Designing ECG monitoring healthcare system with federated transfer learning and explainable AI

Ali Raza, Kim Phuc Tran, Ludovic Koehl, Shujun Li

Designing ECG Monitoring Healthcare System with Federated Transfer Learning and Explainable AI

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

ARTICLE INFO

Keywords:
electrocardiography (ECG)
deep learning
explainable AI (XAI)
privacy
security
federated learning

ABSTRACT

Deep learning plays a vital role in classifying different arrhythmias using electrocardiography (ECG) data. Nevertheless, training deep learning models normally requires a large amount of data and can lead to privacy concerns. Unfortunately, a large amount of healthcare data cannot be easily collected from a single silo. Additionally, deep learning models are like black-box, with no explainability of the predicted results, which is often required in clinical healthcare. This limits the application of deep learning in real-world health systems.

In this paper, to address the above-mentioned challenges, we design a novel end-to-end framework in a federated setting for ECG-based healthcare using explainable artificial intelligence (XAI) and deep convolutional neural networks (CNN). The federated setting is used to solve challenges such as data availability and privacy concerns. Furthermore, the proposed framework effectively classifies different arrhythmias using an autoencoder and a classifier, both based on a CNN. Additionally, we propose an XAI-based module on top of the proposed classifier for interpretability of the classification results, which helps clinical practitioners to interpret the predictions of the classifier and to make quick and reliable decisions. The proposed framework was trained and tested using the baseline Massachusetts Institute of Technology - Boston's Beth Israel Hospital (MIT-BIH) Arrhythmia database. The trained classifier outperformed existing work by achieving accuracy up to 94.5% and 98.9% for arrhythmia detection using noisy and clean data, respectively, with five-fold cross-validation. We also propose a new communication cost reduction method to reduce the communication costs and to enhance the privacy of users' data in the federated setting. While the proposed framework was tested and validated for ECG classification, it is general enough to be extended to many other healthcare applications.

1. Introduction

With the increase of internet of things (IoT) devices being used in the 21st century, massive amount of data has been generated [1]. IoT devices are capable of collecting an enormous amount of data each day [2]. This collection of data and the exponentially increasing computational resources have unlocked new dimensions in the information technology sector, especially in deep learning (DL) [3]. Although deep learning is a quite old concept [4] but owing to limited data and computational resources available in the past its use was limited. However, thanks to the internet, IoT devices and the increasing computational power, nowadays we can see deep learning revolutionizing nearly every field, including healthcare [5], economics [6], manufacturing [7], agriculture [8], and military [9].

In regards to healthcare applications, a lot of data is being generated across the globe and it has quite unique properties. Most of the data related to healthcare are multi-dimensional, this makes the use of classical machine learning (ML) models, for example, decision trees and random forests, quiet challenging and complex. However, the new generation machine learning models, especially the deep learning based ones, can solve issues related to multi-dimensional data due to its capability of self learning [10]. In the healthcare industry, deep learning has played a critical role, e.g., to help diagnose life threatening diseases [11]. Nevertheless, it has some limitations [12]. First, to train a deep learning model a large amount of training data is needed, but each silo (for example, a hospital) can have a very limited amount of data, so a single source of data can be insufficient to train a good deep learning model. A solution to this is to collect data from multiple sources and then train the model on the collected data. One major issue of this approach is about privacy concerns [13]. As medical data are highly sensitive and private data, some individual sources may not be willing to share their data with a central data collector [14].

In 2016 Google came forward with an idea called federated learning to solve the conflict between data availability and privacy concerns [15]. The basic idea behind federated learning is to collaboratively train a machine learning model without centralized training data. Federated learning enables edge devices or servers with sufficient computational power (e.g., home computers, mobile phones, wearables and other IoT devices) to collaboratively learn a shared machine learning model while keeping all the training data on local devices, decoupling the ability to do machine learning from the need to store the data centrally at a single server or in the cloud. Although deep learning with a federated setting can solve the issues mentioned earlier, there exists the

The authors can be contacted via ali.raza@ensait.fr or ar718@kent.ac.uk (Ali Raza), kim-phuc.tran@ensait.fr (Kim Phuc Tran), ludovic.koehl@ensait.fr (Ludovic Koehl), hooklee@gmail.com or S.J.Li@kent.ac.uk (Shujun Li).

*Corresponding author

ORCID(s): 0000-0001-8326-8325 (Ali Raza); 0000-0002-3404-8462 (Ludovic Koehl); 0000-0001-5628-7328 (Shujun Li)

problem of explainability in deep learning. Since the deep learning models are generally black box models, with no reasonable explanation for a given prediction. This ambiguity causes a limitation of deep learning in healthcare, because a clinical practitioner should know the reason for a prediction by a deep learning model [16]. To address the problem of explainability in deep learning models, researchers have proposed different solutions [17, 18, 19]. For instance, Selvaraju proposed a method called Gradient-weighted Class Activation Mapping (Grad-CAM) [20] to visualize input regions that are important for predictions. From such values, we can have an idea about where exactly the machine learning model is focusing while making a prediction and thus the reason. Explainability is important in healthcare, because to convince a clinical healthcare practitioner and a patient we need to give them the reason behind a certain prediction for sample input.

In regards to the application of deep learning in healthcare, electrocardiogram (ECG) classification is a very important routine task. Many machine learning based solutions have been proposed for analyzing and classifying ECG data [19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. However, most of these works are based on a centralized machine learning architecture, thereafter they are prone to issues like privacy concerns and data availability. Moreover, since most of the real-time ECG data is noisy, they cannot perform well in real time because they are being trained on preprocessed (cleaner) data. Furthermore, they do not provide explainability/interpretability (we use explainability and interpretability interchangeably through out this article), which is one of the key requirements in deep learning based clinical healthcare. Hence, this limits their real-time application. To address all of the above-mentioned challenges, in this paper, we propose an end-to-end explainable healthcare framework in a federated setting. The proposed Framework consists of three main parts: an autoencoder, a classifier and an XAI module. Firstly, we propose a novel deep convolutional neural network (CNN) based autoencoder, which is used to denoise the raw ECG signals from the subject directly. Secondly, we propose a novel CNN-based classifier, which uses transfer learning to classify the raw time series of ECG data. Thirdly, we adopt the Grad-CAM model [20] in the framework to explain the classification results in a novel and reliable pattern. Additionally, we propose a custom communication cost reduction approach that reduces the communication cost and increases the privacy protection of the framework.

1.1. Contributions

The main contributions of this paper are as follows:

1. We propose an end-to-end framework which is the first federated transfer learning and explainable-AI based framework for healthcare. It aggregates the data from different edge devices (hospitals, users) without compromising privacy and security, provides relatively personalized model learning through knowledge transfer and provides interpretability of the results, which is one of the key requirements in applications like healthcare. In addition to interpretability, the proposed XAI module can be used to recognize new potential patterns leading to trigger heart arrhythmias.
2. We propose a novel 1-Dimensional CNN-based autoencoder in a federated setting to efficiently denoise the raw time series of ECG signal collected data from patients. The autoencoder provides a denoised version of the input, which we use for further classification and explanation of the predictions.
3. With the help of transfer learning, we use the encoder part of the proposed autoencoder to make a novel 1-Dimensional CNN-based classifier to classify given ECG data into five classes: non-ecotic beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), fusion Beats (F), and unknown beats (Q).
4. We propose a novel module, called XAI module for interpretability of predictions of the proposed classifier. The proposed XAI module is combined with the proposed classifier to interpret and explain the decision making process of the classifier. The XAI module can be used with every updated classier locally at the edge devices in the federated setting, and it does not need any pre-training.
5. We propose a new communication cost reduction method for the federated learning in the proposed framework, which not only reduces the communication costs but also increases the privacy of the classical federated learning method. Furthermore, the proposed method can be integrated into existing cost optimization algorithms to enhance their cost effectiveness and privacy protection level.
6. We used the MIT-BIH Arrhythmia Database [32] to train our proposed framework. It is important to note that to make the data more realistic, we first upsample the data to create more data samples, and then add 10-30% random noise. The proposed framework shows excellent performance by providing an overall accuracy of 94.5% using noisy data and overall accuracy of 98.9% on the clean data in the original MIT-BIH database. Moreover, we evaluated the performance of the proposed framework using four standard metrics: classification accuracy, precision, recall and F1-score.
7. The proposed framework additionally boasts desirable features: interpretability of the results by using the proposed XAI module, and efficient classification of the ECG. Additionally, it provides an enhanced level of privacy protection to users because of the federated setting and the proposed communication cost reduction method.

The rest of the paper is organized as follows. Section 2 presents related work and background. Section 3 discusses detailed description of the proposed framework. Sections 4 and 5 present the experimental setup and performance evaluation, respectively. Section 6 concludes the article.

2. Related Work and Background

Intelligent systems have helped us achieve efficient solutions. Various types of intelligent systems, such as, intelligent systems for modeling uncertainties by robust optimization [33], intelligent parking lots for electrical vehicles [34], and other wide ranges of applications [35] have been a focus of academia and industries. In recent decades, intelligent systems based on ML have been studied in a wide range of applications. For example, its applications have been studied for cyber security [36], economics and agriculture [37], and in healthcare [38]. The use of machine learning in healthcare has been widely studied ranging from detection and diagnosis of different diseases, such as, melanoma [39, 40] and cancer [41, 42]. Owing to the importance of machine learning in healthcare-related applications, in this section we review the literature of machine learning in healthcare, with a special focus on machine learning for ECG analysis.

2.1. Machine Learning in Healthcare

Certain activities in our body are governed by signals of some cognitive diseases [43]. For example, a changing gait may result from a stroke. A number of researchers proposed to monitor users' activities using wearable sensors, with the help of which different human body activities can be recognized [44, 45, 46]. Based on monitoring of such activities, early prognosis of health issues can be identified. In this regards, there has been significant development in the utilization of ML and DL technologies in healthcare. While such technologies will probably never completely replace clinical practitioners, they can transform the healthcare sector, benefiting both patients and providers [11, 47, 48, 49].

In regards to ECG analysis in healthcare, ML and DL play a vital role. Researchers have proposed many methods for ECG classification into arrhythmia types [50, 51, 52, 28, 26]. Rubin et al. [53] applied deep learning to the task of automated cardiac auscultation, i.e., recognizing abnormalities in heart sounds. They described an automated heart sound classification algorithm that combines the use of time-frequency heat map representations with a deep CNN. Their CNN architecture is trained using a modified loss function that directly optimizes the trade-off between sensitivity and specificity. Gjoreski et al. [54] presented a method for chronic heart failure (CHF) detection based on heart sounds. The method combines classic ML and end-to-end DL models. The classic ML model learns from expert features, and the DL model learns from a spectro-temporal representation of the signal. Moreover, in order to enable intelligent classification of arrhythmias with high accuracy, Huang et al. [55] presented an intelligent ECG classifier using the fast compression residual convolutional neural networks (FCResNet).

