significant issue in a smart factory that must undertake huge monitoring and detection in real-time. Within this context, the concept of Edge-Cloud Computing combined with FL shall arise to circumvent this constraint.

- Firstly, the initial model is created by the Cloud Server as a Weight "Federator".
- The VAE model was then applied to solve anomaly detection. It then subscribes to numerous MQTT topics to which the zones will send the weights of their models.
- After the first model's weights are published to the aggregated model topics, the Cloud Server awaits requests from the VAE model configuration from each zone.
- Local models are trained at each edge based on their own dataset.
- In each communication round, the weights of the trained models $\omega_r^{(c)}$ are sent to the Cloud Server for FL.

Algorithm 1: Phase-1: FedVAE

Input: Initial model ω_0

Output: *VAEcomplete* - Trained VAEs model in each client

ρ - the number of local epochs;

Rounds - the number of communication rounds;

C - number of zones;

ι - learning rate;

$\beta_1, \beta_2 \in [0, 1)$ - hyper-parameters;

$\hat{\epsilon}$ - a very small value;

for $r = 0$ **to** $Rounds - 1$ **do**

  Server updates ω_r to C zones;

  for *node* $c \in C$ **do**

    $\omega_{r,0} \leftarrow \omega_r$;

    for $t = 0$ **to** $\rho - 1$ **do**

      Update (θ, ϕ) to minimize $\mathcal{L}_{\theta,\phi}$ using Adam's algorithm:

      $\theta_{r,t+1}^{(c)} \leftarrow \theta_{r,t}^{(c)} - \iota \cdot \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \cdot \frac{\mathbb{E}[\nabla_\theta \mathcal{L}_{\theta,\phi}]_t}{\sqrt{\mathbb{E}[(\nabla_\theta \mathcal{L}_{\theta,\phi})^2]_t + \hat{\epsilon}}}$

      $\phi_{r,t+1}^{(c)} \leftarrow \phi_{r,t}^{(c)} - \iota \cdot \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \cdot \frac{\mathbb{E}[\nabla_\phi \mathcal{L}_{\theta,\phi}]_t}{\sqrt{\mathbb{E}[(\nabla_\phi \mathcal{L}_{\theta,\phi})^2]_t + \hat{\epsilon}}}$

      ▷ $\mathbb{E}$ is the expected value

    end

    send $\omega_r^{(c)} = [\theta_r^{(c)}, \phi_r^{(c)}]$ to the server;

  end

  server calculates $\omega_{r+1} \leftarrow \frac{1}{C}\sum_{c\in C} \omega_r^{(c)}$;

end

VAEcomplete $\leftarrow \omega_{Rounds-1}$;

return *VAEcomplete*

- The Cloud then uses the formula (5) to calculate the weight of the federated global model:

$$\omega_{r+1} = \frac{1}{C}\sum_{c=1}^{C}\omega_r^{(c)} \tag{5}$$

where: C is the number of zones. $\omega_r^{(c)}$ is the weight of the local model of zone c at round r. ω_{r+1} is the federated global model's weight at round $r+1$.

- Finally, the weight from the federated global model is sent downward to update the local model of each zone.

In fact, as described in Algorithm 1, an edge will need to send the weight matrix to the cloud in many iterations during the back propagation process until the VAE model converges. Each iteration will cause a certain bandwidth occupation (or communication cost) on the edge-cloud link. In order to reduce this communication cost, we design the Federated Learning environment as follows:

- Each edge runs ρ local epochs during the back propagation process to minimize loss function $\mathcal{L}_{\theta,\phi}$.
- After every ρ local epochs, the edge sends a matrix of weights to the cloud server for global model aggregation. It is called one communication round.
- The process is repeated until the VAE model is considered as converged.

Note that, for the learning model to converge quicker, the Adam's algorithm [23] is used to update the parameters (θ, ϕ) during the gradient descent process.

D. AUTOMATIC THRESHOLD DETERMINATION - FEDVAE-SVDD

In this subsection, we will describe the automatic mechanism to determine a threshold for the Anomaly Detection model (FedVAE) to work efficiently. In order to do that, Support Vector Data Description (SVDD) is deployed to go over this request accurately while keeping the real-time assurance.

Usually, experts in the industry establish the threshold after attempting a range of values, then select the one that best balances the requirements (performance, true positive, or false negative,.etc). SVDD also works well as an outlier detection algorithm, especially with high-dimensional datasets, but just like all SVMs, it does not scale to large datasets. Therefore, we suggest a combination of FedVAE and SVDD, as a moderate addition: FedVAE serves as the main anomaly detection model for the distributed system, while SVDD, trained with a small set of error vectors from the output of the FedVAE model, can correspond to finding a small region that encompasses all instances.

SVDD is a type of support vector method used for single-class classification and outlier detection. The primary idea behind SVDD is to wrap samples in a high-dimensional space with the smallest volume. For the anomaly detection task in which most of the collected data are normal, the hypersphere is usually taken as the boundary around normal samples, separating them from outliers.

The process of determining a threshold is depicted in Fig.3 and Algorithm 2 - called Phase 2. More specifically, at the end of Phase 1 (i.e., FedVAE), the pairs of original input X and the reconstructed input X' then return a set of loss value vectors $V = (X' - X)$. We use these vectors V as an input to train the SVDD model.

Given a set of training data samples $V = \{v_1, v_2, \ldots, v_n\}$, we need to find the centre I and radius R of a hypersphere to achieve the mimimum volume that could contain all data samples within. The condition should be satisfied as follows:

$$(v_i - I)^T(v_i - I) \leq R^2 \tag{6}$$

where:

$v_i \in V, i = 1, \ldots, n$ represents the training data

R: the radius that represents the decision variable

I: the center, a decision variable

The above optimization problem can be solved by solving the equivalent optimization problem using the Lagrange multipliers, as follows:

$$Max \sum_{i=1}^{n}\alpha_i(v_i \cdot v_j) - \sum_{i,j=1}^{n}\alpha_i\alpha_j(v_i \cdot v_j). \tag{7}$$

$$s.t \;\; 0 \leq \alpha_i \leq \chi \quad and \quad \sum_{i=1}^{n}\alpha_i = 1 \tag{8}$$

where:

$\alpha_i \in \mathbb{R}, \;\; i = 1, \ldots, n$ are the Lagrange coefficients.

$\chi \in (0, 1]$ is a penalty constant.

The position of a data sample v_i relative to the hypersphere induces to the following condition of Lagrange coefficient α_i

- Position of Centre I:

$$\sum_{i=1}^{n}\alpha_i v_i \tag{9}$$

- Position inside the hypersphere boundary:

$$\|v_i - I\| < R \rightarrow \alpha_i = 0 \tag{10}$$

- Position at the Boundary:

$$\|v_i - I\| = R \rightarrow 0 < \alpha_i < \chi \tag{11}$$

- Position outside the hypersphere boundary:

$$\|v_i - I\| > R \rightarrow \alpha_i = \chi \tag{12}$$

The circular data boundary can include an amount of very sparse distribution of training observations space that can increase the probability of false positives.

