An Artificial Intelligence Approach for Atrial Fibrillation Detection in Single-lead Invisible ECG

Rafael Luís Soares da Costa e Silva
rafael.c.silva@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2021

Abstract

Cardiovascular diseases (CVDs) are currently the leading cause of mortality worldwide. Since some CVDs can occur without symptoms and still be harmful, early detection and monitoring of patients at risk outside the hospital environment is crucial to avoid even higher mortality rates. For this reason, a recent paradigm of continuous monitoring systems is based on integrating physiological monitoring with the patient's daily life, using wearable and invisible technologies. One of the most relevant CVDs is Atrial Fibrillation (AF), associated with an increased risk of stroke. Due to its increasing prevalence and costs to healthcare systems, several approaches to detect it have been developed in recent years. Using single-lead electrocardiographic (ECG) recordings from PhysioNet Computing in Cardiology Challenge 2017 (CinC2017), an artificial neural network-based algorithm was developed to distinguish AF from Normal Sinus Rhythm (NSR). The proposed model involves coupling a compressed version of ECG segments generated by an unsupervised Autoencoder (AE) and a Machine Learning (ML) classifier. A Sparse Autoencoder (SpAE) and a Multilayer Perceptron (MLP) obtained an F1-score of 82.2%. By adding a feature concerning local changes of RR intervals around an R peak, the F1-score improved to 88.2%. Although simple, this approach proves that AEs can outperform algorithms using the same features, and that these can be improved to achieve even higher performance rates.
Keywords: Atrial Fibrillation Detection, Single-lead ECG, Autoencoders, Machine Learning

1. Introduction

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the leading cause of mortality worldwide, with 17.9 million deaths estimated in 2019, representing 32% of the world's total deaths [1]. CVDs include several health conditions involving blood vessels and heart function, such as heart failure, coronary heart disease, stroke, myocardial infarction, and arrhythmia.

To avoid higher mortality rates, home and remote monitoring of cardiac functions have gained more relevance over the years [2], becoming increasingly pervasive through the integration of cardiovascular assessment sensors in smartwatches and other wearable devices. Especially in wearables, a more granular monitoring can be achieved by recording the heart's activity, using techniques such as Photoplethysmography (PPG) or Electrocardiography (ECG). Due to the nature of this particular context, ECG recordings often use single-lead configurations, which means that only one view of the heart's electrical activity is recorded.

One of the most relevant cardiovascular diseases is Atrial Fibrillation (AF). Because of its high prevalence and costs to the healthcare systems [3], the detection of AF was chosen to be the main focus of this work. Also, in recent years, there has been an increased interest in developing algorithms for its detection [4], making it possible to compare different approaches and evaluate their performance.

Over the years, researchers built many approaches to detect AF, including algorithms based on Decision Trees (DTs), regression analysis, k-Nearest Neighbors (k-NNs), Support Vector machines (SVMs), and Artificial Neural Networks (ANNs) [4]. From the period 2016 to 2020, SVMs and ANNs represent almost 50% of all developed algorithms, which, in general, present high accuracy rates. Many ANN-based algorithms are built with deep architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which are known for their ability to capture complex patterns in the data. However, very few approaches use simple architectures such as the Autoencoders (AEs), which have been proven to be very useful to extract meaningful features from data.

Therefore, this work aims to develop an AF de-

tection algorithm based on AEs using data from single-lead ECG records suited for wearable and invisible applications [5].

2. Background

2.1. Physiological Concepts

Numerous technological advancements over the last decades have led to the development of techniques that can monitor vital signs in the human body, examples of which PPG and ECG.

ECG can be defined as the recording of the electrical activity of the heart, which has a conduction system responsible for propagating the electrical "pacemaker" pulses generated at the sinus node, throughout the atrial and ventricular myocardium. This provokes timed-muscle contractions, enabling the heart to pump blood through the cardiovascular system. In a clinical environment, ECG recordings are often made with several electrodes placed on predefined locations of the body to obtain the 12-lead ECG, allowing 12 different views of the heart's electrical activity. However, as already mentioned, wearable and invisible approaches rely in much simple acquisition systems, based on single-lead ECG that only requires as few as 2 or 3 electrodes. Although the single-lead ECG does not convey as much information as the 12-lead ECG, several ECG anomalies can be detected, such as AF.

Atrial Fibrillation (AF) is a type of arrhythmia, an uncoordinated activity of the heart chambers, in which the heart presents an irregular and high-rate electrical activity of the atria. When observing the ECG, AF is characterized by having "irregularly irregular" heart rhythms, absence of P-waves, nonexistence of an isoelectric baseline and variable ventricular rate [6]. Also, P-wave activity can be replaced by fibrillatory waves (f-waves).

2.2. Technical Concepts

ANNs are able to solve very complex problems of non-linear, multivariate, and/or stochastic nature. The basic unit of a ANN is an artificial neuron, that receives a set of input values $\bar{x} \in \mathbb{R}^n$ and outputs a real value $\hat{y} \in \mathbb{R}$. By connecting artificial neurons through weight values into a network, many architectures are possible. To be able to train, an algorithm is used to update the neurons connections by comparing the prediction with the desired output. The goal of the training algorithm is to find the optimal set of parameters that minimize a predefined cost function C, which measures how accurately the model predicts the data.

