Computer Engineering and Applications Vol. 12, No. 3, October 2023

Classification of Atrial Fibrillation In ECG Signal Using Deep Learning

Muhammad Fachrurrozi[1], Muhammad Naufal Rachmatullah[2], Raihan Mufid Setiadi[1]

[1]Department of Computer Engineering,Faculty of Computer Science, Universitas Sriwijaya
[2]Intelligent System Research Group, Computer Engineering,Faculty of Computer Science, Universitas Sriwijaya
*mfachrz@unsri.ac.id

ABSTRACT

Atrial fibrillation is a type of heart rhythm disorder that most often occurs in the world and can cause death. Atrial fibrillation can be diagnosed by reading an Electrocardiograph (ECG) recording, however, an ECG reading takes a long time and requires specialists to analyze the type of signal pattern. The use of deep learning to classify Atrial Fibrillation in ECG signals was chosen because deep learning has 10% higher performance compared to machine learning methods. In this research, an application for classification of Atrial Fibrillation was developed using the 1-Dimentional Convolutional Neural Network (CNN 1D) method. There are 6 configurations of the 1D CNN model that were developed by varying the configuration on the learning rate and batch size. The best model obtained 100% accuracy, 100% precision, 100% recall, and 100% F1 Score.

Keywords: Atrial Fibrillation, CNN 1D, Normal, Non-Atrial Fibrillation, ECG Signal

1. INTRODUCTION

Atrial fibrillation is the most common type of arrhythmia. Atrial fibrillation occurs when the muscles in the heart malfunction and cause an irregular heartbeat. Irregular heartbeat can form blood clots in the chambers of the heart and inhibit the process of blood circulation so that it becomes a factor in the emergence of cardiovascular disorders [1]. Characteristics of Atrial Fibrillation represented using ECG can be an anomaly in the P waveform and irregular RR-interval [2]. However, diagnosing atrial fibrillation is not easy and can take a long time. This obstacle occurs because the rhythm of the ECG signal can change from an Atrial Fibrillation rhythm to a normal rhythm, and many rhythms that are not Atrial Fibrillation but have irregular RR intervals the same as Atrial Fibrillation[3].

The process of automating Atrial Fibrillation diagnosis can be made easier thanks to the rapidly evolving fields of machine learning and deep learning. Machine learning and deep learning can be used to analyze and study the features of the ECG recording data, to classify the Atrial Fibrillation signal patterns and non-Atrial Fibrillation signal patterns. However, deep learning algorithms such as CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory) have a higher average performance of 10% when compared to machine learning algorithms such as Support Vector

Machines and Logistic Regression [4]. Deep learning models can perform automatic feature extraction on input data and extract only those features that are considered important [4]. Meanwhile, machine learning algorithms are more traditional because they require data whose features are manually selected by humans and feature extraction is done manually [5].

CNN (Convolutional Neural Network) is a deep learning architecture composed of input layer, convolution layer, pooling layer, and fully connected layer [6]. The CNN model is proven to have the best performance in the case of ECG signal classification when compared to other deep learning models such as LSTM and RNN [7]. However, the CNN architecture requires high computing power, because to find features in the CNN architecture data, it uses a 2-dimensional filter in the form of a matrix that scans the data [8]. The 2-dimensional CNN architecture is suitable for use in visual data such as image classification. However, to classify the 1-dimensional CNN ECG signal data it is more effective to use because it requires lower computing power, because 1-dimensional CNN uses a 1-dimensional filter to find features in the data [9].

The main problems with traditional Atrial Fibrillation diagnosis method is only experts can process, analyse, and diagnose certain type of signal in ECG recordings needed and the process can be time consuming. The time consuming problem of this process was caused by many signal type in one ECG recordings. Based on the described problem the purpose of this study such as follow;

- Building a classification model for Atrial Fibrillation signal patterns using a 1-dimensional CNN algorithm that can classify Atrial Fibrillation, Non-Atrial Fibrillation, and Normal signal patterns.
- Analysing the accuracy and performance of the Atrial Fibrillation signal pattern classification model using the 1-dimensional CNN algorithm.

2. RELATED WORKS

This Section of paper describe other existing similar study that used as based for this study. In previous study Nurmaini and Others [8] proposed 1 dimensional CNN with 13 convolution layer and 10 fold cross validation scheme to classify 3 classes, Atrial Fibrillation, Non Atrial Fibrillation, and Normal Signal. The proposed preprocessing method was using Discreete Wavelete Transform to remove noise from 4 data sources MIT-BIH Atrial Fibrillation, Physionet Atrial Fibrillation , MIT-BIH Malignant Ventricular Ectopy, and ECG data from an Indonesian hospital.

Xia and others [10], proposed 2 dimensional Deep CNN to classify MIT-BIH *Atrial Fibrillation* dataset. Short Term Fuorier Transform and Stationary Wavelet Transform was proposed as the preprocessing method to shape 1 dimensional input data to 2 dimensional input data.

In previous study Chandra and others [3] proposed beat stack classifier on 1-dimensional CNN model to classify stacks on R peak wave in MIT-BIH Arhythmia dataset.

