Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 605~615
e-ISSN: 2548-6861

Ensemble Voting Method for Phonocardiogram Heart Signal Classification Using FFT Features

Adisaputra Zidha Noorizki [1]*, Heri Pratikno [2]*, Weny Indah Kusumawati [3]*
* Computer Engineering, Faculty of Technology and Informatics, Dinamika University, Surabaya, Indonesia
hi.zidha@gmail.com [1], heri@dinamika.ac.id [2], weny@dinamika.ac.id [3]

Article Info

Article history:

Received 2024-10-16
Revised 2024-10-31
Accepted 2024-11-06

Keyword:

Phonocardiogram,
Ensemble learning,
FFT features,
Soft voting,
LSTM,
GRU,
TCN.

ABSTRACT

Heart disease is still one of the leading causes of death worldwide, hence the need for effective diagnostic tools. Phonocardiogram (PCG) signals have been explored as a complementary approach to electrocardiogram (ECG) to detect cardiac abnormalities. This research investigates the classification of PCG signals using Fast Fourier Transform (FFT) features and deep learning models, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN). Hyperparameter tuning, particularly learning rate adjustment, is applied to optimize the performance of the models. The results show that the GRU and TCN models outperform the LSTM, achieving up to 92% accuracy at a learning rate of 0.0001. Ensemble learning with soft voting was also applied to combine the strengths of each model. Although the ensemble model showed strong performance with 92% accuracy and ROC AUC of 0.9636, it did not provide significant improvement over the base model. This finding highlights the importance of hyperparameter tuning in model optimization, with GRU and TCN showing slightly better performance in the time series classification task. This study concludes that ensemble learning offers stability but does not significantly improve classification accuracy beyond a well-tuned base model.

I. INTRODUCTION

The heart is a vital organ in the human body that is responsible for pumping blood throughout the body and returning it after passing through the lungs [1]. In addition to the natural aging process, there is a tendency for cardiac function to decline [2]. It is therefore imperative to maintain a healthy heart in order to reduce the risk of disease and ensure a good quality of life. In 2021, the World Health Organization (WHO) published data indicating that heart disease is the underlying cause of 17.8 million deaths annually, representing one-third of all global deaths [3]. Furthermore, the American Heart Association (AHA) has indicated that cardiovascular disease represents the leading cause of mortality globally [4]. In Indonesia, the Institute for Health Metrics and Evaluation (IHME) has reported that from 2014 to 2019, heart disease was the leading cause of mortality. Basic Health Research data indicates an increase in the incidence of heart disease to 1.5% in 2018, with the highest health costs reaching IDR 7.7 trillion, according to the Health Social Security Organizing Agency [5].

In the medical field, techniques such as electrocardiography (ECG) and phonocardiography (PCG) are employed for the monitoring of cardiac activity and the diagnosis of cardiovascular diseases. An ECG is capable of assessing the condition of the heart directly, but it is not always able to detect all abnormalities, such as heart murmurs [6][7]. This study employs PCG signal data as an alternative methodology. In the traditional practice of medicine, medical practitioners utilize a stethoscope to assess heart sounds. However, PCG offers a more comprehensive representation of the cardiovascular system, thereby making it a valuable tool in the diagnosis of heart disease [6][8][9]. PCG records the sounds and murmurs produced by the heart during its duty cycle. Murmurs represent additional sounds that indicate the presence of a cardiac disorder [6][7]. In normal conditions, PCG records the principal heart sounds, namely S1 and S2.

However, in abnormal conditions, murmurs that manifest as S3 and S4 sounds are also recorded [6][10].

The utilization of a stethoscope for conventional diagnosis is inherently challenging, as it necessitates the expertise and experience of medical professionals to accurately interpret heart sounds. Consequently, technology plays a pivotal role in streamlining and expediting the diagnostic process. Machine learning is a subfield of artificial intelligence (AI) that enables systems to learn from data and enhance their performance without the necessity for explicit programming [11]. In the field of machine learning, ensemble learning represents a powerful methodology that combines multiple models with the objective of improving prediction accuracy and robustness [12]. One of the most commonly utilized ensemble techniques is soft voting, wherein the prediction outcomes from each model are integrated and weighted to ascertain the final classification based on the most prevalent result [12][13]. This approach leverages the strengths of individual models, thereby enhancing the overall accuracy and reliability of predictions. Consequently, soft voting is gaining prominence in object classification tasks due to its capacity to optimize the performance of machine learning models.

A number of previous studies have demonstrated the significant potential of PCG signal classification. These studies employed a straightforward diagnostic approach that utilized Computer-Aided Diagnosis (CAD) and Multi-Layer Perceptron (MLP) classification, resulting in more efficient and rapid calculations with satisfactory accuracy. Subsequently, the amplified cardiac PCG signal was subjected to a Fast Fourier Transform (FFT), which yielded characteristics in the form of a fundamental frequency and maximum amplitude. A total of 55 data points were tested, resulting in an accuracy rate of 90%, a sensitivity rate of 80%, a positive predictive value (PPV) of 100%, and a negative predictive value (NPV) of 83.33% [9].

