Federated Intrusion Detection in Medical IoT: Client-Side Feature Learning with Variational Autoencoders vs Autoencoders

Ahlem Harhad, Cyril Drocourt, David Durand, Guillaume Muller, Kamal Singh, Gil Utard

Federated Intrusion Detection in Medical IoT: Client-Side Feature Learning with Variational Autoencoders vs Autoencoders

Ahlem HARHAD
MIS Laboratory
University of Picardie Jules Verne
Amiens, France
ahlem.harhad@u-picardie.fr

Cyril DROCOURT
MIS Laboratory
University of Picardie Jules Verne
Amiens, France
cyril.drocourt@u-picardie.fr

David DURAND
MIS Laboratory
University of Picardie Jules Verne
Amiens, France
david.durand@u-picardie.fr

Guillaume MULLER
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
guillaume.muller@emse.fr

Kamal SINGH
Lab. Hubert Curien
Université Jean Monnet Saint-Etienne, CNRS
F-42023 Saint-Etienne, France
kamal.singh25@gmail.com

Gil UTARD
MIS Laboratory
University of Picardie Jules Verne
Amiens, France
gil.utard@u-picardie.fr

***Abstract*—The rapid adoption of Internet of Things (IoT) technologies in the medical field has introduced new challenges in securing sensitive data against increasingly sophisticated cyber threats. To address the limitations of intrusion detection systems that rely on centralized machine learning models, federated learning has emerged as a more privacy-conscious alternative, allowing distributed devices to collaboratively train models without sharing raw data. Building upon existing federated semi-supervised intrusion detection frameworks, this work proposes an enhanced client-side model that replaces the conventional autoencoder with a Beta-Variational Autoencoder (β-VAE). This substitution yields a more discriminative latent space, as illustrated by higher silhouette scores and 3D visualizations on the BoT-IoT dataset in our experiments. These improvements in the latent representation correlate with enhanced detection performance across all three IoT-specific datasets—BoT-IoT, SCADA, and wustl-ehms-2020—demonstrating significant improvements.**

***Index Terms*—Intrusion Detection System, Federated Learning, Semi-Supervised Learning, β-VAE, Medical IoT**

I. Introduction

By 2025, the number of connected IoT devices has grown from about 15 billion in 2015 to over 75 billion[1]. This growth continues to be fueled by the arrival of 5G and beyond. As the number of devices rises, intrusion detection has become a critical concern.

Primary IDS approaches relied on signature-based techniques, which identified known attack patterns by comparing network traffic with a database of attack signatures. While effective at detecting established threats with minimal false positives, this method struggled to identify novel attacks. With the rise of Machine Learning (ML), more adaptive approaches have emerged. Anomaly detection, typically implemented using unsupervised ML, involves establishing a baseline of normal behavior and flagging deviations as potential intrusions. These methods are well-suited for detecting novel threats but can suffer from higher false positive rates, mistakenly identifying legitimate activity as malicious [1]. In contrast, supervised ML techniques, often used for classification, rely on labeled datasets to learn to distinguish between normal and malicious traffic, achieving high accuracy on known attacks. However, most existing ML-based IDS rely on centralized architectures, where data from all nodes is collected and processed on a central server. As highlighted in [2], this approach raises serious concerns in privacy, latency, and communication overhead, especially in resource-constrained environments such as the Internet of Medical Things (IoMT).

To address these limitations, Federated Learning (FL) has emerged as a promising approach. Initially introduced by Google in 2016 [3], FL allows training machine learning models directly on devices where data is produced, thus eliminating the need to share raw data.While this paradigm addresses several privacy and communication challenges, it still suffers from critical security issues, such as model poisoning attacks, particularly in non-i.i.d. settings, where malicious contributions may be difficult to distinguish from legitimate ones. [4]. However, this work does not address these security challenges and focuses instead on other aspects.

Although FL has been successfully applied to intrusion detection, existing methods generally fall into two categories: fully supervised or unsupervised, as highlighted in the review by [5]. Since labeled data is often limited due to the manual effort required by domain experts, there has been a growing interest in semi-supervised approaches, which aim to leverage the small amount of labeled data available while exploiting

Identify applicable funding agency here. If none, delete this.

[1]https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

the abundance of unlabeled data.

FLuIDS (Federated Learning for semi-sUpervised Intrusion Detection Systems) [6] was proposed as a method particularly suited for IoT contexts. One key advantage of FLuIDS is its ability to operate without any data labeling. In [6], the method is described and tested, demonstrating performance comparable to that of centralized settings, while operating in a federated manner. Leveraging the potential of this approach, we first began by re-implementing the original version as baseline for comparison purposes. Then, with the objective of improving the quality of the latent representation and enabling synthetic data generation, we introduce an enhancement by replacing the traditional autoencoder (AE) on the client side with a β-variational autoencoder (β-VAE).