An AE is type of ANN which is divided into two elements: an encoder and a decoder. The encoder is responsible for generating a feature vector $\bar{z}_i$ (also called code) from the input $\bar{x}_i$, generally by compression, while the decoder is responsible for reconstructing the model input from the feature vector (Figure 1). The cost function is usually proportional to the reconstruction error using, for instance, the mean squared error (MSE) between the input $\bar{x}_i \in X$ and output $\hat{x}_i$:

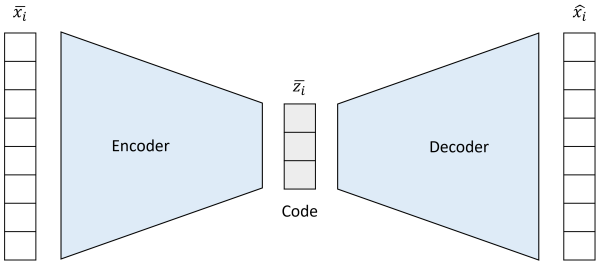


Figure 1: Schematic representation of an Autoencoder model. It is composed of an encoder, responsible for compressing the data into a latent representation called code, and a decoder, which aims to reconstruct the input from the code.

$$C\left(X; \bar{\theta}_e, \bar{\theta}_d\right) = \frac{1}{M} \sum_{i=1}^{M} \|\bar{x}_i - \hat{x}_i\|^2 \qquad (1)$$

where $\bar{\theta}_e$ and $\bar{\theta}_d$ are the set of the encoder and decoder parameters, respectively, and M is the cardinality of the dataset X.

By enforcing the input to be compressed into a latent representation, and by using a cost function that favors the output to be as close as possible to the input, the feature vector within the code should hold relevant information about the data's structure, provided that the model converged and that no overfitting occured. Besides the standard AE, a number of other architectures are possible, including Denoising Autoencoders (DAEs), Sparse Autoencoders (SpAEs) and Variational Autoencoders (VAEs). All AE variations aim to promote meaningful latent representations of the data, but using different approaches. For example, DAEs aim to reconstruct the original version of an input from a noisy corrupted version, while SpAEs try to enhance the generalization ability by applying a L1 penalty to the code layer, meaning that the penalty equals to the absolute value of the weights. In VAEs, the latent code is assumed to be a random variable of a distribution, and the goal is to obtain the parameters of a probabilistic encoder, assumed to be a Gaussian distribution, that minimize the Kullback-Leibler divergence with the posterior distribution. Other types of AE are possible such as deep, contractive and convolutional AEs. For further details, the reader is referred to [7].

2.3. Classification Metrics

Classification metrics evaluate the performance of a specific characteristic of a classifier, and they are an essential tool to compare different classifiers with

the same task. In this work we reduce the problem to a binary classification, in which two classes separate the data: one called positive and the other called negative. Some of the most common classification metrics are:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

where TP, TN, FP and FN represent the number of true positive, true negative, false positive and false negative instances, respectively.

Another method to evaluate a classifier is the Receiver Operating Characteristic (ROC) curve, and the respective area under the curve (AUC). They can be useful if the classifier has parameters that affect classification or if its output consists of probability values. In such case, the threshold to separate the classes will affect its performance metrics. The ROC curve shows then the trade-off between the sensitivity and specificity, by setting different parameters or threshold values, and the AUC, because it does not depend on a specific point of the curve, is capable of measuring the ability of the classifier to predict the classes accurately.

3. State of the Art

3.1. Atrial Fibrillation Detection Algorithms

Algorithms greatly rely on what information is fed into them and, in the case of AF detection, different ECG feature inputs can be used. Atrial features to detect AF rely on ECG properties inherent to atrial activity, such as the absence of P-waves and or appearance of f-waves. The F1-score for an algorithm of such category developed by Christov et al. was estimated at 83.8% [4].

Ventricular features are based on ventricular activity by obtaining information regarding the QRS complexes, which are generally more pronounced than P-waves and f-waves. The R-peaks are the main focus of such features because the presence of irregularities in RR-intervals are a key sign to diagnose AF. Examples of ventricular features are standard deviation (SD), coefficient of variance, root mean square of succesive differences (RMSSD), Poincaré plots, sample entropy, Shannon entropy, turning point ratio (TPR) and Lyapunov exponents. The review made by Wesselius et al. [4] reported a median F1-score of 96.9% from 38 algorithms that only use ventricular features, and represented the highest score from the other types and combinations of features (Table 1).

Table 1: Median F1-score by type of feature groups. It includes atrial, ventricular and signal features, as well as the combinations between them. Extracted from [4].

Feature groups	Number of algorithms	Median F1-score
Atrial	1	83.8%
Ventricular	38	96.9%
Signal	34	95.2%
Atrial + Ventricular	10	85.6%
Atrial + Signal	1	88.9%
Ventricular + Signal	6	91.1%
Atrial + Ventricular + Signal	13	81.0%
Overall	**103**	**94.0%**

Another frequent type of features to be extracted are signal properties, which include a series of measures such as basic statistics, signal power, kurtosis and derivatives. More complex signal features involve performing power spectral analysis, phase space analysis, computing wavelet transform and measuring signal quality. The same review reports a median F1-score of 95.2% from a total of 34 algorithms.