3. MATERIAL AND METHODS

3.1 DATASET

MIT-BIH Atrial Fibrillation Database [11] and MIT-BIH Normal Sinus Rhythm Database [12] was used as Raw ECG datasets for this study. MIT-BIH *Atrial Fibrillation* Database contains 4 signal type such as AFIB for Atrial Fibrillation, AFL for Atrial Flutter, J for AV Junction, and N for others signal. For MIT-BIH *Atrial Fibrillation* Database N label was not used because to avoid ambiguity on model training. AFL and J laber were merged for Non AF label. MIT-BIH Normal Sinus Rhythm Database contain normal sinus heart bear for Normal label data. The samples of MIT-BIH Atrial Fibrillation Database and MIT-BIH Normal Sinus Rhythm Database are shown in fig 1.

3.2 PREPROCESSING

The preprocessing methods that used in this study such as DWT Denoising, data normalization, data segmentation, and train – test split. Discrete Wavelet Transform (DWT) is a signal decomposition method consisting of a low pass filter (LPF) and a high pass filter (HPF). The two filters will break the signal into LPF or approximation components and HPF or detail components [13]. Thresholding is a function that is used to clean the Raw ECG Data from noise by changing the value of detail coefficients based on the Threshold calculation [14]. New detail coefficients will be merged with approximation components in inverse DWT to create new clean signal [14]. In this study soft threshold calculation was used to calculate new detail coefficients ($softD_j'$):

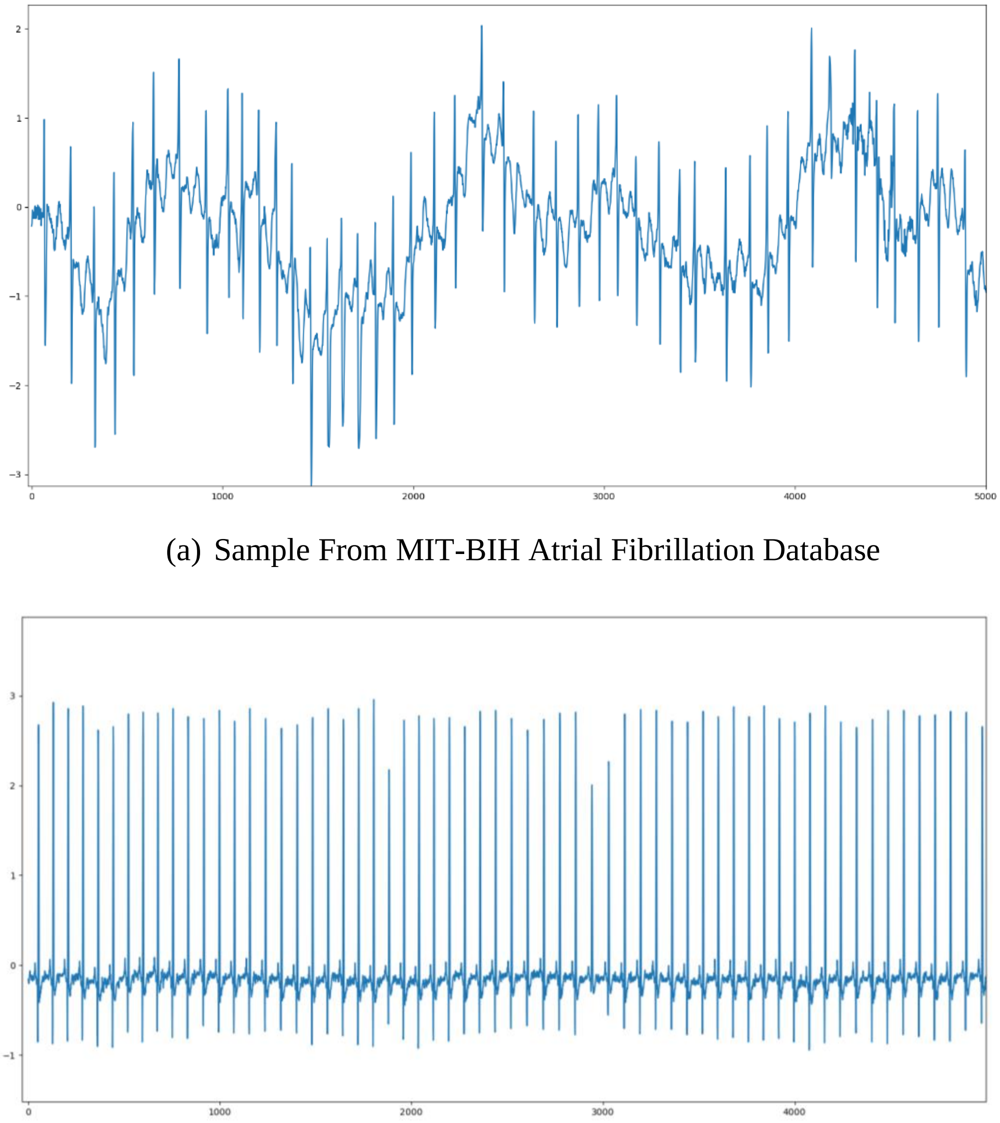
$$softD_j' = \begin{cases} sing(D_j\ )(D_j\ - D_j\ ),\ if|D_j\ | > \lambda \\ 0,\ if|D_j\ | \leq \lambda \end{cases} \tag{1}$$

$$\lambda = \sigma\sqrt{2log_e(N)} \tag{2}$$

Where σ is noise deviation = (median|cDj |/0.6457), and N is length of the signal.