Our key contributions are:

- Enhanced FLuIDS by replacing the traditional autoencoder with a β-VAE, resulting in improvements of 7%, 10%, and 15% in Macro F1 score on the SCADA, Wustl-ehms-2020, and BoT-IoT datasets, respectively.
- Enhancing model explainability through the analysis and visualization of latent space metrics, which provide deeper insights into the representation learned by the two architectures.
- Demonstrating the effectiveness and generalizability of the method by evaluating on three IoT-suited datasets, including a dataset specifically designed for medical IoT environments.
- Introducing a model capable of synthetic data generation through the use of the β-VAE, opening possibilities for future data augmentation.

The remainder of this paper is organized as follows:

Section 2 reviews related work in intrusion detection, focusing on federated semi-supervised methods. Section 3 provides background information to help the reader understand the FLuIDS approach, including the general IoT architecture considered, the fundamentals of federated learning, and the role of autoencoders used in FLuIDS. Section 4 overviews the FLuIDS method, outlining its design, functionality, and advantages for IoT environments. Section 5 details the proposed enhancements to FLuIDS by integrating the β-variational autoencoder and its expected impact. Section 6 presents the experimental setup, including the datasets used and evaluation metrics. Finally, Section 7 discusses the results of the experiments, comparing the original FLuIDS method with the proposed enhancement, as well as with existing literature.

II. Literature Review

The work in [5] highlight the necessity of transitioning from centralized to federated learning for intrusion detection in IoT networks. This shift is crucial to address challenges like data privacy, latency, and network overhead. Most existing intrusion detection methods are either fully supervised or unsupervised. Fully supervised approaches depend on labeled data, which is often difficult or expensive to obtain in IoT networks. Unsupervised methods, while attempting to detect anomalies without labeled data, often face difficulties in achieving high accuracy in complex environments.

Few studies have focused on semi-supervised methods for intrusion detection. In [7], a semi-supervised federated learning method is proposed that leverages entropy-based Active Learning to identify and label only the most uncertain data points, thereby reducing the labeling effort while preserving high performance by focusing on the most informative samples. Building on this approach, METALS [8] incorporates similar Active Learning strategies, such as entropy, margin, or least confidence sampling, to select the most informative unlabeled samples in each round. These samples are then labeled and used to train local models, which are aggregated on the server through FedAvg or FedProx. Thus, METALS extends the principles of entropy-based Active Learning from [7] while incorporating additional strategies to further optimize the process for intrusion detection in IoT environments.

FEDUPS [9] turns network traffic data into gray-scale images so they can be processed using convolutional neural networks (CNNs). Its main innovation is Uncertainty-aware Pseudo-Labeling (UPS), which assigns both positive and negative pseudo-labels based on model confidence. It uses temperature scaling to adjust softmax outputs and Monte-Carlo Dropout [10] to estimate uncertainty during prediction. Instead of relying only on entropy-based filtering, FEDUPS takes advantage of both highly confident correct and incorrect samples. On the server side, client models are aggregated using confidence-weighted averaging giving more importance to updates from clients with higher average confidence in their pseudo-labels.The method in [11] uses knowledge distillation: each client trains a CNN on private labeled data and uses a local discriminator to filter low-confidence predictions on public unlabeled data. High-confidence predictions are sent to the server, which aggregates them via majority voting to create global pseudo-labels for central training and client distillation. FedMSE [12] combines a Shrink Autoencoder (SAE) and a Centroid-based one-class classifier. The SAE introduces a shrinkage loss to the classical autoencoder loss to encourage compact latent representations of normal behavior and the Centroid-based one-class classifier detects anomalies based on distance from learned centroids. A new aggregation strategy, MSEAvg, prioritizes more accurate local models using mean squared error.

FLuIDS [6] is a semi-supervised method that assumes there is no labeled data on the client side and only a small portion of labeled data on the server side. It uses a classic autoencoder trained in an unsupervised manner on the clients, which is then aggregated on the server. Only the encoder part is kept, and a fully connected layer is added to build a supervised model, which is trained on the server using the small labeled data portion. Finally, a more recent method introduced in [13] follows a conceptually similar strategy, further supporting the relevance of this method for IoT. These approaches avoid the complexity of mechanisms like active learning and knowledge distillation, and unlike label-free anomaly detection techniques, they support multi-class classification, enabling

more accurate identification of different attack types.

In current paper, we enhance FLuIDS, by replacing the autoencoder with a (β-VAE) on the client side, with the goal of improving its performance. In this context, the study [14] demonstrates that the latent space learned by a VAE is more informative compared to the latent spaces produced by PCA, AEs, or Deep Belief Networks (DBN). This supports our choice of VAE as a more promising approach for improving the feature extraction process in FLuIDS. Furthermore, a notable advantage of VAEs is their ability to generate high-quality synthetic data. This capability is particularly valuable in our case of intrusion detection using federated learning with limited labeled data, for several reasons: (1) synthetic data can be shared without compromising user privacy, (2) it serves as a viable alternative when labeled data are costly to obtain, and (3) the generated samples tend to be more realistic than conventionally simulated network data.