Although the aforementioned work seems promising, they may find limited applicability in real world because they use centralized data collection techniques. As discussed earlier that it may cause privacy concerns among users and data owners. Thereafter, traditional centralized healthcare applications find limited applicability due to privacy concerns [56, 57, 58]. To address the privacy issues in machine learning, researchers have been working on Federated learning (FL) and Transfer learning (TF). Federated learning (FL) was introduced by Google [15]. The key idea is to train ML models with privacy by design at the architectural level. FL trains a machine learning model in a distributed architecture, where the edged devices train their own ML model on their local data and a central global server aggregates all of the locally trained models and distribute the aggregated model back to all nodes on the network (more details about FL can be found in Section 2.3). Due to its privacy preserving and efficient communication constraints, FL finds a number of applications in healthcare [59]. Xu et al. [60] summarized the general solutions to the statistical challenges, system challenges, and privacy, and point out the implications and potentials of FL's application in healthcare. They show that training the model in the federated learning framework leads to comparable performance to the traditional centralized learning setting. Transfer learning (TF) aims at transferring knowledge from an existing trained model to a new model. The key idea is to reduce the distribution divergence between different models. To this end, there are mainly two general approaches: instance reweighting [61] and feature matching [62]. Recently, deep transfer learning methods have made considerable success in many application fields. Chen et al. [24] proposed FedHealth, the first federated transfer learning framework for wearable healthcare to tackle privacy and security challenges. FedHealth performs data aggregation through federated learning, and then builds relatively personalized models by transfer learning. FedHealth makes it possible to do deep transfer learning in the federated learning framework without accessing the raw user data. However, there are certain limitations to it. Firstly, it does not provide the explainability of the predictions, which is often required in sensitive domains like healthcare. Secondly, it does not accommodate any mechanism to denoise the raw signals, which often contain random noise and dealing with the random noise is quite challenging.

In other words, regarding the application of ML and DL healthcare, a lot of promising work has been done as discussed above. However, some of those works are vulnerable to privacy issues. Research work like FedHealth tries to address the issues of privacy concerns using FL and TL architecture. Nevertheless, works like FedHealth have the limitation of explainability, as discussed earlier. Thus there is a need for research work to address such challenges.

2.2. Autoencoder

Autoencoder [63] is an unsupervised neural network that learns the best encoding-decoding scheme from data. In general, it consists of an input layer, an output layer, an encoder neural network, a decoder neural network, and a latent space. When the data is fed to the network, the encoder compresses data into a latent space, whereas the decoder decompresses the encoded representation into the output layer. The encoded-decoded output is then compared with the initial data and the error is backpropagated through the architecture to update the weights of the network [64]. Given the in-

put $x \in R^m$, the encoder compresses x to obtain an encoded representation $z = e(x) \in R^n$. The decoder reconstructs this representation to give the output $x' = d(z) \in R^m$. The autoencoder is trained by minimizing the reconstruction error L, defined by the following equation:

$$L = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i')^2, \tag{1}$$

where Y_i is the true label, Y_i' is the predicted label, and n is the total number of samples. An ideal autoencoder simply copies the input to the output, whereas keeping the latent space to have a smaller dimension than the input. The autoencoder learns the most salient features of the training data, i.e., it reduces the data dimensions while keeping the important information of the data.

Since being proposed, many researchers have proposed many optimized approaches of autoencoder, such as sparse autoencoder, denoising autoencoder, contractive autoencoder, and convolutional autoencoder [65]. We can achieve two main tasks from autoencoders: denoising and dimensionality reduction. In this study, we build a denoising autoencoder, which is an extension of simple autoencoders. It is worth noting that denoising autoencoders were not originally meant to automatically denoise an input. Instead, the denoising autoencoder procedure was invented to help:

1. the hidden layers of the autoencoder learn more robust filters,
2. reduce the risk of overfitting in the autoencoder, and
3. prevent the autoencoder from learning a simple identity function.

In denoising autoencoders noise is stochastically (i.e., randomly) added to the input data, and then the autoencoder is trained to recover the original, non-perturbed signal.

2.3. Federated Learning

Federated machine (FL) learning was first proposed by Google [15], an overview of FL is shown in Figure 1. In FL settings machine learning models are trained based on distributed edge devices all over the world. The key idea is to protect user data during the process. FL has the ability to resolve the data islanding problems by privacy-preserving model training in the network.

It works like this: an edge (client) device downloads the current model, improves it by learning from data on its local data, and then summarizes the changes as a small focused update. Only this update to the model is sent to the cloud, using encrypted communication, where it is aggregated with other user updates to improve the global shared model. All the training data remains on local devices, and no individual updates are stored in the cloud. Federated Learning allows for smarter models, lower latency, and less power consumption, while ensuring privacy. This approach has another benefit: in addition to providing an update to the global shared model, the improved model on the local edge device can also be used immediately, powering experiences personalized by the use of IoT devices.

Figure 1: Architecture of Federated Learning

2.4. Transfer Learning

Transfer learning aims at transferring knowledge from existing domains to a new domain. The key idea is to reduce the distribution divergence between different domains. Here are mainly two types of transfer learning: instance reweighting [61] and feature matching [62]. Recently, deep transfer learning methods have made considerable success in many application fields.

2.5. Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) [16] lets humans understand and articulate how an AI system made a decision. XAI is a set of processes and methods that allows human users to comprehend and trust the results and output created by machine learning algorithms. XAI is used to describe an AI model, its expected impact and potential biases. It helps characterize model accuracy, fairness, transparency and outcomes in AI-powered decision making. XAI is crucial for an organization in building trust and confidence when putting AI models into production. AI explainability also helps an organization adopt a responsible approach to AI development. There are many advantages to understanding how an AI-enabled system has led to a specific output. Explainability can help developers ensure that the system is working as expected, it might be necessary to meet regulatory standards, or it might be important in allowing those affected by a decision to challenge or change that outcome. Recent research suggest that it will be of key importance in marketing [66], healthcare, manufacturing, insurance, and automobiles [67].

3. The Proposed Framework

Before describing our proposed framework in detail, let us explain the research problem first. Given data on N different edge nodes (since we are using cross-silo federated

learning, each edge node can represent a different organization, i.e., hospital) represented by $E = \{E_1, E_2, \ldots, E_N\}$ and the data of each E_i (here $i = 1, 2, \ldots, N$) is given by $\{D_1, D_2, \ldots, D_i\}$, respectively. A conventional machine learning model, denoted by ConMOD, can be trained by combining all the data $D = \{D_1, D_2, \ldots, D_i\}$. The data from different edge nodes have different distributions. However, in our problem, we want to collaborate all the data to train a federated transfer learning model, denoted by FedMOD, where any user E_i does not expose its data D_i to others. Assume that AccFed represents the accuracy of FedMOD and AccCon_i represents the accuracy of each locally trained model of E_i, then one of the objectives of our proposed method is to ensure that the accuracy of AccFed is close to or superior to each AccCon_i.

The proposed framework aims to achieve accurate and efficient personal healthcare through federated transfer learning and XAI without compromising privacy. Figure 2 gives an overview of the proposed method. The proposed method consists of three major parts, the autoencoder, the classifier and the XAI module, which are discussed below in the following three sub-sections. The final sub-section 3.4 discusses the learning process.

3.1. CNN-based Autoencoder

In order to denoise the raw input signal from ECG devices, we proposed an autoencoder. The proposed autoencoder is shown in Figure 3. It consists of an input layer, an output layer and 12 hidden layers. Among the hidden layers, there are 6 convolutional layers, 3 maxpooling layers and 3 upsampling layers. Furthermore, the CNN-autoencoder is virtually divided into two parts: Encoder and Decoder. The encoder consists of the input layer, 3 maxpooling layers and 3 convolutional layers in an alternate fashion. On other hand, the decoder consists of 3 upsampling layers, 3 convolutional layers and a convolutional output layer. In the proposed autoencoder, we use a varying learning rate to keep the training process efficient while keeping the reconstruction loss L as small as possible. Equation (2) gives the mathematical representation of the learning rate (lr) used.

$$\text{lr} = \begin{cases} 0.01, & \text{if epoch} \leq 40, \\ \text{lr} \times e^{-0.1}, & \text{otherwise.} \end{cases} \tag{2}$$

3.2. CNN-based Classifier

The proposed classifier is composed of 4 convolution layers, 3 max pooling layers, 2 fully connected layer and 1 softmax layer for classification, as shown in Figure 4. The classifier is designed for classifying an input ECG signal into one of the five classes, as shown in Table 1. We use transfer learning to transfer the encoder part of the trained autoencoder into the proposed classifier, because these convolution layers aim at removing the noise from raw input data and the next layers in the classifier aim to classify the input ECG signal. Hence, the first 3 convolutional layers do not need to be trained while training the individual local classifiers. In other words, we keep the first 3 convolutional layers static during the classifier training phase, which means that no parameters are updated during back propagation in the first 3 convolutional layers. This provides each local node E_i with the trained parameters for denoising the signal while training the classifier, which increases the performance of the classier. As for the last 2 convolution layers and the fully connected layers, since they are at a higher level, they focus on learning specific features for the classification task. Therefore, we update their parameters during the classifier training phase. The softmax serves as the classification function, and is given by the following equation:

Table 1
The five classes of ECG signals

Class description	Single-letter symbol
Non-ecotic beats (normal beat)	N
Supraventricular ectopic beats	S
Ventricular ectopic beats	V
Fusion Beats	F
Unknown Beats	Q

$$y_i = \frac{\exp^{z_c}}{\sum_{c=1}^{C} \exp^{z_c}}, \tag{3}$$

where C is the total number of classes, z_c denotes the learned probability for a specific class c, and y_i is the final classification result for a sample i. Our classifier uses categorical cross-entropy (CE) as the loss function. This gives probability over the C classes for each input sample, given by Eq. (4). Where t_c is the ground truth for each class c.

$$\text{CE} = -\sum_{c}^{C} t_c log(y_i) \tag{4}$$

3.3. XAI with Grad-CAM

Inspired by the work in [20] and [68], we decided to use Gradient-weighted Class Activation Mapping (Grad-CAM) and modified it for time series data on top of our classifier, which uses class-specific gradient information to localize important regions. We combine these localized regions with an existing time-series visualization map to create a high-resolution heatmap visualization. Using this visualization, practitioners can understand the reason of a certain prediction given by the classifier. The XAI with GRAD-CAM module is shown in Figure 5.

The creation of this heatmap visualization consists of the following steps:

1. In the first step, we compute the gradient of y^c (where y^c is the score for any class c) with respect to the feature map activations A^k for kernel k of the last convolution layer. If G_c represents the gradients for any class c, it can be represented as follow:

$$G_c = \frac{\partial y^c}{\partial A^k}. \tag{5}$$

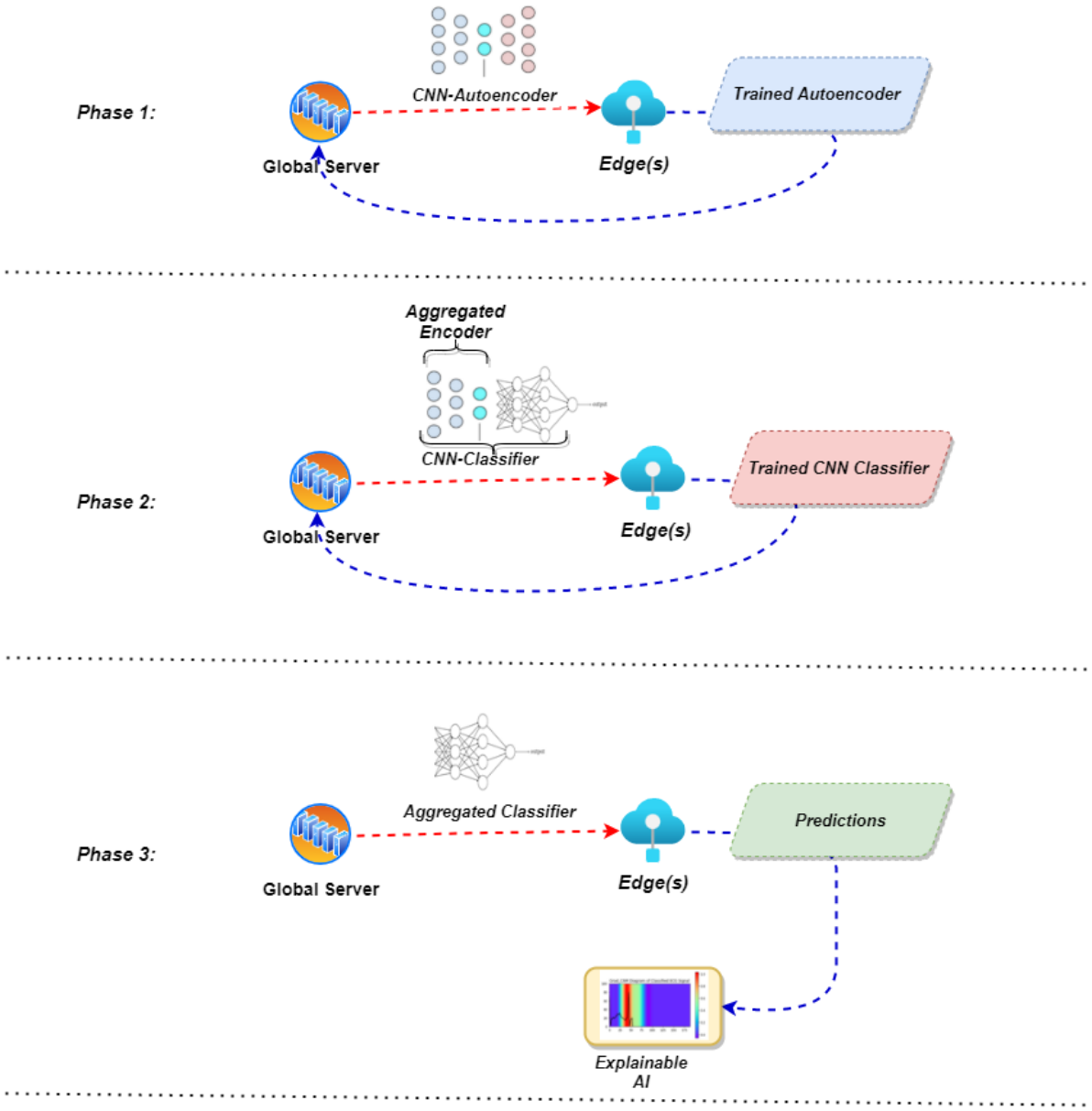

Figure 2: An overview of the proposed framework

Any particular value calculated in this step depends on the input ECG signal (sample input). The weights of the classifier are fixed at this stage. We first reshape an input sample into the batch size and feed it into the classifier, since the input determines the feature maps A_k as well as y^c.

2. The second step consist of global average pooling of the gradients G_c, both along height h and width w to obtain the neuron importance weights α_k^c also called alpha values, given by Eq. (6).

$$\alpha_k^c = \frac{1}{Z} \sum_h \sum_w \frac{\partial y^c}{\partial A^k} \tag{6}$$

These alpha values for class c and feature map k will be used later as a weight applied to the feature map A^k.

3. The third step consist of weighted linear combination of the feature map activations A^k and α_k^c is calculated using the alpha values, given by Eq. (7).

$$\text{Grad\_CAM}^c = \text{ReLU}(\sum_k \alpha_k^c A^k) \tag{7}$$

This gives us the final Grad-CAM heatmap. A rectifier linear Unit (ReLU) function is applied to emphasize only the positive values and turn all the negative values into 0.

4. The classifier's last convolutional layer's features are quite small, and it is difficult to visualize them for analysis. To address this problem, we upsample the heatmap to the size of the input sample in width. Moreover, we feed the input sample to the autoencoder and