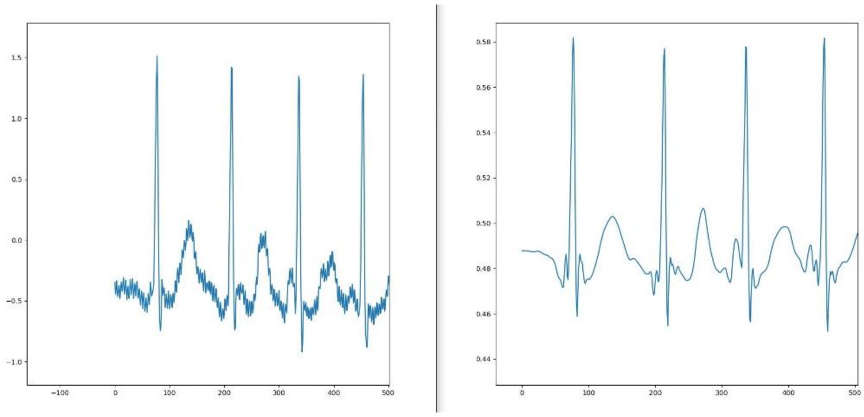
This study use 8 level DWT decomposition. Sym mother wavelet is selected because it has the highest SNR (Signal to noise ratio) compared to other mother wavelet [14]. The excact Sym5 mother wavelet function is selected because it gave good result on previous similar denoising case [8].

Normalization need to be applied on data because every data has different upper and lower bound. Normalization process resulted on data with 1 as upper bound limit and 0 as lower bound limit. Fig 2 shown AFIB, AFL, J, and Normal data before and after denoising and normazilation process applied.