Moreover, SVDD becomes more flexible by replacing the inner product $(v_i \cdot v_j)$ with an appropriate kernel function $K(v_i, v_j)$, which actually does not change the results of statement from Formula (9) to Formula (12).

Then the threshold R to detect which is normal and which is abnormal is calculated by using a Kernel function $K(.)$ as follows:

$$R = \sqrt{K(v_l, v_l) - 2\sum_{i}\alpha_i K(v_i, v_l) - \sum_{i,j}\alpha_i\alpha_j K(v_i, v_j)} \tag{13}$$

Algorithm 2: Phase-2: FedVAE-SVDD

Input: X - NormalTrainData;
χ - penalty constant;
$K(.)$ - Kernel function;
Reconstructed Data X';
Set of error vectors $V = X' - X$;
V' is a subset of V;
Output: Threshold;
for $v_i, v_j \in V$ **do**
  if $(\alpha_i < \chi \ and \ \alpha_j < \chi)$ **then**
    calculate
    $R = \sqrt{K(v_l, v_l) - 2\sum_i \alpha_i K(v_i, v_l) - \sum_{i,j} \alpha_i \alpha_j K(v_i, v_j)}$;
    ▷ $v_l \in V'$, satisfying $\alpha_l < \chi$
  end
end
Threshold = R;
return *Threshold*;

using any $v_l \in V'$ where V' is a subset of V that satisfies the condition $\alpha_l < \chi$.

For the selection of kernel function, any function that meets the Mercer condition can be used as a kernel function. Some commonly used kernels can be listed as such as the Gaussian kernel, Exponential kernel, and Laplacian kernel. Among of which, the Laplacian Kernel is completely equivalent to the exponential kernel, except for being less sensitive for changes in the σ parameter. In our case study, we decide to use the Laplacian kernel since it allows the SVDD model to run faster.

1) LAPLACIAN KERNEL

$$K(v_i, v_j) = exp(-\frac{\|v_i - v_j\|}{\sigma}) \tag{14}$$

where, σ is the width constant ($\sigma > 0$).

Finally, for each new arriving instance x, as depicted in Fig.3, the incoming data $u = x' - x$ is fed into the trained SVDD model to define whether x is a normal sample or not. The distance between u and the centre I of the hypersphere - $d(u, I)$ - can be calculated as follows:

$$d(u, I) = \sqrt{K(u, u) - 2\sum_i \alpha_i K(v_i, u) + \sum_{i,j} \alpha_i \alpha_j K(v_i, v_j)} \tag{15}$$

If $d(u, I) > R$, then x is indicated as an anomaly.

E. EXPLAINABLE ARTIFICIAL INTELLIGENCE - FEDEX

In anomaly detection, although algorithms related to neural network models tend to be more beneficial than signature-based methods, its drawback is insufficient interpretability. Therefore, the reason why an instance is predicted to be abnormal can not be easily discovered in such cases. This renders researchers time-consuming and vague in analyzing model-predicted anomalies, causing the reliability of anomaly detection models to become degraded. To overcome this limitation, a widely-used approach called Explainable Artificial Intelligence (XAI) can be adopted.

The aim of XAI is to assist humans to understand the results of solutions using black-box models by the assessment of feature attributions, thereby demonstrating how much each feature participated in making a decision for each data point of the model. With simple machine learning (ML) models, such as logistic regression and linear regression, the importance of features can be assessed via the coefficient of each feature in the data set. Meanwhile, as aforementioned, for several complicated models related to neural networks such as VAE, it is difficult to measure or compute the influence level of each feature on an output decision. Because there is simply a large number of parameters engaging in this model. In fact, with the advent of some state-of-the-art XAI frameworks, this problem has been handled.

There are several effective XAI frameworks such as Local Interpretable Model-agnostic Explanations (LIME) [24] and Deep Learning Important Features (DeepLIFT) [25]. However, within the scope of this study, the main XAI approach is based on SHapley Additive exPlanations (SHAP) [26] using the Shapley values, which comes from the theory of the cooperative game.

As the average marginal contribution of a player across all possible coalitions, the Shapley value tells us the payout to which the player is allocated fairly in a game. For a decision made by a black-box model, each feature of a data point can be considered as a player.

Mathematically, the Shapley value of a feature in a learning model can be generally defined as:

$$\xi_n(f, a) = \sum_{\mathcal{P} \subseteq a'} \frac{|\mathcal{P}|!(m - |\mathcal{P}| - 1)!}{m!} [f_a(\mathcal{P}) - f_a(\mathcal{P} \setminus n)] \tag{16}$$

where,

ξ_n is the Shapley value of feature n
f is a "black-box" model that needs explaining
a is an input datapoint
a' is the simplified data input, which maps to a via a particular function mapping h_a that satisfies $a = h_a(a')$.
$\mathcal{P}$ is one of all possible subsets of feature set, considered as a coalition
m is the number of features in the dataset

Due to the fixed input size, commonly, the features of a model omitted in the Eq.(16) are substituted with random input values from the background dataset. It can be seen that the total possible subsets of an m-feature set used for interpretation is 2^m, which leads to the massive complexity of computing Shapley values if m increases. Therefore, to deal with this issue, without calculating all combinations, Kernel SHAP [26] can be employed by sampling feature subsets and

then fitting them into a linear regression model:

$$Y = \gamma_0 + \gamma_1 a_1 + \gamma_2 a_2 + \gamma_3 a_3 + \ldots \gamma_m a_m \quad (17)$$

In this linear regression model, the variables a_i ($i = 1, 2, \ldots, m$) are features encoded according to their presence ($a_i = 1$) or absence ($a_i = 0$), and the output value or label is the prediction value of the model f. After the training, the coefficients γ_i can be interpreted as approximated Shapley values.

In this work, our FedeX architecture is applied with SHAP to explain the impact level of features on the anomalies that are predicted by the FedVAE-SVDD model, via their SHAP values. As depicted in Fig.3, the Kernel SHAP takes the FedVAE model and the test data as inputs to construct a local linear regression explanation model. Subsequently, the explanatory model computes the SHAP values of classified anomalies and displays them visually. As feature values are measured by sensors, by using this explanation, operators or domain engineers can easily determine the sensors likely causing the abnormality and make a faster detection response. Through such actual maintenance, experts can validate and trust the proposed anomaly detection model more. These will be demonstrated in the case study of a SCADA liquid storage infrastructure in Section IV-C2.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the FedeX architecture in various aspects. From the detection performance perspective, the FedVAE-SVDD learning model is evaluated in comparison with different cutting-edge solutions for Anomaly detection in ICSs. The resource requirement of FedVAE-SVDD over the embedded edge device is also taken into account. Moreover, the results of using the XAI-SHAP technique to interpret the predicted results of FedVAE-SVDD are also described with the main case study in a SCADA liquid storage infrastructure dataset [27].

A. ICS CASE STUDY

In our main case study, we consider the SCADA liquid storage infrastructure dataset [27], which simulates a fuel storage system supplying an automated production line monitored by an ICS system. The high-level overview of the testbed system is shown in Fig. 6.

