AnoFed: Adaptive Anomaly Detection for Digital Health Using Transformer-Based Federated Learning and Support Vector Data Description[☆]

Ali Raza[a,b,*], Kim Phuc Tran[a], Ludovic Koehl[a], Shujun Li[b]

[a] *University of Lille, ENSAIT, GEMTEX–Laboratoire de Génie et Matériaux Textiles, F-59000 Lille, France*

[b] *School of Computing & Institute of Cyber Security for Society (iCSS), University of Kent, UK*

Abstract

In digital healthcare applications, anomaly detection is an important task to be taken into account. For instance, in ECG (Electrocardiogram) analysis, the aim is often to detect abnormal ECG signals that are considered outliers. For such tasks, it has been shown that deep learning models such as Autoencoders (AEs) and Variational Autoencoders (VAEs) can provide state-of-the-art performance. However, they suffer from certain limitations. For example, the trivial method of threshold selection does not perform well if we do not know the reconstruction loss distribution in advance. In addition, since healthcare applications rely on highly sensitive personal information, data privacy concerns can arise when data are collected and processed in a centralized machine-learning setting. Hence, in order to address these challenges, in this paper, we propose AnoFed, a novel framework for combining the transformer-based AE and VAE with the Support Vector Data Description (SVDD) in a federated setting. It can enhance privacy protection, improve the explainability of results and support adaptive anomaly detection. Using ECG anomaly detection as a typical application of the framework in healthcare, we conducted experiments to show that the proposed framework is not only effective (in terms of the detection performance) but also efficient (in terms of computational costs), compared with a number of state-of-the-art methods in the literature. AnoFed is very lightweight in-terms of number of parameters and computation, hence it can be used in applications with resource-constrained edge devices.

Keywords: anomaly detection, federated learning, transformer, autoencoder, Support Vector Data Description, explainable AI

[☆] Please cite this paper as follows: Ali Raza, Kim Phuc Tran, Ludovic Koehl and Shujun Li, "AnoFed: Adaptive anomaly detection for digital health usingtransformer-based federated learning and support vector data description," *Engineering Applications of Artificial Intelligence*, 121:106051, 16 pages, Elsevier, 2023, https://doi.org/10.1016/j.engappai.2023.106051.

[*] Corresponding author

Email addresses: ali.raza@ensait.fr (Ali Raza), kim-phuc.tran@ensait.fr (Kim Phuc Tran), ludovic.koehl@ensait.fr (Ludovic Koehl), S.J.Li@kent.ac.uk (Shujun Li)

1. Introduction

One of the important tasks we often encounter when analyzing real-world data is to determine whether a given instance is normal or an anomaly for a given environment and task. The formal process of detecting or classifying all such data instances (anomalous data points) in a data-driven fashion is known as anomaly detection or outlier detection (Chandola et al., 2009, 2012). Anomaly detection is important because it can be used to detect different types of important real-world problems such as health issues, fraud, and security breaches. In some critical applications such as many related to healthcare, anomaly detection can help avoid catastrophic outcomes, such as loss of human lives. For example, anomalous cells in a Magnetic Resonance Image (MRI) or an irregular segment of an Electrocardiogram (ECG) may indicate the presence of a specific disease such as a malignant tumor or an impeding heart attack (Ahmed et al., 2016; Li and Boulanger, 2020; Fernando et al., 2021), respectively.

Detection of anomalies or outliers has been of great interest to the statistics and Machine Learning (ML) research communities. Many anomaly detection techniques have been developed, including general techniques and more application-specific ones. For example, an ECG (Electrocardiogram) is a quick, safe and painless way to monitor heart conditions (e.g., arrhythmias). Nevertheless, to detect arrhythmias, longer-term ECG monitoring is often required to track the patients' heart conditions for an extended period of time (e.g., 24 hours) (Libby et al., 2021). Recent development in sensing technologies has enabled such longer-term monitoring of patients. Smart and portable devices, such as smart watches, Omron HeartScan (Kaleschke et al., 2009) and the recently developed Hexoskin Smart Garments (Haddad et al., 2020), are revolutionizing cardiac diagnostics by monitoring cardiac activities and transmitting longer-term ECG signals to cloud services for remote analysis by medical professionals. However, such signals are often too long for medical professionals to inspect, who simply cannot spend too much time (hours or longer) on looking at the ECG signal in order to detect possible abnormal signals.

To address the above-mentioned challenges about longer-term ECG analysis, machine learning based anomaly detection methods have been proposed (Adler et al., 2015). For such methods to work, the collected data usually need to be sent to a central cloud service. Such a centralized approach has certain limitations. First, the communication costs can be too high since the data volume is high. Secondly, due to the sensitive nature of healthcare data, there are privacy concerns among patients, their family members, legal guardians and caregivers (Jin et al., 2019; Al-Janabi et al., 2017). Thirdly, to train a sufficiently accurate and robust machine learning model, we normally need a lot of well-labeled data, which can take a long time to collect from a single silo (organization) especially if one or more target health conditions are not very common. Fourthly, providing only results from an anomaly detector cannot magically make people trust the results. Instead, the results need to be explained in a way the user (mainly medical professionals but sometimes patients and carers) can understand (e.g., where exactly is the problem and why the model gave a specific result) (Asan et al., 2020; Xu et al., 2019). Achieving the explainability is often difficult, due to the complexity of many deep learning models and the target health conditions. Fifthly, one limitation of most threshold-based anomaly detection methods is that they do not determine the threshold adaptively. Instead, they rely on the standard deviation (Maussang et al., 2007) or the absolution deviation around the mean (Leys et al., 2013) to determine the threshold.

However, this approach works only when we know the (at least approximate) underlying distribution of data. Unfortunately, most real-time data are not normally distributed so the distribution has to be estimated first. Additionally, in addition to privacy concerns mentioned above, data owners (individual silos) often have other reasons to be unwilling to share data with a central authority, e.g., market competition. All such problems call for more adaptive and privacy-preserving machine learning in a non-centralized setting, and federated learning has been proposed to address such a need (McMahan et al., 2017).

In this paper, we propose AnoFed, a new framework for ECG analysis in federated settings that can achieve the above-mentioned aims. Although the framework is proposed for ECG analysis, the general idea can be extended to other digital health applications. In AnoFed, we apply federated learning for two goals – to provide enhanced data privacy and to reduce communication costs. The use of federated learning allows incremental updates of the global model while new data from participating nodes (edge devices) come in, therefore achieving adaptivity. To support resource-constrained edge devices as local nodes, we propose novel lightweight transformer-based Autoencoders (AE) and Variational Autoencoders (VAE) as building blocks of AnoFed. Moreover, we combine the proposed models with the Support Vector Data Description (SVDD) (Tax and Duin, 2004) with kernel density estimation for adaptive anomaly detection, for each global round of training in the federated setting. We also provide an eXplainable AI (XAI) module to trace the most critical part(s) of the ECG that is/are responsible for the detected anomaly.

The key contributions of this paper are summarized as follows.

1. To the best of our knowledge, AnoFed is the first lightweight (in terms of the number of parameters and the number of local/global training rounds required for the desired efficacy) VAE and an AE based on transformers in federated setting (for enhanced privacy of user) for ECG anomaly detection. Owning to the use of transformers, AnoFed can combine the merits of both CNN- and RNN-based models.
2. We propose a new framework which is a combined design of the VAE/AE and SVDD with kernel density estimation for adaptive anomaly detection. The VAE/AE extracts features from the input data in the form of error vectors which are then used to train the SVDD with kernel density estimation, which allows the proposed framework to provide state-of-the-art results even when the data distribution changes in the local clients in federated setting.
3. We design an XAI module to improve explainability of the results of our framework, which helps enhance the trust of users on the proposed framework.
4. We trained and tested our proposed framework using two datasets from PhysioNet (Goldberger et al., 2000) and a testbed with three edge devices and a global server, and the results show that our framework could achieve state-of-the-art detection accuracy with the proven ability of automatically adapting to different distributions of the data.

The rest of the paper is organized as follows. Section 2 presents related work and background. Section 3 presents the proposed framework. Sections 4 and 5 presents the experimental setup and performance analysis, respectively. Comparisons with some state-of-the-art methods and an analysis of the time complexity of the proposed framework are also included in Section 5. Section 6 presents the limitation and future work. Finally, the last section concludes the paper.

2. Related Work and Background

In this section, we discuss the background knowledge and related work for this study.

2.1. Machine Learning for Anomaly Detection

The use of machine learning for anomaly detection has been steadily growing in academia as well as in industry due to its proven performance. It finds its application in many domains such as cyber security (Vanerio and Casas, 2017), telecommunication and networking (Kwon et al., 2019), and healthcare (Krichen, 2021). An extensive survey of such anomaly detection methods can be found in (Chandola et al., 2009), which gives a board review of different methods including those based on machine learning as well as those that do not use machine learning. Moreover, the survey also discusses applications of anomaly detection in cyber security, medical image analysis, natural language processing, wireless sensing, etc. In the cyber security research literature, intrusion detection has been the topic of many researchers. For example, a comprehensive study on anomaly detection-based intrusion detection techniques was presented in (Yu, 2012), covering statistical and machine learning-based techniques. Kwon et al. (2019) presented network anomaly detection based on restricted Boltzmann machine-based deep belief networks and deep recurrent neural networks, as well as other methods based on more traditional machine learning algorithms. Durga et al. (2019) presented anomaly detection using machine learning (including deep learning) algorithms in the context of the Internet of Things (IoT) based healthcare. Also focusing on healthcare-related applications, Wang et al. (2016) applied deep learning to analyze physiological signals that allow doctors to identify latent health risks. Similarly, some researchers have investigated the potential of using smartphones and wearable devices to capture data in this regard, and the latter is seen as a promising solution for healthcare (Amin et al., 2016; Banaee et al., 2013). However, there are still many challenges that need addressing, such as privacy preservation, explainability, and adaptive anomaly detection with evolving data. Furthermore, RNN and CNN-based models are currently being used for sequence-to-sequence modeling. However, RNNs are costly for sequence-to-sequence modeling because of their serial computation and they generally require more computational time. CNNs reduce the cost of sequence-to-sequence modeling because they are easy to parallelize, which is not possible in RNNs. However, one disadvantage of CNNs is that they require a very large number of layers to capture the long-term dependencies in the sequential data, eventually making the model so large that would be impractical to use in resource constraint devices. Transformers address the long-term dependencies with the self-attention mechanism with positional encoding, which can be easily parallelized. Hence, they can achieve merits of both CNNS and RNNs, which is the motivation for the use of transformers in our proposed framework.