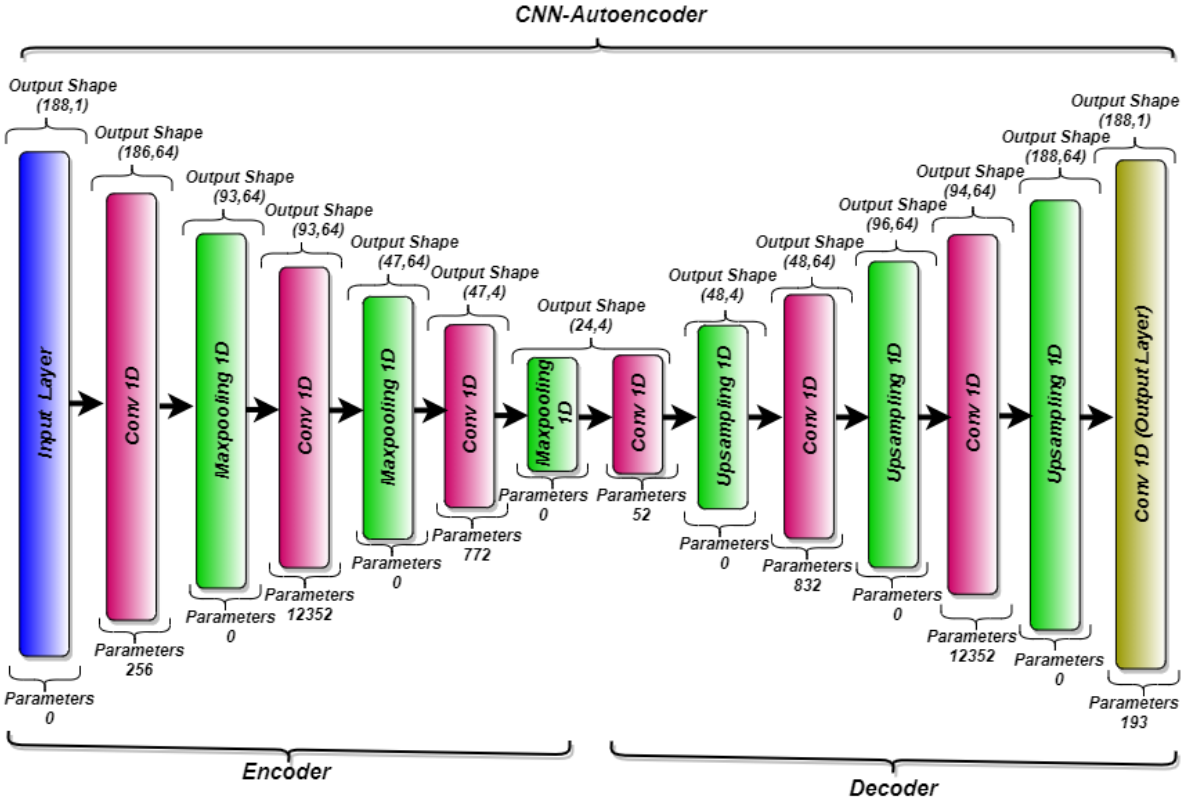

Figure 3: The architecture of the proposed denoising autoencoder

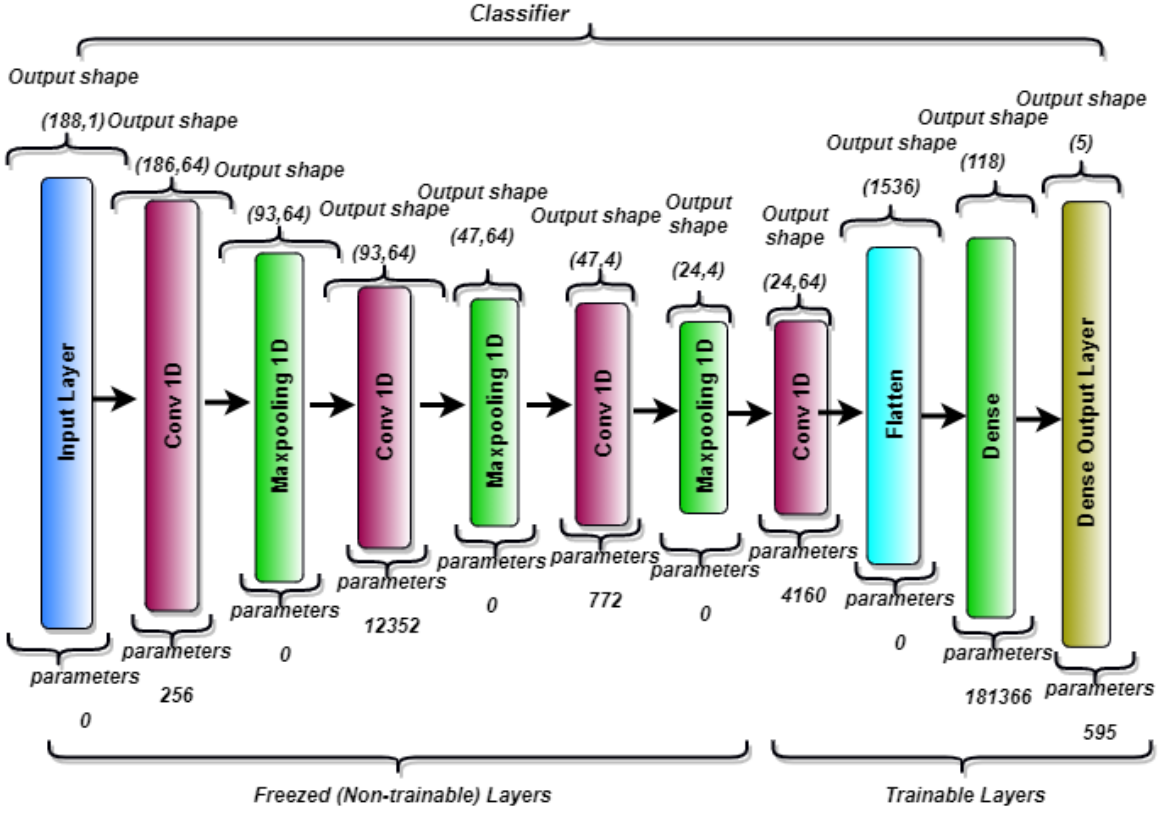

Figure 4: The proposed CNN-based classifier

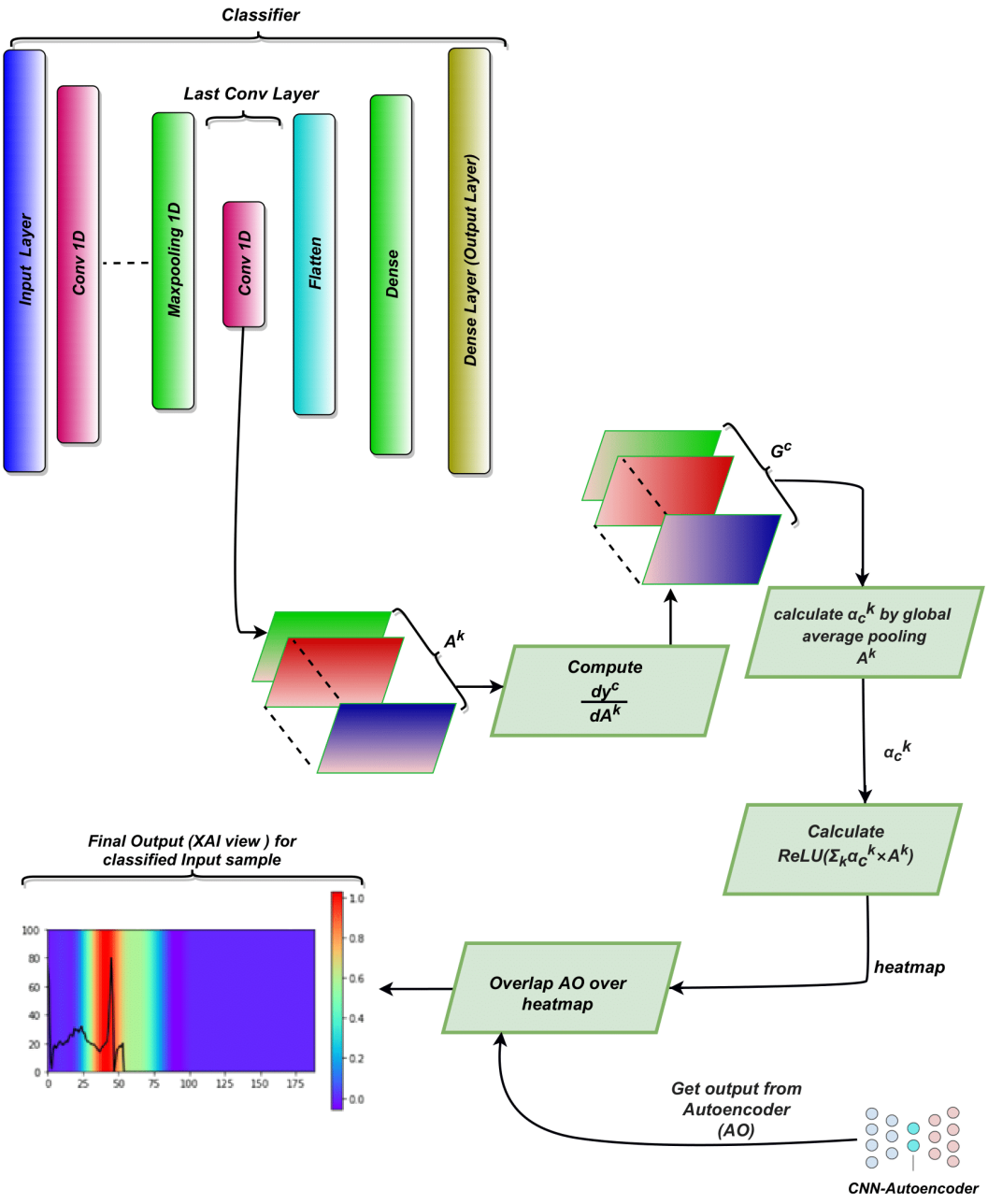

Figure 5: Overview of the proposed XAI module in our framework

receive a denoised version of the input sample and overlap it on the heatmap. In the resulting heatmap, regions overlapping between the heatmap and the ECG signal show the point of focus during prediction. This gives a detailed picture to the practitioners to understand which region of the ECG input signal the classifier is looking at while making a prediction.

3.4. Learning Process

The learning process of the proposed method has been depicted in Figure 2. For a clearer explanation, we present the learning procedure in Algorithm 1. It should be noted that the algorithm works continuously with new emerging data. Optionally, if an E_i wants to personalize the classifier C, it can be done by keeping all the convolution layers of the final updated classifier static and by training the dense layers for personalization. This is because the convolution layers aim at extracting low-level features about activity recognition and for the densely connected layers, since they are at a higher level, they focus on learning specific features for the task and the user.

The global server G_s (Aggregation Server) creates an autoencoder AE with predefined hyper-parameters. It should be noted that we use Keras auto-tuner to get the best possible hyper-parameters. Keras auto-tuner empirically tries to find the best possible hyper-parameters. After creating the AE, G_s waits for the clients' request. When clients request G_s, it

Algorithm 1: Training procedure of Proposed method

Input: Data from edge nodes $D_1, D_2, \ldots, D_n$
Output: Trained aggregated and updated model

Global Server G_s constructs the initial Global Autoencoder AE and compiles it using the predefined hyper-parameters
G_s waits for the E_i to request. If request received, send AE to the E_i
E_i receives the AE, and trains it on its local data D_i and sends trained weights of AE back to G_s
G_s, wait for n E_i to send back their locally train AE.
if *weights received form n* E_i **then**
    $F(w) = \sum_{k=1}^{n} \frac{n_k}{n_t} w_{r+1}^k$
G_s constructs a classifier C
for *For* $i = 1, 2, 3$ **do**
    set Weight of convolutional $Layer_i$ of C = Weight of convolutional Layer$_i$ of $F(w)$.
    set convolutional Layer$_i$ of C trainable = False
G_s sends C to E_i
E_i trains C on D_i and sends back the trained C to G_s.
G_s, wait for n E_i to send back their locally trained C.
if *weights received form n* E_i **then**
    $F(w) = \sum_{k=1}^{n} \frac{n_k}{n_t} w_{r+1}^k$
G_s send $F(w)$ to E_i
E_i set $F(w)$ as weight of C and makes predictions
Repeat with continuously emerging data.

sends the AE to the client. It is worth noting that, each global round is divided into two tiers, for the first tier G_s sends the AE and for the second tier G_s sends the classifier C. Hence while requesting, each client mentions the tier as well. On receiving the autoencoder, client E_i trains the autoencoder on its local data D_i. When the training is completed the client sends back the trained weights of the autoencoder to the global server. The server waits for a fixed number n of clients to send the weights of their locally trained AE. Here, n can be decided by mutual consensus among administrators. When the desired number of clients send their weights and are received by G_s, it aggregates the weights of all the clients by using the formula for aggregation given by Eq. (8) from [69].

$$F(w) = \sum_{k=1}^{n} \frac{n_k}{n_t} w_{r+1}^k, \text{ where } F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w)). \quad (8)$$

Here, $F(w)$ are the aggregated weights, n_t is number of data samples of all participants and n_k is the number of samples of k^{th} participant. For a machine learning problem, typically $f_i(w) = (x_i, y_i; w)$, that is, the loss of the prediction on example x_i, y_i made with model parameters w. There are n clients over which the data is partitioned, with P_k the set of indexes of data points on client k, n is total number of participants in each round and r is the global round number.

After aggregation, G_s creates a new CNN-based classifier C for classification. Here, again we use the Keras auto-tuner for best hyper-parameters for the newly created C. Furthermore, we use the encoder part of the autoencoder for transfer learning. We transfer the weights of the updated and aggregated encoder part of AE to C and set the transferred layers to static. After this, G_s sends C to each client E_i. Upon receiving C, each E_i trains the classifier using its local data and sends it back to G_s. G_s collects the weight of n clients and aggregates them using Eq. (8). After aggregation, it sends the aggregated weights back to each E_i. Clients set the aggregated weights as new weights of their local C, which can be further used for predictions. During predictions, the XAI module taps the gradients and outputs the visual explanation.

3.5. Communication Cost Reduction and Privacy Enhancement

In the federated learning setting, training data remain distributed over a large number of clients each with unreliable and relatively slow network connections. For the synchronous protocols in federated learning [70], the total number of communication bits that are required during uplink and downlink communication by each of the N clients during training of the global model is given by:

$$\tau^{\text{up/down}} \in \mathcal{O}(U \times |w| \times (H(\Delta w^{\text{up/down}}) + \tau)), \quad (9)$$

where U is the total number of updates by each client, $|w|$ is the model size and $H(\Delta w^{\text{up/down}})$ is the entropy of transmitted weights during communication, τ is the difference between the update size and the minimal update size, given by entropy [71]. Generally, there are three ways to reduce the communication costs: (1) reducing the number of clients N, (2) reducing the update size, (3) reducing the number of updates U. Hence, research on communication-efficient federated learning can be divided into four groups: model compression, client selection, update reduction, and peer-to-peer learning [60]. In order to provide communication-efficient federated learning, we provide a new approach for our proposed architecture called layer selection, which comes under the model compression group. Moreover, layer selection can be added to all of the existing approaches to further reduce the communication costs. The proposed layer selection (communication cost reduction) method is shown in Figure 6, with more details given below.

Suppose that W1 and W2 represent the weights of all layers of encoder and decoder of the autoencoder, respectively, trained at edge devices. Since we are only concerned with the encoder part of the autoencoder, the edge devices select the weights of the encoder part (W1) and send them to the global server. The global server aggregates the received weights to obtain the global weights, represented as AW1 and sent to the edges. After receiving AW1, the edge devices use transfer learning to transfer these global weights to their local classifier and freeze the transferred layers, as

mentioned earlier. The edge devices train the local classifier using their local data. Suppose that WC1 and WC2 represent the weights of the trainable lower (convolutional) and higher (dense) layers of a local classifier, respectively. As the higher layers learn specific features about the underlying data [72], each edge sends only WC1 to the aggregation server that carries common and low-level features about the training data. The aggregation server performs weighted aggregation of all WC1 weights received to obtain AWC1, which are then sent to edge devices. The edge devices use AWC1 along with their individual WC2 for a more localized classification of the ECG. Since we share few weights compared to the classical method, this makes the communication lighter and reduces the communication costs. Moreover, since the FL framework continuously performs global training with emerging data, our communication cost reduction method can significantly reduce the overall communication costs.

Furthermore, recall that features in deep neural networks are highly transferable in the lower levels of the network since they focus on learning more common and low-level features. As the edge devices only send the weights of lower layers, the privacy of underlying data at each edge is enhanced. To be more precise, the weights of lower layer, weights of the encoder part in the autoencoder (WC1), contains more common and low-level features about the underlying data, while the weights of higher layers, weights of the decoder part of the autoencoder (WC2), contains more specific features about the underlying data. Hence, by not communicating WC2, we can increase the privacy of the local data by sharing only weights (WC1) that contains more common and low-level (i.e., less private) features.

4. Experimental Results

4.1. Dataset

For the experimental purpose, we used the widely used MIT-BIH Arrhythmia Database [32] as our baseline dataset. This database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. The dataset includes 109,446 samples. Twenty-three recordings were chosen at random from a set of 4,000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. In our experiment, we have used ECG lead II re-sampled to the sampling frequency of 125 Hz as the input. It should be noted that this dataset has unbalanced classes. Figure 7 shows the distribution of the original dataset. This highly unbalanced data can cause problems like overfitting. Hence to balance the classes we used upsampling. The resulting data distribution after upsampling is shown in Figure 8. Furthermore, this dataset is highly preprocessed, but in real-world scenarios, the ECG data collected is always noisy. Hence, to simulate more realistic data we introduced 10-30% noise into the original dataset and trained the proposed framework on the noisy version of the dataset, too. A comparison of the original (clean) and noisy datasets is shown in Figure 9.

4.2. Implementation Details

The Framework was implemented using Python and TensorFlow. Secure socket layer communication was used for communication between the server and edge devices. Both the autoencoder and the classifier were trained locally only on three local Raspberry Pi devices (Pi 3 Model B+ with 1.4GHz, 64-bit quad-core ARMv8 CPU and 1GB LPDDR2 SDRAM), denoted by Edge_1, Edge_2 and Edge_3. Furthermore, a workstation with an Intel core i-6700HQ CPU and 32 GB RAM was used as the global server G_s. It should be noted that FedHealth [24] initially trained their model at G_s, which may cause security risks in the case of a malicious global server. If the models (AE and C) are trained initially on G_s this may cause biased training. Hence, to avoid such risks, we performed only aggregation at the G_s. Furthermore, AE adopted a convolution size of 3. It uses a Root Mean Square Propagation (RMSProp) as the optimizer. Each E_i device uses 80% of data for training and 20% of data for evaluation. We distributed the dataset randomly at each edge device and introduced random noise. In this case, the data in Edge_1 contains 20% random noise, the data in Edge_2 contains 30% random noise,the data in Edge_3 contains 10% random noise. Furthermore, each edge used a fixed batch size of 100, and was trained for 50 training epochs. Moreover, each edge used an evolving learning rate, given by Eq. (2).