As depicted in Fig 6, the system is composed of the main tank and secondary tank with a capacity of 9 and 7 liters, respectively. Data is collected by connecting the sensors to a PLC. Four discrete sensors in the main tank (corresponding to features IN0, IN1, IN2, and IN3) and one in the secondary tank (corresponding to features R4 and Slope) are used to measure the level of fuel. Pump1 and Pump2 control the flow of fuel between two tanks, represented by the pair feature PG-VG and PP-VP, respectively. PLC registers 2 through 4 provided output data elaborating the state of the system used for analysis. Register 2 contains the bits that indicate the discrete sensors' binary status. To extract the state of each sensor separately, a population count can be performed on

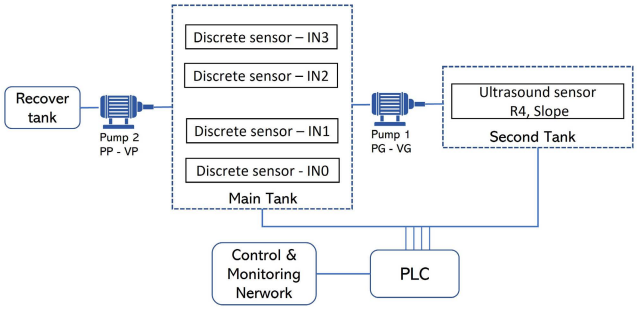


FIGURE 6. High level architecture of the SCADA liquid storage infrastructure system.

the register. Register 3 holds the pump's active or inactive state, whereas Register 4 holds the ultrasound sensors' step value from 0 to 10,000. (e.g. Step 3,000 represents 2.1 liters of liquid in the tank).

As described in [27], the data set consists of 14 distinct scenarios. Each scenario includes one of 5 operational situations (such as sabotage, breakdown, accident, or cyber-attack) as well as 6 affected components. The affected components are those parts of the system that are directly impacted by the abnormality.

B. EXPERIMENT SETUP

To implement the proposed FedeX architecture, we set up a small-scale testbed as follows:

- 4 Raspberry-Pi-4-Model-B kits acting as edge devices; Raspberry-Pi-4 equipped with quad-core 1.5 GHz ARM Cortex-A72 processor and 4 GB RAM with 32-bit Raspbian OS.
- 1 Dell Precision 3640 Tower Workstation serves as Cloud Server; the workstation with Intel Core i7–10700K 3.8 GHz (up to 5.1 GHz), 16 GB RAM, working on Linux operating system.
- All edge devices and the Cloud Server are connected by a router through a WIFI interface.

At the edge devices (i.e. Raspberry-Pi-4), we implement our FedeX framework in Python 3 with the TensorFlow 2 platform, which is built with the support of the FL framework - FedML [6]. In the FedeX architecture, the edge devices and cloud server exchange the weights and bias matrix of the VAE model using the standardized MQTT protocol for an IoT environment [28]. EMQ X Broker (2021) is hosted on the cloud server as an MQTT broker for better long-term performance. We discover EMQ X Broker as the most scalable open-source broker that could accept more advantageous devices linked to the server.

As our proposed architecture leverages edge computing, it is also important to assess the edge efficiency during training in the ICS context. For this purpose, on edge devices, we utilize tool *bmon* to measure bandwidth occupation on a upstream link to Cloud Server, tool *resmon* to monitor computational resources of the edge devices. Besides, we measure power consumption with the support of an external

monitoring gadget *UM25C USB Tester* which is directly connected to Raspberry Pi 4.

The experimental data set has 10 features, namely IN0, IN1, IN2, IN3, R4, Slope, PG, VG, PP, and VP. The characteristics of the features presented in Subsection IV-A, in which data is considered as non-purely time series data. We use the normal data to perform the training model for all scenarios.In order to simulate data of the 4 distributed zones, we split the original data set into 4 independent subsets, each of which is used for local training at each edge device. To evaluate the model performance, 4 separate test sets are split from the original test set, containing both normal and abnormal data points.

For detection performance evaluation, we further perform training with the centralized version of the proposed solution.

As for the centralized FedVAE-SVDD implementation, the algorithm is also written in Python 3 using the TensorFlow 2 platform. In these centralized settings, the full original data set is used for training at the Cloud Server (i.e., Dell Workstation) in our testbed.

C. FEDEX PERFORMANCE EVALUATION

1) DETECTION CAPABILITY

To evaluate the detection performance of the FedVAE-SVDD model in each distributed edge zone, the common detection metrics such as F1 score, Accuracy, Recall, Precision are measured. These metrics can be defined in short as follows:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (18)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePostive} \quad (19)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (20)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (21)$$

where:

- *TruePositive*: number of outcomes correctly predicted as positive.
- *FalsePositive*: number of wrong predictions of actual negative as positive.
- *FalseNegative*: number of wrong predictions of actual positive as negative.

The detection performance is evaluated in 2 main scenarios based on a single run only.

- Scenario 1: FedVAE-SVDD versus its centralized counterpart in our main case study of ICS (i.e. the SCADA liquid storage infrastructure dataset [27])
- Scenario 2: FedVAE-SVDD and its centralized counterpart versus other previously proposed AD solutions in our main case study of ICS and different SCADA datasets.

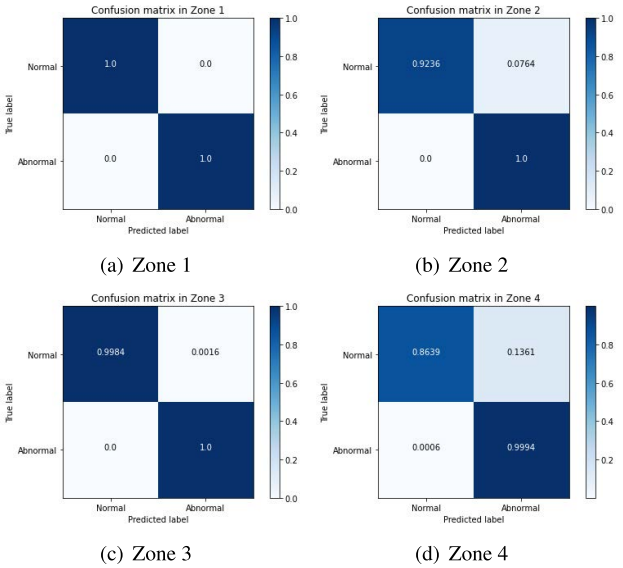
Also note that, since the SVDD model is used to automatically determine an optimal detection threshold for the FedVAE model on each different training dataset. In our experiments, the optimal thresholds found for 4 distributed zones (i.e., Zone 1, Zone 2, Zone 3, and Zone 4) are 0.11, 0.09, 0.09, and 0.09 respectively. In case of centralized learning (i.e., the whole original SCADA data set is used), the threshold is found 0.26. Since the learning model converges after 3 communication rounds, the results shown in the following subsections are retrieved after the 3 rounds.

a: SCENARIO 1: FEDVAE-SVDD VS. ITS CENTRALIZED VAE-SVDD

In this scenario, we measure the detection performance of our FedVAE-SVDD solution in 4 separated manufacturing zones. The detection performance is also measure for the Centralized VAE-SVDD in which the training process is supposed to be carried out at the Central Cloud. The results are shown in Table. 2.