III. Background

A. IoT Architecture

In this work, a three-layer IoT architecture is considered: the Connected Objects Layer, the Proximity Network Layer (IoT gateways and routers), and the Internet Layer (cloud storage and processing). The intrusion detection system is deployed at the IoT gateway, serving as a critical point for monitoring network traffic and detecting potential intrusions early. This architecture is illustrated in Fig. 1.

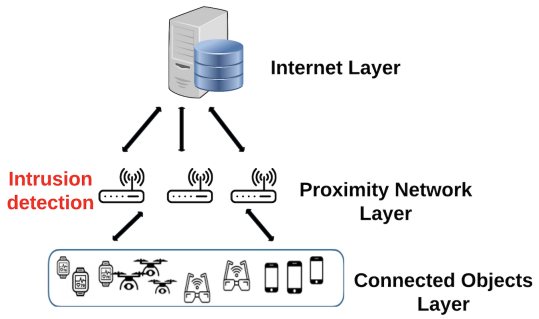


Fig. 1. FLuIDS IoT Architecture

B. Federated Learning

Federated Learning is a decentralized machine learning approach where multiple devices collaboratively train a model without sharing raw data. The steps involved are illustrated in Fig. 2 and described below.

- **Step 1: Model Initialization/Distribution** – The central server sends the latest model parameters to the nodes, or initial parameters if it is the first round.
- **Step 2: Data Selection** – Each node selects local data for training.
- **Step 3: Local Training** – Each local model is trained using the received parameters and local data.
- **Step 4: Parameter Upload** – Clients send the updated local model parameters back to the server.
- **Step 5: Aggregation** – The server aggregates the received parameters to update the global model.
- **Step 6: Iteration** – The process restarts from Step 1 for the next training round. The aggregated model from Step 5 can also be deployed to production.

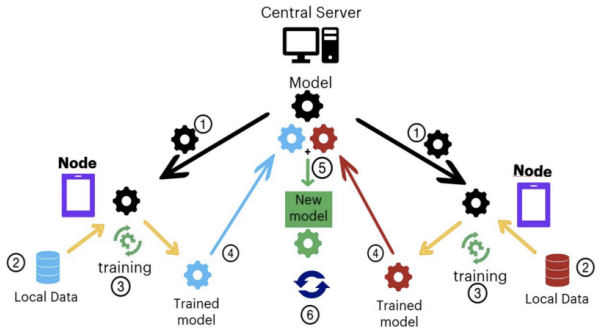


Fig. 2. . Federated Learning Architecture. [15]

C. Autoencoder

An *autoencoder* is a neural network that learns a compressed representation of input data. It has two parts:

- **Encoder:** maps input $x \in \mathbb{R}^d$ to latent code $c \in \mathbb{R}^p$:

$$c = e(x)$$

- **Decoder:** reconstructs $\hat{x} \in \mathbb{R}^d$ from c:

$$\hat{x} = d(c)$$

Training minimizes the reconstruction loss, typically the Mean Squared Error (MSE):

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|_2^2 = \|x - d(e(x))\|_2^2$$

This loss encourages the model to capture essential input features in c for accurate reconstruction.

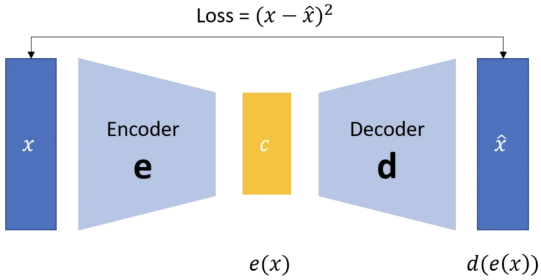


Fig. 3. Autoencoder Architecture.

[2]

[2]https://medium.com/geekculture/variational-autoencoder-vae-9b8ce5475f68

IV. FLuIDS Method Overview

The proposed method in [6] aims to train a semi-supervised model using only a small amount of labeled data at the server while preserving data privacy. It follows a federated architecture composed of two main parts: the client side and the server side.

On the client side, each client receives an initial autoencoder model and trains it locally in an unsupervised manner using only its own unlabeled data. After minimizing the reconstruction error over several epochs, clients send the updated model weights back to the server without sharing raw data.

On the server side, the models received from the clients are aggregated using FedAvg to produce a new global autoencoder. The server then modifies the architecture by removing the decoder and adding a fully connected layer for supervised training using a small labeled dataset. Currently, this labeled dataset must be manually annotated by experts, but by adopting a VAE approach, it will be possible to complement these labels with synthetic data generated by the model.

The updated global autoencoder is sent back to the clients for further unsupervised training, as well as the trained supervised model to be used for inference on the clients' local network data.

Fig. 4 illustrates the architecture of the supervised model trained on the server side.