The classifier C used a batch size of 100. The learning rate was set to 0.001 with 150 training epochs. The accuracy of each of the locally trained C was calculated by using the following equation:

$$A_{cc}^{i} = \frac{|x : x \in D_i \wedge y^{'}(x) = y(x)|}{|x : x \in D_i|}. \tag{10}$$

In regards to the execution time, given the above setting, it took an average of 745 seconds to complete one global round of training. Furthermore, it took an average of 2.32 seconds to generate prediction and XAI results.

5. Performance Analysis of the Proposed Method

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

5.1. Reconstruction of Autoencoder

We introduced noise in to the dataset and used the noisy sample as the input in the autoencoder and the cleaned samples as labels. The performance of the autoencoder was measured using reconstruction mean absolute error (MAE). Reconstruction MAE for each locally trained AE in each of

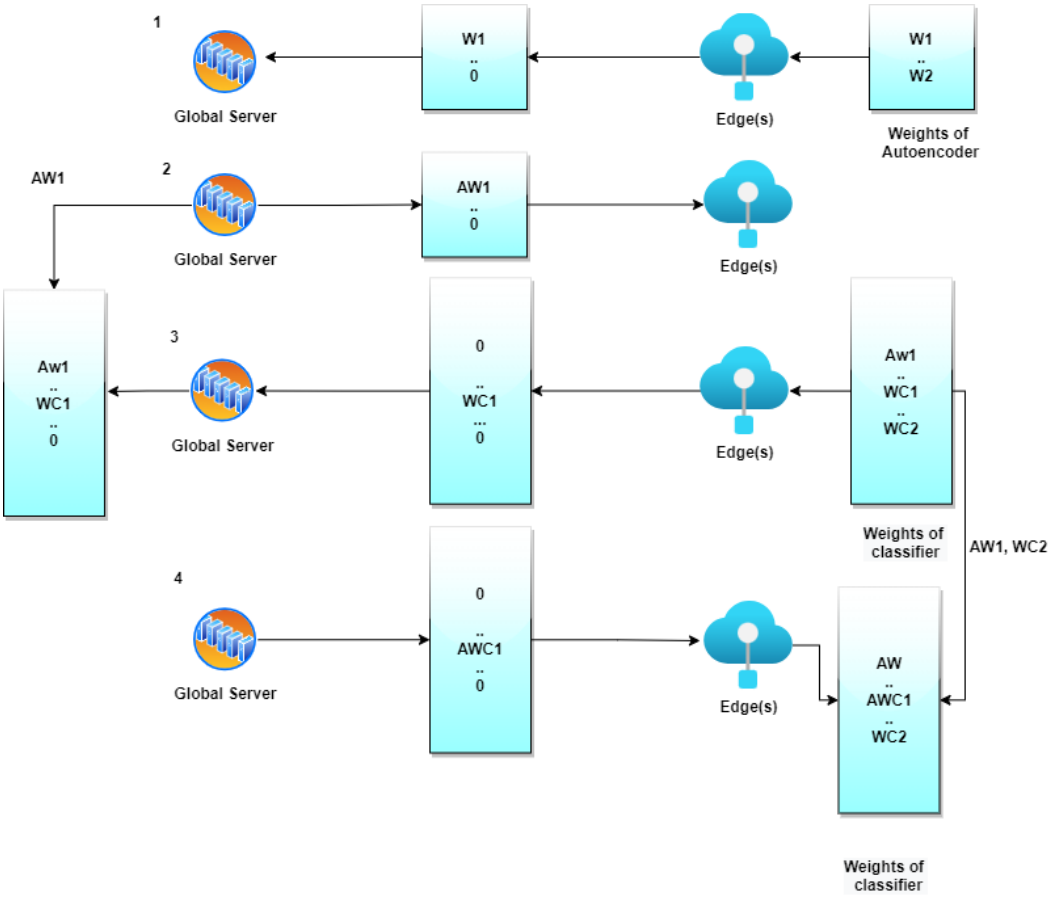

Figure 6: The layer selection method for communication cost reduction

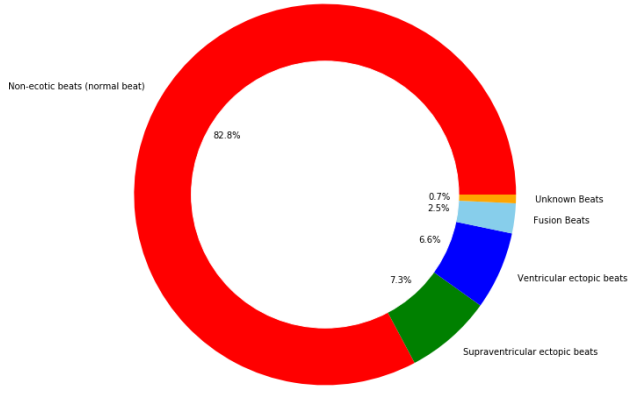

Figure 7: The distribution of the original dataset

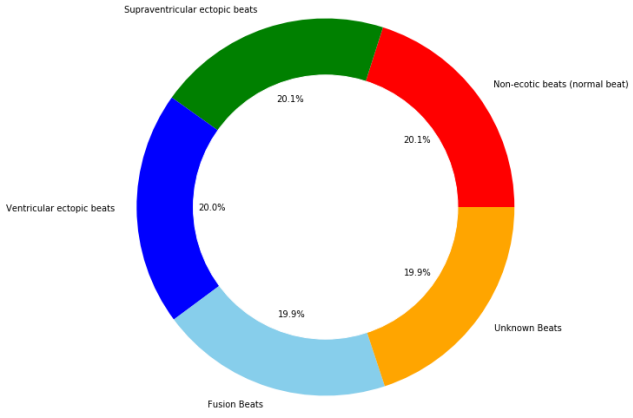

Figure 8: The distribution of the upsampled (re-balanced) dataset

$Edge_1$, $Edge_2$, $Edge_3$ and aggregated AE is given in Figure 10. It can be seen that the reconstruction MAE of the aggregated autoencoder is nearly 0, which means that our autoencoder reconstructed the original signal very well. Moreover, it can be seen that reconstruction MAE aggregation AE is less than or nearly equal to the reconstruction MAE of each locally trained AE.

5.2. Classification Performance

Classification performance was measured using the four standard metrics found in the literature [73]: classification accuracy, precision, recall and F1-score. While accuracy

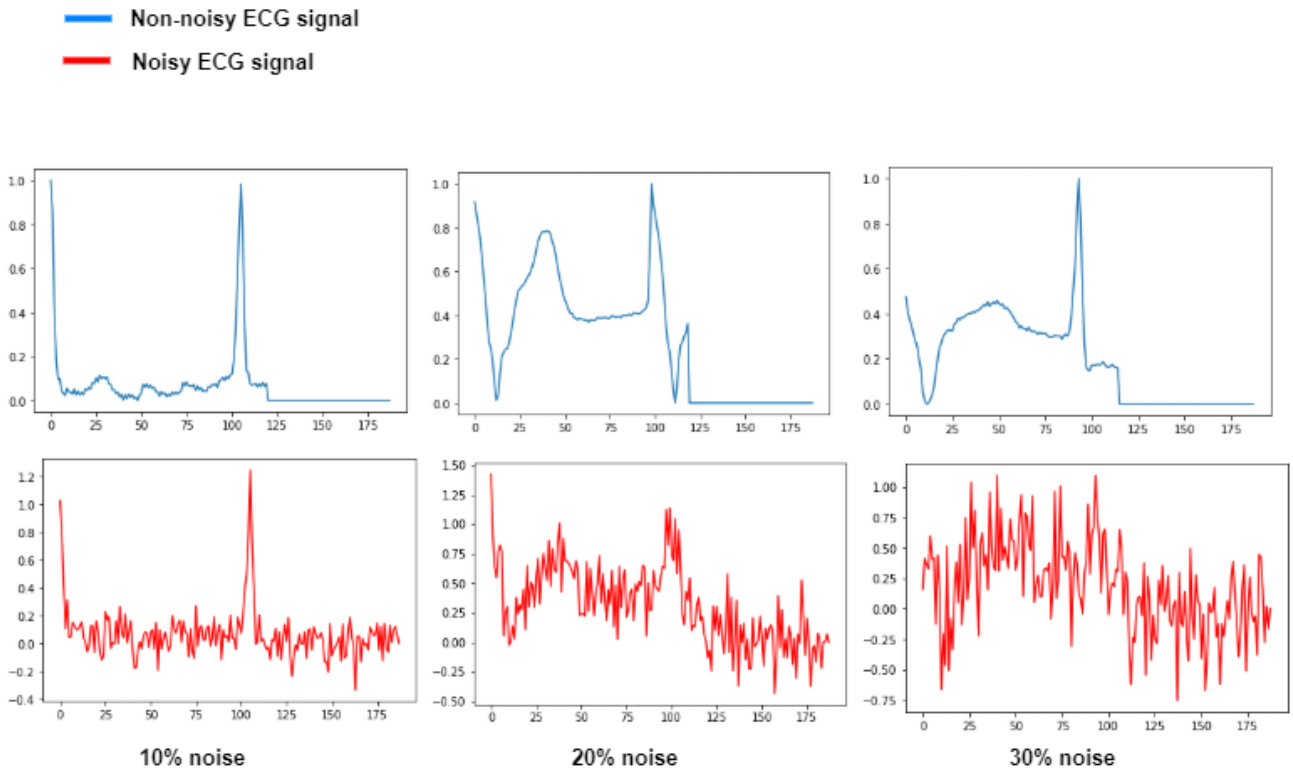

Figure 9: Comparison of the original and the noisy version of the dataset

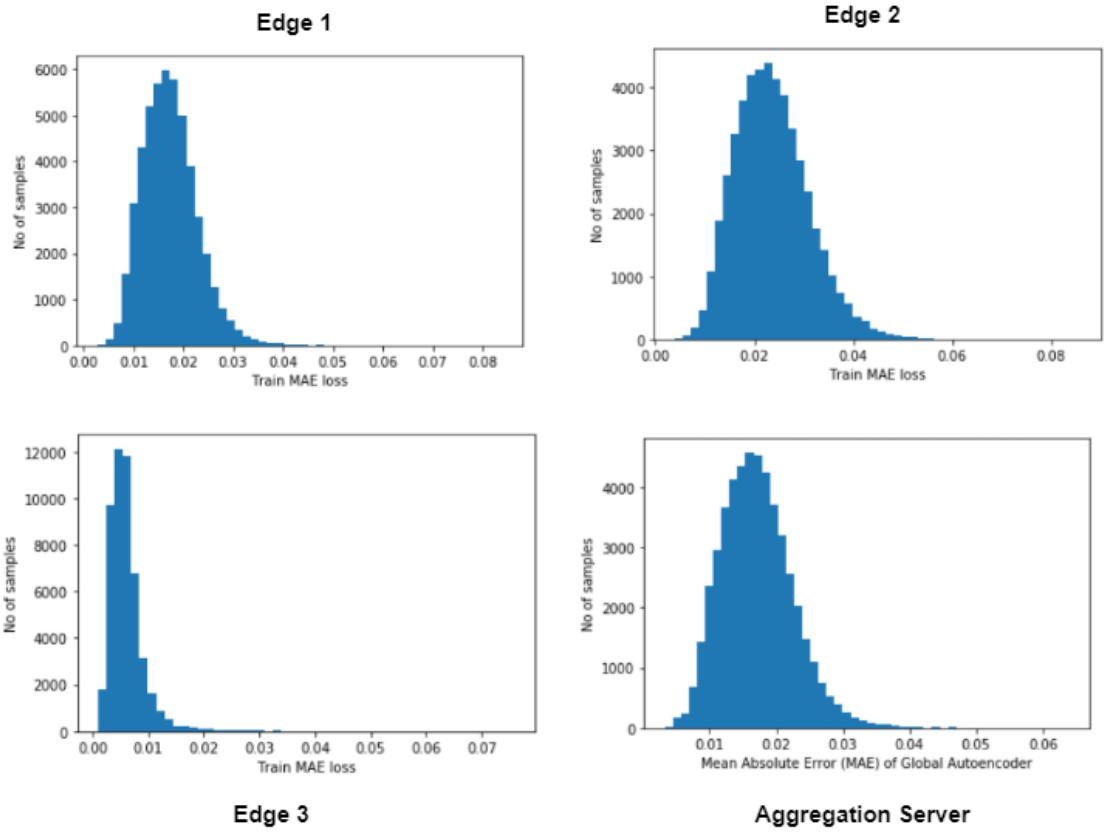

Figure 10: Reconstruction MAE

measures the overall system performance over all classes in the dataset, the other metrics are specific to each class, and they measure the ability of the classification algorithm to distinguish certain events. For a binary classifier, each of the metrics is defined as follows:

1. **Accuracy** is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations, as defined below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \tag{11}$$

   where TP, TN, FP and FN refer to the numbers of true positives, true negatives, false positives, and false negatives, respectively.
2. **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations, as defined below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{12}$$

3. **Recall** is the ratio of correctly predicted positive observations to all the observations in actual positive class, as defined below:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{13}$$

4. **F1-score** is the harmonic mean of precision and recall, as defined below:

$$\text{F1-Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{14}$$

The above definitions can be easily extended to multi-class classifiers with $n > 2$ classes. For instance, accuracy is defined as the ratio between the number of total correct predictions and the total number of samples. For other metrics, i.e., precision, recall, and F1-score, we can derive n binary classifiers, one for each given class versus all remaining classes (one-vs-rest), and then use the above definitions of the three metrics as usual for each of the n binary classifiers.

Precision, recall, F1-score metrics of each binary classifier (one for each of five class labels) at the three edge devices (Edge_1, Edge_2, and Edge_3) and the global server are given in Table 2. We also show the accuracy of the five-class classifier in the last column. It should be noted that the results shown in Table 2 are computed using the noisy data which we prepared earlier. We also tested the proposed classifier using the original (clean) data. With this data it provided $98 \pm 0.9\%$ accuracy. Other metrics, such as precision, recall and F1-score, are shown in Table 3. However, for real-time use we expect the data to be noisy, which is why we proceeded with the noisy data.

5.3. Qualitative Analysis

Understanding the reasons for predictions of the model decision is very important in healthcare applications. In order to validate that the decisions made by the proposed XAI module are interpretable, we use visualization to demonstrate that clinically important beats in the ECG wave are used for classification. Figure 12 illustrates the importance for each beat that the ECG classifier is giving while performing classification of some instance ECG signal inputs.

In order to achieve the interpretability/explainability of the XAI module, it is important to understand the ECG signal [74]. Generally, the amplitude and width of the p-wave, QRS complex and the T-wave are important features of an ECG graph, as shown in Figure 11. These regions play a vital role in ECG analysis [75]. The XAI module in the proposed framework shows that the proposed classifier looks at these critical features of the input sample. In Figure 12 , the red segments show more important regions of the heartbeat for the network's decision while predicting a particular class. In other words, the red segments of the heartbeat have more influence on the detection process of the classifier while classifying the input ECG signal.These results can be used to help clinical practitioners to diagnose the underlying health issues. However, we strongly advise that these results should not be used for any medical consultation without prior discussion with a clinical professional. In other words, heat maps should be cross-checked by clinicians with prior expert knowledge.

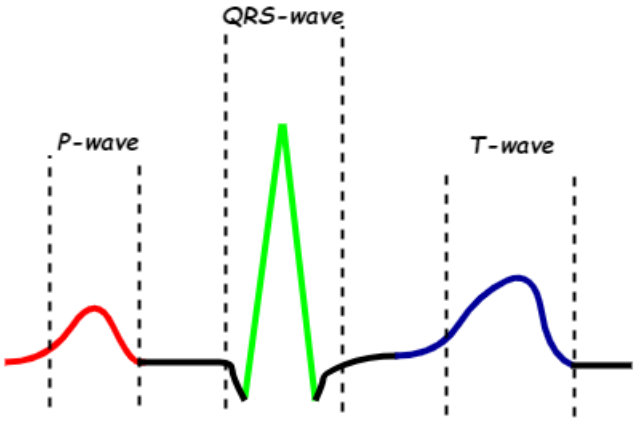

Figure 11: The major waves of a single normal ECG pattern

5.4. Comparison With Other State-of-the-Art Methods

We compared our proposed framework with the state-of-the-art methods reported in 2020 [24, 25, 26, 27, 28, 29, 30, 19, 31]. First we compare the previous work with ours to show that the proposed framework provides all of the desirable properties like interpretability, privacy preserving, and working with raw data. Table 4 shows the comparison between our proposed method and the other methods. It can be seen that the proposed scheme outperforms others by providing all desirable properties, while others lack some of the desirable properties. Moreover, we also compare our work with existing works for ECG classification. It should be noted that other methods used the baseline MIT-BIH dataset (without noise), with which better accuracy re-

Table 2
The classification performance of the proposed framework, with the noisy version of the dataset

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
N	89%	91%	90%	94.9%
S	94%	89%	92%	
V	93%	96%	94%	
F	95%	94%	95%	
Q	99%	99%	99%	

(a) Edge 1 (20% noise)

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
N	85%	87%	86%	91.9%
S	91%	87%	88%	
V	91%	94%	92%	
F	93%	93%	93%	
Q	98%	98%	98%	

(b) Edge 2 (30% noise)

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
N	94%	98%	96%	97.9%
S	98%	92%	95%	
V	95%	99%	97%	
F	99%	94%	96%	
Q	97%	100%	98%	

(c) Edge 3 (10% noise)

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
N	90%	92%	91%	94.5%
S	94%	89%	91%	
V	93%	96%	94%	
F	96%	96%	95%	
Q	99%	99%	99%	

(d) Global/Aggregation Server

Table 3
The classification performance of the proposed framework, with the original (clean) dataset

Class	*Precision*	*Recall*	*F1-Score*	*Accuracy*
N	95%	99%	97%	98.9%
S	98%	97%	98%	
V	97%	99%	98%	
F	99%	93%	96%	
Q	100%	100%	100%	

sults can be achieved. Contrastingly, we introduced (10%-30%) noise into the data to make it more realistic. Table 5 shows the comparison of classification performance between our proposed method and the other methods. It can be seen that, our proposed method outperformed the methods in others. It should be noted that the proposed classifier deals with the classification tasks of five classes, while others deal with fewer classes (some with two and some with three). Additionally, the proposed method works in federated architecture and performs better compared to others. Our proposed method provides explainability as an additional feature. Moreover, the proposed method provides data privacy to the users via the federated setting, which is not the case for other methods. Furthermore, the proposed method can denoise raw signals without any preprocessing, followed by classification and explainability.

5.5. Privacy Enhancement

As mentioned earlier, digital healthcare data is like digital finger prints that carry a lot of personal information. Hence, we should protect such data as much as possible while using them in machine learning algorithms. Most past studies on ECG classification do not provide privacy protection of such data because they are centralized and data are shared with the central model directly. Recently, Chen et al. [24] used federated learning to provide privacy protection, by only sharing the learned parameters without sharing the data. Although the shared parameters can protect privacy, there are still chances to recover some information from the shared parameters of higher level layers in the classifier, since they can contain more data-specific information as discussed previously. As a comparison, in our proposed framework we only share the learned parameters from lower-level layers that carry only more common and low-level (i.e., less privacy-sensitive) features. Thus, our proposed framework can enhance privacy even further, and at the same time can reduce communication costs as fewer parameters are shared between the edge/local and global servers.

Table 4
Comparison with the state-of-the-art work

Scheme	Interpretability	Raw Input	Privacy Preserving
[21]	✗	✗	✗
[22]	✗	✗	✗
[76]	✗	✗	✗
[24]	✗	✗	✓
[25]	✗	✗	✗
[26]	✗	✗	✗
[27]	✗	✗	✗
[28]	✗	✗	✗
[29]	✗	✗	✗
[30]	✗	✗	✗
[19]	✓	✗	✗
[31]	✗	✗	✗
Proposed	✓	✓	✓

A comparison between with existing work in federated setting for healthcare is shown in Table 6.

5.6. Communication Cost Reduction

Here, we show the communication cost reduction using the proposed communication cost reduction method. The number of total parameters communicated between an edge device and the global server, for one global round, denoted

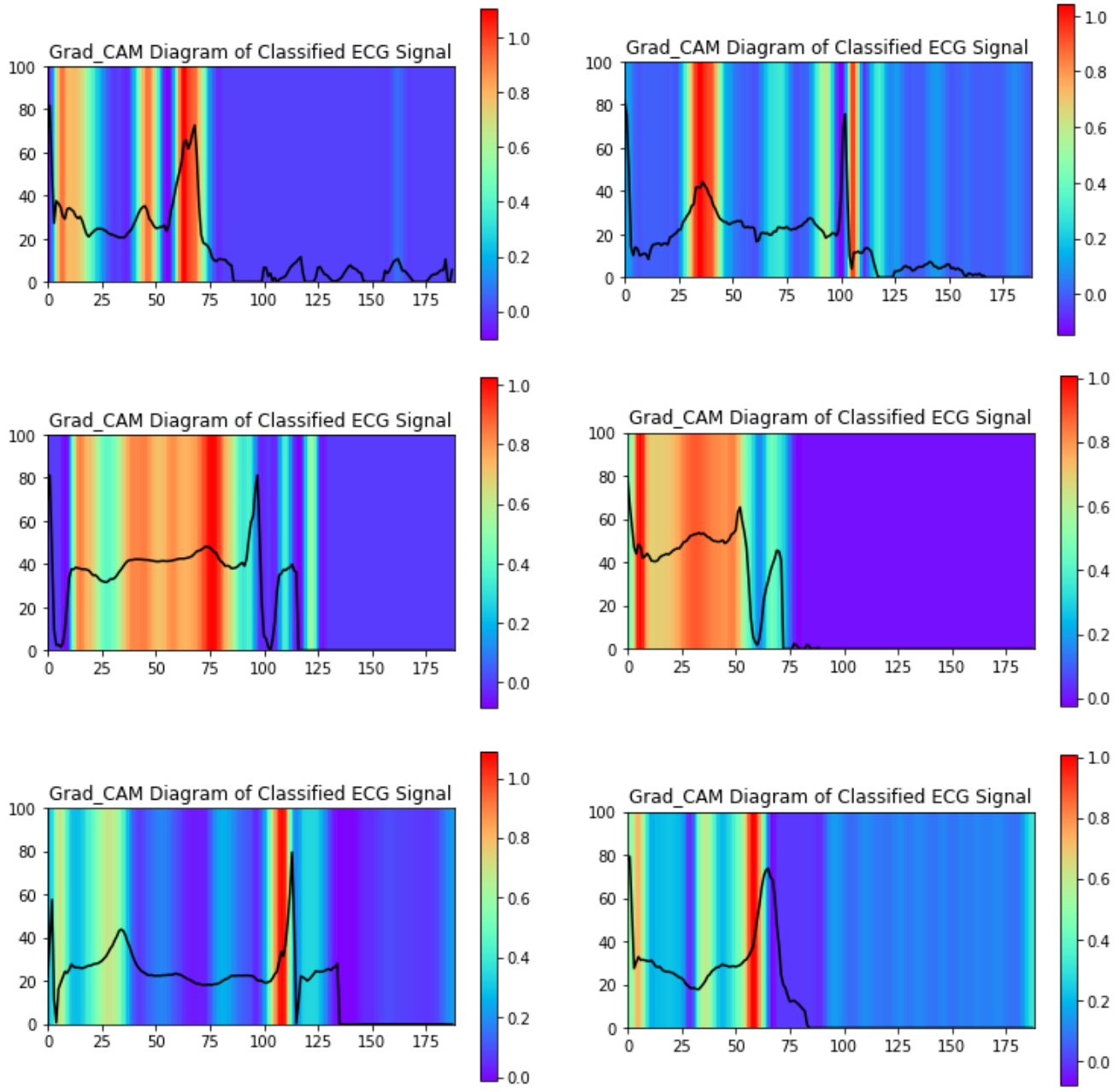

Figure 12: The outputs of the XAI module

by TPC, is given as follow:

$$\begin{aligned}\text{TPC} = \text{W1} + \text{W2} + \text{WC1} + \text{WC2} \\ + \text{AW1} + \text{AW2} + \text{AWC1} + \text{AWC2}, \quad (15)\end{aligned}$$