3.2. Autoencoders for Atrial Fibrillation Detection

Very few approaches use ANN architectures such as the AEs to detect AF. For instance, Yuan et al. [8] developed an approach for AF detection from ECG records using a stacked SpAE based on 84 selected features extracted from the RR-intervals and p-wave measurements of a 10-second-window. The AE used to achieve data compression had 84 input nodes and 2 hidden layers with 300 nodes each. AF detection was made by stacking a softmax classifier to the extracted features of the AE. Using ECG records from the MIT-BIH databases[1], the model first achieved a detection accuracy of 75,6%, and, after fine-tuning the model, a 98,3% accuracy was reported. A similar approach was followed by [9], where a stacked SpAE receives 19 features extracted from the ECG records, including statistical mea-

[1] https://ecg.mit.edu/

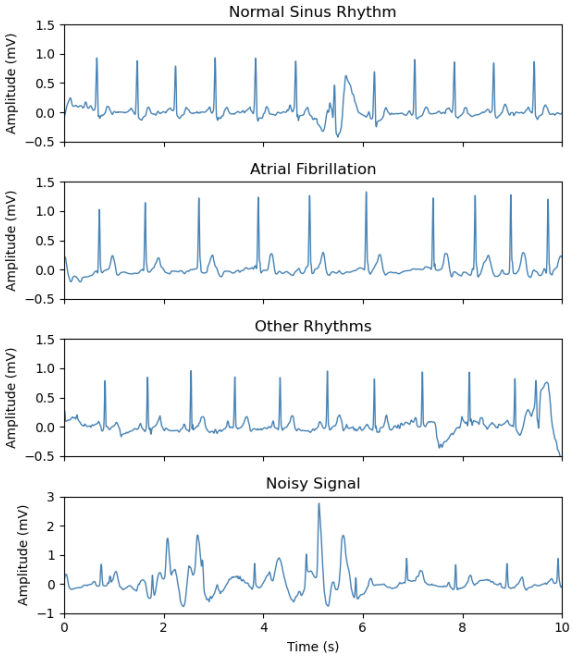


Figure 2: Examples of Normal Sinus Rhythm, Atrial Fibrillation, other rhythms and noisy acquisitions in the Computing in Cardiology Challenge 2017 dataset.

sures, parameters from the Hilbert-Huang transform, and wavelet decomposition features. After training, the AE is then coupled to a softmax classifier to detect AF. A 96% accuracy was achieved.

3.3. ECG Datasets for Atrial Fibrillation Detection

The increased interest in AF detection in recent years led to the development of publicly available databases to train and test new models.

The AF database from the Computing in Cardiology Challenge 2017 (CinC2017)[2] is considered to be one of the main sources of new publications related with AF detection algorithms [4]. The CinC2017 database has 12186 short single-lead records, varying from 9 to 60 seconds. Record labels include Normal Sinus Rhythm (NSR), AF, other rhythms, and noisy acquisitions (Figure 2). The records were obtained with a Left Arm – Right Arm lead configuration, equivalent to a Lead I, using AliveCor's[3] single channel ECG devices, including the AliveCor® KardiaMobile. The acquisitions were stored using a 300 Hz sampling frequency with 16-bit resolution over a ±5 mV dynamic range and a 0.5 – 40 Hertz (Hz) bandwidth interval [10].

[2] https://physionet.org/content/challenge-2017/
[3] https://www.kardia.com/

4. Proposed Approach

4.1. Rationale

As already discussed, AEs are an unsupervised learning model in which it is possible to retrieve a meaningful and compact representation of a signal. The interest in studying AEs to detect AF arose because few approaches are using them; thus, its use is not well documented. Also, the great diversity of possible AE architectures allows exploring and creating innovative solutions to a real-world challenge, for which there is no current gold standard approach for mainstream use. Also, besides anomaly detection, AEs can be used for classification by feeding classifiers with their compact representations, i.e. using AEs for feature extraction and dimensionality reduction.

Moreover, in today's implementations of heart monitoring technologies and algorithms, responsiveness is highly valued. Systems with high computational speeds using simple algorithmic approaches, such as the AEs, are preferred. Also, when designing classification algorithms, there is a great investment in choosing the features that allow the best classification performance. However, AEs do not require feature engineering, and they are flexible in terms of complexity, thus requiring fewer computation resources for possible real-time implementation.

In heart monitoring systems, preprocessing steps often include beat extraction using segmentation techniques; thus ECG waveform fiducials and several metrics can be used to classify beats. However, most AF detection systems rely on information extracted from ECG records containing several beats, namely because they focus on ventricular activity (e.g., RR-intervals), which would extend the dataflow pipeline and increase resource usage. By using ECG segments, instead of long records, to detect AF, the respective implementation becomes lighter and it converts the problem into a beat classification one. Since the AE receives the ECG waveform as its input, it can be called a morphological AE.