TABLE 2. FedVAE-SVDD performance measured in 4 zones vs. Centralized VAE-SVDD over the SCADA liquid storage infrastructure dataset [27].

	Zone 1	Zone 2	Zone 3	Zone 4	Centralized
Threshold	0.11	0.09	0.09	0.09	0.26
Accuracy	1	0.9587	0.9992	0.9210	0.9017
Precision	1	0.9237	0.9985	0.864	0.9059
Recall	1	1	1	0.999	0.9806
F1	1	0.96	0.9992	0.9269	0.9418
AUC	1	1	1	0.92	0.9



(a) Zone 1 (b) Zone 2

(c) Zone 3 (d) Zone 4

FIGURE 7. Normal-Abnormal confusion matrix for FedVAE-SVDD measured over the SCADA liquid storage infrastructure dataset.

In fact, besides the benefits of distributed learning, we have always thought about trade-offs of its detection performance. However, as we can see, in our ICS main case study, the hybrid FedVAE-SVDD solution even outperforms the Centralized learning manner. FedVAE-SVDD reaches ideal results at zone 1, the max Precision of 0.9985, Recall of 1, F1-score of 0.9992, AUC of 1, Accuracy of 0.9992 at the rest of zones; while the centralized learning achieves

TABLE 3. Comparison among different AD methods over two ICS datasets: SCADA liquid storage infrastructure and SWaT.

Method	SCADA liquid storage infrastructure dataset [27]				SWaT dataset [22]		
	Accuracy	Precision	Recall	F1-score	Precision	Recall	F1-score
LR [8]	0.87	0.78	0.51	0.49	0.9953	0.722	0.8369
LDA [8]	0.88	0.87	0.53	0.53	0.9953	0.7254	0.8392
KNN [8]	0.91	0.85	0.7	0.75	**0.9978**	0.7711	0.8699
CART [8]	0.94	0.86	0.86	0.86	0.9684	0.8847	0.9247
NB [8]	0.67	0.63	0.8	0.6	0.96	0.7338	0.8318
SVM [8], [9]	0.91	0.9	0.68	0.74	0.925	0.699	0.796
DNN [9]	0.897	0.863	**1**	0.926	0.9829	0.6784	0.8028
MLP [10]	0.887	0.851	**1**	0.92	0.967	0.696	0.812
CNN [10]	0.857	0.819	**1**	0.9	0.952	0.702	0.808
RNN [10]	0.816	0.806	0.943	0.869	0.936	0.692	0.796
LSTM [11]	0.906	0.873	**1**	0.932	0.986	0.698	0.8175
MADICS [12]	0.912	0.885	0.992	0.935	0.984	0.75	0.851
1D CNN [13]	0.87	0.831	0.987	0.902	0.968	0.791	0.871
LAD-ADS [14]	—	—	—	—	0.939	0.891	0.914
FedVAE-SVDD @Zone 1	**1**	**1**	**1**	**1**	0.942	0.9999	0.97
FedVAE-SVDD @Zone 2	0.9587	0.9237	**1**	0.96	0.9718	**1**	**0.9857**
FedVAE-SVDD @Zone 3	0.9992	0.9985	**1**	0.9992	0.9427	**1**	0.9705
FedVAE-SVDD @Zone 4	0.921	0.864	0.999	0.9269	0.9433	**1**	0.9708
Centralized VAE-SVDD	0.9017	0.9059	0.9806	0.9418	0.9751	0.9962	0.9855

0.9059, 0.9806, 0.9418, 0.9, and 0.9017 respectively. It can be explained that Federated learning offers the improvement of generalizability of the VAE-SVDD model through the collaboration of multiple edge devices by taking advantage of separate data sources when compared to a single global model under data heterogeneity. FL eliminates a single point of failure due to its distributed nature. This can be considered as an advantage of Decentralized Learning, so the results in comparison with the Centralized learning version are slightly higher.

Considering the performance of FedVAE-SVDD only, we can see that all detection metrics are very good. Only Precision in Zone 4 gets a bit low at 0.864. However, in a smart factory, even the smallest abnormal incident can adversely affect the entire factory. So in general, we need to avoid discarding anomalies (i.e. Recall is important) and accept that sometimes the model can miss detecting a normal sample to be abnormal (i.e. Precision). Because engineers can easily test it and then operate the factory properly. Therefore, the Recall results of our model prove that this model can be a very good candidate to be deployed in a smart factory.

To deeper investigate the detection performance, a confusion matrix is shown for the proposed one-class classifier across two classes: normal and abnormal. As illustrated in Fig. 7, FedVAE-SVDD achieves high TP (True Positive) of 1 and TN (True Negative) of 1 at maximum, as well as very small FN (False Negative) in most of the zones (just around 0.0016, 0.0764, 0.1361). This means that our architecture is able to detect even very small anomalies within an IIoT-based industrial control system.

b: SCENARIO 2: FEDVAE-SVDD VS. OTHER ANOMALY DETECTION SOLUTIONS

In this experiment scenario, we will study the performance of FedeX in comparison with 14 other reference solutions comprehensively, including: machine-learning reference methods is reported by [8], namely LR, LDA, KNN, CART, NB, and SVM; SVM and DNN reported by [9]; MLP, CNN, and RNN reported by [10]; LSTM reported by [11]; MADICS [12] based on LSTM, 1D-CNN [13], and LAD-ADS [14] using the rule-based method. Our experiments are run with 2 cases: with the SCADA liquid storage infrastructure dataset [27]), and (2) with the well-known SWaT dataset [22]. Since SWaT has been considered as an imbalanced dataset, the accuracy metric is not preferred in most of the recent researches as we have investigated.

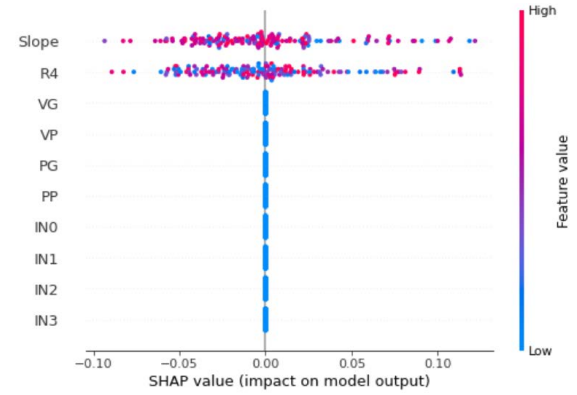
The performance of all 14 other AD solutions, our solution and its centralized learning counterpart can be seen in Table 3, in which FedVAE-SVDD outperforms 13 other reference methods on the SCADA liquid storage infrastructure dataset in all metrics, even with its centralized counterpart (i.e the centralized VAE-SVDD). Only the performance of LAD-ADs on the SCADA liquid storage infrastructure dataset has not been reported since its code is not available and can hardly be reproduced the same as the original paper. Therefore the performance of LAD-ADs can only be compared over the Swat dataset with the same experiment setting. But overall, the results show that our learning model is suitable for such a case study.