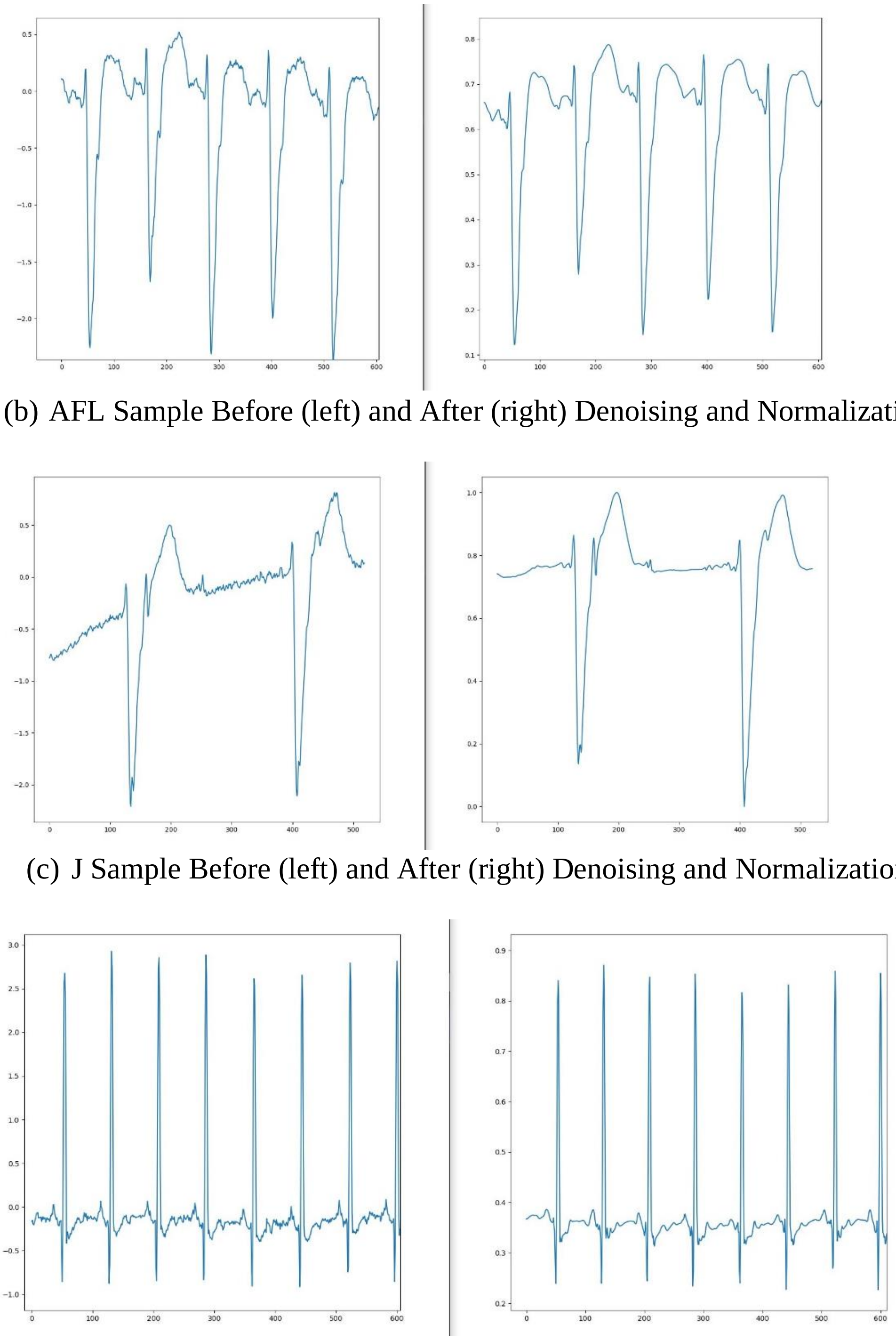


(a) Sample From MIT-BIH Atrial Fibrillation Database

(b) Sample MIT-BIH Normal Sinus Rhythm Database
FIGURE 1. Raw ECG Samples



(a) AFIB Sample Before (left) and After (right) Denoising and Normalization



(b) AFL Sample Before (left) and After (right) Denoising and Normalization

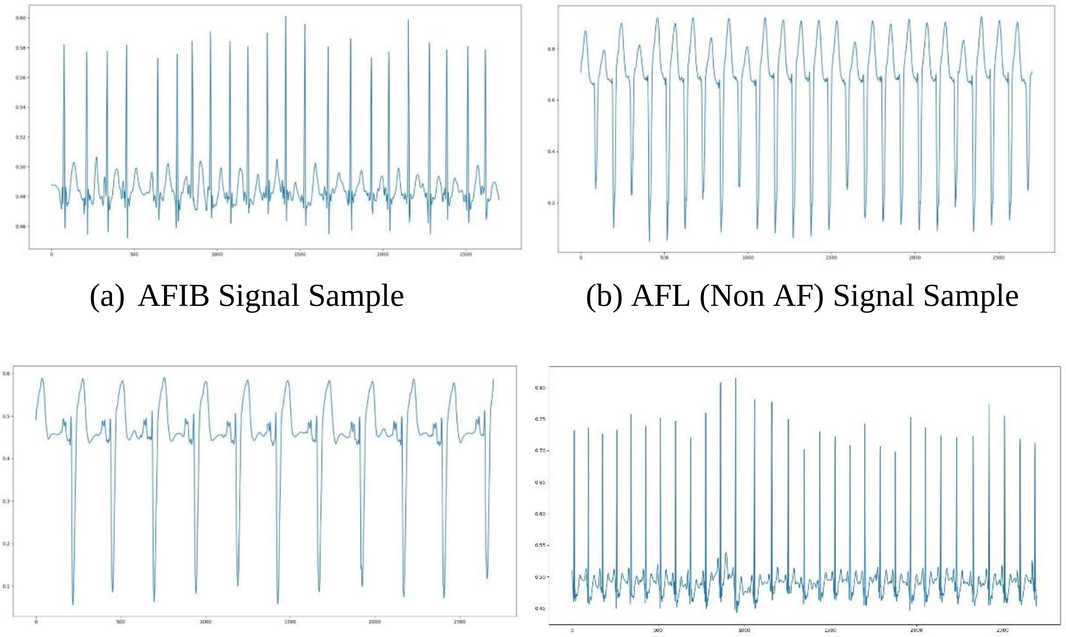
(c) J Sample Before (left) and After (right) Denoising and Normalization

(d) Normal Sample Before (left) and After (right) Denoising and Normalization

FIGURE 2. Samples Before and After Denoising and Normalization

After each of ECG recordings upper and lower bound normalized segmentation is applied to normalized each of ECG recordings length. Segmentation used to simplified the process of constructing input layer of 1 dimensional CNN model, to deal with varying sample frequencies from both of dataset, and to simplified 1 dimensional CNN calculation. Dataset is segmented at fixed length of 2700 indexes

[8]. Fig 3 shown the samples of AFIB, AFL, J, and Normal signal after segmentation. Table 1 shown total array data for every labels.



(a) AFIB Signal Sample

(b) AFL (Non AF) Signal Sample

(c) J (Non AF) Signal Sample

(d) NormalSignal Sample

FIGURE 3. Signal Samples After Segmentation

In order to create the dataset that can be use to train and test the 1 dimensional CNN model, train – test split is applied to segmented data. 90% of data is used as training data and 10% data is used as testing data. Table 2 shown amount of data for training and testing.

TABLE 1.
Total Array For Every Label

NO	Label	Total Signal Array	Total Label Array
1	AFIB	3.409	3.409
2	Non AFIB	320	320
3	Normal	4428	4428
Total		**8158**	**8158**

TABLE 2.
Total Array For Every Variables

Variabel	Total Array
X_train	7342
Y_train	7342
X_test	816
Y_test	816

3.3 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks or CNN is a deep learning architecture. In general, CNN is composed of 4 main layers, namely input layer, convolution layer, pooling layer, and fully connected layer [15]. CNN generally uses a kernel and 2-dimensional map filter which is generally used in image classification, but in the case of ECG recording data classification, a kernel and 1-dimensional map filter are used to maximize performance optimization and reduce computational complexity [16].

