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 play a vital role in classifying different arrhythmias using the electrocardiography (ECG) data. Nevertheless, training deep learning models normally requires a large amount of data and it 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, we design a new explainable artificial intelligence (XAI) based deep learning framework in a federated setting for ECG-based healthcare applications. The federated setting is used to solve issues such as data availability and privacy concerns. Furthermore, the proposed framework setting effectively classifies arrhythmia's using an autoencoder and a classifier, both based on a convolutional neural network (CNN). Additionally, we propose an XAI-based module on top of the proposed classifier to explain the classification results, which help clinical practitioners make quick and reliable decisions. The proposed framework was trained and tested using the MIT-BIH Arrhythmia database. The classifier achieved accuracy up to 94% and 98% for arrhythmia detection using noisy and clean data, respectively, with five-fold cross-validation.

1. Introduction

With the increase of internet of things (IoT) devices being used in the 21^{st} century, tons 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 quiet 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 quiet unique properties. Most of the data related to healthcare is 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 diagnosis of 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 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 of doing machine learning from the need to store the data centrally at a single server or in the cloud. Although deep learning with federated setting can solve the issues mentioned earlier, there exists the 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, researcher have proposed different solutions [17, 18]. For instance, Selvaraju proposed a method called Grad-CAM [19] to visualize input regions that are important for predictions. From such val-

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); 0000-0001-5628-7328 (Shujun Li)

ues we can have an idea about where exactly the machine learning model is focusing at 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 with the reason behind a certain prediction for a 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 [20, 21, 22]. 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 EEG 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, 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 explainable healthcare framework in a federated setting. Firstly, we propose a deep convolutional neural network (CNN) based autoencoder, which is used to denoise the raw ECG signals from the subject directly. Secondly, we propose a CNN-based classifier, which uses transfer learning to classify the raw time series of ECG data. Thirdly, we adopt the Grad-CAM model [19] in the framework to explain the classification results in a novel and reliable pattern. Additionally, the proposed framework provides an enhanced level of privacy protection to patients via the federated setting. The contributions of our work are summarized below.

1. We develop a CNN-based autoencoder in a federated setting to denoise the raw time series of ECG signal collected data from patients. The autoencoder provides a denoised version of input, which we use for explanation of the predictions.
2. With the help of transfer learning, we use the encoder part of the proposed autoencoder to make a CNN-based classifier which 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).
3. We combine the proposed classifier with an XAI module to explain the decision making process of the classifier. The XAI module can be used with every updated classier in the federated setting, as it can be used with any classifier architecture and it does not needs retraining.
4. We used the MIT-BIH Arrhythmia Database [23] 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 provides an overall accuracy of 94% using noisy data and an overall accuracy of 98% on the clean data in the original MIT-BIH database. Moreover, we evaluated the performance of proposed framework using four standard metrics: classification accuracy, precision, recall and F1-score.
5. The proposed framework provides an enhanced level of privacy protection to users because of the federated setting.

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

2.1. Machine Learning in Healthcare

Certain activities in our body are governed by signals of some cognitive diseases [24]. 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 [25, 26, 27]. 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, 28, 29, 30].

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 [31, 32, 33, 34, 35]. Rubin et al. [36] 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. [37] 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. [38] 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 [39, 40, 41]. 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 is privacy preserving and efficient communication constraints, FL finds a number of applications in healthcare [42]. Xu et al. [43] 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 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 [44] and feature matching [45]. Recently, deep transfer learning methods have made considerable success in many application fields. Chen et al. [46] 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 provides 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 contains random noise and dealing with the random noise is quiet 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 has the limitation of explainability, as discussed earlier. Thus there is a need of research work to address such challenges.

2.2. Autoencoder

Autoencoder [47] is unsupervised neural network which 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 [48]. Given the input $x \in R^m$, 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:

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

where Y_i is the true label, $Y_i^{'}$ is 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 [49]. 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 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.

2.4. Transfer Learning

Transfer learning aims at transferring knowledge from existing domains to a new domain. The key idea is to reduce

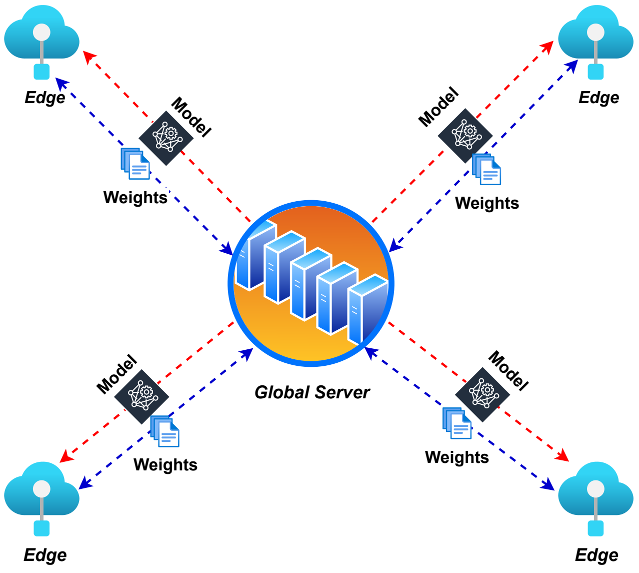


Figure 1: Architecture of Federated Learning

the distribution divergence between different domains. Here are mainly two types of transfer learning: instance reweighting [44] and feature matching [45]. 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. Research suggest that it will be of key importance in healthcare, manufacturing, insurance, and automobiles [50].

