Weighted Convolutional Neural Networks Rare Electrocardiogram Detection for Real-Time Heart Monitoring

Colyn Jonker
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.h.c.jonker@student.utwente.nl

ABSTRACT

The electrocardiogram(ECG) plays a vital role to reduce the high mortality rate from cardiovascular diseases in computer-aided arrhythmia detection. Many arrhythmia classification research is devoted to developing classifiers that attain high prediction accuracies from the MIT-BIH Arrhythmia dataset. However, the complex variations and high imbalance in this dataset make this a demanding issue. Current state-of-the-art research achieves high overall accuracy in classifying regular ECG beats but receives less satisfactory results in the classification of rare classes. This research proposes to apply weights to the minority classes in the loss function of convolutional neural networks. The study will show that the proposed method can achieve high performance on rare ECG beat detection on embedded devices. The research will be compared to the state-of-the-art using the Supraventricular Ectopic beats(SVEB) and Ventricular Ectopic Beats(VEB) evaluation metrics, the research will be compared to the state-of-the-art. The best performing ECG classifier presented in this paper achieved an SVEB- and VEB-accuracy of 99.7% and 99.6%, respectively. The proposed classifier required around $2ms$ for classification per sample, which is suitable for real-time application.

Keywords

ECG classification, Convolutional Neural Network, Embedded Devices, Class Imbalance, Deep Learning, AAMI, Weighed Convolution Neural Network

1. INTRODUCTION

Cardiovascular diseases (CVDs) are the leading cause of mortality worldwide [26]. According to the World Health Organization, an estimated 17.9 million people died from CVDs in 2016. An electrocardiogram (ECG) is a measurement of electrical activity in the heart. An ECG is commonly represented as a graph with voltage on the y-axis versus time on the x-axis. All over the world, ECGs are used to detect numerous kinds of cardiac abnormalities. However, as manual identification of ECG irregularities is time and resource-consuming, there has been a call for automatic methods for analysing heartbeats.

35^{th} Twente Student Conference on IT July. 2^{nd}, 2021, Enschede, The Netherlands.

Heartbeat classification has been the subject of many types of research over the last two decades. The rise of deep learning is anticipated to provide promising results in the future of bioinformatics [15, 16]. Various methods for the automatic classification of ECGs have been proposed. The type of ECG beat can be distinguished by the use of auto-encoders [19, 27], discrete wavelet transform [2], support vector machines (SVM) [24, 20], neural networks [21], or other classification methods. Although the classification methods mentioned above achieve high overall prediction accuracy, these performances might be biased due to an extreme class imbalance present in the datasets used for training the models[12]. As a result, using accuracy as the sole evaluation metric might give a distorted view of the prediction performance on minority classes. Other methods rely heavily on data preprocessing, such as noise removal and feature extraction, which are computationally more expensive, thus limiting their use on low-cost embedded ECG devices. [22]

In general, for real-time arrhythmia detection classifiers using public datasets like the MIT-BIH Arrhythmia Database [8], we note that the number of regular beats is significantly more prevalent than abnormal beats. However, these minority classes are of most interest when it comes to arrhythmia detection. Within the ECG classifying research community, there are no general guidelines on the classes of ECG beats to be identified. As a result, many researchers use different classes and data splits, making it difficult to compare other ECG classifiers fairly. The variation in focus and classes in other research is likely caused by the large number of possible cardiac disorders derived from an ECG.

This paper proposed a novel method for dealing with class imbalance. The method involves altering the weights per class in the loss function of convolutional neural networks to increase the importance of abnormal ECG beats. A variety of different weighted methods will be evaluated and compared to the situation where no weights are applied. Furthermore, as accuracy might be a biased metric when dealing with imbalanced data, the method will be evaluated on other metrics to determine rare ECG class detection performance. Additionally, the performance of the classifiers will be tested on embedded devices to test the suitability of the classifier for real-time heart monitoring.

1.1 Research Questions

The research can be summarised by the following research question: *How could weighting minority classes in a CNN increase the classification performance of rare ECG heartbeats on embedded devices, given an imbalanced dataset?*

This question can, in turn, be answered by the following

sub-questions(SQ).

- *SQ1: Which weighting methods can be used to achieve better performance for rare classes in a convolutional neural network?*
- *SQ2: What metrics give a more concise overview of the classifier's performance on an imbalanced dataset?*

2. BACKGROUND

This section will first explain the concepts of the machine learning technique of convolutional neural networks. After that, an introduction to the topic of class imbalance and existing techniques will be made. Finally, a comparison to state-of-the-art literature on ECG analysis will be made.