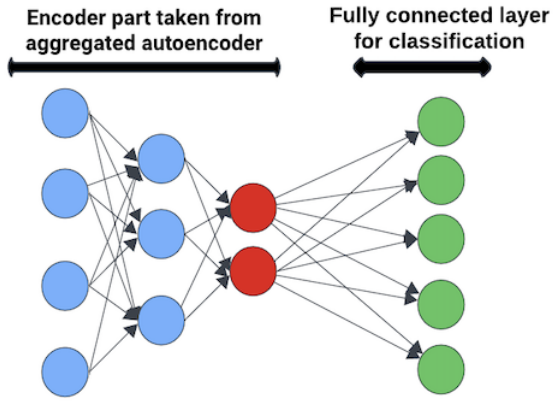


Fig. 4. FLuIDS's server side model

V. Substituting the Autoencoder with a β-Variational Autoencoder

Unlike the deterministic encoding of a traditional Autoencoder, the Variational Autoencoder [16] adopts a probabilistic framework. The encoder produces a distribution $q(z|x)$ rather than a single point, which approximates the posterior distribution of the latent variable z given the observation x. This distribution is usually modeled as a Gaussian with a mean and variance dependent on the input. The decoder models the conditional distribution $p(x|z)$, which represents the probability of reconstructing x from z.

The total VAE loss is the sum of the reconstruction loss (e.g., mean squared error between x and $\hat{x}$) and the KL divergence between the approximate posterior $q(z|x)$ and the prior $p(z)$, usually a standard normal $\mathcal{N}(0, I)$, ensuring simple and well-structured latent representations. :

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{KL}} \tag{1}$$

Fig. 5 illustrates the architecture of a variational autoencoder.

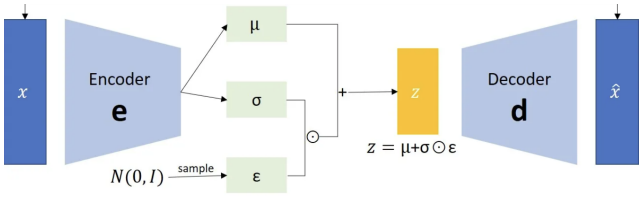


Fig. 5. Variational Autoencoder.[3]

The β**-VAE** introduces a weighting factor β on the KL term in (1).

$$\mathcal{L}_{\beta\text{-VAE}} = \mathcal{L}_{\text{reconstruction}} + \beta \cdot \mathcal{L}_{\text{KL}} \tag{2}$$

By adjusting β, one can control the extent to which the model emphasizes the regularization of the latent space versus the fidelity of data reconstruction.

The work [17] focuses on generating sentences from a continuous space using variational autoencoders, address the problem of KL divergence vanishing, where, during training, the KL term can become close to zero. This indicates that the latent space distribution is already very close to the standard normal distribution. Although this might seem like a good result, it can lead to a less informative latent space representation, making the generated data less useful. Essentially, the model may focus too much on minimizing the KL divergence at the expense of learning a rich latent space.

To mitigate this, the authors propose a technique called **annealing**. This involves starting training with a low β value, allowing the model to focus primarily on data reconstruction. Over time, β is gradually increased, guiding the model to shift its focus from reconstruction to regularizing the latent space distribution.

In our implementation, we adopted a linear annealing strategy, progressively increasing β from 0 to 1 over the course of training epochs on the clients. This adjustment was necessary because, with a fixed β, training stalled as early as the second communication round.

The introduction of a β-VAE should lead to more discriminative latent representations, allowing the model to better separate points from different classes. This would improve the model's ability to distinguish between different types of data, thereby enhancing classification performance. Moreover, using an annealing schedule for β—where it increases progressively during training—allows the model to first focus on learning detailed reconstructions before gradually shaping the latent space.

[3]https://medium.com/geekculture/variational-autoencoder-vae-9b8ce5475f68

VI. EXPERIMENTAL SETUP AND EVALUATION METRICS

A. Hardware Setup and Code Repository

Experiments are conducted on a high-performance computing cluster using a partition that provides up to 28 CPU cores and 125 GB of RAM. The simulation of the federated learning scenario was implemented with Python and TensorFlow, leveraging the Flower framework[4]. The specific software versions used are detailed in the code repository available at this link[5].

B. Dataset

For this study, three datasets tailored to IoT are used: Bot-IoT, SCADA and wustl-ehms-2020.

1) The Bot-IoT dataset: [18] is tailored for assessing intrusion detection systems within IoT environments. It emulates five distinct IoT scenarios (e.g., weather station, thermostat) and includes both benign and malicious network traffic, such as Distributed Denial of Service(DDOS), Denial of Service(DoS), reconnaissance, and data theft attacks. Additionally, the dataset offers statistical metrics like mean and standard deviation.

2) wustl-ehms-2020: The wustl-ehms-2020 dataset [19] was generated using a testbed representative of an Internet of Medical Things (IoMT) environment. It includes a multi-sensor board collecting biometric data such as Electrocardiogram, oxygen saturation, body temperature, and blood pressure. This data is transmitted to a server for medical analysis, while another computer simulates cyberattacks. The dataset combines network and biometric features, with attacks categorized as spoofing and data alteration.