On the well-known SWaT dataset, FedVAE-SVDD obtains the highest performance in comparison with all classic and deep learning references in terms of Recall and F1-score. Again, let us note that Recall and F1-score are the 2 important metrics for ICSs. In the case of this imbalanced dataset, a good F1-score figure is necessary since the number of abnormal samples is so much different from the number of normal samples. Accordingly, both FedVAE-SVDD and its centralized counterpart are prominent with the maximum F1-score of 0.9857 and 0.9855, respectively. Besides, our solution performance is also higher than the average of

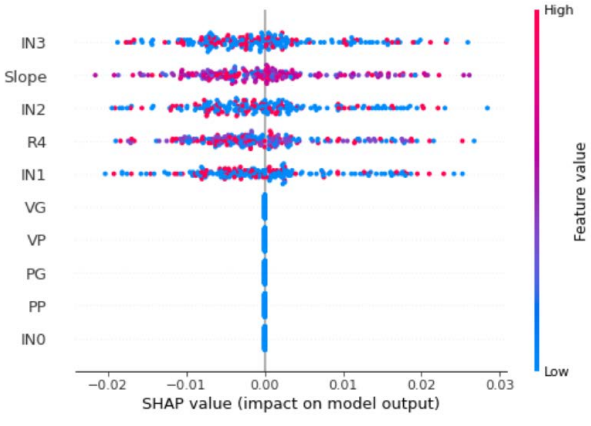
the remaining references regarding to the Precision metric. In particular, it is worth mentioning that LAD-ADS is, perhaps, one of the best methods in recent researches over the SWaT dataset, with F1-score of 0.914. But FedVAE-SVDD still outperforms it in this scenario.

2) EXPLAINABLE AI

Although the above results demonstrate that FedeX achieves good anomaly detection performance, we want to investigate the reasons why they are predicted so. Since our case study is based on the data set gathered in a liquid storage infrastructure [27] as described in Section IV-A, we expect that FedeX could support domain engineers quickly and visually in finding and checking abnormal behavior of those sensors or actuators. Therefore, SHAP is employed to identify how features contribute to the anomalies predicted by the FedVAE-SVDD model. Thanks to this, decisions and priorities in checking and maintaining systems can be made effectively, allowing operators to save more time.



(a) Explanation for detected anomaly 1



(b) Explanation for detected anomaly 2

FIGURE 8. Summary plot of SHAP values.

From a practical perspective, anomalies can arise from various threats such as accidents, sabotage, breakdown, and cyber-attack. This promotes us to perform two explanation scenarios, where SHAP is employed to explain two sets, corresponding to two different intervals, drawn randomly from anomalous samples predicted in the test set at Zone 1. The results of both scenarios are visualized in Fig.8, a summary plot for the distribution of SHAP values over whole computed data points, pointing out the importance of features through their impact. In the visualizations, the dots in each feature correspond to the SHAP values of each data point, accumulating up to depict density. The position on the x-axis is denoted by the Shapley values and on the y-axis by the features ordered as per importance. Besides, the value of the features from low to high is displayed by color gradation. As depicted in Fig. 8(a), with Shapley values in the range of from -0.1 to above 0.1, Slope and R4 are two critical features, while the other features do not contribute to the anomaly. Consequentially, it can be inferred that the ultrasound sensor which measures the physical values of the R4 and Slope features may be broken down. This incident can come from some weather factors like humidity. Therefore, by checking the ultrasound sensor quickly, domain engineers can make reasonable solutions, without verifying other physical components in the system. On the other hand, if the sensor still works properly, i.e., false alarm occurs, the operator can consider retraining the model for higher anomaly detection accuracy. For the remaining scenario, Fig. 8(b) shows that Shapley values range from -0.02 to 0.03, and IN3 is the most crucial feature; while Slope, IN2, R4, and IN1 have a remarkable influence on the anomaly. Based on these signs, as an engineer, we could determine that the anomaly is most likely to arise from sabotage impacting physical components such as the discrete sensors in the main tank and the ultrasound sensor.

In both of these scenarios, SHAP suggests that the R4 feature has a significant impact on predicted anomalies, similar to the analyses mentioned in a SCADA dataset research [29]. The authors confirm that the ultrasound sensor badly affects most of the abnormal scenarios in the dataset and the value of R4 measured by this sensor is most significant. Accordingly, it can be seen that our XAI-based explanatory solution is capable of precisely identifying the primary cause related to the anomaly in reality.

In conclusion, based on these positive findings, we would like to make some comments and recommendations. Firstly, our scheme can make a comprehensive explanation for detected anomalies, boosting the reliability of FedeX. Besides, if there are the occurrence of unknown threats, FedeX will still support operators to determine affected physical components and come up with timely responses rather than inspecting the entire system. This issue may not be solved by other multi-class classification-based anomaly detection solutions. Furthermore, based on data records, we recommend that domain engineers should run SHAP periodically, for example, once per week, to check and schedule system maintenance depending on attack types, or to retrain the model for higher detection performance.

3) EDGE COMPUTING CAPACITY

Deploying a learning model at the edge is challenging due to the limited capacity of embedded devices. Therefore,

to get insight into the efficiency and feasibility of the FedeX architecture, we conduct a few experiments for the FedVAE-SVDD training phase to evaluate the edge performance during the training, based on some metrics such as bandwidth consumption, model running time, power consumption, CPU usage, and memory usage.

In practice, with a vast amount of training data, the training process could burden such resource-constrained edge devices. While the detection process for each single incoming data sample leaves no significant impact, because the learning model has been already exported for the security system to use. Hence, in this paper, only the edge performance during the training phase is measured and presented.

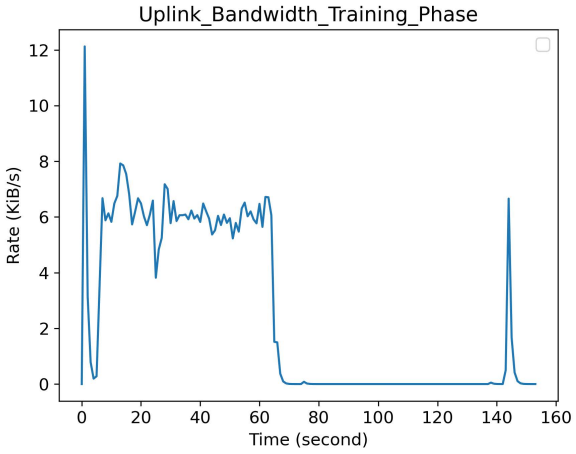


FIGURE 9. Bandwidth occupied during the FedVAE-SVDD training in the edge-cloud link.