The convolutional layer functions to perform feature extraction and study the features contained in the input data [6]. The ReLU (*f*) activation function is used to map the eigenvectors (*x*) of the scanned features into a feature vector [15].

$$x_j^l = f(\sum_{i \in (M_j)} x_j^{l-1} * W_{ij}^l + b_j^l) \tag{3}$$

$$ReLU = f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{4}$$

Pooling layer serves to perform down sampling (*down*) feature dimension reduction on the feature map [6].

$$x_j^l = f(\beta_j^l * down\left(x_j^{l-1}\right) + b_j^l) \tag{5}$$

The softmax activation function is used to calculate the numerical output probability of the input. The numerical output probability itself represents a predetermined classification class [15] Softmax function can be expressed as,

$$Y_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}} \tag{6}$$

By using CNN 1 dimension, forward and backward propagation mechanism can be implemented effectively. This mechanism can also help make the output of a 1-dimensional CNN architecture more adaptive to the given input [16]. Forward propagation is a mechanism in which input data is processed sequentially by each layer on the network [16]. Forward propagation can be expressed as,

$$X_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) \tag{7}$$

By entering the input (X_k^l) into the activation function, it will get the output value (y_k^l), where $\downarrow ss$ is down sampling operation [16]. Activation function can be expressed as,

$$y_k^l = f\left(X_k^l\right) \tag{8}$$

$$s_i^l = y_i^l \downarrow ss \tag{9}$$

Backpropagation is a mechanism to get the best weight and bias configuration values on the network by propagation of the error (E_p) starting from the output layer ($\left[y_1^L, \dots, y_{N_l}^L\right]'$) iteratively using the Mean Squared error method (MSE) [16]. Backpropagation can be expressed as,

$$E_p = MSE\left(\boldsymbol{t^p}, \left[y_1^L, \dots, y_{N_l}^L\right]'\right) = \sum_{i=1}^{N_l}(y_i^L - t_i^p)^2 \tag{10}$$

Updates on the weights and biases of a neuron can be done by finding the value of the delta error derivative of the weights (∂w_{ik}^{l-1}) and biases (∂b_{ik}^l) [16]. Delta error function can be expressed as,

$$\frac{\partial E}{\partial w_{ik}^{l-1}} = \Delta_k^l \, y_i^{l-1} \tag{11}$$

$$\frac{\partial E}{\partial b_{ik}^{l}} = \Delta_k^l \tag{12}$$

The calculated bias sensitivity and weight values can be used to update the bias and weight values using the learning factor (ε) [16]. Caltulation to find new weight and new bias can be expressed as,

$$w_{ik}^{l-1}\,(t+1) = w_{ik}^{l-1}(t) - \varepsilon \frac{\partial E}{\partial w_{ik}^{l-1}} \tag{13}$$

$$b_k^l\,(t+1) = b_k^l(t) - \varepsilon \frac{\partial E}{\partial b_{ik}^{l}} \tag{14}$$

3.4 PROPOSED 1 DIMENSIONAL CNN MODEL

To find the best layer configuration for 1 dimensional CNN model, several layer convolutional configuration were tested on training and testing. Result for every convolutional layer configuration is shown in table 3.

TABLE 3.
Performance Comparison for Every Convolutional Configuration

MODEL NAME	CLASS	EVALUATION METRICS			
		Accuracy	Precission	Recall	F1 Score
3 CONVOLUTIONAL LAYER	AFIB	90.07%	81%	99%	89%
	Non AFIB	99.63%	92%	100%	96%
	Normal	90.2%	99%	83%	90%
4 CONVOLUTIONAL LAYER	AFIB	98.3%	100%	96%	98%
	Non AFIB	98.67%	76%	86%	86%
	Normal	99.15%	99%	99%	99%
5 CONVOLUTIONAL LAYER	AFIB	100%	100%	100%	100%
	Non AFIB	100%	100%	100%	100%
	Normal	100%	100%	100%	100%
6 CONVOLUTIONAL LAYER	AFIB	100%	100%	100%	100%
	Non AFIB	99.88%	97%	100%	99%
	Normal	99.88%	100%	100%	100%

The 1D CNN architecture (Figure 4) used consists of 14 layers, which are composed of 1 input layer, 5 Convolutional 1D layers, 4 Dropout layers, 1 max pooling layer, 1 Flattern layer, 1 Dense layer as a Fully Connected Layer, and 1 Dense layer as an Output Layer. The input layer functions as a layer that receives input from training data with array shape 1 x 2700. After the input layer is added 2 convolutional 1D layers with 64 filters and 3 x 1 kernels which function to study the features of the data and produce a feature map that will be used as input for the next layer. The next layer is a dropout layer and a 1D convolutional layer with 128 filters, a dropout layer is added to prevent the architecture from being overfitted by not updating the weights on 0.2 random neurons [17]. while the number of filters in the convolutional layer is increased so that the architecture can study features at a higher level. The max pooling layer is added to reduce the complexity of the feature map. The flatten layer is used to convert the input data into a new form that can be read by the fully connected layer. The fully connected layer used in this architecture is 128 nodes. The output layer used is a dense layer with a softmax activation function used for multiclass classification. The training optimizer used is the Adam optimizer. This optimizer can be used to adjust the learning rate of the architecture. Adam also excels in minimizing the loss function [17].