3) SCADA: We conduct experiments on the Gas Pipeline SCADA dataset [20] primarily to compare our results with the original method. This dataset is widely used as a benchmark for cybersecurity research in industrial control systems. It contains both legitimate (benign) and malicious network activities. Benign data corresponds to normal, authorized network traffic, while the malicious activities are categorized as follows:

- NMRI: Naive Malicious Response Injection
- CMRI: Complex Malicious Response Injection
- MSCI: Malicious State Command Injection
- MPCI: Malicious Parameter Command Injection
- MFCI: Malicious Function Command Injection
- DoS: Denial of Service
- Rec: Reconnaissance

As shown in Table I, the datasets exhibit a strong class imbalance. For instance, the Bot-IoT dataset exemplifies this with majority classes like DDoS and DoS having over a million samples each, while minority classes such as Theft have fewer than one hundred samples. Similar imbalances are observed across the other datasets as well. This is typical in cyber-security, and evolves over time, with some types of attacks becoming more or less used by attackers.

[4]https://flower.dev

[5]https://anonymous.4open.science/r/VAE-Classif_IDS-01EA

TABLE I
SUMMARY OF TRAFFIC TYPES AND SAMPLE SIZES IN DIFFERENT DATASETS

Dataset	Label	Count	Train	Test
Bot-IoT	Benign	477	370	107
	DDoS	+1.9M	154131	38530
	DoS	+1.6M	132014	33011
	Recon	91082	72919	18163
	Theft	79	65	14
Wustl-ehms-2020	Benign	14269	11414	2855
	Spoofing	1124	899	225
	Data Alteration	922	738	184
SCADA	Benign	61156	48815	12341
	NMRI	2763	2200	563
	CMRI	15466	12383	3083
	MSCI	782	639	143
	MPCI	7637	6175	1462
	MFCI	573	467	106
	DoS	1837	1489	348
	Rec	6805	5447	1358

C. Performance evaluation metrics

To evaluate the classification performance, we use accuracy, precision, recall, and F1-score, considering both macro and weighted averages. In our case, where all attack types are equally important regardless of their frequency, we focus more on the macro F1-score, as it offers a fair evaluation across all classes. A detailed definition of these metrics and their applications is provided in [21]. Additionally, we use the Wilcoxon signed-rank test [22] to determine whether the performance improvement of the variational autoencoder over the standard autoencoder is statistically significant.

D. Latent Space Evaluation Metrics

1) Silhouette Score: To quantitatively assess the quality of the latent space, we use the silhouette score, a metric that evaluates how well each data point fits within its assigned cluster compared to other clusters. A detailed definition of this metric is given in [23].

2) t-SNE Visualization: In addition to the silhouette score, we employ t-distributed Stochastic Neighbor Embedding (t-SNE) [24] to visually explore the latent space. t-SNE is a non-linear dimensionality reduction technique that projects high-dimensional data into two or three dimensions while preserving local neighborhood relationships. Its purpose is to provide an intuitive visual indication of class separability, potential overlaps, and cluster structures.

E. Model Architecture

The numbers in each architecture of Table II indicate the number of neurons in each layer of the encoder and decoder. For VAE models, the latent representation is defined by two parallel layers corresponding to the mean and variance vectors Marked with || in the table. Additionally, the final classification layer is specified for each dataset.

F. Federated Learning Parameters:

The federated learning setup is described in Table III, Note that training epochs and Label ratio differ for BoT-IoT compared to SCADA and WUSTL. This distinction is

TABLE II
MODEL ARCHITECTURES

Dataset	AE Architecture	VAE Architecture
BoT-IoT	19-14-14-10-14-14-19	19-14-14-10-(10 ‖ 10)-14-14-19
	Classification Layer: 5	
WUSTL	36-30-30-20-30-30-36	36-30-30-20-(20 ‖ 20)-30-30-36
	Classification Layer: 3	
SCADA	26-20-20-15-20-20-26	26-20-20-10-(15 ‖ 15)-20-20-26
	Classification Layer: 8	

motivated by the inherent differences in dataset size and complexity. For SCADA, parameters were aligned with prior works to enable fair comparison.

TABLE III
FEDERATED LEARNING PARAMETERS

Parameter	Value
Number of rounds (BoT-IoT)	30
Number of rounds (SCADA)	18
Number of rounds (WUSTL)	20
Number of clients	10
Fraction of clients sampled for training	100%
Fraction of clients sampled for evaluation	100%
Minimum clients for training	10
Minimum clients for evaluation	10
Minimum available clients	10
Epochs per client (BoT-IoT)	5
Epochs per server (BoT-IoT)	20
Epochs per client (SCADA, WUSTL)	20
Epochs per server (SCADA, WUSTL)	100
Batch size	64
Learning rate	0.0001
Label ratio (BoT-IoT)	10%
Label ratio (SCADA, WUSTL)	30%

VII. RESULTS AND ANALYSIS

In this section, we present the multiclass classification results, highlighting performance improvements and supporting these improvements through visualizations of the latent space and silhouette scores. The analysis is conducted on the test portions of the datasets described in the previous section, with results averaged over 10 executions.