a: BANDWIDTH OCCUPATION

In a distributed edge computing environment, a detection algorithm can be totaly fulfilled right at a local edge device itself. In that case, the edge does not need to send any information to the cloud from the perspective of the Anomaly Detection task, thereby not consuming bandwidth of the edge-cloud link. But in the FL-based edge-computing environment, each edge needs to send its local model to the central cloud for global model updating until the algorithm at each edge converges. This model transmission obviously causes some communication cost over both of the edge-cloud uplink and downlink. Therefore, we measure this communication cost during the FedVAE-SVDD training phase in the uplink. As illustrated in Fig.9, the FedVAE-SVDD architecture occupies a small amount of bandwidth over the period of 150 seconds. It is notable that the bandwidth consumption is approximately zero throughout the period from 70 to 140 seconds and just around 7 KiB per second during the remaining intervals (1KiB is equal to 1024 Bytes). This result shows the advantage of the FedVAE-SVDD architecture in terms of bandwidth consumption. It results in low communication cost which leaves more free bandwidth resources for other data transmission tasks in Industrial IoT networks.

b: MODEL RUNNING TIME

Using deep learning to detect anomalies inside ICS is common worldwide, but we have usually experimented the training time on the scale of hours for the whole data set. And those figures mean that the system should be only retrained periodically on the scale of an hour, day, or week, since it can not capture any sudden change of traffic patterns in real-time. However, the testbed result shown in Fig. 10 indicates that Raspberry-Pi-4 takes relatively little time of 150 seconds to run the FedVAE-SVDD model (with just about 70 and 80 seconds in the FedVAE phase and SVDD phase respectively) in each communication round. In our case study, it just needs to run 3 communication rounds for the training to converge. Therefore, the FedVAE-SVDD model takes only roughly 450 seconds (i.e 7.5 minutes) overall to produce such high detection performance. Basically, it overcomes the running time problem in a trade-off for high performance of a previous work [30].

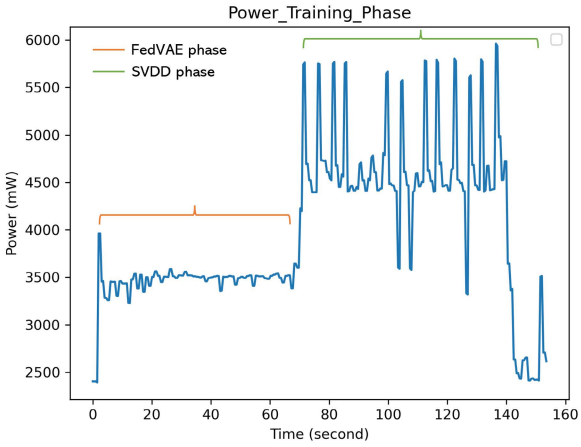


FIGURE 10. Power consumption of an edge device in one communication round.

c: POWER CONSUMPTION

Moreover, we want to dive into investigating the energy at the edge to know whether there should be a trade-off here or not. Fig.10 illustrates the consumed power level during the FedVAE-SVDD training process in one communication round at an edge device (i.e the Raspberry-Pi-4), comprising two successive phases: FedVAE and SVDD. The measurement shows that the power consumption ranges from under 3500mW to 6000mW in the whole training process. Power consumption at the SVDD phase fluctuates strongly and is much higher than the FedVAE phase. These real-world metrics give us a better idea of how deploying distributed machine learning models on edge devices will consume more energy for that computation.

d: CPU USAGE

Fig.11 shows the proportion of CPU usage during the FedVAE-SVDD process in one communication round at the

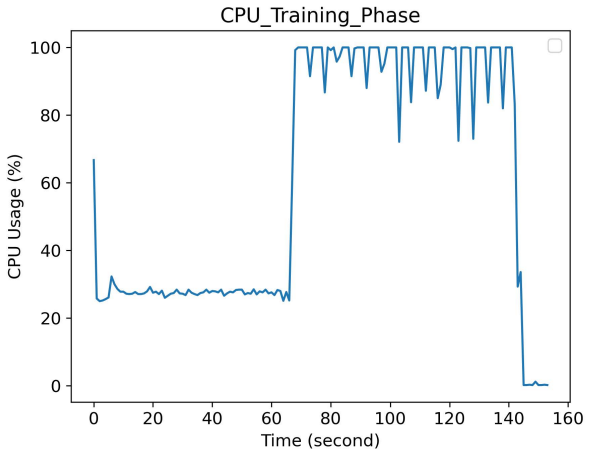


FIGURE 11. CPU usage of an edge device in one communication round.

edge device. It is conspicuous that the running time in the whole process is very fast, but in the worst case, the SVDD phase accounts for 100% of the CPU usage while this ratio of the FedVAE phase is just over 20%. Based on these findings, we would like to make a few recommendations. Firstly, in reality, with a runtime of only 70 seconds, the threshold update process (i.e., SVDD phase) can be retrained during system maintenance time or the night on schedule, rather than implemented on a real-time scale (i.e minutes or seconds scale). Thanks to this, other services would not be interrupted on the edge device every update time. From another perspective, these findings seem to be an acceptable trade-off between the running time for high detection performance and the hardware resource. Furthermore, it is possible to consider upgrading to edge hardware devices with higher processing capacity than Raspberry-Pi-4. With more powerful edge hardware, the FedVAE-SVDD model will be the effective detection model for such a factory.

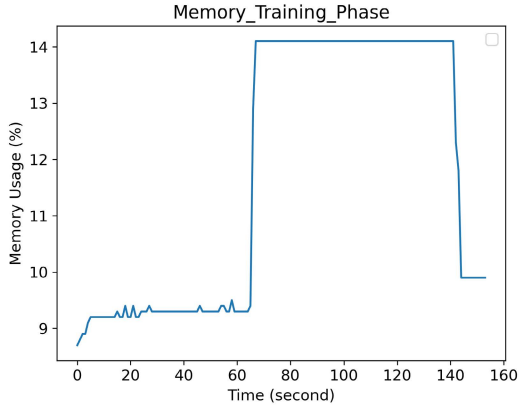


FIGURE 12. Memory usage of an edge device in one communication round.

e: MEMORY USAGE

In the same experimental setup with the power measurement, the percentage of memory usage in the VAE-SVDD phase at an edge device is demonstrated in Fig.12. It can seen that throughout the period of 150 seconds of the VAE-SVDD training process with one communication round, the memory usage of the VAE and SVDD phase is quite steady, with just over 9% and 14%, respectively. With these ratios, it can be inferred that in the training process, the memory resource is still available for other tasks.

Technically, the aforementioned measured values are likely to vary according to hardware configuration, but their normalized representations are considered as a benchmark in different hardware setups. Therefore, we apply min-max scaling based normalization of power consumption, CPU usage, and memory usage to illustrate the results in a more general form. The general formula for a min-max of [0, 1] is given as:

$$val_{norm} = \frac{val - min(val)}{max(val) - min(val)} \tag{22}$$

where val is an original value, val_{norm} is the normalized value. The normalized values of power consumption, CPU usage, and memory usage of an edge in one communication round are shown in Fig.13.

V. DISCUSSIONS

A. CONTRIBUTIONS

Based on the mentioned motivations, FedeX is a framework intended to achieve high detection performance, learn new data patterns fast, have lightweight, and improve the interpretability of the model. To the best of our knowledge, existing works have not yet achieved all of these features in the literature.