2.2. Transformers

Vaswani et al. (2017) introduced transformers for sequence-to-sequence learning. Transformers work primarily based on the attention mechanism. The attention mechanism maps the importance of each part of the input by looking at the entire input sequence. The transformation mechanism of the transformers functions in the same way as that of the RNNs (recurrent neural networks), i.e., by using an encoder and a decoder, with the exception that transformers do not employ any recurrent networks. The encoder and the decoder both consist of a stack of multi-head attention, addition, normalization, and feed-forward layers. Figure 1 presents the multi-head attention module (Vaswani et al., 2017). Transformers also use a positional encoding layer to retain the positions of different parts of the input and output sequence as there are no recurrent networks to remember such information.

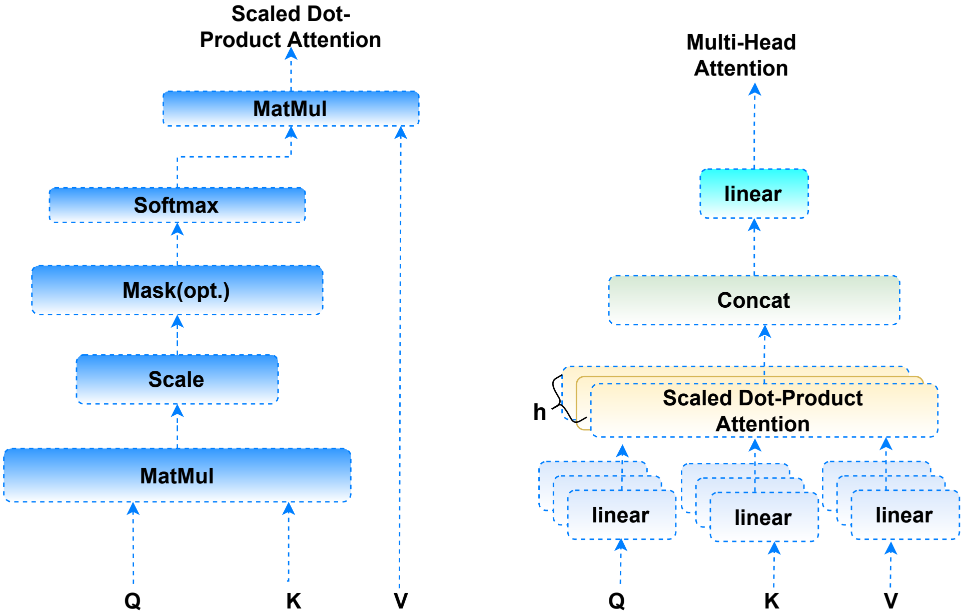


Figure 1: Left: The scaled dot-product attention mechanism, Right: Multiple attention layers of the multi-head attention mechanism proposed by Vaswani et al. (2017).

The attention mechanism is given by the following equation:

$$\text{Attention}(Q, K, V) = \text{Sofmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where Q is a query matrix (the input sequences as vectors), K represents the keys (sequences represented as vectors) and V are the values (sequences represented as vectors). In order to learn different representations of Q, K, and V, which can be beneficial to the model, the attention mechanism is employed multiple times with projections of Q, K, and V. The parallelization of the attention mechanism is shown in Figure 1 (the sub-figure on the right). The linear representations are obtained by multiplying Q, K, and V with the weight matrices W that are computed during the training process. The multi-head attention module, which is responsible to join the encoder and the decoder, makes sure that the input sequence of the encoder is taken into account with the input sequence of the decoder up to a given location in sequence. A feed-forward layer is employed in both the encoder and the decoder following the multi-attention head mechanism.

The ability of transformers to work without any RNNs improves the results in various applications, e.g., for language translation (Devlin et al., 2018) and for HAR (human activity recognition) (Li et al., 2021).

2.3. Federated Learning

Federated learning (FL) (McMahan et al., 2017) employs distributed clients (also called edge devices) to train a combined global model, without requiring clients to directly share local data with a central repository. This is achieved by training a model at each participating client locally and sharing the trained model parameters (e.g., weights or gradients of a neural network model) with a central or global server **GS**, which combines the shared model parameters of the locally trained models to achieve a robust global model. FL aims to train a global model **GM** using the updated model parameters of locally trained models of $K > 1$ participating clients $\{\mathbf{E}_k\}_{k=1}^{K}$. Each client $\mathbf{E}_k$ participates in the r-th global round with its local data $\mathbf{D}_k^r$ to train a local model $\mathbf{LM}_k^r$. There are two main approaches commonly used to make local updates: (1) federated stochastic gradient descent (SGD), where clients send an update after each training epoch, and (2) federated averaging, where clients send an update after multiple training epochs. Past studies have shown that federated averaging outperforms federated SGD and the former is more efficient since it needs fewer communication rounds. Federated averaging can be described by the following two equations: Eq. (2) gives the local updates and Eq. (3) gives the global updates:

$$\mathbf{LM}_k^{r+1} = \mathbf{LM}_k^r - \alpha g_k^r, \tag{2}$$

$$\mathbf{GM}_{r+1} = \sum_{i \in D_k} \frac{n_k}{n_t} \mathbf{LM}_k^{r+1}, \tag{3}$$

where $\mathbf{LM}_k^{r+1}$ is the locally updated model at the r-th global round, g_k is the gradient of backpropagation, $\mathbf{GM}_{r+1}$ is the updated global model after the r-th global round, α represents the learning rate, n_t is the total number of training samples of all the participating clients, and n_k is the number of training samples of the k-th client.

Federated learning provides various advantages compared to the centralized approach. For instance, one of its advantages over the centralized approach is that, in case of processing sensitive personal data of human users such as patients, it can provide more protection to such sensitive data. This is because of a key property of FL: the central server trains the global model without seeing locally sensitive data, which remains locally with local edge devices at the client side. Moreover, it can significantly reduce communication costs because each local client only needs to share parameters of the locally trained model, instead of a huge amount of data with the central server. McMahan et al. (2017) showed that federated averaging can reduce communication costs more (sometimes up to 100%) than federated SGD, so it has been used more often in many applications (Yang et al., 2019).

2.4. Autoencoders and Variational Autoencoders

Autoencoders (AEs) (Baldi, 2012) are neural networks trained mainly using unsupervised learning. They have been extensively used for data denoising and compression (Smys et al., 2020; Yildirim et al., 2018). An AE usually consists of two main components: an encoder and a decoder. The encoder learns latent space vector representations during the training phase, while the decoder learns to reconstruct the original input given the latent vectors. We depict an AE with a single hidden layer (Cozzolino and Verdoliva, 2016) in Figure 2.

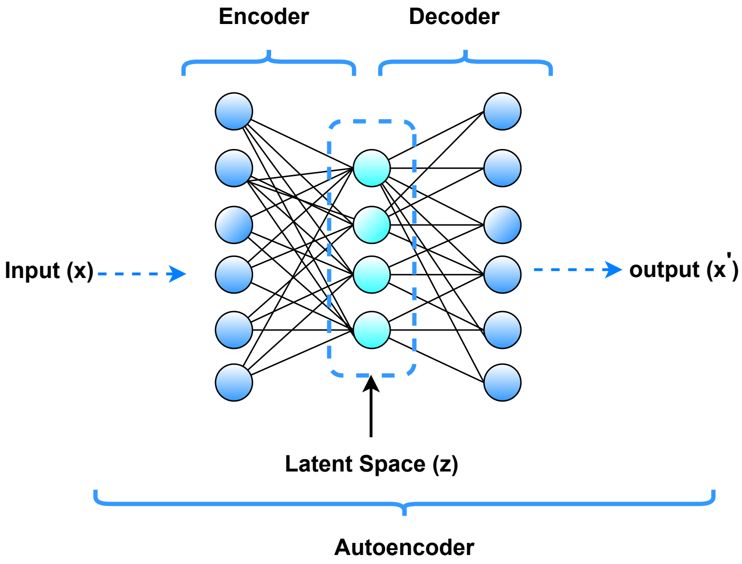


Figure 2: An AE with a single hidden layer.

The mathematical formulation of an encoder is given by the equation below:

$$Z = \phi_1(W_1 X + B_1), \tag{4}$$

where $X \in R^k$ is a k-dimensional input vector, z is a latent space vector, ϕ_1 is an activation function and W_1 represents the weights matrix of the encoder and B_1 is the bias vector. The parameters W_1 and B_1 are randomly initialized at the start and updated during the training phase. For the decoder, the following equation shows how it works:

$$X' = \phi_2(W_2 Z + B_2), \tag{5}$$

where $X' \in R^k$ is a k-dimensional output vector obtained using the latent representation given by the encoder, ϕ_2 is an activation function, W_2 and B_2 are weights matrix and the bias vector of the decoder, respectively. Each input X of the encoder part of an AE is mapped into a latent space vector. This latent space vector is used as the input of the decoder that produces X' (a reconstructed version of X) as the output. The model's internal parameters are trained by minimizing the reconstruction loss L with a suitable optimizer, given by the following equation:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} ||X_i - X_i'||^2, \tag{6}$$

where N is the total number of input vectors, θ denotes the model's parameters, and X_i and X'_i are the i-th input and output vectors, respectively. Generally, the latent representations have a lower dimensionality than the input vector, let us say k, so that the AE will keep the important and relevant information necessary to reconstruct the input (Seyfioğlu et al., 2018). When a trained AE is used for anomaly detection, the reconstruction loss is normally used to detect the anomalies (Baur et al., 2018; Oh and Yun, 2018).