2.1 Convolution Neural Networks

A neural network(NN) is a machine learning model that closely resembles a biological neural network. The NN is build up out of multiple connected neurons in a layered structure. Convolutional neural networks are a class of neural networks and are most commonly applied to visual imagery. Typically, the CNN consists of an input layer, several convolutional layers with possibly some pooling layers, and fully connected layers, including the output layer. Unlike fully connected layers, convolutional layers are only connected to a small receptive field of its input, where the weights of its connections define a *filter bank* [10]. A convolutional operation is a process of sliding this filter bank across the layer's input, producing activations at each receptive field to create a *feature map* of the input. Combining multiple of these feature banks in a convolutional layer, the model can learn to detect specific features in the input and pass these on to the subsequent layers. Pooling layers are added after one or more convolutional layers to merge semantically similar features and reduce dimensionality [29]. Generally, after these layers, the output is flattened and fed to fully connected layers for classification. A schematic overview of a basic convolutional neural network is shown in figure 1.

More recently, convolutional neural networks also have been applied with 1-dimensional signals such as time series, financial data and ECG analysis. Kiranyaz *et al.*[11] were the first to use these 1-dimensional convolutions in the field of ECG detection.

2.2 Class Imbalance

The most prominent methods for addressing the class imbalance in datasets can be split into two categories; data-level and algorithm-level methods [10]. In data-level methods, techniques such as over- and under-sampling can be applied to adjust the number of samples of the majority class. Oversampling for minority classes entails the repetition of samples associated with the minority classes. On the other hand, under-sampling removes samples from the majority class to balance out the dataset.

Contrary to data-level methods, the algorithm-level methods aim to adjust the loss function in the training stage of the model to overcome the balance. One way of doing so is by altering the loss function to give more importance to minority classes. This can be achieved by setting weights per class. By using this cost-sensitive learning, the neural network may learn features of the minority class more. It is a proven method[18] of dealing with imbalanced datasets and has been applied to many classifier models, such as k-Nearest Neighbour, SVMs and decision trees. However, throughout our literature study, we have not seen this concept be applied to convolutional neural networks.

Although either of the two data-level methods balances out the dataset, they do not directly tackle the issues caused by class imbalance. Instead, it risks introducing new problems. For example, oversampling introduces duplicate samples, which can slow down training time and cause the overfitting of the model. On the other hand, under-sampling can cause the model to miss out on learning certain features that it could have learnt from. For our research, an algorithm-level method is more appropriate as this leaves out the task of data preprocessing, which is more suitable for computationally limited devices [3].

2.3 Related work

This section will go over some of the related work in the automatic analysis of ECG signals.

With the rise of new machine learning techniques, there has been a need for powerful computational techniques that can maximise the information extracted from the ECG data [14]. Rahhal *et al.* [19] observed that the feature representation of ECG signals in most state-of-the-art methods relied on handcrafted features. They found an automatic way of doing this using Denoising Auto-Encoders(DAE).

The work of Acharya *et al.* [1] found that convolutional neural networks could also efficiently be used in ECG classification.

In the 2019 work of Sellami *et al.* [22], a convolutional neural network was realised to classify heartbeat abnormalities. In their study, they were able to address the imbalanced nature of the MIT-BIH [8] database by using a dynamic batch-weighted loss function for the deep learning CNN. As a result, they achieved an accuracy of 88.34% on classifying the five classes of heartbeats defined by the Association of the Advancement of Medical Instruments (AAMI).

The 2021 work by Wang *et al.* [25] addressed the imbalance issue by using the Synthetic Minority Oversampling Technique (SMOTE) in combination with feature selection. As a result, their random-forest classifier reached an accuracy of 98.68% on the UCI arrhythmia dataset. Likewise, Xiaolin *et al.* [28] used SMOTE to balance the classes in training data. The synthesised data was then passed down to train a deep learning convolutional neural network.

The work of Lima *et al.*[13] shows a novel approach that uses Generative Adversarial Networks(GAN) to synthesise rare classes from the MIT-BIH database, which were then augmented back to the training dataset to balance the classes. Their method has shown to outperform traditional oversampling techniques. However, they were unable to receive satisfactory results in the classification of minority classes.

Gao *et al.*[7] have had great success in the detection of arrhythmia given imbalanced datasets. Their work down-weights easily identified regular ECG beats by using a focal-loss (FL) function. In addition, the adoption of such a function avoids the reduction of effective information caused by the under-sampling method or the increase of total network training time with oversampling techniques.

3. ECG CLASSIFICATION

This section will cover the details of the MIT-BIH dataset, including its records, the frequency of these records, and the method of splitting the data for model training. In addition, the section starts with an introduction to existing standards in the field of ECG classification.