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 $E = \{E_1, E_2, \dots, E_N\}$ and the data of each E_i (here $i = 1, 2, \dots, N$) is given by $\{D_1, D_2, \dots, D_i\}$, respectively. A conventional machine learning model, denoted by ConMOD, can be trained by combining all the data $D = \{D_1, D_2, \dots, D_i\}$. 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 objective 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 consist of three major parts, the autoencoder, the classifier and the XAI module, which are discussed in 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 consist 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 consist of the input layer, 3 maxpooling layers and 3 convolutional layers in alternate fashion. On other hand, the decoder consist of 3 upsampling layers, 3 convolutional layers and an 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

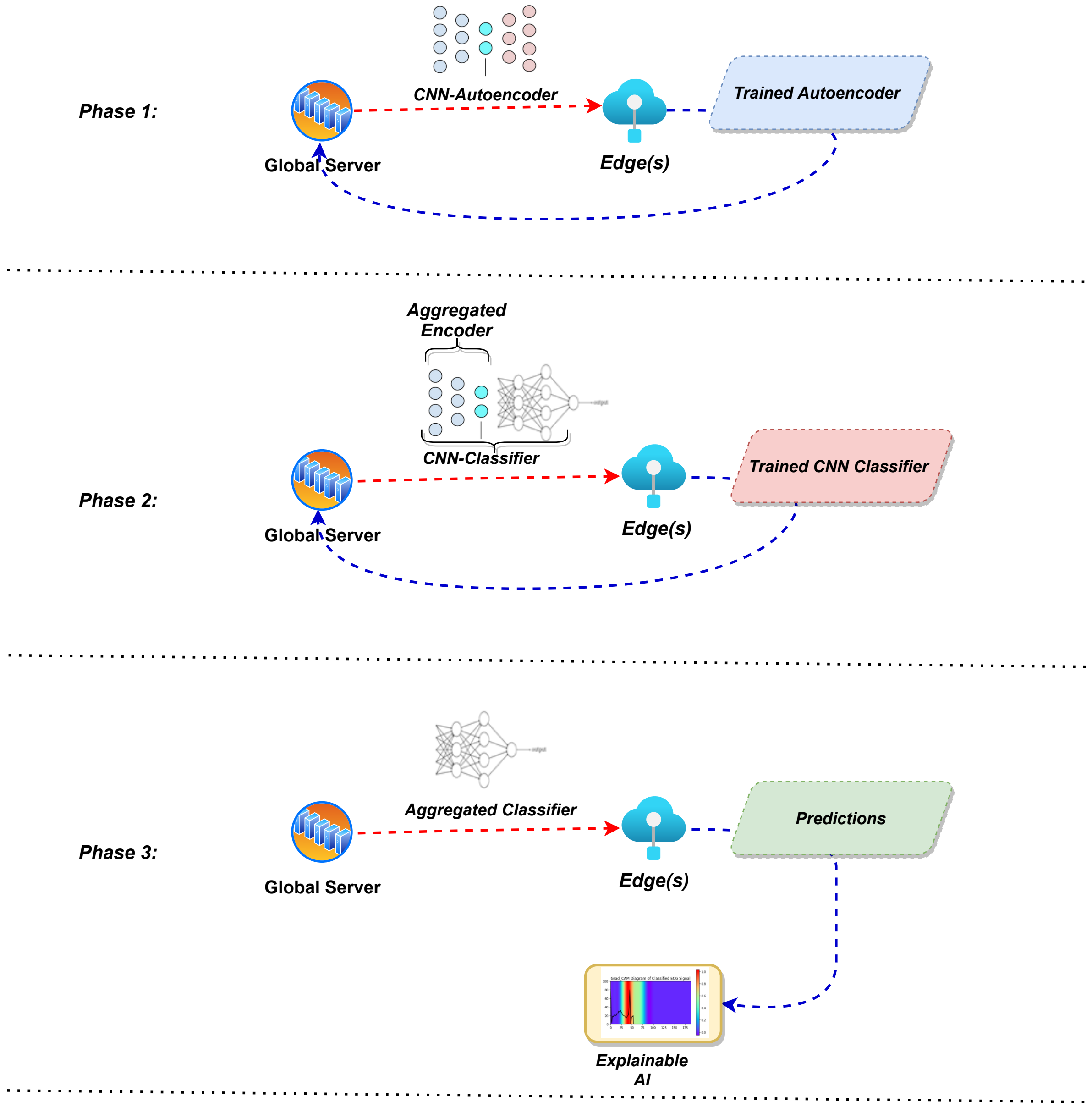


Figure 2: An overview of the proposed framework

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:

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

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 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 [19] and [51], 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 pre-