In order to verify the performance of FedeX, we utilized an SCADA liquid storage infrastructure dataset as the main case study and another well-known dataset-SWaT. The obtained results show that FedeX achieves remarkable detection performance that outperforms many existing anomaly detection methods, whilst still attaining a fast training time of 7.5 minutes. This facilitates frequent retraining in ICSs.

With edge computing, our proposal enables a faster system response against attacks since the detection module is near attack sources. Through our comprehensive experiments on edge computing capacity, FedeX is proven to be lightweight in terms of bandwidth, power consumption, and memory occupation. Our testbed contributes practical implications because it demonstrates the feasibility of the model in ICS edge-computing environments. A point worth mentioning is that edge-computing-based FL proposals in the literature are hardly assessed on realistic testbeds, such as [17] and [18].

Last but not least, by integrating XAI, FedeX provides a comprehensive explanation of detected anomalies. This enables experts to respond to anomalies quickly, based on the relationship between the features and respective physical components at each manufacturing zone. Through that, FedeX-predicted anomalies also become more reliable. These findings could be useful for other authors in designing their black-box model for reliable, effective anomaly detection purposes in distributed ICSs.

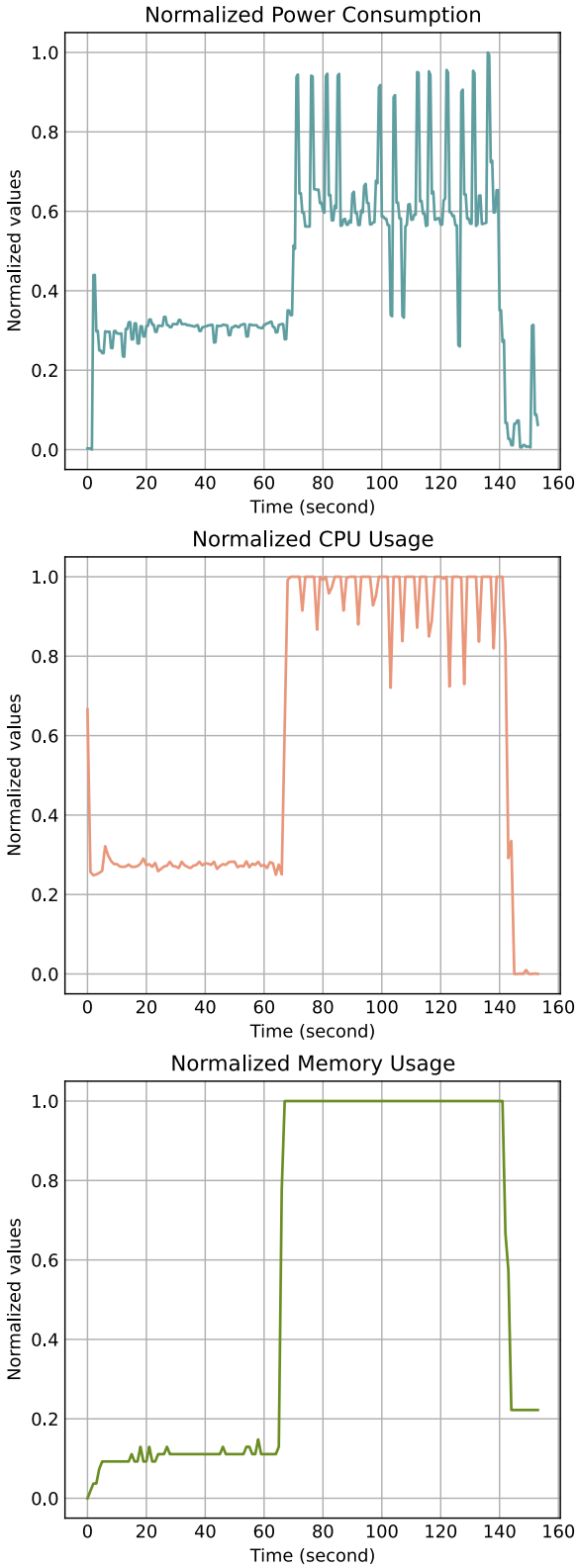


FIGURE 13. Normalization of hardware values of an edge in one communication round.

B. LIMITATIONS AND FUTURE WORK

It is proven that Federated learning is a more effective learning solution for distributed IIoT. However, in the FL environment, the ML models must be sent from the edges of distantly distributed zones to the Cloud, so privacy and security on those very communication channels can be a critical issue. Attackers can steal information transmitted on the edge-cloud communication channel instead of information sources before the edge (our research scenario). In the future, we will investigate how to secure or encrypt the information sent on the edge-cloud link in the Federated Learning environment to prevent tap-in. The information. The information should be designed in a more secure, lightweight way.

Another future work that needs to be examined is the similarity of manufacturing zones, for example, the number of machines in each zone. It poses an imbalanced distributed-learning issue for local learning models at the edge. As local data of different zones could not be similar, it raises the bias of the global model aggregated by the cloud server. In the future, we will find a way to improve our FedeX architecture when imbalanced data distribution occurs.

VI. CONCLUSION

In this paper, we have elaborated our proposed hybrid model based on VAE and SVDD with the Federated-Learning technique. The hybrid model is enabled to perform efficiently on weak edge devices installed in the IoT-based system of a Smart Factory. With the FL architecture design, the detection task is distributed to smaller local zones located in the last premise of traffic senders. Therefore, anomalies or attacks can be quickly identified and quarantined in each separate zones. This FL architecture also helps to deal with Big Data created from a variety of devices inside a huge smart Factory 4.0 of the future. In addition to achieving prominent performance, fast runtime (7.5 minutes), and lightweight, the proposed architecture solves the black-box issue and improves its reliability by integrating XAI. This brings the benefit of allowing experts to analyze and respond to anomalies quickly in distributed ICS environments.

REFERENCES

[1] Y. Lu, X. Xu, and L. Wang, "Smart manufacturing process and system automation—A critical review of the standards and envisioned scenarios," *J. Manuf. Syst.*, vol. 56, pp. 312–325, Jul. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861252030100X

[2] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of Things security: Requirements, challenges, and solutions," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100129. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660519302288

[3] N. Tuptuk and S. Hailes, "Security of smart manufacturing systems," *J. Manuf. Syst.*, vol. 47, pp. 93–106, Apr. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612518300463

[4] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2020.