In the proposed framework, the TPC is given as follow:

$$\begin{aligned}\text{TPC} = 13386 + 13429 + 4160 + 181961 + \\ 13386 + 13429 + 4160 + 181961 = 425872, \quad (16)\end{aligned}$$

With the proposed communication cost reduction method, the TPC is given as follow:

$$\text{TPC} = \text{W1} + + \text{WC1} + \text{AW1} + + \text{AWC1}, \quad (17)$$

In the proposed framework, the TPC is given as follow:

$$\text{TPC} = 13386 + 4160 + 13386 + 4160 = 35092, \quad (18)$$

From Eqs. (16) and (18), we can calculate that the proposed communication cost reduction method reduces the communication cost by 8.2%.

5.7. Time Complexity of proposed Algorithm

In this section, we provide the time complexity of the proposed Algorithm 1. In a CNN-based network, the number of features in each feature map is at most a constant times the number of input features let us say n (typically the constant is < 1). Convolving a fixed-size filter across an input

Table 5
Comparison with previous studies for ECG classification

Scheme	Centralized or Federated	Acc (clean data)	Acc (noisy data)
[21]	Centralized	86.0%	-
[22]	Centralized	96.9%	-
[25]	Centralized	98.1%	-
[26]	Centralized	96.5%	-
[28]	Centralized	93.1%	-
[27]	Centralized	98.7%	-
[29]	Centralized	94.9%	-
[30]	Centralized	98.1%	-
[77]	Centralized	98.6%	-
[76]	Centralized	98.3%	-
[19]	Centralized	98.8%	-
Proposed	Federated	98.9%	94.5%

Table 6
Comparison with the state-of-the-art work in federated setting for healthcare

Scheme	Communication Cost Reduction	Privacy Enhancement
[24]	✗	✗
Proposed	✓	✓

signal with n features takes $O(n)$ time, since each output is just the sum product between some features let's say k in the input, and a fixed number of weights w in the filter, and w and k do not vary with n. Similarly, any max or average pooling operation does not take more than a linear amount of time in the input size. Moreover, the edge node can compute in parallel, therefore, the overall runtime is still linear i.e., $O(n)$.

5.8. Limitations and Future work

In this section, we discuss some limitations of the proposed framework. Our proposed framework provides a unique way of diagnosing arrhythmias with the explanation of the predicted results. However, there are certain limitations, which can be explored in the future to make the proposed framework more reliable. First, in federated learning, data is not collected at a single server and there are multiple devices for collecting and analyzing data. Such distributed settings increases the chances of data poisoning attacks, hence, methods should be developed for data integrity and authentication. Second, the proposed framework considers the data and devices in distributed edges to be homogeneous, but in some cases, the data and devices may be heterogeneous, and device-specific characteristics may limit the generalizability of the local models from device to device and may reduce the accuracy of the aggregated model. Furthermore, from a security perspective, malicious internal attackers have been not considered in our work, who may attack other peers by inserting hidden back-doors into the joint global model. Last but not the least, in case the global server goes down, the whole system may stop working. As part of our future work, we aim to improve the current framework by considering the above-mentioned limitations to produce a more secure and robust framework.

6. Conclusions

In this paper, we proposed a privacy-preserving, efficient and interpretable/explainable AI-based end-to-end framework to address the limitations of deep learning applications for ECG signal classification. Firstly, we proposed a CNN-based autoencoder in a federated architecture to denoise the raw ECG signal from patients. When trained on the baseline dataset, The proposed autoencoder provided an excellent reconstruction of the raw input signals and improved the overall performance when applied in federated settings. Secondly, we proposed a new classifier for ECG signals. When the classifier was trained in federated settings it was able to improve the overall classification performance of the edge devices. Moreover, the experimental results on the baseline database revealed that the proposed framework achieved outperformed existing algorithms, including both centralized and federated ones. Furthermore, we extended the usability of our framework by providing a novel explainable module on top of the classifier, whose usefulness is visually demonstrated by showing that clinically meaningful heartbeat segments of ECG signals are indeed behind the classification results. Additionally, we also proposed a communication cost reduction method, which can significantly reduce communication costs while increasing the level of privacy protection of users' ECG data against the global server. Hence, the proposed framework shows its applicability by providing many desirable properties including interpretability, privacy protection, communication cost reduction, and high accuracy in classification. Such a combination of such properties does not hold for other existing solutions, therefore making the proposed framework a unique solution for real-world healthcare applications where ECG signal classification is an important task.

Eventually, the proposed framework will encourage (1) more healthcare data owners to participate in training a good machine learning model for patients and health professionals, with fewer privacy concerns, (2) more accurate diagnostic assistance in places with scarce access to cardiologists or healthcare facilities, (3) more interpretable classification results that can be used to identify new potential patterns leading to trigger heart arrhythmias. Hence, the proposed framework has great potential to be added to hospital computer software platforms to support the work of health professionals and ultimately reduce mortality and save human lives.

As a future research direction, we aim at applying the proposed framework to more healthcare applications, especially human activity recognition and anomaly detection in the context of home care, and other types of arrhythmia to extract new patterns that might be helpful for their diagnosis and monitoring. We also aim at extending the applicability of the proposed work by addressing the limitations of the proposed framework as mentioned previously.

Acknowledgments

This research work was supported by the I-SITE Université Lille Nord-Europe 2021 of France under grant No. I-COTKEN-20-001-TRAN-RAZA.

References

[1] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, I. Yaqoob, Big IoT data analytics: architecture, opportunities, and open research challenges, ieee access 5 (2017) 5247–5261. doi:10.1109/ACCESS.2017.2689040.

[2] D. Mourtzis, E. Vlachou, N. Milas, Industrial big data as a result of IoT adoption in manufacturing, Procedia CIRP 55 (2016) 290–295. doi:10.1016/j.procir.2016.07.038.

[3] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, IEEE Communications Surveys & Tutorials 20 (4) (2018) 2923–2960. doi:10.1109/COMST.2018.2844341.

[4] J. W. Shavlik, T. Dietterich, T. G. Dietterich, Readings in Machine Learning, Morgan Kaufmann, 1990.

[5] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, Nature Medicine 25 (1) (2019) 24–29. doi:10.1038/s41591-018-0316-z.

[6] A. M. Ozbayoglu, M. U. Gudelek, O. B. Sezer, Deep learning for financial applications: A survey, Applied Soft Computing (2020) 106384:1–106384:29doi:10.1016/j.asoc.2020.106384.

[7] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications, Journal of Manufacturing Systems 48 (2018) 144–156. doi:10.1016/j.jmsy.2018.01.003.

[8] A. Kamilaris, F. X. Prenafeta-Boldú, Deep learning in agriculture: A survey, Computers and electronics in agriculture 147 (2018) 70–90. doi:10.1016/j.compag.2018.02.016.

[9] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, H. Laga, A comprehensive survey of deep learning for image captioning, ACM Computing Surveys 51 (6) (2019) 118:1–118:36. doi:10.1145/3295748.

[10] T. Georgiou, Y. Liu, W. Chen, M. Lew, A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision, International Journal of Multimedia Information Retrieval 9 (3) (2020) 135–170. doi:10.1007/s13735-019-00183-w.

[11] R. Miotto, F. Wang, S. Wang, X. Jiang, J. T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Briefings in Bioinformatics 19 (6) (2018) 1236–1246. doi:10.1093/bib/bbx044.

[12] V. Kumar, M. Garg, Deep learning in predictive analytics: A survey, in: Proceedings of the 2017 International Conference on Emerging Trends in Computing and Communication Technologies, IEEE, 2017. doi:10.1109/ICETCCT.2017.8280331.

[13] Z. Ji, Z. C. Lipton, C. Elkan, Differential privacy and machine learning: a survey and review (2014).
URL https://arxiv.org/abs/1412.7584

[14] L. Van Zoonen, Privacy concerns in smart cities, Government Information Quarterly 33 (3) (2016) 472–480. doi:10.1016/j.giq.2016.06.004.

[15] J. Konečnỳ, H. B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, arXiv preprint arXiv:1610.02527 (2016).
URL https://arxiv.org/abs/1610.02527

[16] D. Gunning, D. Aha, DARPA's explainable artificial intelligence (XAI) program, AI Magazine 40 (2) (2019) 44–58. doi:10.1609/aimag.v40i2.2850.

[17] W. Samek, T. Wiegand, K.-R. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models (2017).
URL https://arxiv.org/abs/1708.08296

[18] J. Choo, S. Liu, Visual analytics for explainable deep learning, IEEE Computer Graphics and Applications 38 (4) (2018) 84–92. doi:10.1109/MCG.2018.042731661.

[19] S. Mousavi, F. Afghah, U. R. Acharya, HAN-ECG: An interpretable atrial fibrillation detection model using hierarchical attention networks, Computers in Biology and Medicine 127 (2020) 104057:1–104057:9. doi:10.1016/j.compbiomed.2020.104057.

[20] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-CAM: Why did you say that? (2016).
URL https://arxiv.org/abs/1611.07450

[21] B. Pyakillya, N. Kazachenko, N. Mikhailovsky, Deep learning for ECG classification, in: Journal of Physics: Conference Series, Vol. 913, IOP Publishing, 2017, pp. 012004:1–012004:5. doi:10.1088/1742-6596/913/1/012004.

[22] S. M. Mathews, C. Kambhamettu, K. E. Barner, A novel application of deep learning for single-lead ECG classification, Computers in Biology and Medicine 99 (2018) 53–62. doi:10.1016/j.compbiomed.2018.05.013.

[23] F. Murat, O. Yildirim, M. Talo, U. B. Baloglu, Y. Demir, U. R. Acharya, Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review, Computers in biology and medicine (2020) 103726:1–103726:14doi:10.1016/j.compbiomed.2020.103726.

[24] Y. Chen, X. Qin, J. Wang, C. Yu, W. Gao, Fedhealth: A federated transfer learning framework for wearable healthcare, IEEE Intelligent Systems 35 (4) (2020) 83–93. doi:10.1109/MIS.2020.2988604.

[25] S. L. Oh, E. Y. Ng, R. San Tan, U. R. Acharya, Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats, Computers in biology and medicine 102 (2018) 278–287.