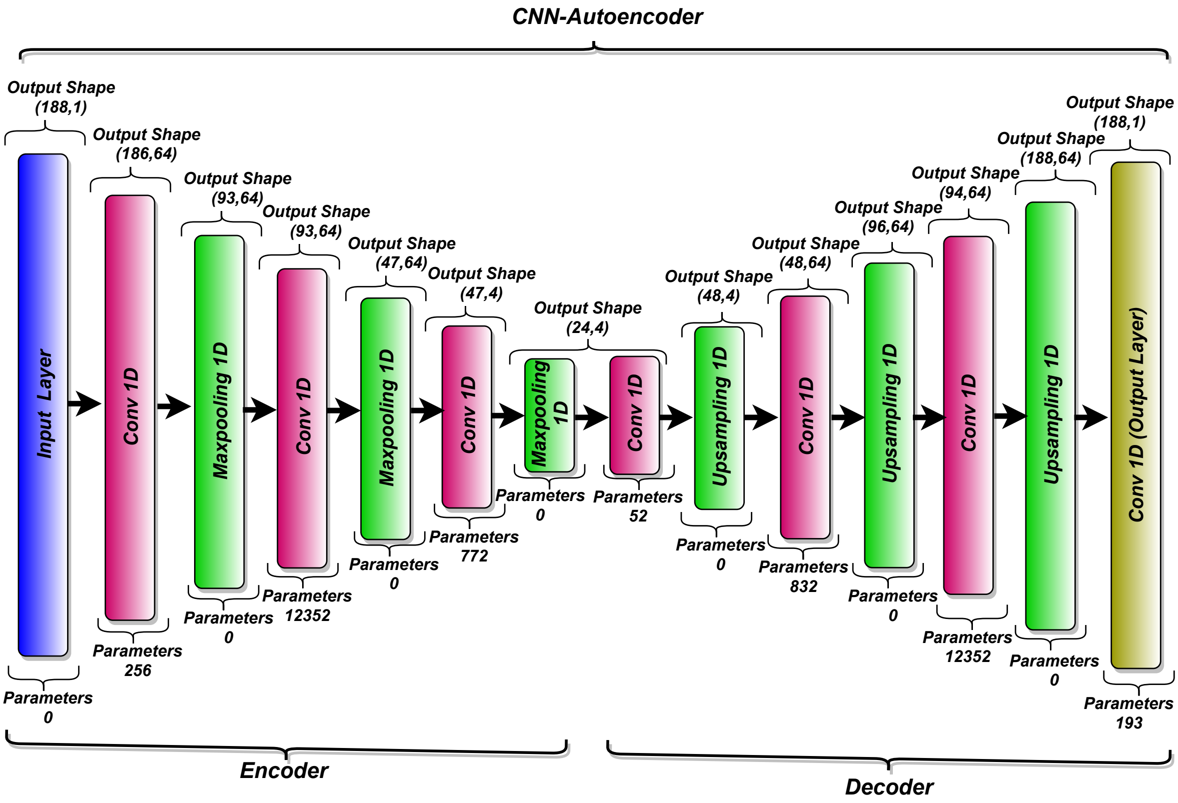


Figure 3: The architecture of the proposed denoising autoencoder

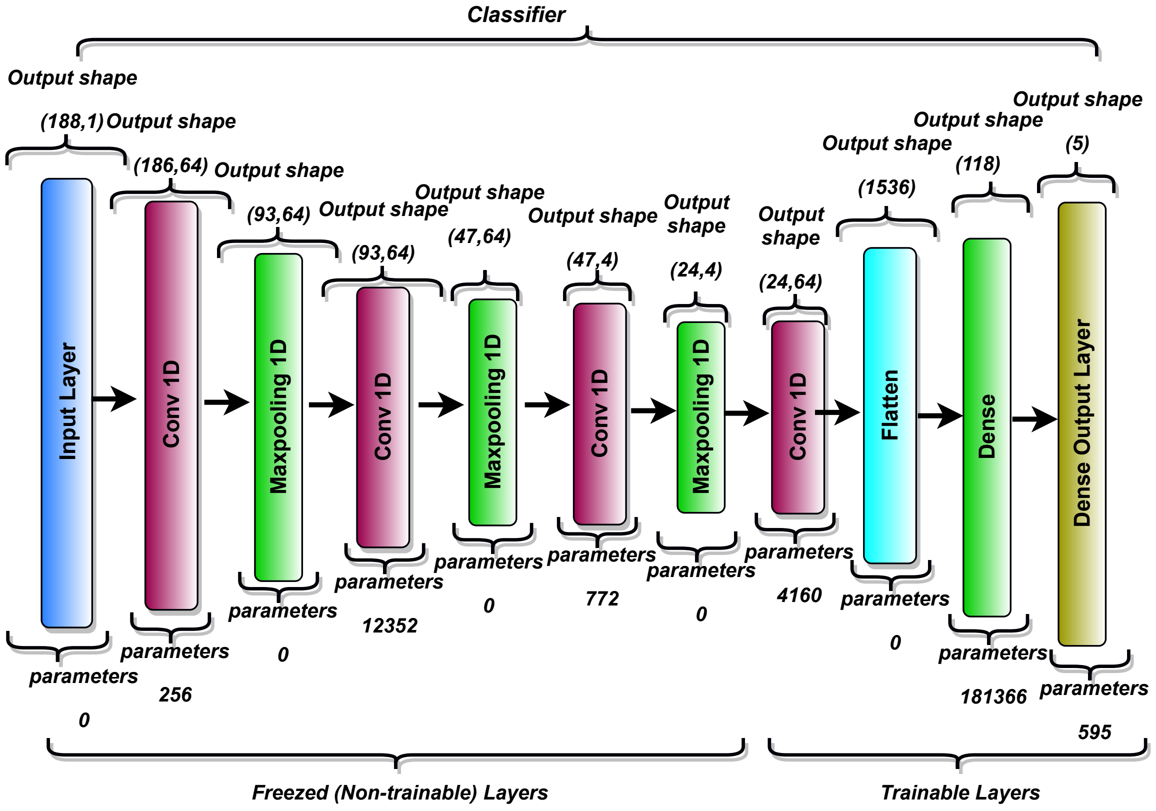


Figure 4: The proposed CNN-based classifier

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

diction given by the classifier. The XAI with GRAD-CAM module is shown in Figure 5.

The creation of this heatmap visualization consist of he following steps:

1. In first step, we compute 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:

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

   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 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 quiet small, and it is difficult to visualize them for analysis. To address this problem, we upsample the heatmap to the size of input sample in width. Moreover, we feed the input sample to the autoencoder and receive a denoised version of 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 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 aims at extracting low-level features about activity recognition and for the dense connected layers, since they are at a higher level, they focus on learning specific features for the task and the user.

Algorithm 1: Training procedure of Proposed method

```
Input: Data from edge nodes $D_1, D_2, \dots, D_n$
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.
```

