An Efficient and Private ECG Classification System Using Split and Semi-Supervised learning

Ahmad Ayad, Mahdi Barhoush, Marian Frei, Benedikt Völker
and Anke Schmeink

June 12, 2023

Implementation Details

The following sections show the research pipeline overview that we used in our paper [1] and the implementation details that are left unmentioned in the paper.

1 Research Pipline Overview

[1]- A. Ayad, M. Barhoush, M. Frei, B. Vo¨ lker and A. Schmeink, "An Efficient and Private ECG Classification System Using Split and Semi-Supervised Learning," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2023.3281977.

2 Test IoT setup

Table 1: Description of the hardware used in the experiments

	Client	**Server**
Description	Nvidia Jetson Nano Developer Kit	Dell Precision 3650
CPU	Quad-core ARM A57 @ 1.43 GHz	Intel Core i9-10900K 3,7 GHz
GPU	128-core Maxwell	Nvidia GeForce RTX 2080 Super 6GB
RAM	4 GB LPDDR4	64 GB DDR4

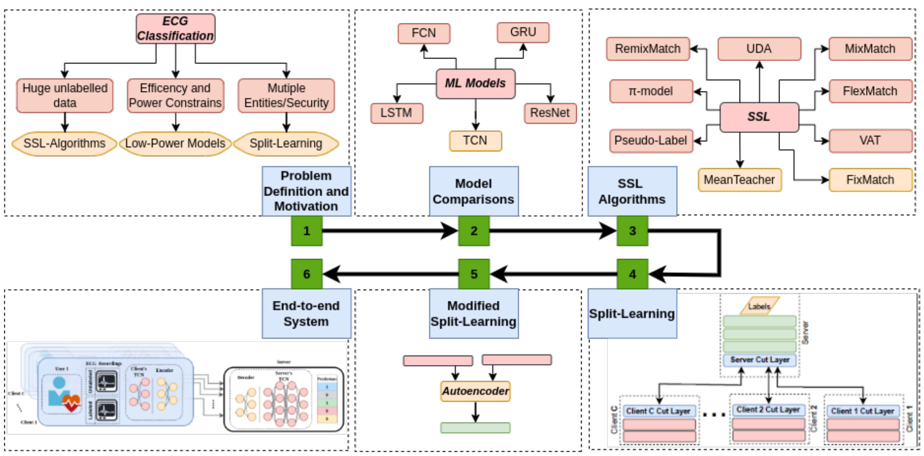

Figure 1: The ECG Pipeline Used in Our Paper [1]

3 SSL algorithms complexity

Table 2: Comparison of complexity between two SSL methods on the CIFR-10 data set to reach 80% accuracy [2]

SSL Algorithm	Time [Sec]	Memory [GB]	Batch Iteration	Time/Iteration [ms]	PFLOPS [10^6 GFLOPS]
MeanTeacher (MT)	540	9.88	6183	87.4	3.4
FixMatch (FM)	1341	24.55	4242	316.1	9.1
MT/FM Ratio	40%	40%	146%	28%	37%

[2]- M. Barhoush, A. Ayad, and A. Schmeink, "Semi-supervised algorithms in resource-constrained edge devices: An overview and experimental comparison," in IEEE Int. Conf. on Smart Data (SmartData), 2022.

4 Model Hyper-parameters

Table 3: Training Parameters and General Values

	Parameter	Value
Training	Optimizer	Adam
	Learning Rate (α, β_1, β_2)	(0.001,0.9,0.99)
	Batch Size	64
	Epochs	30
	Loss	Binary Cross Entropy
General Values	**Parameter**	**Value**
	Kernal Size (K_T)	5
	Number of Features (F_T)	11
	Number of Timestamps (T)	250
	Number of Classes (C)	5

Table 4: TCN5 Hyper-parameters, Block_0 to Block_2 reside in the Client's side, whereas Block_3 to Block_6 reside in the Server's Side

Block	**Layer**	**Parameter**	**Value**
Block_0	Input	Size	$T \times F_T$
	Padding	(Left, Right) Padding Size	$(K_T - 1, 0)$
	Conv1D	(Input,Output) Channels	$(F_T, F_T + 1)$
	Activation	Relu	
	BatchNorm1d	Size	L+1
Block_1 (TCN1)	–	Dialation1 (D1)	1
	Padding	(Left, Right) Padding Size	$((K_T - 1) \times D1, 0)$
	Conv1D	(Input,Output) Channels	$(L + 1, F_T)$
	BatchNorm1d	Size	F_T
	Activation	Relu	
	Dropout	Dropout Rate	0.3
	Padding	(Left, Right) Padding Size	$((K_T - 1) \times D1, 0)$
	Conv1D	(Input,Output) Channels	(F_T, F_T)
	BatchNorm1d	Size	F_T
	Activation	Relu	
	Dropout	Dropout Rate	0.3
Block_n (TCNn), $n = 2, 3, 4, 5$	–	Dialation (Dn)	2^{n-1}
	Padding	(Left, Right) Padding Size	$((K_T - 1) \times Dn, 0)$
	Conv1D	(Input,Output) Channels	(F_T, F_T)
	BatchNorm1d	Size	F_T
	Activation	Relu	
	Dropout	Dropout Rate	0.3
	Padding	(Left, Right) Padding Size	$((K_T - 1) \times Dn, 0)$
	Conv1D	(Input,Output) Channels	(F_T, F_T)
	BatchNorm1d	Size	F_T
	Activation	Relu	
	Dropout	Dropout Rate	0.3
Block_6	Flatten	Size	1
	Linear	(Input,Output) Features	$(F_T \times 1000, C)$
	Activation	Sigmoid	
	Output	Size	C

Table 5: Autoencoder Hyper-parameters,The Encoder part resides in the Client's side, whereas The decoder part resides in the Server's Side

Autoecncoder	Layer	(Input,Output)	Kernal Size	Stride	Padding
Encoder	Conv1D	$(T_T, F_T) = (11, 11)$	4	2	1
	Conv1D	$(8, 11)$	4	2	1
	Conv1D	$(11, 8)$	4	2	1
Decoder	ConvTranspose1D	$(5, 8)$	4	2	1
	ConvTranspose1D	$(8, 11)$	4	2	1
	ConvTranspose1D	$(T_T, F_T) = (11, 11)$	4	2	1