4.2. Data Preprocessing

To train the AEs with ECG segments, the records from the CinC2017 database were first preprocessed using the Python toolbox BioSPPy[4]. The data were filtered using a 90th-order high-pass Finite Impulse Response (FIR) filter with a cutoff frequency of 0.5 Hz, and the R-peak detector by Hamilton available in the toolbox was used.

Because some of the records were inverted in amplitude, an approach to correct them was set in place: the median of the R-peak amplitudes was chosen as the main criterion. The inverted records

[4] https://github.com/PIA-Group/BioSPPy

were identified by having a lower median R-peak amplitude than their correspondent inverted versions, that is, the correct ones. In fact, the peaks of the inverted signals are S-waves, with lower amplitude than the R-peaks. An example of such correction is shown in Figure 3. After correcting the inverted ECG records, the signals were then clipped from -200 to 600 milliseconds around the R-peaks, as defined by the toolbox and used by a real-time implementation in [11]. To avoid the AE being affected by the appearence of second R-peaks, zero padding was applied 50 milliseconds before the second R-peak to cover the Q-wave, as represented in Figure 4(a). This approach to segment the signals was preferred because the number of input nodes in an AE must be fixed from the beginning.

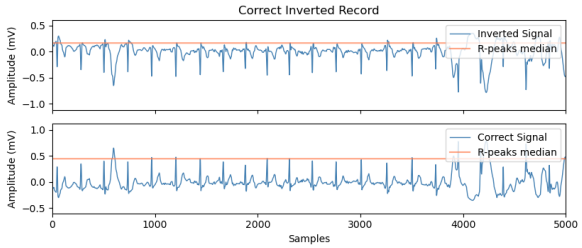


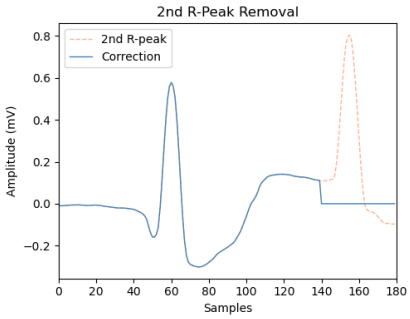
Figure 3: Example of an inverted ECG record that was identified by having a lower median of the correspondent peak amplitudes (top) than the inverted version (bottom), which is the correct one. The peaks of the inverted signal are S-waves, and the peaks of the correct signal are the real R-peaks.

4.3. Autoencoder Models

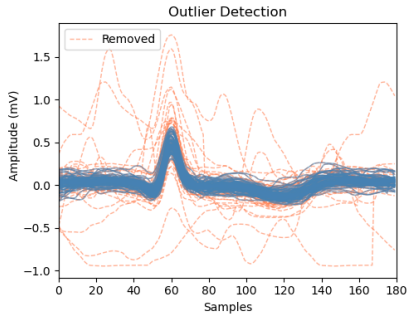
Since the architecture of an AE is based on ANNs, AE design is very versatile. The simplest AE structure contains a single hidden layer, where one needs to define the compression factor to be achieved, since the input and output layers have the same nodes as the data samples. Although there are no strict rules when choosing the number of nodes in the hidden layer, frequent compression factors are 25%, 50%, and 75% [12]. Adding more hidden layers is also a common choice for AEs.

Aside from choosing an AE structure, inherent parameters of ANN training are still of utmost importance, since they can affect the overall ability of the AE to properly reconstruct the inputs through the choice of the networks' weights and biases (e.g., optimizer, learning rate, loss function, activation function). Furthermore, the choice of the batch size (number of samples fed into the network in one training iteration) and the number of epochs (number of times that the whole dataset passes through the network) are also relevant parameters to achieve the best performance.

To study the best approach to reconstruct ECG



(a)



(b)

Figure 4: (a) Example of 2nd R-peak removal on a segment using zero-padding. (b) Use of DMEAN to detect outliers in a set of ECG segments from a record.

segments, different symmetric AE structures were tested, varying the compression factor and number of hidden layers. Under adequate learning parameters, different AEs were evaluated based on their capacity to reconstruct the inputs with a high compression factor. The MSE loss was monitored in each AE training using validation data of subjects not seen by the AE.

The proposed AE models are briefly described in Table 2, indicating the AE type, structure and loss function. Aside from input and output dense layers with $N\_segment$ nodes, the Standard Autoencoder (SAE) consists of a single hidden layer with N_hidden nodes, having a linear activation function for the output layer and a MSE loss function (i.e., the reconstruction error). The DAE adds a Gaussian noise layer to the input, defined by its standard deviation, and the SpAE model has a L1 regularization in the hidden layer that allows the model to become sparse. The Robust Autoencoder (RAE) model is similar to a DAE in its ability to resist noise, but uses a loss function to do it. The correntropy loss function proposed by Liu et al. [13] was used. The Contractive Autoencoder (CAE) is also a model that aims to extract useful features by its loss function. In CAEs, a penalty term is added to the MSE, which is defined as the Frobenius norm

of the Jacobian matrix of the hidden mapping of the AE [14]. The VAE model has an intermediate layer with $N\_intermediate$ nodes before the bottleneck layer. The bottleneck consists of two steps, one involving two layers to encode the mean and the covariance matrix of the data as Gaussian distributions, and another involving a layer that allows backpropagation of the algorithm.