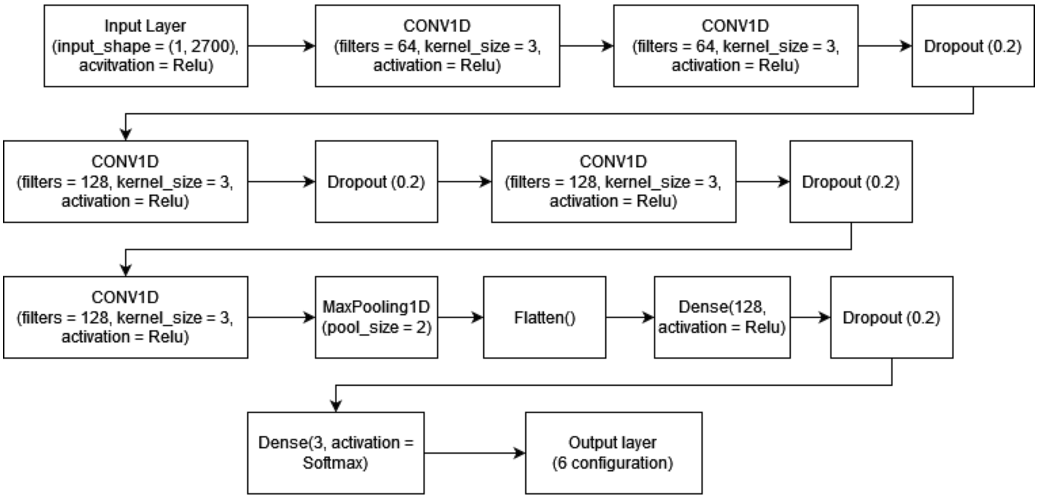

Input Layer (input_shape = (1, 2700), acvitvation = Relu)
CONV1D (filters = 64, kernel_size = 3, activation = Relu)
CONV1D (filters = 64, kernel_size = 3, activation = Relu)
Dropout (0.2)
CONV1D (filters = 128, kernel_size = 3, activation = Relu)
Dropout (0.2)
CONV1D (filters = 128, kernel_size = 3, activation = Relu)
Dropout (0.2)
CONV1D (filters = 128, kernel_size = 3, activation = Relu)
MaxPooling1D (pool_size = 2)
Flatten()
Dense(128, activation = Relu)
Dropout (0.2)
Dense(3, activation = Softmax)
Output layer (6 configuration)

FIGURE 4. Proposed 1 Dimensional CNN Architecture

There are 6 configurations for ouput layer in this architecture. Configuration variations are used to find the best output layer configuration. Every output layer configuration is using 100 epoch and variations of batch size and learning rate. Configurations details can be seen in table 4.

TABLE 4.
Configurations of Tuning Parameters

Name	*Learning Rate*	*Batch Size*
Model 1	10^{-3}	32
Model 2	10^{-4}	32
Model 3	10^{-3}	24
Model 4	10^{-4}	24

Model 5	10^{-3}	16
Model 6	10^{-4}	16

The data used to train the 1D CNN architecture is x_train which contains 7342 arrays of AFIB, Non AFIB, and Normal data, along with y_train which stores the actual labels of the 7342 arrays. CNN architecture will study the features of x_train and try to guess the label of each array, the classification results will be compared with the actual label in y_train, the better the classification results, the greater the accuracy value and the smaller the loss value of the architecture.

The data used to test the 1D CNN model is x_test which contains 816 arrays of AFIB, Non AFIB, and Normal data, along with y_test which stores the actual labels of the 816 arrays. CNN architecture will study the features of x_test and try to guess the label of each array, the classification results will be compared with the actual label in y_test, the better the classification results, the greater the accuracy value and the smaller the loss value of the model.

4. RESULTS AND DISCUSSION

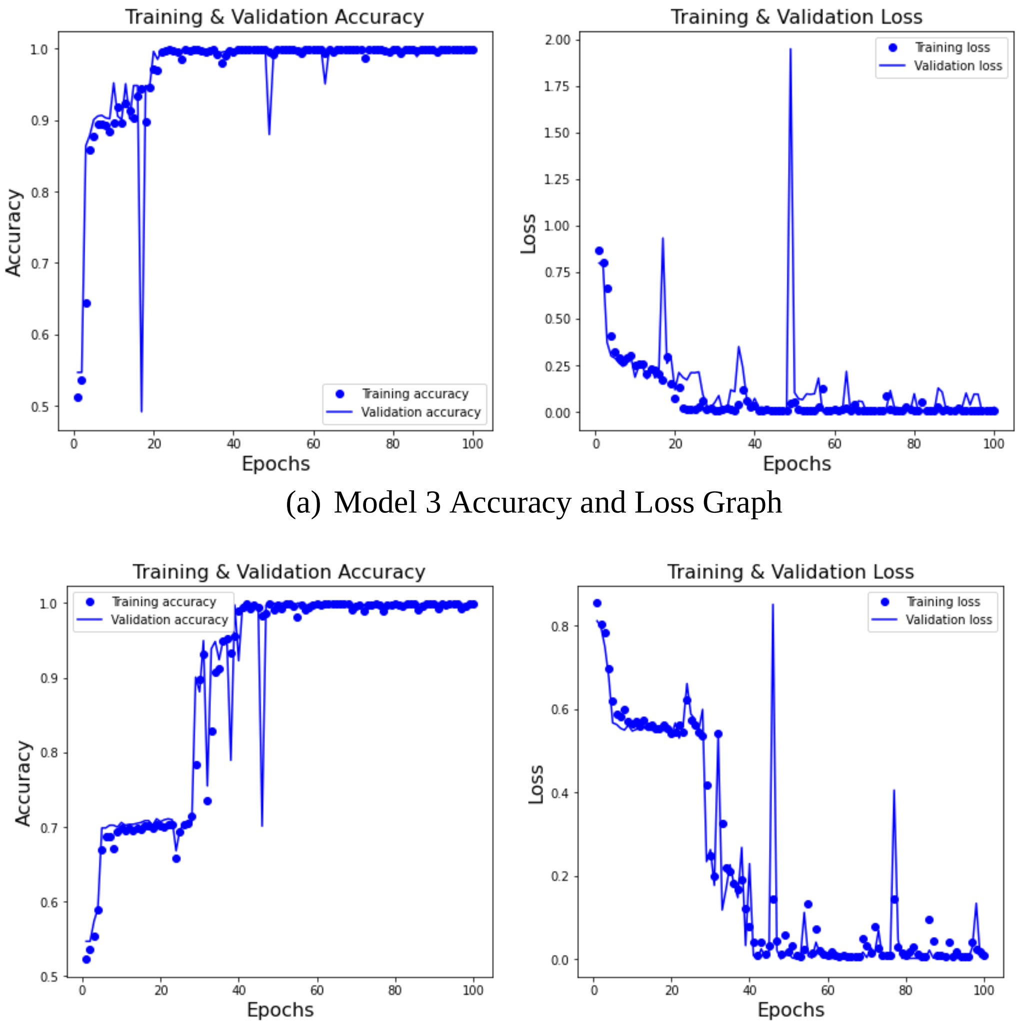
Variations of each configuration of the CNN 1D as shown in table 3, model shave different training and validation performance in the classification of Atrial Fibrillation in the ECG signal. To find the model with the best performance, measurements will be made on the measurement metrics for each 1D CNN model. The results of the comparison of the measurement metrics value from each model are presented in table 5.