[5] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[6] G. Li, Y. Shen, P. Zhao, X. Lu, J. Liu, Y. Liu, and S. C. H. Hoid, "Detecting cyberattacks in industrial control systems using online learning algorithms," *Neurocomputing*, vol. 364, pp. 338–348, Oct. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219309762

[7] C.-P. Chang, W.-C. Hsu, and I.-E. Liao, "Anomaly detection for industrial control systems using K-means and convolutional autoencoder," in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2019, pp. 1–6.
[8] G. E. I. Selim, E. E.-D. Hemdan, A. M. Shehata, and N. A. El-Fishawy, "Anomaly events classification and detection system in critical industrial Internet of Things infrastructure using machine learning algorithms," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 12619–12640, Mar. 2021.
[9] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 1058–1065, doi: 10.1109/ICDMW.2017.149.
[10] D. Shalyga, P. Filonov, and A. Lavrentyev, "Anomaly detection for water treatment system based on neural network with automatic architecture optimization," 2018, *arXiv:1807.07282*.
[11] G. Zizzo, C. Hankin, S. Maffeis, and K. Jones, "Adversarial attacks on time-series intrusion detection for industrial control systems," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Jan. 2021, pp. 899–910, doi: 10.1109/TrustCom50675.2020.00121.
[12] Á. L. P. Gómez, L. F. Maimó, A. H. Celdrán, and F. J. G. Clemente, "MADICS: A methodology for anomaly detection in industrial control systems," *Symmetry*, vol. 12, no. 10, p. 1583, Sep. 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/10/1583
[13] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy*, Jan. 2018, pp. 72–83.
[14] T. K. Das, S. Adepu, and J. Zhou, "Anomaly detection in industrial control systems using logical analysis of data," *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101935. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820302121
[15] H. R. Ghaeini, D. Antonioli, F. Brasser, A.-R. Sadeghi, and N. O. Tippenhauer, "State-aware anomaly detection for industrial control systems," in *Proc. 33rd Annu. ACM Symp. Appl. Comput.* New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1620–1628, doi: 10.1145/3167132.3167305.
[16] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "GEE: A gradient-based explainable variational autoencoder for network anomaly detection," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2019, pp. 91–99.
[17] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.
[18] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for IoT security attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, 2021.
[19] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965–83973, 2020.
[20] K. Amarasinghe, K. Kenney, and M. Manic, "Toward explainable deep neural network based anomaly detection," in *Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI)*, Jul. 2018, pp. 311–317.
[21] C. Hwang and T. Lee, "E-SFD: Explainable sensor fault detection in the ICS anomaly detection system," *IEEE Access*, vol. 9, pp. 140470–140486, 2021.
[22] J. Goh, S. Adepu, K. N. Junejo, and A. P. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. CRITIS*, 2016, pp. 88–99.
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," *CoRR*, vol. abs/1602.04938, pp. 1–10, Aug. 2016.
[25] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," *CoRR*, vol. abs/1704.02685, pp. 1–9, Apr. 2017.
[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.
[27] P. M. Laso, D. Brosset, and J. Puentes, "Dataset of anomalies and malicious acts in a cyber-physical subsystem," *Data Brief*, vol. 14, pp. 186–191, Oct. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340917303402
[28] R. A. Light, "Mosquitto: Server and client implementation of the MQTT protocol," *J. Open Source Softw.*, vol. 2, no. 13, p. 265, May 2017, doi: 10.21105/joss.00265.
[29] H. Hindy, D. Brosset, E. Bayne, A. Seeam, and X. Bellekens, "Improving SIEM for critical SCADA water infrastructures using machine learning," in *Proc. Int. Workshop Secur. Privacy Requirements Eng.*, in Lecture Notes in Computer Science, 2019, pp. 3–19, doi: 10.1007/978-3-030-12786-2_1.
[30] T. T. Huong, T. P. Bac, D. M. Long, T. D. Luong, N. M. Dan, L. A. Quang, L. T. Cong, B. D. Thang, and K. P. Tran, "Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach," *Comput. Ind.*, vol. 132, Nov. 2021, Art. no. 103509.

TRUONG THU HUONG (Member, IEEE) received the B.Sc. degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2001, the M.Sc. degree in information and communication systems from the Hamburg University of Technology, Germany, in 2004, and the Ph.D. degree in telecommunications from the University of Trento, Italy, in 2007. Her research interests include network security, artificial intelligence, traffic engineering in next generation networks, QoE/QoS guarantee for network services, green networking, and development of the Internet of Things ecosystems and applications.

TA PHUONG BAC received the B.Sc. degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Vietnam, in 2020. He is currently pursuing the master's degree with Soongsil University, South Korea. He was a Research Assistant at the Future Internet Laboratory, School of Electronics and Telecommunications, HUST for three years. Currently, he is a research Assistant with the Distributed Cloud and Network Laboratory, Soongsil University. His research interests include network security, artificial intelligence, and the Internet of Things ecosystems and applications.

KIEU NGAN HA is currently pursuing the degree in electronics and telecommunications engineering with the School of Electronics and Telecommunications, Hanoi University of Science and Technology. She is a baccalaureate from the gifted class in mathematics. She was a Research Assistant at the Future Internet Laboratory for one and half years. Her research interests include the IoT, network security, machine learning, AI, and its application.

NGUYEN VIET HOANG is currently pursuing the degree in electronics and telecommunications engineering with the School of Electronics and Telecommunications, Hanoi University of Science and Technology. He has been a Research Assistant with the Future Internet Laboratory, since 2020. His research interests include machine learning and network security.

NGUYEN XUAN HOANG is currently pursuing the degree of the talented program in smart electronics systems and the IoT with the School of Electronics and Telecommunications, Hanoi University of Science and Technology. He is a baccalaureate who won the Third Prize of the Vietnam National Physics Olympiad. He has been a Research Assistant with the Future Internet Laboratory, since 2021. His research interests include data science, anomaly detection, and network security in the Internet of Things ecosystems.

NGUYEN TAI HUNG received the master's and Ph.D. degrees in communication engineering from the Hanoi University of Science and Technology, in 2001 and 2007, respectively. In 2010, he spent a half of the year with Fraunhofer FOKUS, Berlin, Germany for conducting a research project on service development for 3G/NGN networks. He is an Associate Professor with the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam. His research interests include traffic engineering and tomography, service platforms for future networks, QoS/QoE, resource management and the future internet.

KIM PHUC TRAN received the engineering and M.E. degrees in automated manufacturing, the Ph.D. degree in automation and applied informatics from the University of Nantes, and the HDR (Dr. Habil.) degree in computer science and automation from the University of Lille, France. He is currently an Associate Professor of artificial intelligence and data science with the ENSAIT and the GEMTEX Laboratory, University of Lille. He has published more than 60 papers in peer-reviewed international journals and proceedings of international conferences. His research interests include real-time anomaly detection with machine learning with applications, decision support systems with artificial intelligence, enabling smart manufacturing with IIoT, federated learning, and edge computing. He has edited three books with Springer Nature and Taylor & Francis. He is the Topic Editor and a Guest Editor for Sensors Journal. He has supervised eight Ph.D. students and two postdoctoral researchers. In addition, as the Project Coordinator (PI), he is conducting one regional research project about healthcare systems with federated learning. He has been or is involved (Co-PI or member) in five regional research and European projects. He is an Expert and an Evaluator of the Research and Innovation Program with the Government of the French Community, Belgium. He received the Award for Scientific Excellence (Prime d'Encadrement Doctoral et de Recherche) from the Ministry of Higher Education, Research and Innovation, France for four years, from 2021 to 2025, in recognition of his outstanding scientific achievements. Since 2017, he has been the Senior Scientific Advisor with Dong A University and the International Research Institute for Artificial Intelligence and Data Science (IAD), Danang, Vietnam.