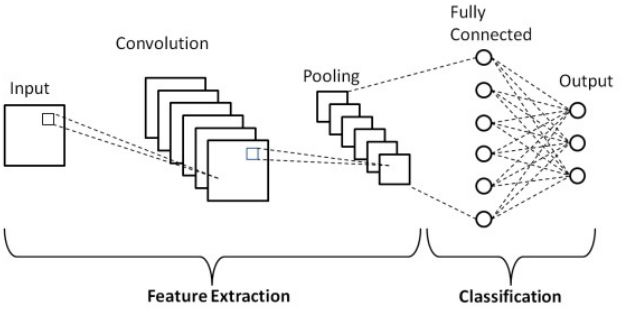


Figure 1: A schematic overview of a basic convolutional neural network structure

3.1 Classes

Classifying ECG beats is often challenging as there are a multi-fold of categories we could identify heartbeats as. In addition, there are sometimes subtle differences in heartbeat classes. According to the Association for the Advancement of Medical Instruments, or AAMI, we can categorise these annotated beats into five superclasses. These classes are Normal(N), Supraventricular(S/SVEB), Ventricular(V/VEB), Fusion of normal and ventricular(F) and Unknown beats(Q) [6]. This study mainly focuses on the SVEB and VEB class because these might indicate possible cardiac irregularities [17]. Nonetheless, samples of the other two remaining classes will be used for both training and testing as these data points can increase the learning of the classifier.

3.2 Materials

The MIT-BIH Arrhythmia Database consists of 48 annotated records collected from 47 clinical patients[8]. Each record is approximately 30 minutes long and sampled at 360 Hz by a 0.1-100 Hz bandpass filter. There are more than 109,900 annotations per beat originating from 16 heartbeat categories. As mentioned earlier, classifying heartbeats can be pretty challenging, and therefore the entire dataset is annotated by multiple cardiologists. In addition, 23 of the 48 records belong to the "100" series, containing samples of the general population. On the contrary, the "200" series include relatively more abnormal categories of heartbeats [8].

In table 1, an overview is shown of the frequency of all beat annotations(BA) present in the dataset. These beat annotations are then categorised according to the AAMI standards, as explained in section 3.1. The data has been split randomly into a training, testing and validation set. All the 44 records are divided into 80% training and 20% testing sets. Another 25% split from this training set splits the training into a training and validation set. This ensures a global 60/20/20% data split. In both training, testing, and validation sets the number of samples from each of the five AAMI classes also followed the proportions mentioned above. Throughout this research, we will denote this data split as A. Throughout the existing ECG classification literature, some researchers followed the notion of splitting the data by the individual records of the MIT-BIH database. To make a fair comparison, we will follow this split proposed by Chazal *et al.*[5] to train the classifier with a training and testing set. They split the data into the following two sets; $DS1$ = {101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230} and $DS2$ = {100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234}. Throughout this research, we will denote this data split as B.

4. METHODOLOGY

This section will cover the methodology taken throughout the research. It will give insights into choices of the classifier architecture, signal preprocessing, loss- and weight functions, and finally, the evaluation metrics.

4.1 Signal

To properly train the classifier, we must find a proper window size, which can be more complicated than expected. The reason for this is the variance in cardiac cycles of each record. Some patients heartbeats last $650ms$, whereas others can last up to $1500ms$. Therefore, we have chosen to take a window of $2.84s$. Since the ECG signals are sampled at $360Hz$, we have approximately $2.84 \cdot 360 \approx 1024$ measurements per window. As our study focuses on the weighting methods in convolutional neural network models on embedded devices, we have opted not to pre-process the data. Instead, we take the raw ECG signal and put it straight into the model.

4.2 Weighting loss

Several different methods have been used and evaluated to test the effects of adjusting the weights in the loss function. Firstly, a study was done on existing weighting methods used in other types of classifiers, such as SVMs, k-Nearest Neighbour and decision tree models. From this study, the following weighting methods have been implemented in Python to calculate the respective weight per class c. In the following equations, we will denote the weight per class c as w_c. Furthermore, the number of labelled samples in class c is denoted as n_c, the total number of samples denoted as n, and the total number of classes represented as a.

First, a test has been conducted with no weighting method applied. This is used as a baseline to evaluate the effect of weighting versus not weighting class respective to their number of samples. Next, we study the case where the weighting methods assigns the class weight inversely proportional to their respective frequencies. This method implicitly balanced out the dataset in the training of the classifier. In equation 1, the balanced weighting methods are shown.

$$\text{Balanced: } w_c = \frac{n}{a \cdot n_c} \tag{1}$$

Two other popular weighting methods are the Inverse Number of Samples(INS) and the Inversely Squared Number of Samples(ISNS). Their calculations are shown in equations 2 and 3, respectively.