TABLE 5.
Performance Comparison for Every Models

MODEL NAME	CLASS	EVALUATION METRICS			
		Accuracy	Precission	Recall	F1 Score
MODEL 1	AFIB	100%	100%	100%	100%
	Non AFIB	99.88%	100%	97%	99%
	Normal	99.88%	100%	100%	100%
MODEL 2	AFIB	92.77%	88%	94%	91%
	Non AFIB	99.8%	100%	97%	99%
	Normal	92.65%	95%	92%	93%
MODEL 3	AFIB	100%	100%	100%	100%
	Non AFIB	100%	100%	100%	100%
	Normal	100%	100%	100%	100%
MODEL 4	AFIB	99.75%	99%	100%	100%
	Non AFIB	99.88%	100%	97%	99%
	Normal	99.63%	100%	100%	100%
MODEL 5	AFIB	94.49%	88%	99%	93%
	Non AFIB	100%	100%	97%	100%
	Normal	94.49%	99%	92%	95%
MODEL 6	AFIB	100%	100%	100%	100%
	Non AFIB	100%	100%	100%	100%
	Normal	100%	100%	100%	100%

Based on table 4 which contains a comparison of the performance values of each CNN 1D model, Model 3 and Model 6 are CNN 1D models with the best performance compared to other models. However, if a comparison is made based on the accuracy and loss curve, the Model 3 curve in Figure 5(a) has a more stable accuracy and loss value at each epoch when compared to the Model 6 curve in Figure 5(b). Therefore, Model 3 with a learning configuration rate 10^{-3} and batch size 24 with

100% accuracy, 100% precision, 100% recall, and F1 Score 100% is the best CNN 1D model, confusion matrix of model 3 can be seen at fig 6.



(a) Model 3 Accuracy and Loss Graph

(b) Model 6 Accuracy and Loss Graph

FIGURE 5. Accuracy and Loss Graph for Model 3 and Model 6

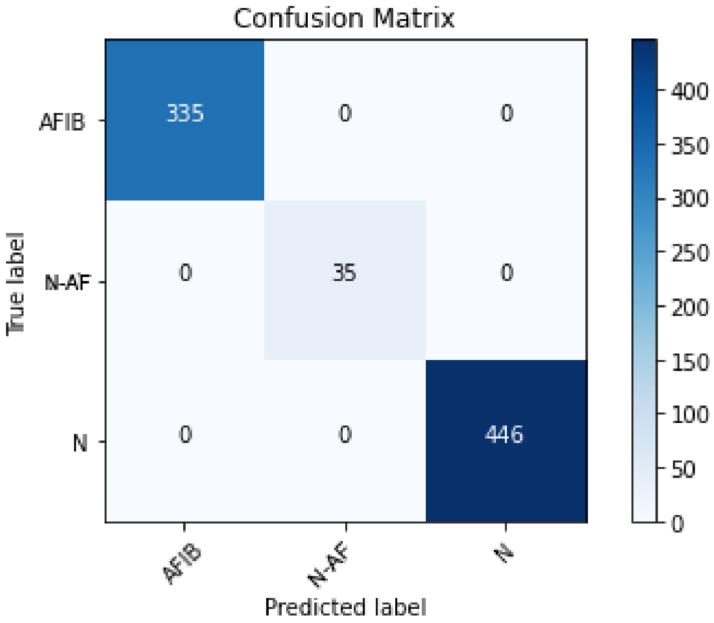


FIGURE 6. Confussion Matrix of Model 3

In addition the Model 3 as the proposed model is compared to another model with 3 classes classifiers (AFIB, Non – AFIB, Nomal). The Class Similarities classifier was chosen as the comparison criterion because models with the same number and type of classes are easier to find than models with datasets that are exactly the same with each other. The results of the comparison from different models are presented in table 6.