A. BotIot Results

Fig. 6 compares β-VAE and a standard AE using macro F1-score over 30 federated rounds. It is evident that the VAE consistently outperforms the AE throughout the entire training process. In the early rounds, the VAE exhibits a steeper improvement, surpassing the AE as early as round 2 and maintaining a significant margin thereafter. While the AE shows steady but slower progress, its macro F1-score plateaus around 0.67, whereas the VAE continues to improve and reaches approximately 0.82 by the final round. The Wilcoxon test also confirms that the VAE model is significantly superior to the AE model at the 0.1% significance level, which is notably lower than the commonly accepted significance threshold of 0.5%.

To offer a more detailed evaluation, we provide the classification reports for both the autoencoder (Table IV) and the β-variational autoencoder (Table V) after 30 training rounds.

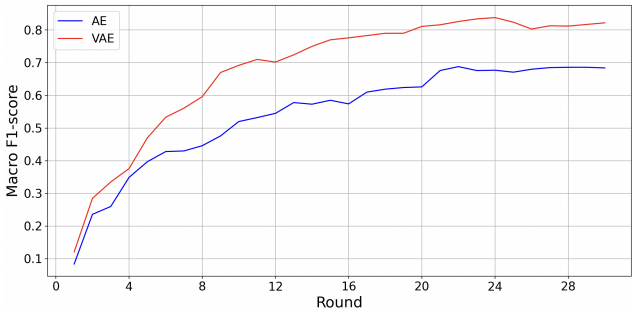


Fig. 6. Macro F1 score comparison between AE and β-VAE (Bot-IoT)

Overall, the β-VAE outperforms the standard AE in most classes, achieving notably higher precision, recall, and F1-scores. However, the AE demonstrates better precision on the Theft class (0.38 vs. 0.21), suggesting fewer false positives in that category. Additionally, AE achieves a slightly higher recall on the Normal class (0.95 vs. 0.93), indicating it identifies a few more true normal instances than the β-VAE; that said, the difference remains minor since the primary focus is on attack detection rather than normal traffic classification. Despite these minor exceptions, the β-VAE attains significantly higher overall accuracy (0.98 vs. 0.82) and macro F1-score (0.82 vs. 0.67), confirming its superior detection capabilities on the Bot-IoT dataset.

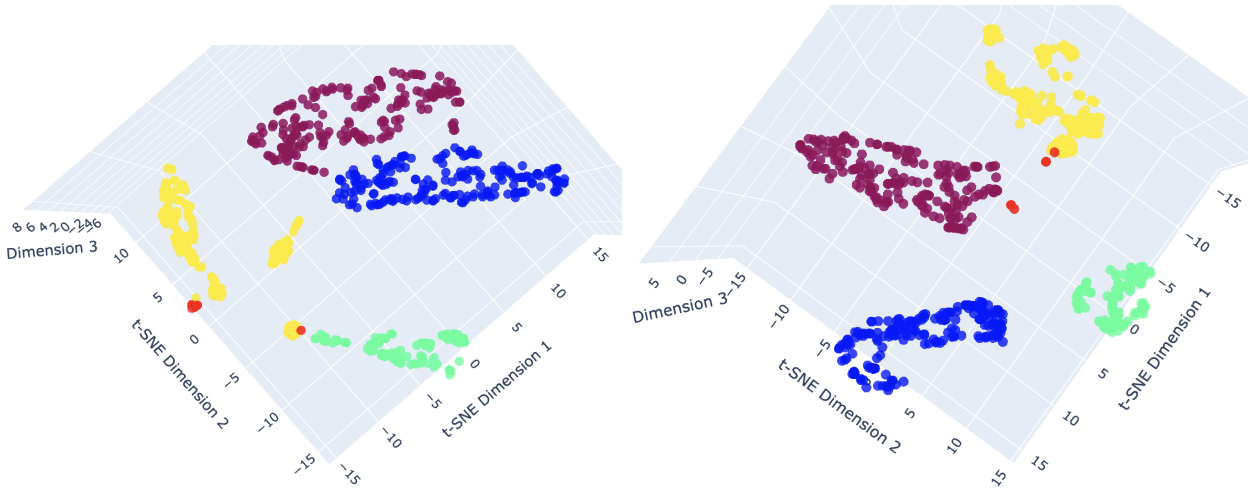
TABLE IV
CLASSIFICATION REPORT FOR THE AE (BOT-IOT)

Class	Precision	Recall	F1-score
DDoS	0.87	0.72	0.74
DoS	0.83	0.95	0.86
Normal	0.74	0.95	0.75
Reconnaissance	0.98	0.78	0.82
Theft	0.38	0.30	0.19
Accuracy		**0.82**	
Macro avg	0.76	0.74	**0.67**
Weighted avg	0.86	0.82	0.80