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

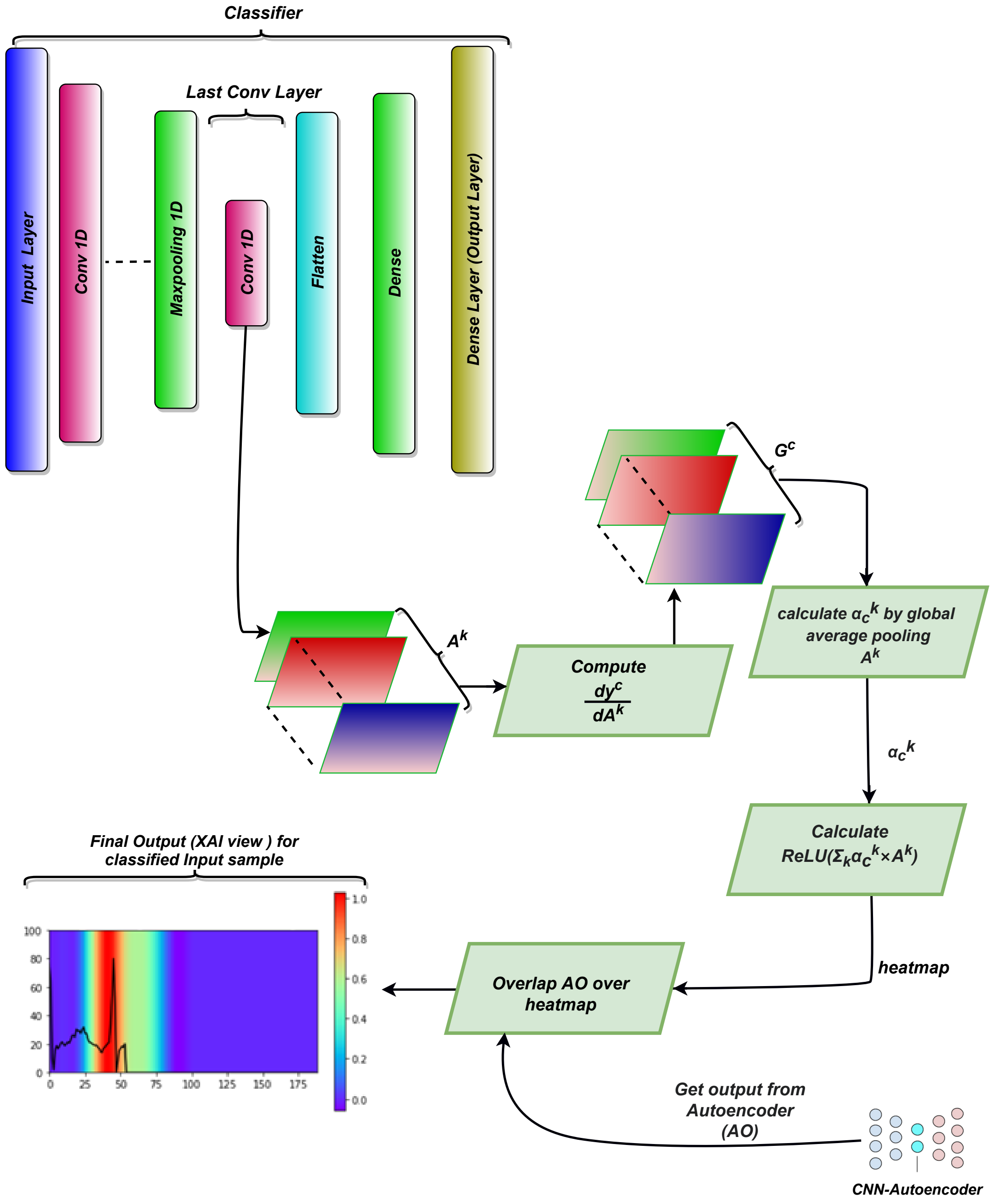

Classifier
Last Conv Layer
Input Layer
Conv 1D
Maxpooling 1D
Conv 1D
Flatten
Dense
Dense Layer (Output Layer)
G^c
A^k
Compute $\frac{dy^c}{dA^k}$
calculate α_c^k by global average pooling A^k
α_c^k
Calculate ReLU($\Sigma_k \alpha_c^k \times A^k$)
heatmap
Final Output (XAI view) for classified Input sample
100
80
60
40
20
0
0
25
50
75
100
125
150
175
1.0
0.8
0.6
0.4
0.2
0.0
Overlap AO over heatmap
Get output from Autoencoder (AO)
CNN-Autoencoder

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

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 clients 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 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 [52].

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

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 autotuner 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 updated and aggregated encoder part of AE to C and set the transfered 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.

4. Experimental Results

4.1. Dataset

For experimental purpose we used the widely used MIT-BIH Arrhythmia Database [23] 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 125Hz as the input. It should be noted that this dataset has unbalanced classes. Figure 6 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 7. Furthermore, this dataset is highly preprocessed, but in real-world scenarios the EEG 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 8.