$$w_c = \frac{1}{n_c} \tag{2}$$

AAMI	BA	Description	Frequency	Count
Normal	N	Normal Beat	68.25%	75052
	L	Left bundle branch block beat	7.34%	8075
	R	Right bundle branch block beat	6.85%	7529
	e	Atrial escape beat	0.01%	16
	j	Nodal (junctional) escape beat	0.21%	229
Supraventricular	A	Atrial premature beat	2.32%	2546
	a	Aberrated atrial premature beat	0.14%	150
	S	Supraventricular premature beat	0.00%	2
	J	Nodal (junctional) premature beat	0.08%	83
Ventricular	V	Premature ventricular contraction	6.48%	7130
	!	Ventricular flutter wave	0.43%	472
	E	Ventricular escape beat	0.10%	106
Fusion Beat	F	Fusion of ventricular and normal beat	0.73%	803
Unknown Beats	/	Paced beat	6.39%	7028
	f	Fusion of paced and normal beat	0.89%	982
	Q	Unclassifiable beat	0.03%	33

Table 1: Frequency of the classes in the MIT-BIH dataset.

$$w_c = \frac{1}{\sqrt{n_c}} \tag{3}$$

Generally, the INS performs poorly in real-life situations with high-class imbalance [4]. Therefore, Cui *et al.* have developed a variant on the Inverse Number of samples. Their paper argues that as the number of samples increases, the added benefit of that new data point diminishes. Thus there exists the notion of an effective number of samples(ENS). The ENS is defined as the volume of samples and can be calculated by a simple formula, as shown in equation 4. In this equation, E_c denotes the effective number of samples for class c, and β is a hyper-parameter ranging from $[0, 1)$. Per the suggestion of the author, we will experiment with the β values of 0.99 and 0.999.

$$w_c = \frac{1}{E_c}$$
$$E_c = \frac{1-\beta^{n_c}}{1-\beta} \tag{4}$$

where $\beta \in [0,1)$

4.3 Classifier Architecture

The basic convolutional neural network model was implemented in Python using Tensorflow and Keras. To find the best performing model, several classifiers with different weighting methods have been trained and evaluated. The network will consist of four convolutional layers that are fed into a max-pooling layer per convolutional block. Then a dropout function is used to avoid over-fitting the model. All layers will be activated by a Rectified Linear Unit(ReLU) function. Finally, the output of the convolutional blocks will be passed onto one fully connected layer and from there to the interpretation layer, which maps the weights to the respective class. The interpretation layer uses a Soft-Max as its activation function.
Given the limited time of this research, only a categorical cross-entropy function has been tested for use as the loss function. The optimizer to be used is Nadam, as this is an effective gradient descent optimization algorithm to calculate adaptive learning rates for different parameters. Overall, Nadam performs better than other gradient descent optimization methods in practical applications [23]. The categories are numerically numbered from 0 to 4, where the numbers represent the Normal, Supraventricular, Ventricular, Fusion and Unknown class, respectively. In addition, we used 5-fold cross-validation(CV) to tune the hyper-parameters of the convolutional neural network.

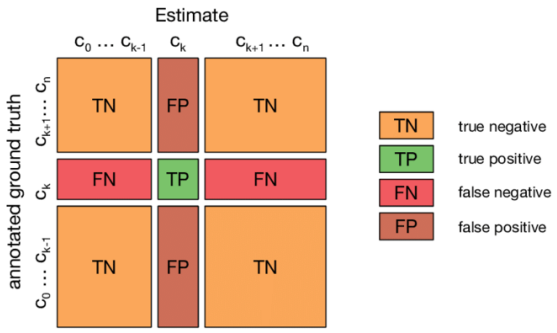


Figure 2: Confusion matrix for multi-class classification

4.4 Evaluation metrics

To evaluate the classifier, we will use the SVEB- and VEB-evaluation metrics as introduced by [17]. The supra-ventricular and ventricular classes are most indicative of potential health risks. The following five evaluation metrics will be used per class: accuracy, positive predictivity, recall, specificity and F1 score. Since the same metrics have been used in other state-of-the-art research, the best performing classifier can be fairly compared to the others. These metrics can all be calculated from the confusion matrix, as shown in figure 2. The FP and FN errors correspond to Type I and Type II errors, respectively. In this context, the Type I error is the case where arrhythmia was detected but was not present. On the contrary, the Type II error represents the situation where an actual arrhythmic beat occurred but was not detected.