Variational autoencoders (VAEs) (Oussidi and Elhassouny, 2018; Zhai et al., 2018) are structurally similar to AEs, with the only difference being that the VAEs learn a latent distribution while the AE learns a point in the latent space. This latent distribution is regularized to be close to a standard normal distribution. For a given input data X, let us assume that X is computed using its corresponding latent variable Z that cannot be observed directly. If we denote the prior distribution of Z as $p(Z)$, and consider that the input Z is sampled from the conditional likelihood $p(X|Z)$, then Bayes theorem gives the link between the prior $p(Z)$, likelihood $p(X|Z)$, posterior distribution $p(Z|X)$, as shown in the following equation:

$$p(Z|X) = \frac{p(X|Z)}{p(X)}. \tag{7}$$

Given an input dataset X defined by an unknown probability function $p(X)$ and a latent vector Z, a VAE learns from the input to get a distribution $p_\theta(X)$, where θ is the set of the network parameters. Equation (8) represents the mathematical formulation of the unknown probability function.

$$p(X) = \int p(X, Z)dz. \tag{8}$$

Unfortunately, $p(X)$ is intractable distribution and hence we cannot compute it directly. However, by leveraging variational inference the problem of intractable distribution can be solved. If we consider $p(Z|X)$ to be approximated by another tractable distribution $q(Z|X)$, then the parameters of $q(Z|X)$ can be defined to be very similar to $p(Z|X)$ to infer the intractable distribution. By minimizing the KL-divergence (a metric describing the difference between two probability distributions), we can ensure that $q(Z|X)$ is similar to $p(Z|X)$,

$$\min\nolimits_{\mathrm{KL}}(q(Z|X)||p(Z|X)), \tag{9}$$

We can minimize the KL divergence by minimizing the following:

$$E_q(Z|X)\log p(X|Z) - \mathrm{KL}(q(Z|X)||p(Z)). \tag{10}$$

The above equation ensures that the learned distribution q is similar to the prior distribution p. A VAE mapping X to Z and reconstructing X from Z is shown in Figure 3.

The decoder learns to reconstruct the input from the latent space vector. Furthermore, re-parameterization is used to calculate the relationship between the model's internal parameters and the loss using backpropagation. Re-parameterization randomly samples ϵ from a unit normal distribution, and then shifts the random sample ϵ with mean μ and scales it by the variance σ of the latent distribution, given by the following equation:

$$Z = \mu + \sigma \times \epsilon. \tag{11}$$

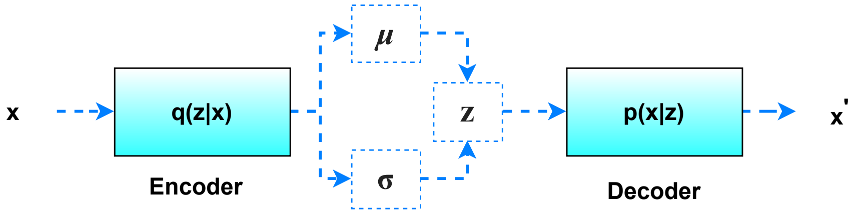


Figure 3: The general structure of a VAE.

The loss function for a VAE consists of two different losses (as shown in the equation below): one is used to penalize the reconstruction loss, and the second (KL-loss) is used to ensure that the learned distribution $q(Z|X)$ is similar to the true prior distribution $p(Z)$, which follows a unit normal distribution, across each dimension j of the latent space.

$$L(X, X') + \sum_{j} \mathrm{KL}(q_j(Z|X)||p(Z)). \tag{12}$$

2.5. Support Vector Data Description (SVDD)

The Support Vector Data Description (SVDD) (Tax and Duin, 2004) leverages a support vector classifier (Chang and Lin, 2001) to construct a spherical boundary around a given distribution of a dataset, with a minimum volume containing as much as possible data samples from the given data distribution. Let us suppose that $\{X_i \in \mathbb{R}^d\}_{i=1}^N$ are a set of N d-dimensional training samples, a and R denote the center and the radius of a sphere covering the training set, respectively. Huang et al. (2011) formulated this goal as a constrained convex optimization problem, given as follows:

$$\min_{R,a,\xi_i} F(R, a, \xi_i) = R^2 + C \sum\nolimits_i \xi_i, \text{ s.t.}$$
$$\begin{cases} ||X_i - a||^2 \leq R^2 + \xi_i, \\ \xi_i \geq 0, \end{cases} \quad i = 1, \dots, N, \tag{13}$$

where the slack variable ξ_i defines the possibility of anomalous (outliers) data in the given training data. The parameter C is used to balance the trade-off between the volume inside the boundary and the errors. The Lagrangian function with Lagrange multipliers α_i and γ gives

$$\begin{aligned} L(R, a, \xi_i, \alpha_i, \gamma_i) &= R^2 + c \sum_i \xi_i \\ &- \sum_i \alpha[R^2 + \xi_i - (X_i - a)^2] - \sum_i \gamma_i \xi_i. \end{aligned} \tag{14}$$

By setting the partial derivatives of a, R, and ξ_i to zero, we can achieve the following constraints:

$$\sum_i \alpha_i = 1, A = \sum_i \alpha_i X_i \tag{15}$$

$$C - \gamma_i - \alpha_i = 0 \implies 0 \geq \alpha_i \geq C. \tag{16}$$

From the above equations, we can get

$$\max L = \sum_i \alpha_i (X_i \cdot X_i) - \sum_{i,j} \alpha_i \alpha_j (X_i \cdot X_j), \text{ s.t.}$$

$$\begin{cases} \sum_i \alpha_i = 1, \\ 0 \leq \alpha_i \leq C, \end{cases} \quad i = 1, \ldots, N, \quad (17)$$

where, $\cdot$ operator denotes the inner product between two vectors. A training sample X_i and its corresponding α_i should follow one of the following conditions:

- $||X_i - a||^2 < R^2 \implies \alpha_i = 0;$
- $||X_i - a||^2 < R^2 \implies 0 < \alpha_i < C;$
- $||X_i - a||^2 < R^2 \implies \alpha_i = C.$

The samples whose coefficients follow $\alpha_i > 0$ are known as support vectors. The center of the sphere can be obtained by Eq. (15). The radius R can be obtained by calculating the distance from any support vector with $0 < \alpha_i < C$ to the center. In order to test if a given sample Z is inside or outside of the defined boundary of the sphere, the distance from the center to Z is calculated. If the distance is smaller than the radius R, then Z is considered inside and not an outlier, given as follows.

$$||Z - A||^2 = (Z \cdot Z) - 2\sum_i \alpha_i (Z \cdot X_i) + \sum_{i,j} \alpha_i \alpha_j (X_i \cdot X_j) \leq R^2. \quad (18)$$

The method can be made more flexible (Tax and Duin, 2004; Vapnik, 1995) by employing new inner products satisfying Mercer's theorem. Moreover, a polynomial kernel and the Gaussian kernel can also be employed to achieve more flexibility as discussed in (Tax and Duin, 2004; Lee et al., 2005).

3. Proposed Framework AnoFed

In this section, we first give an overview of the proposed framework AnoFed and then provide details of different components.

3.1. Overview

An overview of the proposed framework is shown in Figure 4. Let us assume that there are K edge devices participating in FL to jointly train an ECG anomaly detection system. In order to train a joint global model **GM**, all the edge devices connect to a central server or global server **GS**, where an edge device is represented as $\mathbf{E}_k$ and data in each edge device is represented as D_k, $k = 1, \ldots, K$. **GM** represents the global updated model and $\mathbf{LM}_k$ the local model at $\mathbf{E}_k$. We divide one global round into two phases: in Phase one, **GM** is AE/VAE, and in Phase two, **GM** is SVDD. Additionally, we denote the weights of $\mathbf{LM}_k$ in Phase one as W_k and averaged weights of **GM** as AW. In Phase two, we denote the weights of $\mathbf{LM}_k$ as SW_k and averaged weights of **GM** as SAW. In Phase one, all the edge devices train the VAE/AE and use a callback to monitor the reconstruction loss. When the reconstruction loss is not improving anymore, each edge

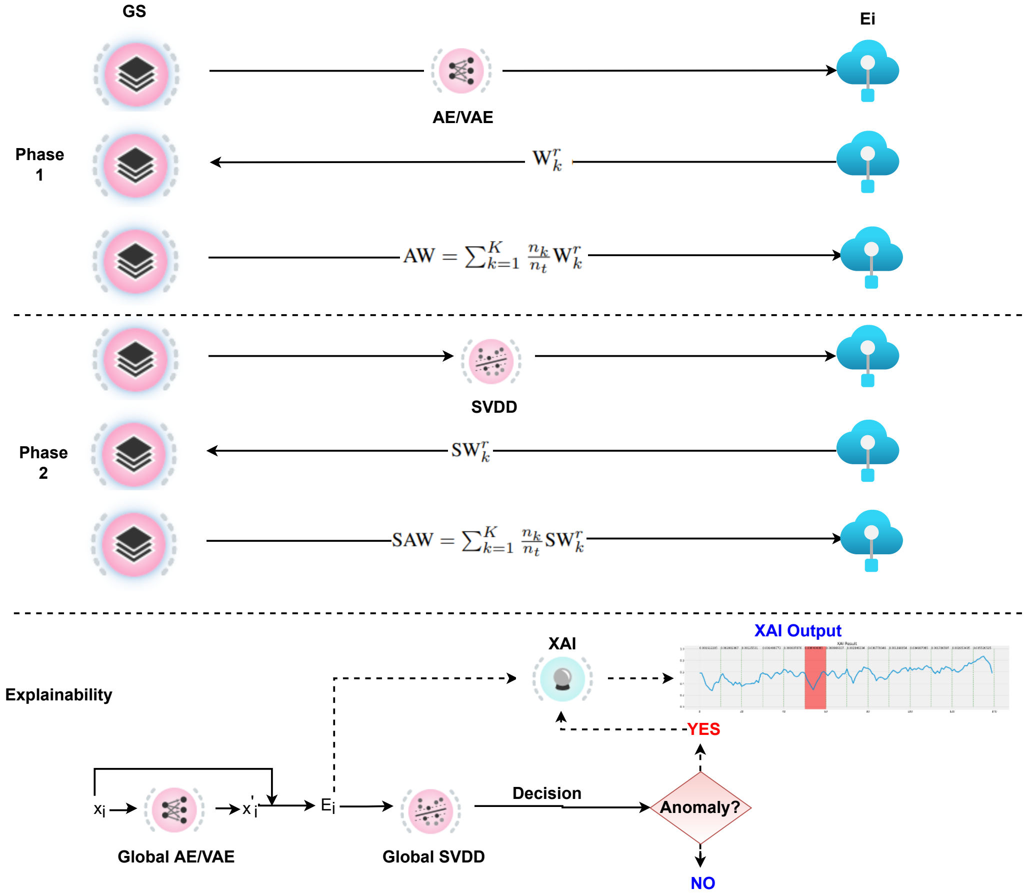


Figure 4: Overview of the proposed framework AnoFed.

device sends the local weights of VAE/AE to the global server and waits for the global server to send the aggregated weights for VAE/AE. After aggregating the updates, the global server sends the aggregated updates to all participating edged devices. Each edge device then computes the error vectors using the aggregated updates and starts Phase two: training of the SVDD. Similar to the first phase, each edge device monitors the classification performance of the local SVDD and sends the updates to the global server once it stops improving, which are then aggregated and sent back to all participating edge devices. The explanations can be achieved in both the global model and the local models. It should be noted that the XAI module does not require training, instead, it simply takes the output of the AE/VAE and that of the SVDD classifier to produce a visualized explanation of the detected anomaly. The training of each global round is described by Algorithm 1. In this algorithm, n_k is the number of training samples of the edge device E_k, and $n_t = \sum_{k=1}^{K} n_k$ is the total number of samples across all edge devices.