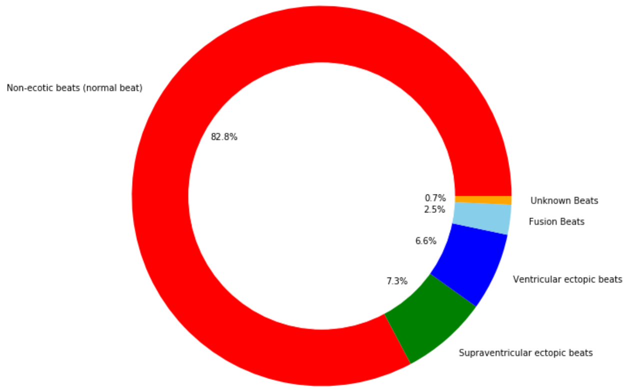


Figure 6: The distribution of the origional dataset

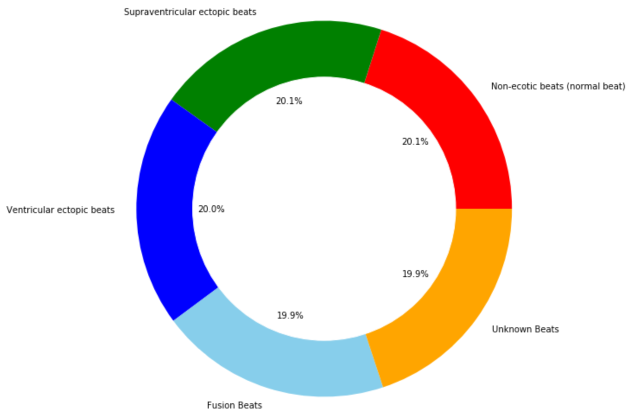


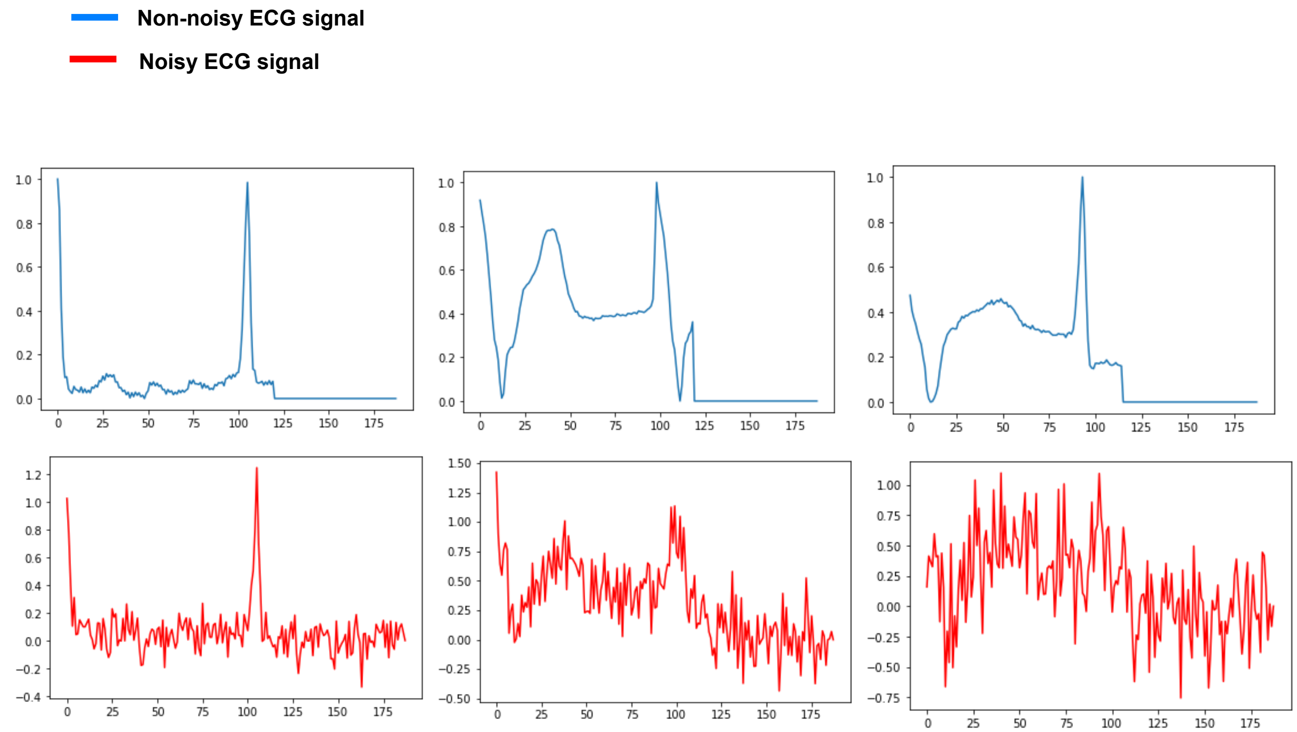
Figure 7: The distribution of the upsampled (re-balanced) dataset

4.2. Implementation Details

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 [46] initially trained their model at G_s, which may cause security risks in 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. Accuracy of each of the locally trained C was calculated by using the following equation:

$$A^i_{cc} = \frac{|x : x \in D_i \wedge y'(x) = y(x)|}{|x : x \in D_i|}. \tag{9}$$



10% noise **20% noise** **30% noise**

Figure 8: Comparison of the original and the noisy version of the dataset

5. Performance Analysis of the Proposed Method

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 Edge_1, Edge_2, Edge_3 and aggregated AE is given in Figure 9. It can be seen that 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 reconstruction MAE of each locally trained AE.

5.2. Classification Performance

Classification performance was measured using the four standard metrics found in the literature [53]: classification accuracy, precision, recall and F1-score. While accuracy 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. Accuracy, precision, recall and F1-score performance for each locally trained C in each of Edge_1, Edge_2, Edge_3 and aggregated C is given in Figure 10. It should be noted that the results shown in Figure 10 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 ± 0.99% accuracy. Other metrics, such as precision, recall and F1-scare, as shown in Figure 11. However, for real-time use we expect the data to be noisy, which is why we proceeded with the noisy data.

5.3. Explainability Using XAI Module

In order to achieve the explainability of the XAI module, it is important to understand the ECG signal [54]. 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 12. These regions play a vital role in ECG analysis [55]. The XAI module in the proposed framework shows that the proposed classifier looks at these critical features of the input sample. The outputs of the XAI module for different ECG samples are shown in Figure 13. 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, heatMaps should be cross-checked with the a prior expert knowledge of clinicians.

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

We compared our proposed framework with three state-of-the-art methods reported in 2020 [46, 35, 34]. It should be noted that other methods used the baseline MITBIH dataset (without noise), with which better accuracy results can be achieved. Contrastingly, we introduced (10%-30%) noise into the data to make it more realistic. Table 2 shows the