Accuracy, as shown in equation 5, is one of the most frequently used metrics when evaluating the performance of classifiers. However, when working with skewed data distributions, the accuracy is dominated by the majority class. Suppose the dataset having a 1% distribution of abnormal heartbeats versus a 99% distribution of normal heartbeats, a classifier can achieve an accuracy of 99% by classifying all beats as normal. Because of the latter, there

Hyper-parameter	Values tested	Best
Dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5	0.2
# epochs	5, 10, 15, 30, 60, 90	30
Batch size	8, 16, 32, 64, 128	32

Table 2: Hyper-Parameter tuning

is a need for other metrics.

$$Acc(\text{accuracy}) = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

Positive predictivity, recall, and specificity have the added benefit of being less biased in evaluating performance in imbalanced datasets. The calculations for these metrics are given in equation 6:

$$\begin{aligned} Re(\text{recall}) &= \frac{TP}{TP + FN} \\ Pp(\text{positive predictivity}) &= \frac{TP}{TP + FP} \\ Sp(\text{specificity}) &= \frac{TN}{TN + FP} \end{aligned} \tag{6}$$

The *F-Measure* is the harmonic mean of precision and recall and can be defined as follows, where the coefficient β is used to adjust the relative importance of the precision versus recall. In our study, we will look at the case where $\beta = 1$, which gives us the *F1-score*. The F-measure can be calculated as follows:

$$F - measure = \frac{(1 + \beta^2) \cdot RE \cdot PR}{(1 + \beta) \cdot RE \cdot PR} \tag{7}$$

4.5 Deployment

To find out the suitability of the classifier for real-time heart monitoring, we tested all trained classifiers on a Raspberry Pi 3B+. The Raspberry Pi consists of a Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit chip clocking at 1.4GHz. It has 1Gb of RAM. During each classification, classification time and energy consumption has been recorded. The latter has been measured by using a USB device that computes the current flowing through the device.

5. EXPERIMENTS

5.1 Setup

Due to the limited amount of time for this research, the most optimal configurations for the classifier architecture have not been studied extensively. E.g. when investigating the optimal number of convolutional layers only 3 and 4 layers have been evaluated. It is possible that using more convolutional layers would have a better effect on gathering ECG signal features. Furthermore, hyper-parameter tuning has only been used for models using training set A. Models trained with set B in this research have not been optimized.

As mentioned in section 4.3, 5-fold cross-validation has been used to optimize the classifier's performance by adjusting the hyper-parameters. The goal of tuning the hyper-parameters was to attain the highest possible total model training accuracy. Firstly, the number of epochs the model has trained for has been tuned. Using early stopping, the stage at which the loss value per epoch is not increasing, the network will stop continuing with training. In the experiment, the network converged to its highest attainable categorical accuracy at epoch 30. Finally, we tuned the dropout layer starting from 0.0 and taking steps of 0.1 up towards a value of 0.5. This test yielded a dropout value of 0.2.

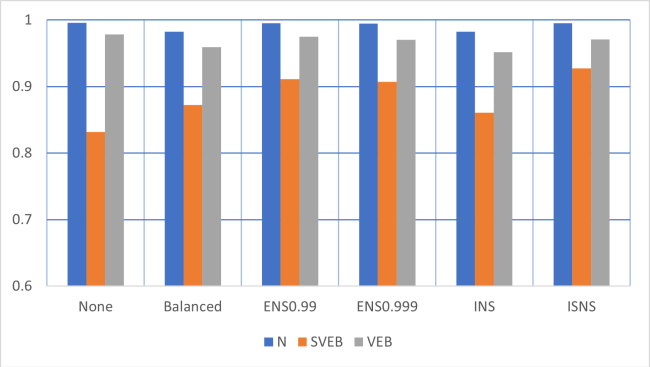


Figure 3: F1 scores of different weighting methods per AAMI class from models created with training set A

5.2 Results

In table 3 the results for determining the optimum weighting methods are shown. The table uses the four common metrics for SVEB and VEB detection to show the influence of weighting the loss function during classifier training. To begin, the performance per weighting method is very diverse. It can be seen that the ENS with $\beta = 0.999$ excels in attaining high positive predictivity rates and specificities. On the other hand, no weights in the loss function proved to result in the least amount of precision for SVEB classes. The latter becomes apparent when looking at the confusion matrices for two classifiers trained during the experiments. Figure 4a presents the confusion matrix of the case where no weights were assigned. Upon first glance, it is easily seen that the model is biased towards the normal heartbeat class, where Type-II errors frequently occur. In figure 4b, ENS0.999 weighting has been applied. Here, the effect of weighting shows its merits with a tremendous amount of reduction in Type-I and Type-II errors.

In figure 3, the F1 scores are shown per class; N, SVEB, and VEB, per weighting method, used. As one might expect, in the case of no weighting being applied, the F1 score for normal classes are significantly higher than those of the SVEB class. One unexpected outcome is the overall high F1 score for VEB class detection, retaining its F1 score of $\approx 97\%$ throughout each classifier. Using the inversely squared number of samples had by far the highest F1 score for the SVEB class, prompting us to believe it to be one of the best weighting methods to be used.

Table 6 shows the power consumption and the classification time. Here, the power consumption is measured as the variance in current measured when running the classifier for one sample. Furthermore, the time it takes to classify one sample has been measured and recorded. Throughout the experiments, the classification time on average lies around *2ms* and the average power consumption around *2-3 mAh*. In addition, each model has 135.835 parameters and takes up a total disk size of 220 kB.