4.4. Classification Algorithms

After training the AE models, the encoders contain the mapping that leads to the latent representations of the signals. This reduced representation can be fed to standard classification algorithms responsible for finding metrics or patterns in the data that separate the classes in a supervised manner. Based on the state-of-the-art, 5 different classifiers were tested, described in more detail in Table 3.

The first consists of a single node, a perceptron. The perceptron model takes the ensemble of the features captured by the encoder and the classification labels (NSR or AF), and it adjusts its weights in the training step to fit the binary data. The output of the model is a real value between 0 and 1, being 0 defined as NSR and 1 as AF. For this reason, the chosen activation function for the output node was the sigmoid function, and the loss function to train the model was the binary cross-entropy. The second model is an multilayer perceptron (MLP). This model has a basic ANN structure with hidden stacked layers and an output node or nodes to perform classification. The advantage of the MLP model is that it can map the features into a nonlinear space. The hidden layers used the rectified linear unit (ReLU) activation function and the output node the sigmoid function.

Other widely used classification models include SVMs, k-NNs and DTs, that were also tested. The SVM model used the radial basis function as its kernel, the k-NN model was defined with the number of neighbors k equal to 5, and the DT classifier used the Classification and Regression Trees (CART) algorithmic implementation.

Figure 5 presents the proposed approach for the AF detection algorithm, using a trained AE and a classifier.

4.5. Training and Evaluating the Models

To check the best combination of AE and classifier to distinguish NSR from AF beats, a series of experiments were conducted. To understand how the number of hidden layers and the compression level in the AE models affect the quality of the reconstructed signal, the SAE model was trained with different configurations. Maintaining the same training parameters and the same training and validation data, the compression levels 25%, 50%, 75%, 90%, and 95% were tested, and, for each, the number of hidden layers varied between 1, 3, and 5. Because these AE parameters change the models' complexity, an early stopping approach was used instead of fixing the number of epochs, consisting of fixing the minimum loss difference *delta* between consecutive epochs to consider an improvement.

Finally, the different models are tested by ex-

Table 2: Layer structure of the proposed Autoencoders and their loss functions.

AE	Layer	Loss Function
SAE	Dense(input) Dense(latent) Dense(output, linear)	MSE
DAE	Dense(input) GaussianNoise(SD=0.05) Dense(latent) Dense(output, linear)	MSE
RAE	Dense(input) Dense(latent) Dense(output, linear)	Correntropy
CAE	Dense(input) Dense(latent) Dense(output, linear)	Contractive
SpAE	Dense(input) Dense(latent, L1 reg.) Dense(output, linear)	MSE
VAE	Dense(input) Dense(intermediate) 2×Dense(latent) Lambda(sampling) Dense(intermediate) Dense(output, linear)	MSE + D_{KL}

SD = Noise standard deviation
MSE = Mean squared error
D_{KL} = Kullback-Leibler divergence

Table 3: Structure and parameters of the proposed classifiers.

Classifer	Structure/Parameters
Perceptron	Dense(1, sigmoid)
MLP	Dense(latent, ReLU) Dense(latent, ReLU) Dense(latent, ReLU) Dense(1, sigmoid)
SVM	RBF kernel
k-NN	$k = 5$ neighbors
DT	CART implementation

tracting the features from the AEs and feeding them into the classifiers. To preserve the models from being biased towards previously seen data, a subject-based data split was performed. Using cross-validation, the data were first split into training and testing sets. The k-Fold Cross-Validation consists of splitting the data into k parts (called splits), where $k-1$ splits are used to train the model, and the remaining one to test it. Each split tries to maintain the same proportion of the various labels. The different combinations of the splits to train and to test are used, and k performance results are obtained. From the training set, 50% of subjects were picked to train the AEs and the other 50% to train the classifiers, as depicted in Figure 6. To evaluate the models, a 2×5 cross-validation was made, which means that a 5-Fold Cross-Validation is performed twice, with different fold compositions [15]. Data balance was guaranteed.

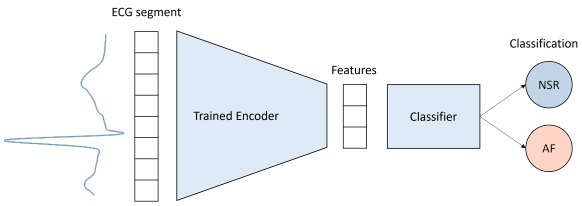


Figure 5: Diagram showing the proposed Atrial Fibrillation detector, by stacking a trained encoder with a classifier.