Algorithm 1: The training procedure of the proposed framework AnoFed (a single global round)

```
Input: Data from edge devices $D_1, D_2, \ldots, D_k$
Output: AW and SAW
GS compiles the initial GM
GS sends GM to the requesting $E_k$
$E_k$ receives GM, trains it using local data $D_k$, and sends updated weights $\mathrm{W}_k^r$ of $\mathbf{LM}_k^r$ to GS
GS waits to receive updates from all $K$ edge devices.
if updates received form $K$ edge devices then
    $\mathrm{AW} = \sum_{k=1}^{K} \frac{n_k}{n_t} \mathrm{W}_k^r$
$G_s$ sends AW to $E_k$
$E_k$ updates its local model with AW
$E_k$ computes error vectors
$E_k$ trains SVDD using error vectors and sends updated weights $\mathrm{SW}_k^r$ to GS
if updates received form $K$ edge devices then
    $\mathrm{SAW} = \sum_{k=1}^{K} \frac{n_k}{n_t} \mathrm{SW}_k^r$
$G_s$ sends SAW to $E_k$
```

3.2. Proposed AE and VAE

Since both AE and VAE performed well according to the literature, we tested an AE and a VAE with the same number of transformer blocks in order to see which one performs better. The proposed AE and VAE are shown in Figure 5.

In both the AE and the VAE, the encoder module consists of the first input layer that takes the ECG segment of 140 stamps as the input. The output of the input layer is passed through the transformer layer. The transformer layer consists of many sub-layers as shown in Figure 5. The first one is an augmentation layer, which applies random (with loss of information) and more realistic (without significant loss of information) transformations to increase the diversity of the training set. The output from the augmentation layer is normalized using a normalization layer, and then a multi-head attention layer

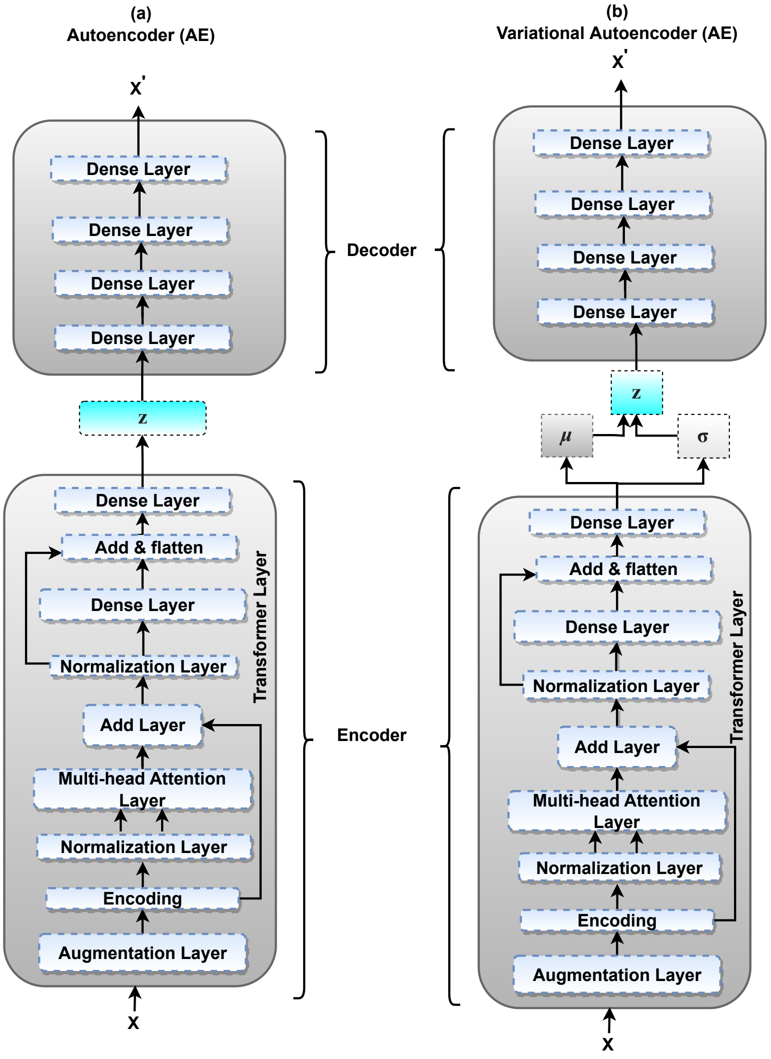


Figure 5: The AE and the VAE for our proposed framework AnoFed.

applies the self-attention mechanism. The self-attention mechanism takes a sequence and outputs a corresponding sequence vector. Let us consider k-dimensional input vectors $X_1, X_2, \ldots, X_t$ and $X_1', X_2', \ldots, X_i'$ as their corresponding output vectors. To compute the vector X_i', the self-attention mechanism computes weights averaged over all the input vectors, given by the following equation:

$$X_i' = \sum_j W_{ij} X_j, \tag{19}$$

where j indexes the whole sequence and the sum of all the indexes is equal to one. The weight W_{ij} is computed using the following function over X_i and X_j:

$$W_{ij} = X_i^T X_j. \tag{20}$$

As the output of a dot product is a real value, in order to map the values in the range of 0 and 1, and to ensure that the sum over the whole sequence sums to 1, we employ a softmax function, given as follows:

$$W_{ij} = \frac{\exp(W_{ij})}{\sum_j \exp(W_{ij})}. \tag{21}$$

The dot product in the self-attention mechanism expresses the correlation of input features. The output features are obtained by computing the weighted sum over the whole input sample. The output of the multi-head attention layer is combined with the output of the preceding normalization layer using an additional layer. Then, the output of the additional layer is fed into a succeeding normalization layer. The output of this normalization layer is then fed into a dense layer that applies a non-linear transformation to extract further features, given as follow:

$$\text{Output} = f_{\text{activation}}(\text{dot}(\text{input}, \text{kernel})), \tag{22}$$

where $f_{\text{activation}}$ is the activation function, and the kernel is a weight matrix. The output of the final dense layer gives latent space representation in the case of AE, i.e., the encoder maps the input into a lower-dimensional feature space Z. Whereas in the case of VAE, the output is a latent distribution with μ as the mean and σ as the standard deviation, expressing the latent space regularization (enforced to be close to a standard normal distribution). For both AE and VAE, the latent representation is then fed into the decoder. The decoder module consists of four simple dense layers and the final layer uses a sigmoid activation function that gives probability distributions of the candidate classes, whereas the activation function of the remaining layers is a rectifier linear unit (ReLU).

3.3. Proposed SVDD Module

In a federated setting, because the distributions of local data and the global data can differ from each other significantly, anomaly detection methods relying on a static threshold (normally manually selected based on the training data, e.g., the mean plus one standard deviation of the reconstruction loss) may not ensure the global model can still work well. To address this challenge, we propose to use SVDD along with density kernel estimation for adaptive anomaly detection, which can avoid the problem of setting a static threshold. We adopted the SVDD classifier from Tax and Duin (2004) to construct a nonlinear SVDD by employing kernel density estimation, as shown in Figure 6. The kernel maps the input into a new higher-dimensional feature space by applying a nonlinear transformation using a special kernel function. After that, we use the SVDD model in this new higher-dimensional feature space. Hence, the SVDD linear model in this new higher-dimensional feature space represents a nonlinear model in the input space. To train the proposed SVDD, $\mathbf{E}_k$ first computes the error vectors E_i, given by the following equation:

$$E_i = X_i - X_i'. \tag{23}$$

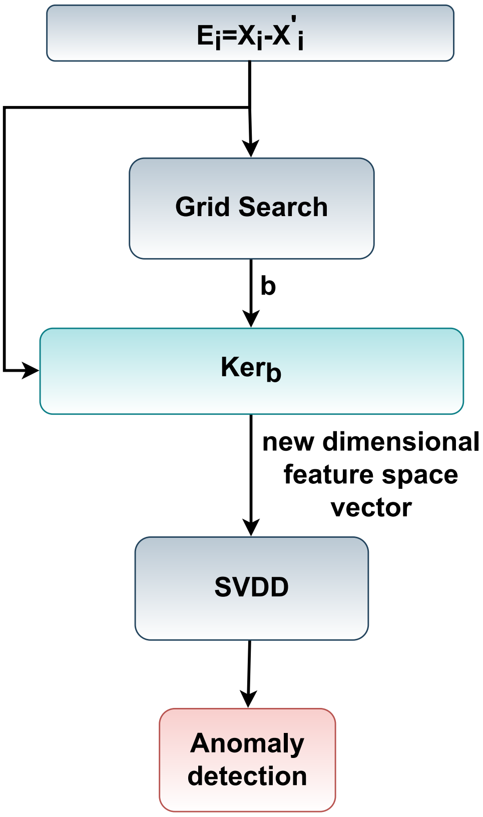


Figure 6: Adaptive anomaly detection using SVDD.

Let $E_1, E_2, \ldots, E_k$ be independent and identically distributed samples with f, where f is the unknown density at any given sample E, then kernel density estimator function of f is given by the following equation:

$$f_b(E) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{Ker}_b(E, E_i), \tag{24}$$

where Ker is the Laplace radial-basis kernel and b is the adaptive bandwidth. A scaled kernel with bandwidth b is defined as follows:

$$\mathrm{Ker}_b(X) = \frac{1}{b}\mathrm{Ker}(\frac{X}{b}).$$

In order to come up with an adaptive bandwidth for the SVDD, we use a grid search mechanism. The grid search mechanism is provided with the error vectors, which apply a grid search using pre-defined hyperparameters and outputs the best bandwidth denoted as b.