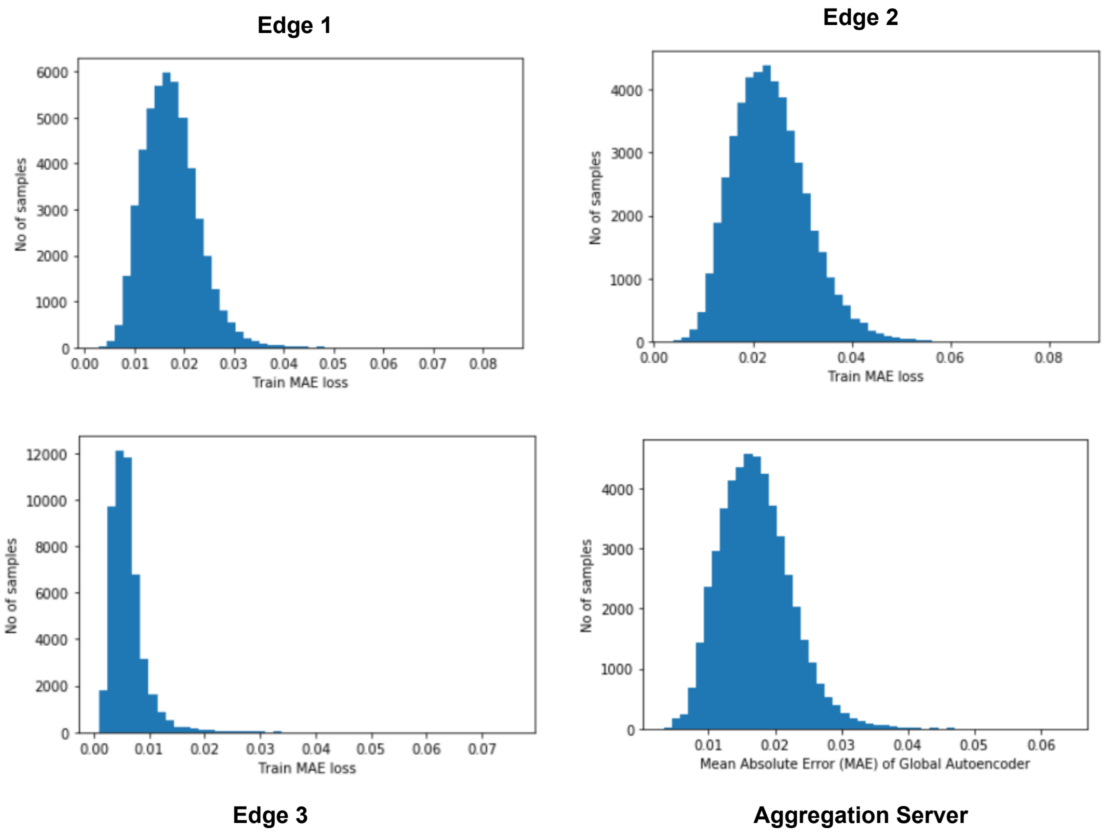


Figure 9: Reconstruction MAE

Table 2
Comparison with the state-of-the-art work

Scheme	XAI	Raw Input	Privacy	Realistic	Accuracy
[46]	✗	✗	✓	✗	99%
[35]	✗	✗	✗	✗	86.5%
[34]	✗	✗	✗	✗	93.19%
Proposed	✓	✓	✓	✓	94%/98.9%

comparison between our proposed method and the three other methods. As can be seen, our proposed method outperformed the methods in [35, 34]. While its accuracy is lower than the method reported in [46], 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.

6. Conclusion

In this paper, we propose a privacy-preserving, effective, efficient and explainable AI-based framework to address the limitations of deep learning applications for EEG signal classification. We propose a CNN-based autoencoder in a federated architecture to train a generalized denoising autoencoder while providing data privacy. Moreover, we propose a CNN-based classifier trained using transfer learning, local training and global aggregation to provide a very good classification accuracy (94% and 98% for noisy and clean data, respectively), compared to three selected state-of-the-art methods reported in 2020. Furthermore, we extended the usability of our framework by providing an explainable module on top of the classifier, which can be used to explain classification results. This makes the proposed framework a unique solution for real-world healthcare applications where ECG signal classification is an important task.

Acknowledgment

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 Com-

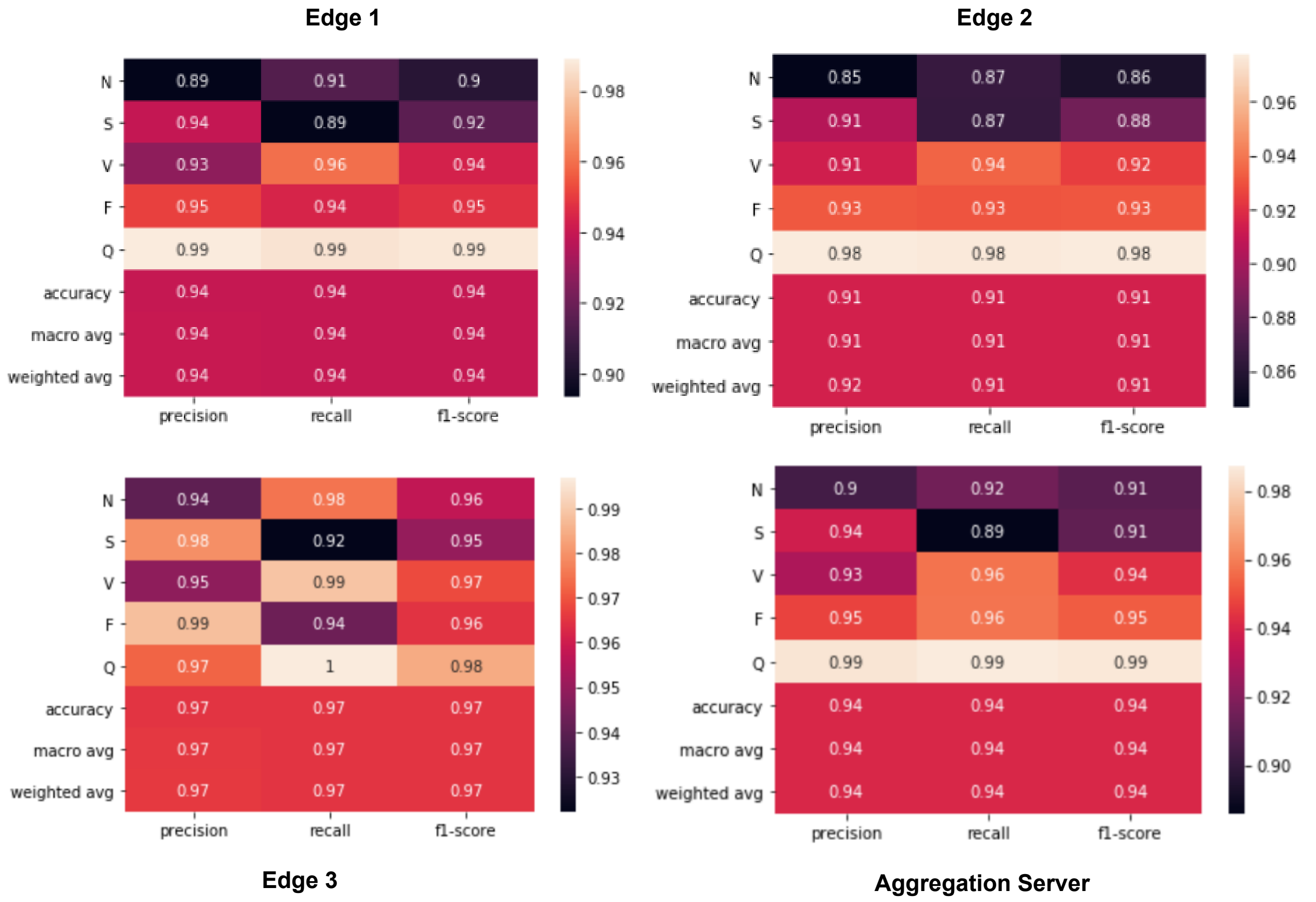


Figure 10: The classification performance of the proposed framework, with the noisy version of the dataset

munications 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.
URL https://doi.org/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) 106384doi:10.1016/j.asoc.2020.106384.
URL https://doi.org/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.
URL https://doi.org/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.
URL https://doi.org/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 (CsUR) 51 (6) (2019) 1–36. doi:10.1145/3295748.
URL https://doi.org/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.
URL https://doi.org/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: 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), IEEE, 2017, pp. 1–6. doi:10.1109/ICETCCT.2017.8280331.