6. DISCUSSION

Table 3 and figures 3 and 2 show that the classifier struggles with the Supraventricular class. This trend can be seen throughout all other state-of-the-art research. Even though the classifier presented in this paper uses more samples from this to learn from, it still receives relatively low F1 and positive predictivity scores compared to the

Weighting method	Acc	SVEB				VEB				Acc
	(training)	Acc	Pr	Re	Sp	Acc	Pr	Re	Sp	(testing)
None	0.997	0.991	0.793	0.852	0.995	0.994	0.970	0.945	0.998	0.993
Balanced	0.988	0.993	0.813	0.941	0.995	0.994	0.935	**0.986**	0.995	0.988
ENS ($\beta = 0.99$)	0.995	0.996	0.936	0.888	**0.999**	0.996	0.972	0.977	0.998	0.996
ENS ($\beta = 0.999$)	0.997	0.996	**0.978**	0.846	**0.999**	**0.997**	**0.979**	0.962	**0.999**	0.996
INS	0.990	0.993	0.786	**0.951**	0.994	0.993	0.938	0.965	0.995	0.988
ISNS	0.973	**0.997**	0.933	0.922	0.998	0.996	0.965	0.976	0.997	0.996

Table 3: Comparison of common SVEB- and VEB-evaluation metrics with different weighting methods used in the loss function using training set A.

Weighting method	Acc	SVEB				VEB				Acc
	(training)	Acc	Pr	Re	Sp	Acc	Pr	Re	Sp	(testing)
None	0.967	**0.962**	**0.123**	0.001	**0.999**	0.971	**0.793**	0.756	**0.986**	0.919
Balanced	0.501	0.867	0.061	**0.178**	0.893	0.898	0.284	0.381	0.934	0.487
ISNS	0.934	0.960	0.001	0.001	0.997	0.927	0.467	**0.902**	0.929	0.846
ENS ($\beta = 0.999$)	0.949	0.961	0.038	0.001	0.997	**0.973**	0.766	0.848	0.982	0.910

Table 4: Comparison of common SVEB- and VEB-evaluation metrics with different weighting methods used in the loss function using training set B.

other classes. In all the records of the MIT-BIH dataset, the Supraventricular class only consists out of approximately 2800 samples, where we argue that given more samples, the performance would increase for this class as well. There exists another dataset, the MIT-BIH Supraventricular Arrhythmia Dataset, which contains more data points of this class. This dataset can be used to enhance the performance of the Supraventricular class. Overall, accuracy is high throughout each class. However, from figure 4, it can be observed that this is not the case. Our experiments support the statement made in section 4.4 about accuracy being biased. The most noticeable result of this study is the general improvement of all the metrics; Accuracy, positive predictivity, recall and specificity, when using weighting. Although not weighing the classifier can achieve high specificity's, using weighting methods balances out the performance in all metrics resulting in generally higher F1 scores, as presented in figure 3.

Table 4 shows the performance of some models trained using the training and testing set of B. Given the limited research time, not all weighting methods proposed in the research have been tested and the models tested have not been tuned. As seen from the table, the weighting methods do not affect the overall score in almost all metrics. Only the ISNS and balanced weighting methods can improve the recall in the SVEB and VEB classes. Another critical difference in the models trained in table 3 and table 4 is that the latter has more fluctuation in overall training and testing accuracy. The difference between training and testing accuracy in table 3 ranges around $0.1 - 2.3\%$, whereas in the models trained with B range around $2.4 - 8.8$. This suggests that most of these trained classifiers are over-fitted.

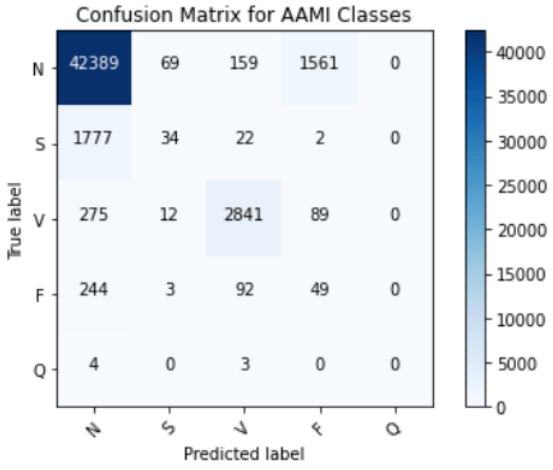
Another key element of the study is the suitability of the classifier on embedded devices. From table 6, it can be observed that the weighting methods do not influence the classification time nor the power consumption. Intuitively, this makes sense, as the total amount of parameters in the classifier remained unchanged. Therefore, the different power consumption for the ENS weighting method may have been caused by human measurement errors. Nonetheless, the experimental results show that this classifier is suitable for embedded devices with less computational power and battery.