[26] S. Liaqat, K. Dashtipour, A. Zahid, K. Assaleh, K. Arshad, N. Ramzan, Detection of atrial fibrillation using a machine learning approach, Information 11 (12) (2020) 549:1–549:15. doi:10.3390/info11120549.

[27] U. Erdenebayar, H. Kim, J.-U. Park, D. Kang, K.-J. Lee, Automatic prediction of atrial fibrillation based on convolutional neural network using a short-term normal electrocardiogram signal, Journal of Korean medical science 34 (7) (2019) e64:1–e64:10. doi:10.3346/jkms.2019.34.e64.

[28] D. K. Atal, M. Singh, Arrhythmia classification with ECG signals based on the optimization-enabled deep convolutional neural network, Computer Methods and Programs in Biomedicine 196 (2020) 105607:1–105607:19. doi:10.1016/j.cmpb.2020.105607.

[29] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, M. Adam, Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network, Information Sciences 405 (2017) 81–90. doi:10.1016/j.ins.2017.04.012.

[30] Z. Yao, Z. Zhu, Y. Chen, Atrial fibrillation detection by multi-scale convolutional neural networks, in: Proceedings of the 2017 20th International Conference on Information Fusion, IEEE, 2017. doi:10.23919/ICIF.2017.8009782.

[31] S. Nurmaini, A. E. Tondas, A. Darmawahyuni, M. N. Rachmatullah, R. U. Partan, F. Firdaus, B. Tutuko, F. Pratiwi, A. H. Juliano, R. Khoirani, Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks, Future Generation Computer Systems 113 (2020) 304–317. doi:10.1016/j.future.2020.07.021.

[32] G. B. Moody, R. G. Mark, The impact of the MIT-BIH Arrhythmia Database, IEEE Engineering in Medicine and Biology Magazine 20 (3) (2001) 45–50. doi:10.1109/51.932724.
URL https://doi.org/10.1109/51.932724

[33] Z. Yang, M. Ghadamyari, H. Khorramdel, S. M. S. Alizadeh, S. Pirouzi, M. Milani, F. Banihashemi, N. Ghadimi, Robust multi-objective optimal design of islanded hybrid system with renewable and diesel sources/stationary and mobile energy storage systems, Renewable and Sustainable Energy Reviews 148 (2021) 111295. doi:10.1016/j.rser.2021.111295.

[34] J. Liu, C. Chen, Z. Liu, K. Jermsittiparsert, N. Ghadimi, An igdt-based risk-involved optimal bidding strategy for hydrogen storage-based intelligent parking lot of electric vehicles, Journal of Energy Storage 27 (2020) 101057.

[35] P. Angelov, D. P. Filev, N. Kasabov, Evolving intelligent systems: methodology and applications, Vol. 12, John Wiley & Sons, 2010.

[36] A. Handa, A. Sharma, S. K. Shukla, Machine learning in cybersecurity: A review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9 (4) (2019) e1306. doi:10.1002/widm.1306.

[37] H. Storm, K. Baylis, T. Heckelei, Machine learning in agricultural and applied economics, European Review of Agricultural Economics 47 (3) (2020) 849–892.

[38] K. Shailaja, B. Seetharamulu, M. Jabbar, Machine learning in healthcare: A review, in: 2018 Second international conference on electronics, communication and aerospace technology (ICECA), IEEE, 2018, pp. 910–914. doi:10.1109/ICECA.2018.8474918.

[39] N. Razmjooy, F. R. Sheykhahmad, N. Ghadimi, A hybrid neural network–world cup optimization algorithm for melanoma detection, Open Medicine 13 (1) (2018) 9–16. doi:10.1515/med-2018-0002.

[40] A. Parsian, M. Ramezani, N. Ghadimi, A hybrid neural network-gray wolf optimization algorithm for melanoma detection., Biomedical Research (0970-938X) 28 (8) (2017).

[41] Z. Xu, F. R. Sheykhahmad, N. Ghadimi, N. Razmjooy, Computer-aided diagnosis of skin cancer based on soft computing techniques, Open Medicine 15 (1) (2020) 860–871. doi:10.1515/med-2020-0131.

[42] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, D. I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Computational and structural biotechnology journal 13 (2015) 8–17. doi:10.1016/j.csbj.2014.11.005.

[43] H. H. Atkinson, C. Rosano, E. M. Simonsick, J. D. Williamson, C. Davis, W. T. Ambrosius, S. R. Rapp, M. Cesari, A. B. Newman, T. B. Harris, S. M. Rubin, K. Yaffe, S. Satterfield, S. B. Kritchevsky, Cognitive function, gait speed decline, and comorbidities: the health, aging and body composition study, The Journals of Gerontology Series A: Biological Sciences and Medical Sciences 62 (8) (2007) 844–850. doi:10.1093/gerona/62.8.844.

[44] Y. Chen, J. Wang, M. Huang, H. Yu, Cross-position activity recognition with stratified transfer learning, Pervasive and Mobile Computing 57 (2019) 1–13. doi:10.1016/j.pmcj.2019.04.004.

[45] S. C. Mukhopadhyay, Wearable sensors for human activity monitoring: A review, IEEE sensors journal 15 (3) (2014) 1321–1330. doi:10.1109/JSEN.2014.2370945.

[46] O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Communications Surveys & Tutorials 15 (3) (2012) 1192–1209. doi:10.1109/SURV.2012.110112.00192.

[47] R. Bhardwaj, A. R. Nambiar, D. Dutta, A study of machine learning in healthcare, in: Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference, Vol. 2, IEEE, 2017, pp. 236–241. doi:10.1109/SURV.2012.110112.00192.

[48] G. Manogaran, D. Lopez, A survey of big data architectures and machine learning algorithms in healthcare, International Journal of Biomedical Engineering and Technology 25 (2-4) (2017) 182–211. doi:10.1504/IJBET.2017.087722.

[49] R. Fakoor, F. Ladhak, A. Nazi, M. Huber, Using deep learning to enhance cancer diagnosis and classification, Proceedings of the WHEALTH Workshop at the 30th International Conference on Machine Learning 28 (2013).

[50] S. H. Jambukia, V. K. Dabhi, H. B. Prajapati, Classification of ECG signals using machine learning techniques: A survey, in: Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications, IEEE, 2015, pp. 714–721. doi:10.1109/ICACEA.2015.7164783.

[51] C. Roopa, B. Harish, A survey on various machine learning approaches for ECG analysis, International Journal of Computer Applications 163 (9) (2017) 25–33. doi:10.5120/ijca2017913737.

[52] S. Sahoo, M. Dash, S. Behera, S. Sabut, Machine learning approach to detect cardiac arrhythmias in ECG signals: A survey, IRBM (2020) 185–194doi:10.1016/j.irbm.2019.12.001.

[53] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, K. Sricharan, Recognizing abnormal heart sounds using deep learning (2017).
URL https://arxiv.org/abs/1707.04642

[54] M. Gjoreski, A. Gradišek, B. Budna, M. Gams, G. Poglajen, Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds, IEEE Access 8 (2020) 20313–20324. doi:10.1109/ACCESS.2020.2968900.

[55] J.-S. Huang, B.-Q. Chen, N.-Y. Zeng, X.-C. Cao, Y. Li, Accurate classification of ECG arrhythmia using MOWPT enhanced fast compression deep learning networks, Journal of Ambient Intelligence and Humanized Computing (2020). doi:10.1007/s12652-020-02110-y.

[56] N. Inkster, China's Cyber Power, The International Institute for Strategic Studies, 2018.
URL https://www.iiss.org/publications/adelphi/2016/chinas-cyber-power

[57] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, Z. Lin, When machine learning meets privacy: A survey and outlook, ACM Computing Surveys 54 (2) (2021) 31:1–31:36. doi:10.1145/3436755.

[58] N. Waheed, X. He, M. Ikram, M. Usman, S. S. Hashmi, M. Usman, Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures, ACM Computing Surveys (CSUR) 53 (6) (2020) 122:1–122:37. doi:10.1145/3417987.

[59] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology 10 (2) (2019) 12:1–12:19. doi:10.1145/3298981.

[60] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, Journal of Healthcare Informatics Research 5 (1) (2021) 1–19. doi:10.1007/s41666-020-00082-4.

[61] P. Huang, G. Wang, S. Qin, Boosting for transfer learning from multiple data sources, Pattern Recognition Letters 33 (5) (2012) 568–579. doi:10.1016/j.patrec.2011.11.023.

[62] X. Qin, Y. Chen, J. Wang, C. Yu, Cross-dataset activity recognition via adaptive spatial-temporal transfer learning, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (4) (2019) 148:1–148:25. doi:10.1145/3369818.

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

[64] H. Nguyen, K. P. Tran, S. Thomassey, M. Hamad, Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management, International Journal of Information Management 57 (2021) 102282:1–102282:13. doi:10.1016/j.ijinfomgt.2020.102282.

[65] W.-J. Jia, Y.-D. Zhang, Survey on theories and methods of autoencoder, Computer Systems & Applications (2018) 05.

[66] R. Yilmazer, D. Birant, Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores, Sensors 21 (2) (2021) 327. doi:10.3390/s21020327.

[67] F. K. Došilović, M. Brčić, N. Hlupić, Explainable artificial intelligence: A survey, in: Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, IEEE, 2018, pp. 210–215. doi:10.23919/MIPRO.2018.8400040.

[68] R. Assaf, I. Giurgiu, F. Bagehorn, A. Schumann, MTEX-CNN: Multivariate time series explanations for predictions with convolutional neural networks, in: Proceedings of the 2019 IEEE International Conference on Data Mining, IEEE, 2019, pp. 952–957. doi:10.1109/ICDM.2019.00106.

[69] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statisticss (AISTATS) 2017, JMLR, 2017, pp. 1273–1282.
URL http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.png

[70] V. Smith, C.-K. Chiang, M. Sanjabi, A. Talwalkar, Federated multi-task learning, arXiv preprint arXiv:1705.10467 (2017).
URL https://arxiv.org/abs/1705.10467

[71] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-IID data, IEEE Transactions on Neural Networks and Learning Systems 31 (9) (2019) 3400–3413. doi:10.1109/TNNLS.2019.2944481.

[72] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al., A closer look at memorization in deep networks, in: Proceedings of the 34th International Conference on Machine Learning, JMLR, 2017, pp. 233–242.
URL http://proceedings.mlr.press/v70/arpit17a.html

[73] Y. H. Hu, S. Palreddy, W. J. Tompkins, A patient-adaptable ECG beat classifier using a mixture of experts approach, IEEE Transactions on Biomedical Engineering 44 (9) (1997) 891–900. doi:10.1109/10.623058.

[74] M. B. Conover, Understanding Electrocardiography, Elsevier Health Sciences, 2002.

[75] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, M. B. Gulmezoglu, A survey on ecg analysis, Biomedical Signal Processing and Control 43 (2018) 216–235. doi:10.1016/j.bspc.2018.03.003.

[76] C. Yuan, Y. Yan, L. Zhou, J. Bai, L. Wang, Automated atrial fibrillation detection based on deep learning network, in: Proceedings of the 2016 IEEE International Conference on Information and Automation, IEEE, 2016, pp. 1159–1164. doi:10.1109/ICInfA.2016.7831994.

[77] Y. Xia, N. Wulan, K. Wang, H. Zhang, Detecting atrial fibrillation by deep convolutional neural networks, Computers in Biology and Medicine 93 (2018) 84–92. doi:10.1016/j.compbiomed.2017.12.007.