TABLE 6.
Performance Comparison for Every Proposed Models

Model	F1 Score			
	AFIB	**Non AFIB**	**Normal**	**Overall**
Proposed Model	100%	100%	100%	100%
Nurmaini et al.[8]	94.68%	98.41%	99.36%	97.48%
Fayyazifar Model [18]	83.87%	72.34%	91.39%	82.45%
Xiong et al.[6]	82%	90%	75%	82%
CNN based R-peak detector[3]	73%	56%	86%	71%

5. CONCLUSION

The conclusions obtained after analyzing the results of the classification of Atrial Fibtillation in ECG signals using deep learning were carried out as follows:

- The CNN 1D model implemented in the Atrial Fibrillation classification application has a learning rate configuration of 10^{-3} and a batch size of 24. The CNN 1D model is a model with the most stable accuracy and loss graphs and gets 100% accuracy, 100% precision, recall 100%, and F1 Score 100% in validation.

The research that has been done certainly has shortcomings in terms of datasets, model performance, and application performance. Suggestions that can be applied for further development are as follows:

- Use of datasets from various reliable sources and ensure the data for each category is balanced, to build a robust model.

- A wider study is needed on layer configuration and fine tuning of the 1D CNN model to obtain accuracy and loss curves from a model that is more stable and has better performance.

REFERENCES

[1] European Heart Network, “Atrial Fibrillation and Cardiovascular Diseases,” *European Heart Network*, 2015.

[2] S. Ghiasi, M. Abdollahpur, N. Madani, K. Kiani, and A. Ghaffari, “Atrial fibrillation detection using feature based algorithm and deep convolutional neural network,” in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4. doi: 10.22489/CinC.2017.159-327.

[3] B. S. Chandra, C. S. Sastry, S. Jana, and S. Patidar, “Atrial fibrillation detection using convolutional neural networks,” in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4. doi: 10.22489/CinC.2017.163-226.

[4] S. Liaqat, K. Dashtipour, A. Zahid, K. Assaleh, K. Arshad, and N. Ramzan, “Detection of Atrial Fibrillation Using a Machine Learning Approach,” *Information*, vol. 11, no. 12, 2020, doi: 10.3390/info11120549.

[5] A. Dey, “Machine Learning Algorithms : A Review,” 2016.

[6] Z. Xiong, M. K. Stiles, and J. Zhao, “Robust ECG signal classification for detection of atrial fibrillation using a novel neural network,” in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4. doi: 10.22489/CinC.2017.066-138.

[7] F. Murat *et al.*, “Review of Deep Learning-Based Atrial Fibrillation Detection Studies,” *Int J Environ Res Public Health*, vol. 18, no. 21, 2021, doi: 10.3390/ijerph182111302.

[8] S. Nurmaini *et al.*, “Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks,” *Future Generation Computer Systems*, vol. 113, 2020, doi: 10.1016/j.future.2020.07.021.

[9] X. Zhou, X. Zhu, K. Nakamura, and N. Mahito, “Atrial Fibrillation Detection Using Convolutional Neural Networks,” *2018 9th International Conference on Awareness Science and Technology (iCAST)*, pp. 84–89, 2018.

[10] Y. Xia, N. Wulan, K. Wang, and H. Zhang, “Detecting atrial fibrillation by deep convolutional neural networks.,” *Comput Biol Med*, vol. 93, pp. 84–92, Feb. 2018, doi: 10.1016/j.compbiomed.2017.12.007.

[11] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH Arrhythmia Database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001, doi: 10.1109/51.932724.

[12] Goldberger *et al.*, “MIT-BIH Normal Sinus Rhythm Database,” *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online].*, vol. 101, no. 23, pp. e215–e220, 2000.

[13] V. Khambhati and M. Patel, “Extraction of a Respiration Rate from ECG Signal using Discrete Wavelet Transform during Exercise,” 2017.

[14] M. Aqil, A. Jbari, and A. Bourouhou, “ECG Signal Denoising by Discrete Wavelet Transform,” *International Journal of Online Engineering (iJOE)*, vol. 13, p. 51, Aug. 2017, doi: 10.3991/ijoe.v13i09.7159.

[15] X. WAN, Z. JIN, H. WU, J. LIU, B. ZHU, and H. XIE, “HEARTBEAT CLASSIFICATION ALGORITHM BASED ON ONE-DIMENSIONAL CONVOLUTION NEURAL NETWORK,” *J Mech Med Biol*, vol. 20, no. 07, p. 2050046, 2020, doi: 10.1142/S0219519420500463.

[16] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech Syst Signal Process*, vol. 151, p. 107398, 2021, doi: https://doi.org/10.1016/j.ymssp.2020.107398.

[17] C.-H. Hsieh, Y.-S. Li, B.-J. Hwang, and C.-H. Hsiao, “Detection of Atrial Fibrillation Using 1D Convolutional Neural Network.,” *Sensors (Basel)*, vol. 20, no. 7, Apr. 2020, doi: 10.3390/s20072136.

[18] N. Fayyazifar, “An Accurate CNN Architecture For Atrial Fibrillation Detection Using Neural Architecture Search,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1135–1139. doi: 10.23919/Eusipco47968.2020.9287496.