Table 5 compares the best performing classifier from our research to other state-of-the-art research. However, this comparison is biased because of the different training and testing sets used in the classifiers. Generally, amongst research in ECG classification, researchers split the data into two sets, each containing distinct records of the MIT-BIH database. As is the case in A, the training and testing set are entirely separate, with no sample of the testing set existing in the training set. Nonetheless, our high performance in models trained with A may be a result of a more significant amount of samples of each class being present than compared to the state-of-the-art research. E.g., in training set A, the number of samples for the Supraventricular class is approximately 1700, whereas, in B, 940 samples of Supraventricular classes have been used for training.

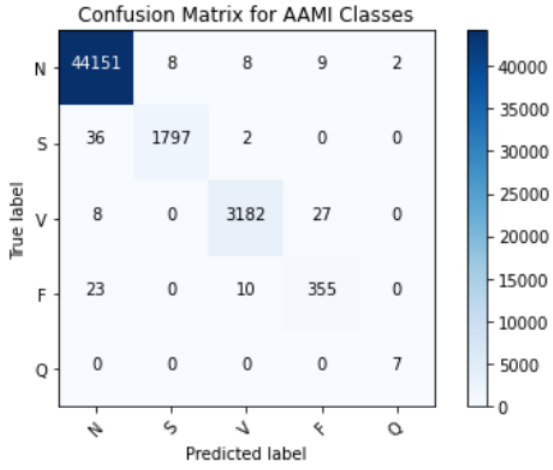
From table 3 and figures 4 and 3, we can conclude that the Effective Number of Samples with hyper-parameter $\beta = 0.999$ and ISNS excel in the performance of supraventricular and ventricular classes. We argue that the ISNS yielded the best classifier for real-time heart monitoring. Although comparatively lacking slightly in positive predictivity and specificity, the higher recall scores in the Supraventricular and Ventricular class make the classifier much more consistent. The reason for this lies in the context of the problem of arrhythmia detection on an embedded device. We argue that a higher recall is slightly more important than having higher positive predictivity rates as the consequences of not identifying the arrhythmia are larger than falsely predicting arrhythmia. Generally, in

Method	Data split	SVEB				VEB			
		Acc	Pp	Re	Sp	Acc	Pp	Re	Sp
Kiranyaz[11]	*B*	0.976	0.635	0.603	0.992	0.990	0.906	0.939	0.989
Heinen [9]	*B*	0.968	0.679	0.586	0.963	0.981	0.213	0.069	0.991
Xia et al[27]	*B*	**0.997**	**0.956**	**0.961**	**0.999**	**0.995**	**0.973**	**0.979**	**0.997**
Sellami et al[22]	*B*	0.924	0.304	0.820	0.928	0.972	0.721	0.921	0.975
This paper	*B*	0.962	0.123	0.001	**0.999**	0.971	0.793	0.756	0.986
This paper	*A*	**0.997**	0.933	0.922	0.998	**0.996**	0.965	0.976	**0.997**

Table 5: Comparison of the common SVEB- and VEB-evaluation metrics to other state-of-the-art research



(a) No weighting



(b) ENS0.999 weighting

Figure 4: Confusion matrices of different weight-loss functions

Weighting method	Power consumption	Classification time
None	2 mAh	2.1ms
ENS0.99	3 mAh	2.1ms
ENS0.999	3 mAh	2.1ms
INS	2 mAh	1.9ms
ISNS	2 mAh	2.1ms

Table 6: Overview of power consumption and classification time per sample per trained classifier

our research the goal is to reduce Type-II errors as when it comes to matters of the heart, people tend to be more cautious.

There is a lot of room for improvement in the classification of rare ECG signals. In our model, we fed the classifier with raw ECG data. However, research has shown that denoising can improve overall performance[27]. Another possible enhancement could be to use feature extraction, as this has helped increase learning[7]. Future studies can be held on the effect of class weighting in other loss functions, as in this study, only cross-entropy has been studied. The extreme difference between recall and precision in tables 3 and 4 suggest that there is an effective amount of training samples needed for the neural network to learn from. Applying the same weighting schemes did not seem to improve the SVEB- and VEB-evaluation metrics with the lower amount of training samples for these rare classes in set *B*. This also poses the question if there is a trade-off between the number of samples and the relative weight of the class. Another interesting topic for future research could be the effect of training the classifier on only the Supraventricular, Ventricular and Normal heartbeat classes. Then feeding this classifier samples from the Unknown and Fusion classes to see what type of AAMI class the classifier predicts from these abnormal beats. Finally, as shown by Heinen[9] and Lima[13], 2D CNNs can be used to classify heartbeats. Research can be done on the effect of weighting in convolutional neural networks that take 2-dimensional inputs.