TABLE V
CLASSIFICATION REPORT FOR THE β-VAE (BOT-IOT)

Class	Precision	Recall	F1-score
DDoS	0.99	0.96	0.97
DoS	0.96	1.00	0.98
Normal	1.00	0.93	0.96
Reconnaissance	1.00	0.89	0.92
Theft	0.21	0.49	0.28
Accuracy		**0.98**	
Macro avg	0.83	0.85	**0.82**
Weighted avg	0.98	0.98	0.97

A visual inspection of the latent spaces (see Fig. VII-A) reveals notable differences in how the AE and the β-VAE organize the data. The comparison was conducted with a fixed random seed (42) and Glorot uniform initialization for both models to ensure fairness, based on a single execution. The latent space produced by the β-VAE achieves a higher silhouette score of 0.3559 compared to 0.2819 for the AE,



(a) AE latent space (Silhouette Score = 0.2819).

(b) β-VAE latent space (Silhouette Score = 0.3559).

Fig. 7. AE vs. β-VAE Latent Spaces (Bot-IoT) Colors: blue — DoS, purple — DDoS, yellow — Reconnaissance, green — Normal, red — Theft.

indicating better separation of classes. More specifically, the blue and purple classes, corresponding to DoS and DDoS attacks respectively, are better separated in the VAE's latent space. Additionally, the VAE represents the red class (Theft) more clearly, with less overlap with the yellow class (Reconnaissance), whereas the AE tends to mix these two classes. The VAE also forms a distinct cluster for the green class (normal traffic), while the AE shows this class scattered and intermixed with Reconnaissance and Theft. These observations, supported by the silhouette scores, suggest that the β-VAE yields a more structured and informative latent representation, making it better suited for classification. Additionally, the β-VAE's latent space can be leveraged for data augmentation – particularly for rare labeled data – or to simulate private client data, enabling training on generated data instead of real data and thereby mitigating risks of model attacks through reverse engineering.

TABLE VI
ACCURACY & F1: LITERATURE VS. OURS (BOT-IOT)

Mode	Model	Source	Acc	F1-Macro
Centralized 100% labels	RNN	[25]	96.76	88.8
	CNN	[25]	96.02	77.8
	DNN	[25]	95.76	77.6
	RF	[26]	91	–
	NB	[26]	71	–
	DT	[26]	91	–
	RF	[27]	98	–
	NB	[27]	70	–
	DT	[27]	91	–
	GradBoost	[27]	**99**	–
Federated 100% labels	RNN	[25]	**94.50**	–
	CNN	[25]	94.06	–
	DNN	[25]	92.02	–
Federated 10% labels	FLuIDS	[6]	82.50	67.0
	FLuIDS+VAE	**Ours**	**98.00**	**82.0**

As shown in Table VI, we highlight the top results in each category: centralized, federated (100% labels) and our method (10% labels). Despite limited supervision, our approach achieves 98% accuracy and 82% F1-macro, approaching centralized performance while clearly outperforming federated baselines.

B. *Wustl-ehms-2020*

Fig. 8 presents a comparison between the performance of the β-VAE and a standard AE in terms of macro-averaged F1-score over 20 rounds of federated training. The results show that the β-VAE converges more quickly and reaches a higher score (0.63), whereas the AE plateaus at a lower value (0.53). A Wilcoxon test further confirms that the performance difference is statistically significant in favor of the β-VAE at the 0.1% significance level.

The method in [28] is the most suitable for comparison because it is federated and reports results using the same number of clients (10). However, it relies on 100% labeled data, achieving a 63% F1 score and 91.40% accuracy. Our approach achieves a 63% F1 score and 92.2%, accuracy while using only 30% of labeled data. The article also includes comparisons with centralized methods and demonstrates that their results outperform DNN (90.52%), LSTM (76.69%), CNN-LSTM (89.08%), and SVM (92.40%), which means our approach surpasses these methods as well.

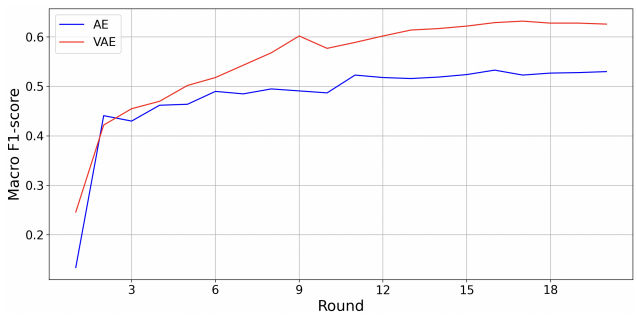


Fig. 8. Macro F1-score comparison between AE and β-VAE (Wustl-2020)

C. SCADA

Fig. 9 compares AE and VAE in terms of Macro F1-score over 18 training rounds. From round 4 onward, VAE consistently outperforms AE, reaching 0.80 versus 0.74. A Wilcoxon test confirms VAE's superiority at the 0.1% significance level.