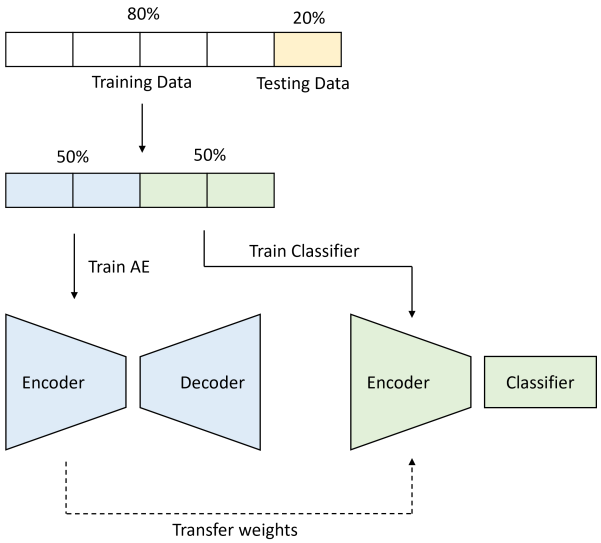


Figure 6: Diagram showing the data split approach to train the Autoencoders and the classifiers. The split avoids using the same data to train the Autoencoder and to train the classifier.

4.6. Alternative Approaches

The approach presented so far is uniquely dedicated to distinguish ECG waveforms in NSR from ECG waveforms in AF. However, an algorithm suited for real-life applications must be able to deal with the variability and complexity of ECG records. Since the impact of other rhythms and waveforms in the proposed AF detection algorithm is unknown, these were also fed into an AE. Because the problem is no longer of binary classification, the MLP had to be modified. Instead of having a single output node, the model was built with three output nodes and the cost function had to be changed from binary cross-entropy to categorical cross-entropy, which is suited for multiclass classification problems.

Because the features extracted from the proposed AE models only rely on the waveform morphology, another alternative approach to enhance it is to provide some information regarding the RR-intervals of the ECG records, at least locally. In ECG records, RR-intervals in AF are more irregular than in NSR. This means that, generally, in relation to a single R-peak, the difference between the RR-intervals around the peak, $RR_i - RR_{i-1}$, is greater in AF than in NSR. The proposed metric, hereinafter called Local Change of Successive Differences (LCSD), is defined for each R-peak R_i as:

$$\mathrm{LCSD}\,(R_i) = \frac{|RR_i - RR_{i-1}|}{\frac{1}{N}\sum_{j=1}^{N} RR_j}, 1 < i < N \qquad (7)$$

where N is the number of RR-intervals. This metric can be particularly relevant in the current context, where the individual waveforms are classified, allowing the classifiers to take into account not only the morphological features generated by the AE but also the local changes of the RR-intervals.

5. Results

5.1. Preliminary Tests

The first preliminary experiment consisted in evaluating the impact of the compression level and the number of hidden layers in the ability of the SAE to reconstruct the ECG segments. All hidden layers used a linear activation function, the *delta* parameter to perform early stopping was set to 1e-6, and the Adam optimizer was used with a 0.001 learning rate. The train and validation data used 77% and 33% of the entire dataset, respectively.

By fixing the number of hidden layers, one can observe that, in general, there is no relevant information loss up to a 75% compression level, that corresponds to 45 nodes of the innermost AE layer. The validation MSE loss starts to steeply rise thereafter, reaching MSE errors close to 1e-3 and higher. Interestingly, as already stated in [16, 17], the increase in the number of hidden layers appears to

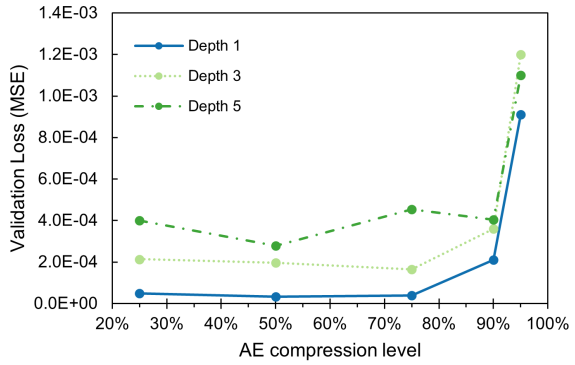


Figure 7: Validation losses varying the number of hidden layers (depth) and the compression level at the bottleneck layer.

worsen the reconstruction capacity of the AEs, since, for each compression level, the MSE increases with the number of layers.

With further training epochs, the 90% compression level was proven to be sufficient. As such, for the next experiments consisted of defining several AEs with a 90% compression level (18 bottleneck nodes) and a single hidden layer. Training of the AEs was made with 600 epochs and 2000 sample batch size, using the Adam optimizer with a 0.001 learning rate.

5.2. Autoencoder-Classifier Combinations

By looking at the first 18 features of Figure 8 generated by the SpAE model, there seems to be no linear relationship to differentiate NSR and AF waveforms from them, since the distributions' medians and interquartile ranges are very similar. This was the case for all AE models, even if a non-linear activation function of the hidden nodes was chosen.

Table 4 presents the best combinations of the AE-classifier models with the respective performance metrics. Some remarks can be made: the DAE and SpAE models, that act directly on the AE's nodes, achieved the best waveform reconstructions and the best classification performances. This suggests that better reconstructions also promote feature meaningfulness; the MLP and SVM classifiers led to the best classification performances across all AE models, and the CART and perceptron models led to the worse performances (Table 4). The best model from all AE-classifier combinations was the SpAE-MLP model, that achieved an F1-score of 82,2% and a AUC equal to 0.902 Table 4).