7. CONCLUSION

In this paper, a novel approach has been presented for weighting convolutional neural networks in the context of ECG analysis. The novel method makes use of assigning weights to the less represented classes in the loss function of the classifier. The study focuses on decreasing the bias towards the majority class in convolutional neural networks trained with skewed data distributions. Early

results show that all the presented weighting methods in the paper outperform the case where no weighting is applied. One of the major shortcomings of this research is the slightly poorer performance of the Supraventricular class, however, this can be improved with feature extraction or more training data. To determine if the novel approach is of value, future research should be done on weighted 2-dimensional convolutional neural networks and the trade-off between the number of samples and the class weight.

8. REFERENCES

[1] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam. Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. *Information Sciences*, 415-416:190–198, 2017.

[2] D. Azariadi, V. Tsoutsouras, S. Xydis, and D. Soudris. ECG signal analysis and arrhythmia detection on IoT wearable medical devices. *2016 5th International Conference on Modern Circuits and Systems Technologies, MOCAST 2016*, pages 7–10, 2016.

[3] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[4] Y. Cui, M. Jia, T. Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:9260–9269, 2019.

[5] P. De Chazal, M. O'Dwyer, and R. B. Reilly. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, 51(7):1196–1206, 2004.

[6] A. for the Advancement of Medical Instrumentation. Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms, 1998.

[7] J. Gao, H. Zhang, P. Lu, Z. Wang, and L. Zou. An Effective LSTM Recurrent Network to Detect Arrhythmia on Imbalanced ECG Dataset. 2019.

[8] R. M. G.B. Moody. MIT-BIH Arrhythmia Database. `https://physionet.org/content/mitdb/1.0.0/`, 2005. [Online; accessed 16-06-2021].

[9] N. Heinen. Using lightweight image classifiers for electrocardiogram classification on embedded devices. *Twents Student Conference on IT 34th*, 2020.

[10] J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 2019.

[11] S. Kiranyaz, T. Ince, and M. Gabbouj. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering*, 63(3):664–675, 2016.

[12] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1), 2018.

[13] J. L. Lima, D. Macêdo, and C. Zanchettin. Heartbeat anomaly detection using adversarial oversampling. *arXiv*, (July):1–7, 2019.

[14] A. Lyon, A. Mincholé, J. P. Martínez, P. Laguna, and B. Rodriguez. Computational techniques for ECG analysis and interpretation in light of their contribution to medical advances. *Journal of the Royal Society Interface*, 15(138), 2018.

[15] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.

[16] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley. Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6):1236–1246, 2017.

[17] R. B. R. P. De Chazal, M. OĎwyer. Automatic classification of heartbeats using ecg morphology and heartbeat interval features, 2004.

[18] J. L. Polo, F. Berzal, and J. C. Cubero. Weighted Classification Using Decision Trees for Binary Classification Problems. *IV Taller de MinerÌa de Datos y Aprendizaje [TAMIDA 2007]*, pages 333–341, 2007.

[19] M. M. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, and R. R. Yager. Deep learning approach for active classification of electrocardiogram signals. *Information Sciences*, 345:340–354, 2016.

[20] M. Reethika, A; Usha Kumari, Ch; Ankita, R; Pavani, T; Arun Vignesh, N; Tarun Varma, N; Aqeel Manzar. Heart Rhythm Abnormality Detection and Classification using Machine Learning Technique. (Icoei):580–584, 2020.

[21] F. A. Rivera Sánchez and J. A. González Cervera. ECG Classification Using Artificial Neural Networks. *Journal of Physics: Conference Series*, 1221(1), 2019.

[22] A. Sellami and H. Hwang. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Systems with Applications*, 122:75–84, 2019.

[23] D. T. 2016.

[24] J. T., K. B., J. M., and L. B.-G. Implementation of a portable device for real-time ECG signal analysis. *BioMedical Engineering Online*, 13(1):1–13, 2014.

[25] T. Wang, P. Chen, T. Bao, J. Li, and X. Yu. Arrhythmia classification algorithm based on smote and feature selection. *International Journal of Performability Engineering*, 17(3):263–275, 2021.

[26] World Health Organization. The top 10 causes of death. `https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`, 2017. [Accessed]: 29-04-2021.

[27] Y. Xia, H. Zhang, L. Xu, Z. Gao, H. Zhang, H. Liu, and S. Li. An Automatic Cardiac Arrhythmia Classification System with Wearable Electrocardiogram. *IEEE Access*, 6:16529–16538, 2018.

[28] L. Xiaolin, B. Cardiff, and D. John. A 1d convolutional neural network for heartbeat classification from single lead ecg. In *ICECS 2020 - 27th IEEE International Conference on Electronics, Circuits and Systems, Proceedings*, 2020.

[29] G. H. Y. LeCun, Y. Bengio. *Deep Learning*. 2015.