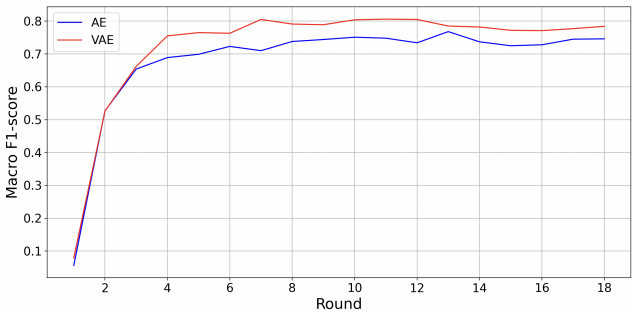


Fig. 9. Macro F1 score comparison between AE and β-VAE (SCADA)

This dataset was used to compare results with the original FLuIDS method [6], which reported an 84% F1-score using 30% labeled and 70% unlabeled data (Ru = 2) with a standard autoencoder. Although the same performance was not replicated, the variational autoencoder achieved comparable results, likely due to the use of different libraries.

VIII. Conclusion and Future Works

Our method, based on a β-variational autoencoder, demonstrated strong performance, with both visual and numerical evaluations supporting its effectiveness. As potential future directions, the framework could be assessed on a wider variety of medical IoT datasets, and temporal modeling could be incorporated to better capture time-dependent attack patterns. Another promising avenue would be the generation of synthetic data to enhance privacy—by reducing risks such as inversion attacks—and to improve performance by increasing the volume of labeled data available to the server.

References

[1] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: a systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021.

[2] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, vol. 195, pp. 346–361, 2022.

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

[4] A. K. Singh, A. Blanco-Justicia, J. Domingo-Ferrer, D. Sánchez, and D. Rebollo-Monedero, "Fair detection of poisoning attacks in federated learning," in *2020 IEEE 32nd international conference on tools with artificial intelligence (ICTAI)*, pp. 224–229, IEEE, 2020.

[5] J. L. Hernández-Ramos, G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, and G. Kambourakis, "Intrusion detection based on federated learning: a systematic review," *ACM Computing Surveys*, 2023.

[6] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semisupervised learning for attack detection in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 286–295, 2022.

[7] F. Naeem, M. Ali, and G. Kaddoum, "Federated-learning-empowered semi-supervised active learning framework for intrusion detection in zsm," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 88–94, 2023.

[8] O. Aouedi, G. Jajoo, and K. Piamrat, "Metals: semi-supervised federated active learning for intrusion detection systems," in *2024 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, IEEE, 2024.

[9] Y. Li and Y. Li, "Semi-supervised federated learning for collaborative security threat detection in control system for distributed power generation," *Engineering Applications of Artificial Intelligence*, vol. 148, p. 110374, 2025.

[10] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, PMLR, 2016.

[11] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semisupervised federated-learning-based intrusion detection method for internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8645–8657, 2022.

[12] R. Beuran *et al.*, "Fedmse: Semi-supervised federated learning approach for iot network intrusion detection," *Computers & Security*, p. 104337, 2025.

[13] T. V. Nguyen and L. B. Le, "Attention-based interpretable semi-supervised federated learning for intrusion detection in iot wireless networks," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pp. 6832–6837, IEEE, 2023.

[14] R. Snoussi and H. Youssef, "Vae-based latent representations learning for botnet detection in iot networks," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 4, 2023.

[15] C. Fachola, A. Tornaría, P. Bermolen, G. Capdehourat, L. Etcheverry, and M. I. Fariello, "Federated learning for data analytics in education," *Data*, vol. 8, no. 2, p. 43, 2023.

[16] D. P. Kingma, M. Welling, *et al.*, "Auto-encoding variational bayes," 2013.

[17] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[18] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[19] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[20] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *Critical Infrastructure Protection VIII: 8th IFIP WG 11.10 International Conference, ICCIP 2014, Arlington, VA, USA, March 17-19, 2014, Revised Selected Papers 8*, pp. 65–78, Springer, 2014.

[21] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[22] R. F. Woolson, "Wilcoxon signed-rank test," *Encyclopedia of biostatistics*, vol. 8, 2005.

[23] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[25] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, "Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis," *IEEe Access*, vol. 9, pp. 138509–138542, 2021.

[26] A. Sharma and H. Babbar, "Bot-iot: Detection of attacks in iot-cybersecurity for smart transportation," in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, pp. 522–527, IEEE, 2023.

[27] K. Ibrahimi and H. Benaddi, "Improving the ids for bot-iot dataset-based machine learning classifiers," in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1–6, IEEE, 2022.

[28] F. Mosaiyebzadeh, S. Pouriyeh, R. M. Parizi, M. Han, and D. M. Batista, "Intrusion detection system for ioht devices using federated learning," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, May 2023.