3.4. Proposed XAI Module

In sensitive applications such as digital healthcare, we need to inform users about why the model reached the output results and which portion(s) of the input is/are responsible for the certain output. In our case, health professionals (end users of the anomaly detection system) would be interested in knowing which portion of the input is responsible for the maximum reconstruction loss because that portion would most influence and contribute to an anomaly. Hence, identifying such a segment in the input would help identify the key pattern(s) of an anomalous ECG signal. However, deep learning-based modules are complex in nature to explain, i.e., a high number of model parameters. To address this challenge, we provide a model-agnostic XAI module to identify the segment (of desired length/window size) of the input ECG sample that contributes the most to the reconstruction loss, thereafter to the anomaly. Let $X = (x_1, \ldots, x_n)$ be the input to the AE/VAE, and $X' = (x'_1, \ldots, x'_n)$ be the corresponding reconstruction, where X is an ECG signal with n time stamps. Then, the proposed XAI module identifies the key segment of the ECG signal with the maximum reconstruction loss by Algorithm 2. In this algorithm, s is the number of sub-segments of the input, S_{pos} is the starting position of a sub-segment, E_{pos} is the end position of the sub-segment, $\max_{\text{loss}}$ is the maximum reconstruction loss, $\max_{\text{loss-position}}$ is the position of the sub-segment in the input, and SLoss is the reconstruction loss of a given sub-segment.

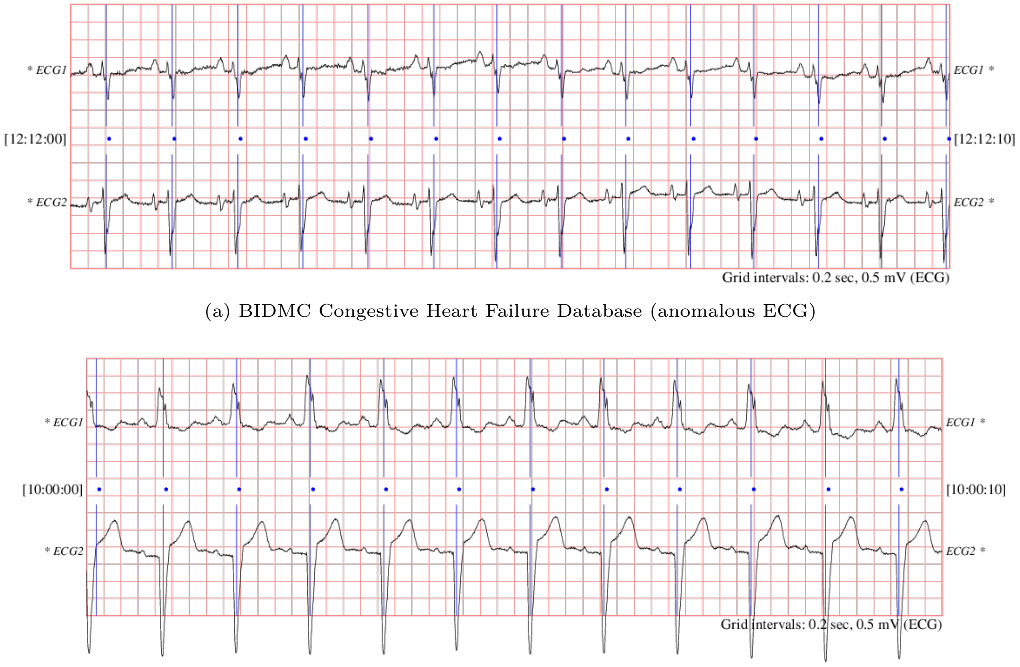
Algorithm 2: The XAI module used in the proposed framework AnoFed

```
Input: Anomalous ECG signal, desired window size $W_s \leq n$
Output: Segment with the maximum reconstruction loss
calculate the possible number of sub-segments $s$ and set the start position
  $S_{\text{pos}} = 0$, end the position $E_{\text{pos}} = W_s$ $\max_{\text{loss}} = 0$, $\max_{\text{loss-position}} = (0, 0)$
for $i=1, 2, \ldots, s$ do
    $\text{SLoss} = \text{abs}(\text{mean}(X[S_{\text{pos}} : E_s] - X'[S_{\text{pos}} : E_s])^2)$
    if $\text{SLoss} > \max_{\text{loss}}$ then
        $\max_{\text{loss-position}} = (S_{\text{pos}}, W_s)$
        $\max_{\text{loss}} = \text{SLoss}$
    $S_{\text{pos}} = S_{\text{pos}} + W_s$
    $E_{\text{pos}} = E_{\text{pos}} + W_s$
Return $\max_{\text{loss}}, \max_{\text{loss-position}}$
```

4. Experimental Setup

4.1. Dataset Description

To train and test the proposed framework AnoFed, we used a combination of two datasets from PhysioNet (Goldberger et al., 2000). For the anomaly class, we used the BIDMC Congestive Heart Failure Database. This database contains longer-term ECG recordings of severe congestive heart failures from 15 subjects, out of which 11 were men (aged 22 to 71), and 4 were women (aged 54 to 63). Further details about the data are available in (Goldberger et al., 2000). The data was pre-processed by



(a) BIDMC Congestive Heart Failure Database (anomalous ECG)

(b) MIT-BIH Normal Sinus Rhythm Database (normal ECG)

Figure 7: An example sample from each of the two used databases.

extracting each heartbeat of equal length using interpolation, and the class values were obtained by automated annotations (Chen et al., 2015). For normal subjects, we use the Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) normal sinus rhythm (Goldberger et al., 2000). It includes 18 longer-term ECG recordings of 18 subjects (5 men, aged 26 to 45, and 13 women, aged 20 to 50) with no significant arrhythmias. We randomly selected 5,000 (2919 normal samples and 2081 anomaly) heartbeats from each dataset to train and test the proposed framework AnoFed. Figure 7 shows one example sample from each of the two selected databases. Table 1 presents additional information about the datasets used. It should be noted that the anomaly class contains different sub-classes of anomalies in it.

Table 1: More information about the datasets used to train and test the proposed framework AnoFed.

Class Name	Class ID	Number of Samples
Normal	1	2919
Anomaly	-1	2081

4.2. Implementation Details

In order to evaluate AnoFed in real time, firstly we developed a testbed using three Raspberry Pi devices (Pi 3 Model B+ with 1.4GHz, 1GB LPDDR2 SDRAM, and 64-bit quad-core ARMv8 CPU) as clients, as shown in Figure 8 and a Dell workstation with 32 GB RAM and an Intel® Core™ i-6700HQ CPU as **GS**. For the initial ($r = 0$) global round, **GS** compiles the AE or the VAE. We used Adam as the optimizer for both the AE and the VAE. We distributed the above-mentioned datasets equally (but randomly selected) among the three edge devices. 75% of the data was used for training by each client or edge, whereas the rest 25% for testing. Secondly, we increase the number of clients to five. In the second setting, we used a non-IID data (unbalanced and skewed) distribution (Edges 3 and 5 with around 630 training samples each, where Edge 3 contains 60% data from the normal class and 40% data from the anomaly class. Edge 5 contains 60% data from the anomaly class and 40% data from the normal class). Each of other clients contain around 1,200 samples selected randomly. 70% of the data was used for training and 30% was used for testing in each client. In both settings, we also kept 1,000 randomly selected samples (not part of the training set in any edge device) to test the global model. Additionally, we used 10-fold cross-validation in both settings. Since our task is to use the reconstruction loss to predict anomalies, we used the normal data only for training the AE and the VAE. Furthermore, each edge used a batch size of 42, and was trained for only three epochs in each global round. The ability of our proposed AE/VAE to achieve the minimum reconstruction loss within three epochs and one global round makes it suitable for resource-constrained devices, which is much needed in many health-related applications. We used a learning rate of 0.001, with a clip value of 0.5. We used mean squared error (MSE) as the loss function. In order to find the best suitable bandwidth for kernel estimation in SVDD, we used sklearn's grid search module with bandwidth space of (3, 0.2,10), and 30-fold cross validation[1].

5. Performance Analysis of the Proposed Framework

In this section, we report the performance of the proposed framework AnoFed using some state-of-the-art metrics.

5.1. Reconstruction Loss

In order to evaluate the performance of AnoFed, we trained both the AE and the VAE in a federated setting following the experimental setup explained in the previous section. Figure 9 shows the reconstruction loss using the proposed AE and Figure 10 shows the reconstruction loss using the proposed VAE. It can be seen that both the AE and the VAE performed very well. We use the blue dotted line to show the point one standard deviation away to the right of the mean of the normal distribution, which can be chosen as a typical static threshold of the classifier. We can optimize this threshold by recursively trying other possible values. However, as mentioned previously, the anomaly detection methods using a static threshold are not compatible with the federated setting due to different distributions of local and global models. Hence, we decided to use SVDD

[1] The cross-validation parameter 30 was empirically determined to get better results for the SVDD classifier.

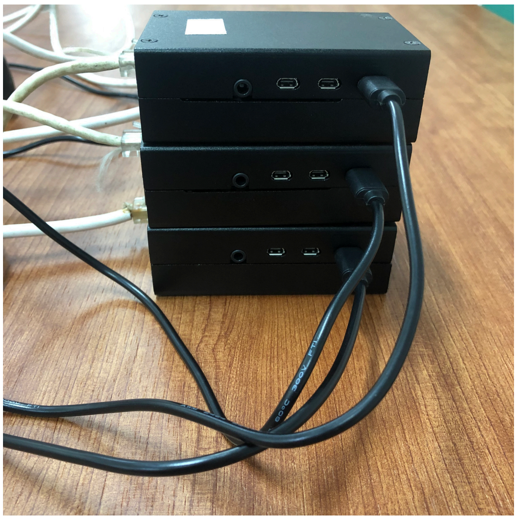


Figure 8: Testbed with 3 edged devices.

along with density kernel estimation for adaptive anomaly detection in federated setting, which allows us to avoid setting a static threshold. Although both the AE and the VAE have similar reconstruction losses, VAEs are considered more generalizable than AEs (San Martin et al., 2019). Therefore, we chose to use the error vectors computed using the VAE's predictions for kernel density estimation and SVDD training.

5.2. classification Performance

To measure the classification performance, we used the following metrics widely used in the machine learning literature (Hu et al., 1997): overall classification accuracy, precision, recall, and F1-score. To measure the classification performance, we adopt the one-vs-rest (OvR) classification method. The definitions of these metrics are given below.