[13] Z. Ji, Z. C. Lipton, C. Elkan, Differential privacy and machine learning: a survey and review (2014).

[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.
URL https://doi.org/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, Explainable artificial intelligence (xai), Defense Advanced Research Projects Agency (DARPA), nd Web 2 (2) (2017).

[17] W. Samek, T. Wiegand, K.-R. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models (2017).

[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] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-CAM: Why did you say that? (2016).

[20] B. Pyakillya, N. Kazachenko, N. Mikhailovsky, Deep learning for ecg

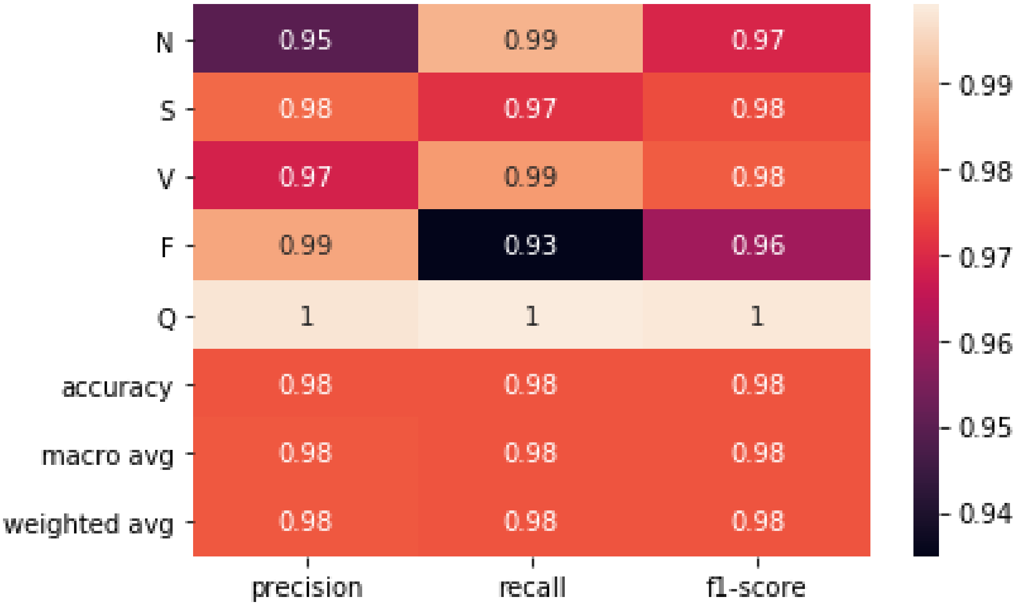


Figure 11: The classification performance of the proposed framework, with the original (clean) dataset

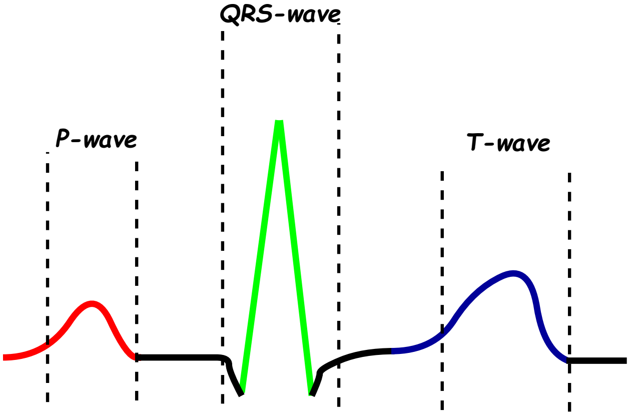


Figure 12: The major waves of a single normal ECG pattern

classification, in: Journal of physics: conference series, Vol. 913, IOP Publishing, 2017, p. 012004. doi:10.1088/1742-6596/913/1/012004.

[21] 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.

[22] 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) 103726doi:10.1016/j.compbiomed.2020.103726.

[23] 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

[24] 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.
URL https://doi.org/10.1093/gerona/62.8.844

[25] 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.

[26] 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.

[27] O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, IEEE communications surveys & tutorials 15 (3) (2012) 1192–1209.

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

[29] 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.
URL https://doi.org/10.1504/IJBET.2017.087722

[30] 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).

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

[32] C. Roopa, B. Harish, A survey on various machine learning approaches for ecg analysis, International Journal of Computer Applications 163 (9) (2017) 25–33.

[33] S. Sahoo, M. Dash, S. Behera, S. Sabut, Machine learning approach to detect cardiac arrhythmias in ecg signals: a survey, IRBM (2020). doi:10.1016/j.irbm.2019.12.001.
URL https://doi.org/10.1016/j.irbm.2019.12.001

[34] 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.