5.3. Alternative Approaches

To test the impact of adding other ECG rhythms to the proposed classifier model, the respective class records were preprocessed with the same procedure as the NSR and AF classes. Because of their higher accuracy rates, only the SVM and MLP were tested. The MLP model, with three output nodes, used the softmax activation function.

From Table 4 it is noticeable that the classification performance suffers when adding a third class. An F1-score of 58.1% was achieved, in contrast with the 82.2% where only AF and NSR were the classifier targets. Because of the model has poor performance, the other test consisted of joining the NSR and other rhythm classes, assigning them the same label. When training in this condition, an improvement of the F1-score was seen, increasing from 58.1% to 64.3%. However, since it does not match the SpAE-SVM model that only classifies NSR and AF, this suggests that some waveforms from other rhythms may be morphologically similar to AF.

As already described, a strategy to feed the classifiers with information regarding the RR-intervals was defined using the LCSD proposed metric. All classifiers were tested using both the features generated by the SpAE model, and the LCSD of the R-peak associated with the waveform. From Figure 8 (feature number 18), it is noticeable that there is a significant difference between the LCSD values of AF and NSR classes. This metric led to an F1-score of 88.2%, an improvement of 6% (Table 4). Besides the SVM, the LCSD feature also increased the performance of the remaining classifiers.

6. Conclusions

The work developed in this thesis allowed the development of an approach for AF detection based on ECG heartbeat waveforms. Taking into account the context of wearable and invisible ECG monitoring systems, a single-lead ECG database for AF detection was selected. The proposed method consisted of using an AE to first achieve a compressed version of the ECG segments, and then the corresponding generated features are used to feed a classifier. Since there are many AE and classifier types, some of them were tested and compared.

From all experiments, the SpAE model in combination with a MLP achieved the best performance in distinguishing NSR and AF segments, with an F1-score of 82.2% and an AUC of 0.902. Moreover, by using a new metric focused on the local RR-interval differences around an R-peak (LCSD), the SpAE-MLP model achieved a 88.2% F1-score (+8.0%) and an AUC of 0.950.

In comparison with state-of-the-art algorithms, the aforementioned F1-score is above the median of current AF detection algorithms based on atrial and ventricular features (85.6%) but stays well below the overall F1-score median of all considered algorithms (94.0%). This comparison, however, is not the most appropriate because the methodology in this work was not developed to classify ECG

Table 4: Summary of the best AE-classifier combinations, and the respective performance metrics. The models with the best F1-score are in bold.

AE-Classifier	Accuracy	Precision	Recall	F1-score	AUC
SAE-MLP	0.816±0.015	0.824±0.018	0.803±0.018	0.813±0.015	0.893±0.014
DAE-MLP	0.822±0.015	0.827±0.017	0.814±0.029	0.82±0.017	0.9±0.014
RAE-MLP	0.817±0.015	0.824±0.015	0.806±0.02	0.815±0.815	0.896±0.012
CAE-MLP	0.771±0.017	0.769±0.018	0.776±0.022	0.772±0.018	-
SpAE-MLP	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**
VAE-SVM	0.805±0.008	0.822±0.008	0.772±0.017	0.8±0.01	-
SpAE-MLP NSR/AF/O	0.586±0.014	0.581±0.014	0.586±0.014	0.581±0.013	-
SpAE-MLP (NSR+O)/AF	**0.776±0.010**	**0.685±0.017**	**0.605±0.026**	**0.643±0.019**	**0.829±0.015**
SpAE-MLP Code + LCSD	**0.880±0.009**	**0.877±0.029**	**0.888±0.027**	**0.882±0.006**	**0.950±0.007**

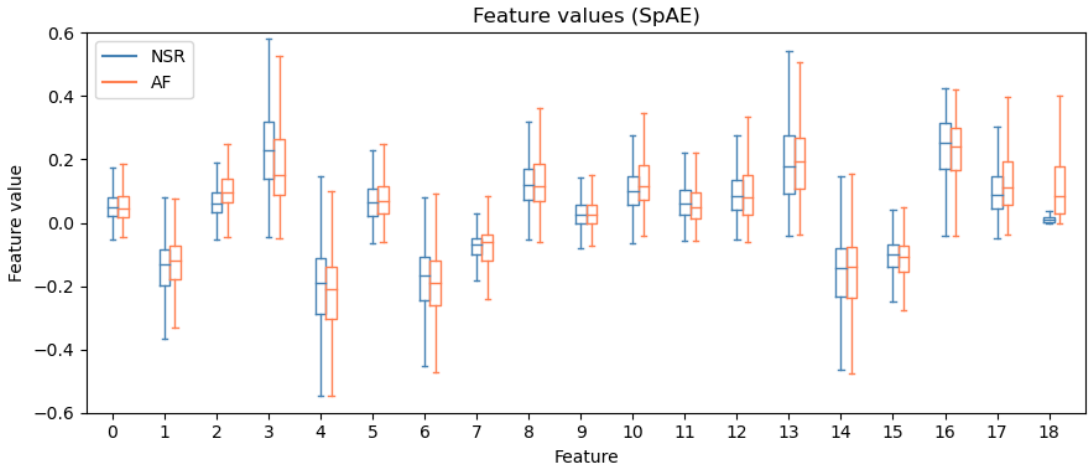


Figure 8: Feature value distributions of the hidden features generated by the SpAE model for NSR and AF test instances. The feature number 18 corresponds to the LCSD distributions, which present significant mean and value range differences.

records, but instead it was developed to classify ECG waveform segments. Thus, the potential to diagnose AF using this approach could be further explored. Also, the algorithm was not tested with other databases.