1. **Accuracy** is the number of correctly predicted samples divided by the total number of samples, mathematically defined as follows:

$$\mathrm{Accuracy} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}, \tag{25}$$

   where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

2. **Precision** quantifies the positive predicted samples that are actually positive, mathematically defined as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{26}$$

3. **Recall** quantifies the correctly predicted positive samples out of all the positive samples, mathematically defined as follow:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{27}$$

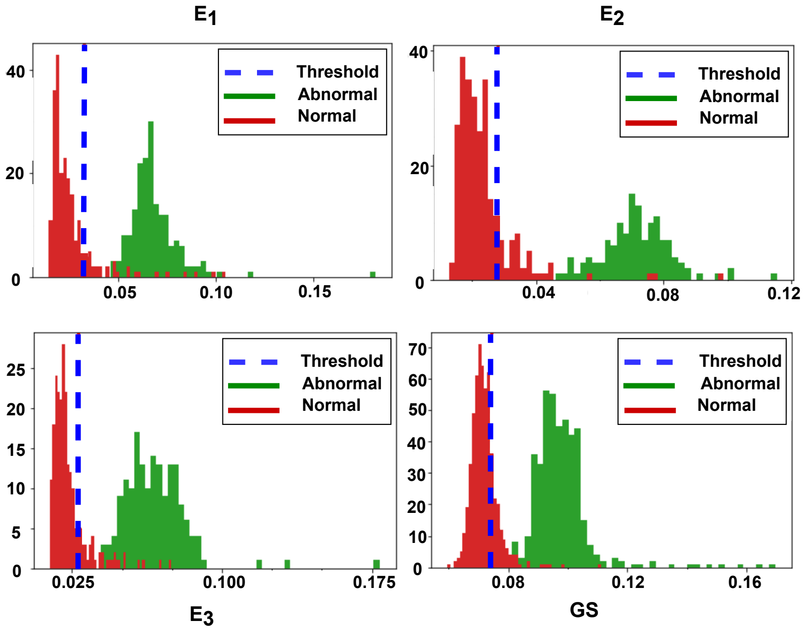


Figure 9: Reconstruction losses of the proposed AE in different models.

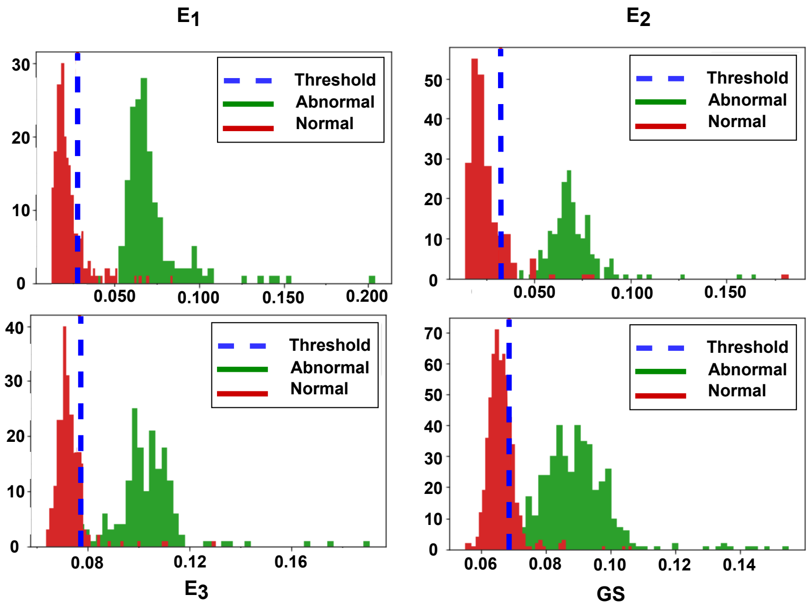


Figure 10: Reconstruction losses of the proposed VAE in different models.

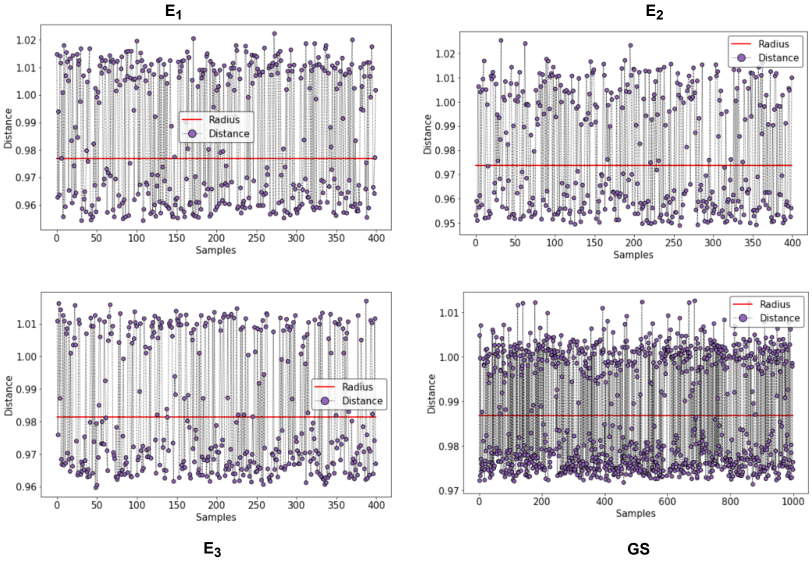


Figure 11: Hyperspheres obtained using AnoFed for IID data.

4. **F1-score** combines recall and precision by taking their harmonic mean, mathematically defined as follows:

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{28}$$

Tables 2 and 3 present the anomaly detection performance of AnoFed using the proposed adaptive anomaly detection method in IID and non-IID data distribution among the clients, respectively, where the number of edge devices is also changing per data distribution setting. It can be seen that the proposed method not only achieved state-of-art performance for local models, but also for the global model. As mentioned before, our method does not require prior knowledge about the distribution of the underlying data, as it can automatically adapt to the changing distribution when new data come in.

Figures 11 and 12 show that AnoFed is able to separate the normal and anomaly ECG test samples efficiently both locally and globally for both IID and non-IID settings, respectively. Any test samples with a distance more than the radius (red line) of the normal class are classified as an anomaly.

5.3. Explainability with XAI module

In order to trace back the segments of the input ECG sample to build trust among the user we proposed an XAI module as discussed previously. In this subsection, we show with a sample test example how efficiently the proposed XAI module can trace back the segments of the ECG signal responsible for maximum reconstruction loss. Figure 13 shows an example output of the proposed XAI module. We used a window size of 10 timestamps. Hence, each input sample is divided into 14 sub-segments. It can be seen that segments 14, 6, and 10 of samples 1, 2, and 3 are highlighted in red showing that

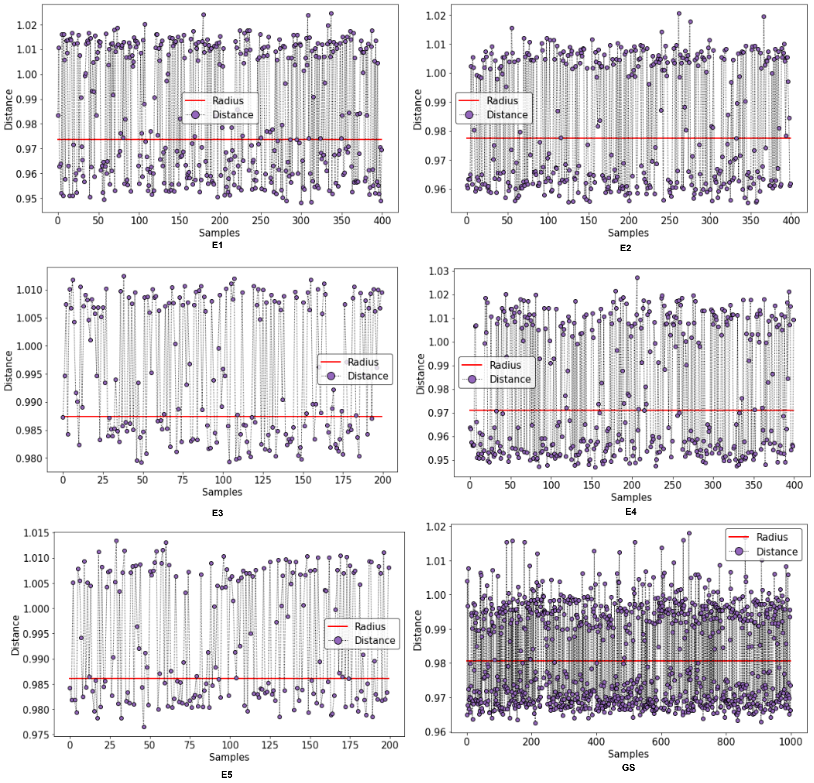


Figure 12: Hyperspheres obtained using AnoFed for non-IID data with increasing number of clients.

Table 2: The classification performance of the proposed framework (the adaptive approach).

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	100%	96.0%	98.0%	97.6%
Anomaly	95.0%	99.0%	97.0%	

(a) Edge 1

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	100%	97.0%	99.0%	98.1%
Anomaly	96.0%	100%	98.0%	

(b) Edge 2

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	99.0%	98.0%	98.0%	98.0%
Anomaly	96.0%	99.0%	97.0%	

(c) Edge 3

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	100%	97.0%	98.0%	98.8%
Anomaly	97.0%	99.0%	98.0%	

(d) Global Server (GS)

Table 3: The classification performance of the proposed framework (the adaptive approach) with Non-IID data.