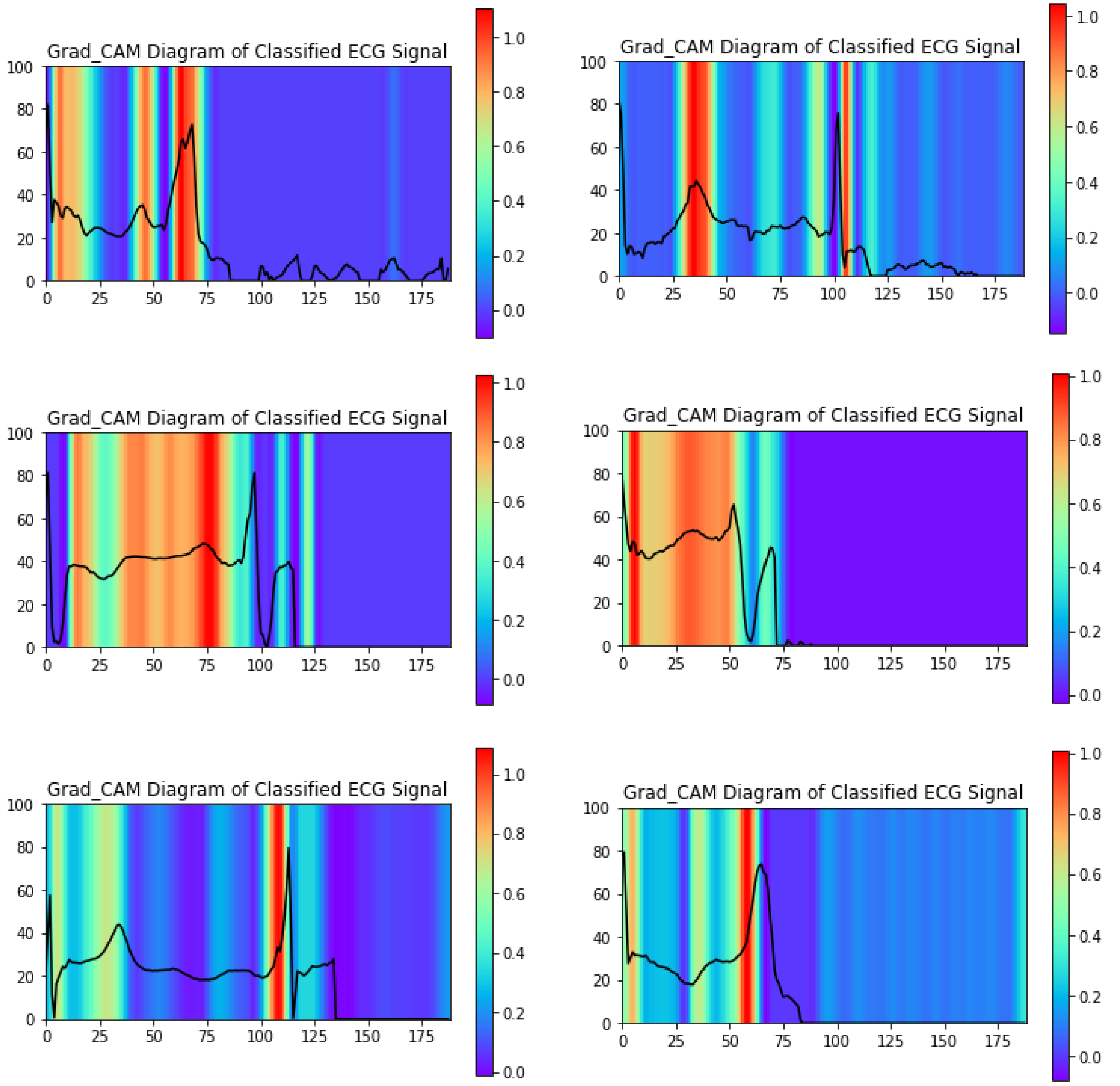


Figure 13: The outputs of the XAI module

doi:10.1016/j.cmpb.2020.105607.
URL https://doi.org/10.1016/j.cmpb.2020.105607

[35] 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. doi:10.3390/info11120549.
URL https://doi.org/10.3390/info11120549

[36] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, K. Sricharan, Recognizing abnormal heart sounds using deep learning (2017).

[37] 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.

[38] 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) 1–18doi:10.1007/s12652-020-02110-y.
URL https://doi.org/10.1007/s12652-020-02110-y

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

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

[41] 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) 1–37. doi:10.1145/3417987.
URL https://doi.org/10.1145/3417987

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

[43] 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.
URL https://doi.org/10.1007/s41666-020-00082-4

[44] 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.

[45] 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.

[46] 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.

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

[48] 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.

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

[50] F. K. Došilović, M. Brčić, N. Hlupić, Explainable artificial intelligence: A survey, in: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, 2018, pp. 0210–0215. doi:10.23919/MIPRO.2018.8400040.

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

[52] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[53] 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.

[54] M. B. Conover, Understanding electrocardiography, Elsevier Health Sciences, 2002.

[55] 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.
URL https://doi.org/10.1016/j.bspc.2018.03.003

Ali Raza received his BS degree in Computer Engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2017, his MS degree in Electronics and Computer Engineering from the Hongik University, Korea, in 2020. Between degrees, Ali has worked as associate researcher at the Information Security and Artificial Intelligence Lab, Gachon University, Korea. He is currently pursuing a cotutelle (dual award) PhD jointly funded by the University of Lille in France and the University of Kent, UK. He is currently studying at the Human Centered Design (HCD) team, GEMTEX, ENSAIT, French Grande Ecole in Roubaix. His research topic is smart healthcare with federated learning and data privacy. His research interests include cryptography, cyber-security, and machine learning.

Kim Phuc Tran is currently an Associate Professor in Automation and Industrial Informatics at the ENSAIT and the GEMTEX laboratory, University of Lille, France. He received the Engineering degree and the Master of Engineering degree in Automated Manufacturing. He obtained a Ph.D. in Automation and Applied Informatics at the Université de Nantes, France. His research works deal with Real-time Anomaly Detection with Machine Learning, Decision support systems with Artificial Intelligence, and Enabling Smart Manufacturing with IIoT, Federated learning, and Edge computing. He has published more than 55 papers in peer-reviewed international journals and proceedings of international conferences. He is the Topic Editor for the Sensors journal. In addition, as the project coordinator, he is conducting 1 regional research project about Smart Healthcare System with Federated Learning. He has been or is involved in 3 regional research and European projects. He is an expert for the Research and Innovation program of the Government of the French Community, Belgium.

Ludovic Koehl has been involved in a great number of projects dealing with optimization of the quality and comfort of textiles by integrating physical measures and human knowledge in the field of technical textiles, quality of textiles as well as new usage and consumers' behaviour studies for textile industry. Since 1998, published more than 92 papers (to March 2019). Research interests include pattern recognition, data mining, Big Data Mgt, Recommendation Systems, Traceability, computer modeling and their applications in textile industry.

Shujun Li (M'08–SM'12) is Professor of Cyber Security at the School of Computing and Director of the Instiute of Cyber Security for Society (iCSS), University of Kent, UK. He has published over 100 scientific papers, including 3 Best Papers and 1 Honorable Mention for Best Paper. His research interests are about inter-disciplinary topics related to cyber security and privacy, human factors, digital forensics and cybercrime, multimedia computing, and data science. Professor Shujun Li is on the editorial boards of 5 international journals and participated in the organization of more than 100 international conferences and workshops.