7. Future Work

Despite satisfactory results in distinguishing AF from NSR waveforms, some limitations of this work can lead to further developments. A model which uses beat-by-beat classification to evaluate ECG records would still be needed to compare with other approaches. For example, an AE with a time window containing several beats could be used, or other ANN structures appropriate for time-series classification, such as Long Short-Term Memorys (LSTMs). Moreover, because of the great diversity of AE models and general ANN structures and training parameters, the proposed approaches may not represent the full potential of AEs to detect AF. Supervised or semi-supervised approaches could possibly enhance the performance rates (e.g., a supervised VAE model).

References

[1] World Health Organization. Cardiovascular diseases (CVDs) fact sheet, November 2021.

[2] Mohamed Adel Serhani, Hadeel T. El Kassabi, Heba Ismail, and Alramzana Nujum Navaz. ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges. *Sensors*, 20(6):1796, January 2020. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[3] Giuseppe Lippi, Fabian Sanchis-Gomar, and Gianfranco Cervellin. Global epidemiology of atrial fibrillation: An increasing epidemic and public health challenge. *International Journal of Stroke*, 16(2):217–221, February 2021. Publisher: SAGE Publications.

[4] Fons J. Wesselius, Mathijs S. van Schie, Natasja M. S. De Groot, and Richard C. Hendriks. Digital biomarkers and algorithms for detection of atrial fibrillation using surface electrocardiograms: A systematic review. *Computers in Biology and Medicine*, 133:104404, June 2021.

[5] Hugo Plácido da Silva. Biomedical Sensors as Invisible Doctors. pages 322–329. January 2019.

[6] P. Podrid, R. Malhotra, R. Kakkar, and P.A. Noseworthy. *Podrid's Real-World ECGs: Volume 4, Arrhythmias: A Master's Approach to the Art and Practice of Clinical ECG Interpretation*. Podrid's Real-World ECGs:A Master's Approach to the Art and Practice of Clinical ECG Interpretation. Cardiotext Publishing, 2015.

[7] Giuseppe Bonaccorso. *Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work, 2nd Edition*. Packt Publishing Ltd, January 2020. Google-Books-ID: M0vODwAAQBAJ.

[8] Chan Yuan, Yan Yan, Lin Zhou, Jingwen Bai, and Lei Wang. Automated atrial fibrillation detection based on deep learning network. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 1159–1164, August 2016.

[9] Lili Chen and Ying He. Identification of Atrial Fibrillation from Electrocardiogram Signals Based on Deep Neural Network. *Journal of Medical Imaging and Health Informatics*, 9(4):838–846, May 2019.

[10] Gari D Clifford, Chengyu Liu, Benjamin Moody, Li-wei H. Lehman, Ikaro Silva, Qiao Li, A E Johnson, and Roger G. Mark. AF Classification from a Short Single Lead ECG Recording: the PhysioNet/Computing in Cardiology Challenge 2017. *Computing in cardiology*, 44:10.22489/CinC.2017.065–469, September 2017.

[11] André Lourenço, Hugo Silva, Paulo Leite, Renato Lourenço, and Ana Fred. Real Time Electrocardiogram Segmentation for Finger Based ECG Biometrics. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing - BIOSIGNALS, (BIOSTEC 2012)*, pages 49–54. SciTePress, 2012. Backup Publisher: INSTICC ISSN: 2184-4305.

[12] Francisco J. Pulgar, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips and guidelines. *Information Fusion*, 54:44–60, February 2020.

[13] Weifeng Liu, P.P. Pokharel, and J.C. Principe. Correntropy: A Localized Similarity Measure. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 4919–4924, July 2006. ISSN: 2161-4407.

[14] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Y. Bengio. *Contractive Auto-Encoders: Explicit Invariance During Feature Extraction*. January 2011. Journal Abbreviation: Proceedings of the 28th International Conference on Machine Learning, ICML 2011 Publication Title: Proceedings of the 28th International Conference on Machine Learning, ICML 2011.

[15] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer, New York, 1st ed. 2013, corr. 2nd printing 2018 edition edition, May 2013.

[16] Francisco J. Pulgar, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. AEkNN: An AutoEncoder kNN-Based Classifier With Built-in Dimensionality Reduction. *International Journal of Computational Intelligence Systems*, 12(1):436–452, February 2019. Publisher: Atlantis Press.

[17] Francisco J. Pulgar, Francisco Charte, Antonio J. Rivera, and María J. del Jesus. Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips and guidelines. *Information Fusion*, 54:44–60, February 2020.