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	98%	97.0%	98.0%	97.0%
Anomaly	97.0%	99.0%	97.0%	

(a) Edge 1

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	100%	96.0%	98.0%	98.0%
Anomaly	96.0%	99%	98.0%	

(b) Edge 2

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	94.0%	97.0%	95.0%	94.0%
Anomaly	95.0%	91.0%	93.0%	

(c) Edge 3

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	99.0%	96.0%	97.0%	97.0%
Anomaly	95.0%	99%	97.0%	

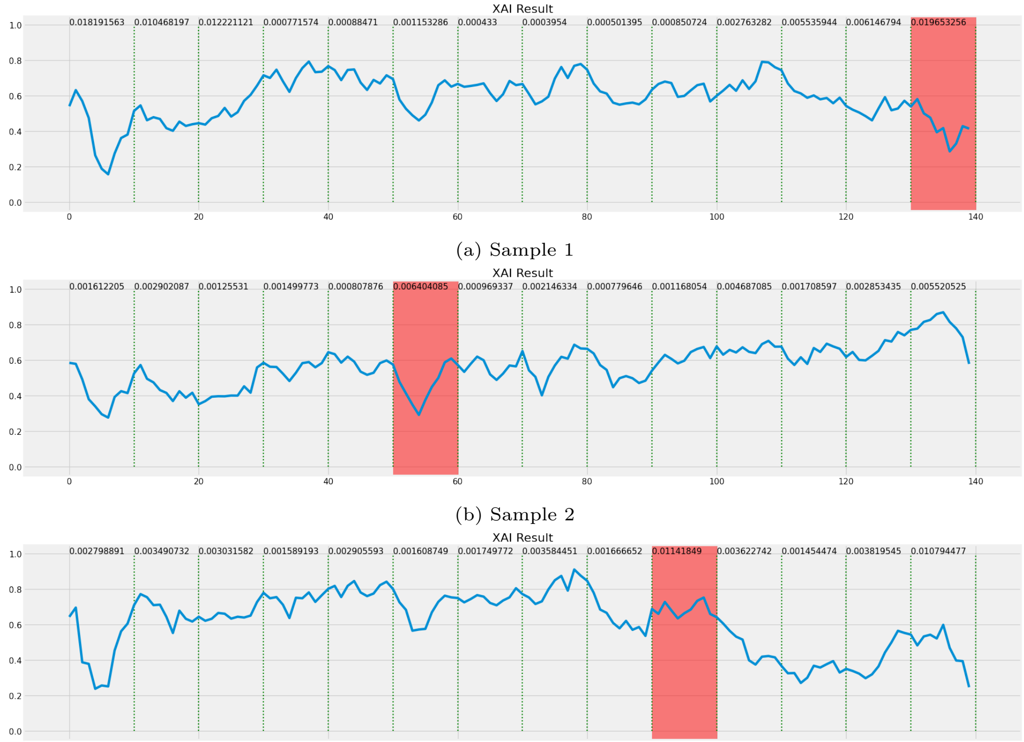
(d) Edge 4

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	91.0%	99.0%	94.0%	93.0%
Anomaly	98.0%	87.0%	92.0%	

(e) Edge 5

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Normal	100.0%	96.0%	98.0%	98.0%
Anomaly	96.0%	100%	98.0%	

(f) Global Server (GS)



(a) Sample 1

(b) Sample 2

(c) Sample 3

Figure 13: An example sample showing how the XAI module helps achieve explainability.

these segments have the maximum reconstruction loss (the reconstruction loss for each segment is given on the top of each column). In other words, the segments highlighted in red contribute more to the reconstruction loss as compared to others, thereafter for the anomaly.

5.4. Comparison

In this subsection, we compare AnoFed with some state-of-the-art methods (Wang et al., 2016; Wess et al., 2017; Shin et al., 2020; Carrera et al., 2019; Lenning et al., 2018; Chauhan and Vig, 2015; Zhou and Kan, 2021), in terms of desired features provided and the overall detection accuracy. Table 4 presents the results of the comparison. It can be seen that AnoFed provides desirable properties such as enhanced privacy protection (because FL employed in the framework allows peers in the network to train a global model without sharing local healthcare data, but only sharing trained parameters that reveal less information compared with the case when the raw local data is shared with the global server directly), explainability, and adaptive anomaly detection, while others lack some of the desired properties. Table 5 presents comparison of selected state-of-the-art methods (Wang et al., 2016; Wess et al., 2017; Shin et al., 2020; Carrera et al., 2019; Lenning et al., 2018; Chauhan and Vig, 2015) with AnoFed in terms of the overall detection accuracy. It can be seen that AnoFed achieved a performance either comparable to or better than the compared methods, with an overall accuracy of 98.8%. Moreover, we achieved state-of-the-art accuracy by training AnoFed with just three local rounds and one global round, which makes it computationally efficient for resource-constrained devices. It should be noted that AnoFed was evaluated in a federated setting, while all others are centralized so less privacy-friendly as mentioned previously.

Table 4: Comparison with selected state-of-the-art methods (key features).

Scheme	*Explainability*	*Adaptivity*	*Enhanced Privacy Protection*
(Wang et al., 2016)	✗	✗	✗
(Wess et al., 2017)	✗	✗	✗
(Shin et al., 2020)	✗	✗	✗
(Carrera et al., 2019)	✗	✓	✗
Lenning et al. (2018)	✗	✗	✗
(Chauhan and Vig, 2015)	✗	✗	✗
(Zhou and Kan, 2021)	✗	✓	✗
Proposed	✓	✓	✓

Table 5: Comparison with selected state-of-the-art methods (detection accuracy).

Scheme	*Centralized or Federated*	*Accuracy (%)*
(Wang et al., 2016)	centralized	–
(Wess et al., 2017)	centralized	99.8
(Shin et al., 2020)	centralized	95.0
(Lenning et al., 2018)	centralized	96.0
(Chauhan and Vig, 2015)	centralized	99.3
(Zhou and Kan, 2021)	centralized	95.0
(Zhang et al., 2020)	federated	70.0 (F1-score)
(Lin et al., 2022)	federated	96.94 (Multi-class)
Proposed	federated	98.8

5.5. Time Complexity of AnoFed

In this subsection, we present the time complexity for the entire pipeline of AnoFed. Since the number of transformer layers and the multi-head attention is constant, i.e., they do not depend on the input size, the dot product in multi-head attention for a given input of size n features takes $O(n)$ time. Moreover, each output is the sum product of k features of the input, with a fixed number of weights, which are not dependent on n. Similarly, computation on the activation function takes a linear amount of time. Furthermore, all the edge or client devices perform in parallel, therefore, the overall run-time is linear, i.e., bounded by $O(n)$.

In our experiments, AnoFed took 74.3 seconds to complete one global round of training. This time is further divided as follows: training the AE/VAE for 3 rounds took around 35.4 seconds and training the SVDD took around 38.9 seconds in a federated setting. Additionally, the framework took 0.93 seconds to test a sample. It should be noted that the actual run-time performance of the entire pipeline can vary depending on the implementation details such as the hardware used, and the number of samples each local model has.

6. Limitations and Future Work

Although AnoFed has many desirable features, some other challenges still need addressing. For example, despite the fact that FL has been successfully applied in different applications, such as healthcare, it is still vulnerable to several attacks, such as privacy inference attacks where a malicious party of the federated learning (the central server or an edge client) tries to infer if a given sample has been used to train a given model (Mothukuri et al., 2021), thereafter violating the key privacy assumption that local data are visible to each local client only, and Byzantine attacks where a group of malicious edge clients colludes to destroy the integrity of the constructed global model (Lyu et al., 2020). In addition, other cyber attacks threatening machine learning models in general such as model poisoning attacks also need addressing before any federated learning methods can be deployed in real-world applications. As part of our future work, we will investigate if and how the proposed method and other similar federated learning methods proposed for healthcare applications may be vulnerable to such more advanced threats, and how they can be enhanced to be more robust.

The results reported in the paper are based on two public datasets from a small population of data subjects $(15 + 18 = 33)$. Validating the performance of our proposed method with a larger dataset covering more patients and people with normal conditions will be useful to consolidate the evidence presented in this paper. Similarly, our experiments were based on a small number of edge devices (3) and simulated local data, so it will be important to re-validate the overall performance of the proposed method in a more real-world setting. Doing both will require close collaboration with healthcare organizations, which will not be trivial to achieve and will be our long-term future work.

Although our proposed method employs XAI to enhance the explainability of the proposed method, we still need to find out if the level of explainability is useful and sufficient for health professionals. In order to conduct such studies, we will need to conduct required empirical studies with recruited health professionals and real-world patient data. This will be another part of our future work. Based on the results, we will investigate how the explainability can be further enhanced, which may require major changes to the proposed method.

Another area for future research is the possibility of combining the anomaly detection method with other machine learning models to construct more complicated e-health systems, e.g., monitoring patients' conditions in the home care context. For anomaly detection, additional input may be available so the machine learning model can be further extended to benefit from such additional information.

7. Conclusions

Anomaly detection is one of the important tasks to address when it comes to digital healthcare with machine learning. Deep learning-based models can achieve state-of-the-art results, but when being applied in a centralized setting, they suffer from data privacy, data availability, trust, and unknown data distribution issues when used for sensitive applications like anomaly detection in healthcare. In this paper, to address such challenges, we proposed AnoFed, a federated learning-based anomaly detection framework, to facilitate collaborative learning with distributed edge servers working with a global server. In order to facilitate federated learning with resource-constraint edge devices, we proposed a lightweight VAE and an AE based on transformers, which are used to minimize the reconstruction loss within three training epochs of each global round. To enhance the performance of the federated learning with a static threshold, we proposed to use kernel density estimation-based SVDD, which can provide adaptive anomaly detection without setting a hard threshold. AnoFed can address issues such as estimating the underlying data distribution automatically with each global round of federated learning for efficient and accurate anomaly detection. Additionally, we proposed an XAI module to provide some level of explainability to the results of AnoFed, by tracing back the major segments of the input that are responsible for a detected anomaly. Lastly, we tested the proposed framework by combining two benchmark datasets from PhysioNet's repository and showed that AnoFed achieved up to 98.8% test accuracy with 10-fold cross-validation with changing distributions of data. We also compared AnoFed with a number of selected state-of-the-art methods, showing comparable results on the performance, but with new desired features such as better privacy protection, adaptive anomaly detection, and enhanced explainability.

Acknowledgments

This study was supported by the grant (REF LABEX/EQUIPEX), a French State fund managed by the National Research Agency under the frame program "Investissements d'Avenir" I-SITE ULNE / ANR-16-IDEX-0004 ULNE.

References

Adler, A., Elad, M., Hel-Or, Y., Rivlin, E., 2015. Sparse coding with anomaly detection. Journal of Signal Processing Systems 79, 179–188. doi:10.3390/s20051461.

Ahmed, M., Mahmood, A.N., Hu, J., 2016. A survey of network anomaly detection techniques. Journal of Network and Computer Applications 60, 19–31. doi:10.1016/j.jnca.2015.11.016.

Al-Janabi, S., Al-Shourbaji, I., Shojafar, M., Shamshirband, S., 2017. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. Egyptian Informatics Journal 18, 113–122. doi:10.1016/j.eij.2016.11.001.

Amin, M.B., Banos, O., Khan, W.A., Muhammad Bilal, H.S., Gong, J., Bui, D.M., Cho, S.H., Hussain, S., Ali, T., Akhtar, U., Chung, T.C., Lee, S., 2016. On curating multimodal sensory data for health and wellness platforms. Sensors 16, 980:1–980:27. doi:10.3390/s16070980.

Asan, O., Bayrak, A.E., Choudhury, A., 2020. Artificial intelligence and human trust in healthcare: Focus on clinicians. Journal of Medical Internet Research 22, e15154:1–e15154:7. doi:10.2196/15154.

Baldi, P., 2012. Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of the 2012 ICML Workshop on Unsupervised and Transfer Learning, ML Research Press. pp. 37–49. URL: https://proceedings.mlr.press/v27/baldi12a.html.

Banaee, H., Ahmed, M.U., Loutfi, A., 2013. Data mining for wearable sensors in health monitoring systems: A review of recent trends and challenges. Sensors 13, 17472–17500. doi:10.3390/s131217472.

Baur, C., Wiestler, B., Albarqouni, S., Navab, N., 2018. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images, in: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries – 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I, Springer. pp. 161–169. doi:10.1007/978-3-030-11723-8_16.

Carrera, D., Rossi, B., Fragneto, P., Boracchi, G., 2019. Online anomaly detection for long-term ECG monitoring using wearable devices. Pattern Recognition 88, 482–492. doi:10.1016/j.patcog.2018.11.019.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Computing Surveys 41, 15:1–15:58. doi:10.1145/1541880.1541882.

Chandola, V., Banerjee, A., Kumar, V., 2012. Anomaly detection for discrete sequences: A survey. IEEE Transactions on Knowledge and Data Engineering 24, 823–839. doi:10.1109/TKDE.2010.235.

Chang, C.C., Lin, C.J., 2001. Training v-support vector classifiers: Theory and algorithms. Neural Computation 13, 2119–2147. doi:10.1162/089976601750399335.

Chauhan, S., Vig, L., 2015. Anomaly detection in ECG time signals via deep long short-term memory networks, in: Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, IEEE. doi:10.1109/DSAA.2015.7344872.

Chen, Y., Hao, Y., Rakthanmanon, T., Zakaria, J., Hu, B., Keogh, E., 2015. A general framework for never-ending learning from time series streams. Data Mining and Knowledge Discovery 29, 1622–1664. doi:10.1007/s10618-014-0388-4.

Cozzolino, D., Verdoliva, L., 2016. Single-image splicing localization through autoencoder-based anomaly detection, in: Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security, IEEE. doi:10.1109/WIFS.2016.7823921.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805, doi:10.48550/arXiv.1810.04805.

Durga, S., Nag, R., Daniel, E., 2019. Survey on machine learning and deep learning algorithms used in Internet of Things (IoT) healthcare, in: Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication, IEEE. pp. 1018–1022. doi:10.1109/ICCMC.2019.8819806.

Fernando, T., Gammulle, H., Denman, S., Sridharan, S., Fookes, C., 2021. Deep learning for medical anomaly detection – a survey. ACM Computing Surveys 54, 141:1–141:37. doi:10.1145/3464423.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101, e215–e220. doi:10.1161/01.CIR.101.23.e215.

Haddad, M., Hermassi, S., Aganovic, Z., Dalansi, F., Kharbach, M., Mohamed, A.O., Bibi, K.W., 2020. Ecological validation and reliability of Hexoskin wearable body metrics tool in measuring pre-exercise and peak heart rate during shuttle run test in professional handball players. Frontiers in Physiology , 957:1–957:8doi:10.3389/fphys.2020.00957.

Hu, Y.H., Palreddy, S., Tompkins, W.J., 1997. A patient-adaptable ECG beat classifier using a mixture of experts approach. IEEE Transactions on Biomedical Engineering 44, 891–900. doi:10.1109/10.623058.

Huang, G., Chen, H., Zhou, Z., Yin, F., Guo, K., 2011. Two-class support vector data description. Pattern Recognition 44, 320–329. doi:10.1016/j.patcog.2010.08.025.

Jin, H., Luo, Y., Li, P., Mathew, J., 2019. A review of secure and privacy-preserving medical data sharing. IEEE Access 7, 61656–61669. doi:10.1109/ACCESS.2019.2916503.

Kaleschke, G., Hoffmann, B., Drewitz, I., Steinbeck, G., Naebauer, M., Goette, A., Breithardt, G., Kirchhof, P., 2009. Prospective, multicentre validation of a simple, patient-operated electrocardiographic system for the detection of arrhythmias and electrocardiographic changes. Europace 11, 1362–1368. doi:10.1093/europace/eup262.

Krichen, M., 2021. Anomalies detection through smartphone sensors: A review. IEEE Sensors Journal 21, 7207–7217. doi:10.1109/JSEN.2021.3051931.

Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I., Kim, K.J., 2019. A survey of deep learning-based network anomaly detection. Cluster Computing 22, 949–961. doi:10.1007/s10586-017-1117-8.

Lee, K., Kim, D.W., Lee, D., Lee, K.H., 2005. Improving support vector data description using local density degree. Pattern Recognition 38, 1768–1771. doi:10.1016/j.patcog.2005.03.020.

Lenning, M., Fortunato, J., Le, T., Clark, I., Sherpa, A., Yi, S., Hofsteen, P., Thamilarasu, G., Yang, J., Xu, X., Han, H.D., Hsiai, T.K., Cao, H., 2018. Real-time monitoring and analysis of zebrafish electrocardiogram with anomaly detection. Sensors 18, 61:1–61:16. doi:10.3390/s18010061.

Leys, C., Ley, C., Klein, O., Bernard, P., Licata, L., 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology 49, 764–766. doi:10.1016/j.jesp.2013.03.013.

Li, B., Cui, W., Wang, W., Zhang, L., Chen, Z., Wu, M., 2021. Two-stream convolution augmented transformer for human activity recognition. Proceedings of the AAAI Conference on Artificial Intelligence 35, 286–293.

Li, H., Boulanger, P., 2020. A survey of heart anomaly detection using ambulatory electrocardiogram (ECG). Sensors 20, 1461:1–1461:33. doi:10.3390/s20051461.

Libby, P., Bonow, R.O., Mann, D.L., Tomaselli, G.F., Bhatt, D., Solomon, S.D., Braunwald, E., 2021. Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine. 12th ed., Elsevier.

Lin, D., Guo, Y., Sun, H., Chen, Y., 2022. Fedcluster: A federated learning framework for cross-device private ecg classification, in: Proceedings of the IEEE Conference on Computer Communications Workshops, pp. 1–6. doi:10.1109/INFOCOMWKSHPS54753.2022.9797945.

Lyu, L., Yu, H., Yang, Q., 2020. Threats to federated learning: A survey. arXiv:2003.02133 [cs.CR]. URL: https://arxiv.org/abs/2003.02133, doi:10.48550/arXiv.2003.02133.

Maussang, F., Chanussot, J., Hétet, A., Amate, M., 2007. Mean–standard deviation representation of sonar images for echo detection: Application to SAS images. IEEE Journal of Oceanic Engineering 32, 956–970. doi:10.1109/JOE.2007.907936.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, ML Research Press. pp. 1273–1282. URL: http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf.

Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G., 2021. A survey on security and privacy of federated learning. Future Generation Computer Systems 115, 619–640. doi:10.1016/j.future.2020.10.007.

Oh, D.Y., Yun, I.D., 2018. Residual error based anomaly detection using auto-encoder in SMD machine sound. Sensors 18, 1308. doi:10.3390/s18051308.

Oussidi, A., Elhassouny, A., 2018. Deep generative models: Survey, in: Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision, IEEE. doi:10.1109/ISACV.2018.8354080.

San Martin, G., López Droguett, E., Meruane, V., das Chagas Moura, M., 2019. Deep variational autoencoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis. Structural Health Monitoring 18, 1092–1128. doi:10.1177/1475921718788299.

Seyfioğlu, M.S., Özbayoğlu, A.M., Gürbüz, S.Z., 2018. Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. IEEE Transactions on Aerospace and Electronic Systems 54, 1709–1723. doi:10.1109/TAES.2018.2799758.

Shin, D.H., Park, R.C., Chung, K., 2020. Decision boundary-based anomaly detection model using improved AnoGAN from ECG data. IEEE Access 8, 108664–108674. doi:10.1109/ACCESS.2020.3000638.

Smys, S., Chen, J.I.Z., Shakya, S., 2020. Survey on neural network architectures with deep learning. Journal of Soft Computing Paradigm 2, 186–194. doi:10.36548/jscp.2020.3.00.

Tax, D.M.J., Duin, R.P.W., 2004. Support vector data description. Machine Learning 54, 45–66. doi:10.1023/B:MACH.0000008084.60811.49.

Vanerio, J., Casas, P., 2017. Ensemble-learning approaches for network security and anomaly detection, in: Proceedings of the 2017 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, ACM. doi:10.1145/3098593.3098594.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer. doi:10.1007/978-1-4757-3264-1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Advances in neural information processing systems 30 (NIPS 2017), pp. 5998–6008.

Wang, K., Zhao, Y., Xiong, Q., Fan, M., Sun, G., Ma, L., Liu, T., 2016. Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals. Scientific Programming 2016. doi:10.1155/2016/5642856.

Wess, M., Sai Manoj, P.D., Jantsch, A., 2017. Neural network based ECG anomaly detection on FPGA and trade-off analysis, in: Proceedings of the 2017 IEEE International Symposium on Circuits and Systems, IEEE. doi:10.1109/ISCAS.2017.8050805.

Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J., 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges, in: Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II, Springer. pp. 563–574. doi:10.1007/978-3-030-32236-6_51.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology 10, 12:1–12:19. doi:10.1145/3298981.

Yildirim, O., San Tan, R., Acharya, U.R., 2018. An efficient compression of ECG signals using deep convolutional autoencoders. Cognitive Systems Research 52, 198–211. doi:10.1016/j.cogsys.2018.07.004.

Yu, Y., 2012. A survey of anomaly intrusion detection techniques. Journal of Computing Sciences in Colleges 28, 9–17.

Zhai, J., Zhang, S., Chen, J., He, Q., 2018. Autoencoder and its various variants, in: Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics, IEEE. pp. 415–419. doi:10.1109/SMC.2018.00080.

Zhang, M., Wang, Y., Luo, T., 2020. Federated learning for arrhythmia detection of non-iid ecg, in: Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications, IEEE. pp. 1176–1180.

Zhou, H., Kan, C., 2021. Tensor-based ECG anomaly detection toward cardiac monitoring in the Internet of Health Things. Sensors 21, 4173:1–4173:17. doi:10.3390/s21124173.