Moreover, a study entitled "Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings" concluded that the model was trained and tested using convolutional neural networks with bidirectional long short-term memory units as well as CNN with BiLSTM units individually. The highest performance was achieved using the CNN-BiLSTM network, with the following values for the statistical parameters: Cohen's kappa, accuracy, sensitivity, and specificity of 97.875%, 99.32%, 98.30%, and 99.58%, respectively. Furthermore, the model demonstrated an average area under the curve (AUC) of 0.998 when a 10-fold cross-validation scheme was employed [14].

The objective of recent research in this area is to classify PCG signals based on a feature extraction method that employs the Short Time Fourier Transform (STFT) and a classification method that utilizes a Convolutional Neural Network (CNN). A series of experiments was conducted to evaluate the efficacy of different windowing techniques, including Hamming, Hann, and Blackman-Harris, in the feature extraction phase, as well as various convolutional layer configurations in the classification phase. The combination of a Hamming window in the feature extraction process and four convolutional layers in the classification process yielded the most optimal results, with an accuracy rate of 88.11% [4].

A number of previous studies have contributed to the development of health diagnosis systems for detecting normal and abnormal conditions of the heart. This research project aims to enhance the performance of models in the classification of phonocardiogram (PCG) signals through the application of ensemble learning techniques. The methodology used is based on soft voting techniques, utilizing features extracted using Fast Fourier Transform (FFT), and applying Early Stopping techniques to optimize the model training process.

II. METHOD

The following flowchart is provided by the researcher as a visual representation of the research process. This diagram illustrates the steps and processes that will be carried out during the research.

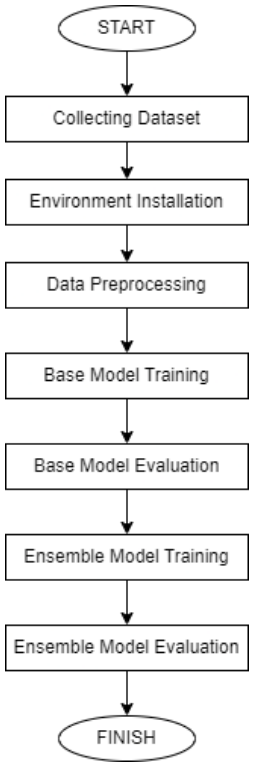

Image 1. Research flow diagram.

A. Collecting Dataset

This research employs open-source data from PhysioNet, specifically the dataset titled ‘Normal/Abnormal Heart Sound Recordings: the PhysioNet/Computing in Cardiology Challenge 2016’ [15]. This dataset comprises heart sound recordings, including normal and abnormal heart sounds,

collected as part of the 2016 Computing in Cardiology challenge. It consists of approximately 3240 audio files with the extension '.wav'. These heart sound recordings have been labelled and divided into segments for further analysis. Information regarding the labels and other metadata related to this dataset is presented in a separate Excel file, which provides a more detailed understanding of the characteristics of each sound recording.

B. Environment Installation

This research employs the use of the Python programming language, which is widely utilized in the field of computer science. At the outset of the research process, the investigator established the requisite working environment for data analysis and processing with Python. This entailed selecting the appropriate version of the programming language and installing libraries and other supporting dependencies through a package manager, designated 'pip'. Furthermore, a file, 'requirements.txt', was created to store all the necessary libraries and the versions employed during this research. This step was undertaken with the objective of facilitating the replication of the work environment by future researchers.

```
python --version
pip install --upgrade pip
pip install virtualenv

python -m venv myenvironment
myenvironment\Scripts\activate
pip install -r requirements.txt
```

Image 2. Environment installation scripts.

C. Data Preprocessing

The objective of this stage is to guarantee that the data utilized for model training is of optimal quality and representative. In this research, data preprocessing is conducted in five principal stages, including data segmentation, denoising, normalization, fast fourier transform (FFT), and augmentation.

1) *Segmentation:*

The segmentation process on the PCG signal entails the identification of discrete phases within the cardiac cycle, thereby facilitating the partitioning of the PCG signal into segments that correspond to each predefined class [16].

TABEL I
THE AMOUNT OF DATA BY CLASS DURING THE SEGMENTATION PROCESS

Class Name	Before process	After process
abnormal	2575	2464
normal	665	848

At this juncture, the data undergo a filtration process based on duration, with only data meeting a minimum duration of 15 seconds being retained. In the event that the duration of the recording exceeds 15 seconds, the signal will be divided into segments of equal length, with each segment being a multiple of 15 seconds. Consequently, the quantity of data within each class within the dataset will be subject to alteration.

The rationale behind the decision to assign a duration value of 15 seconds is based on a straightforward analysis that encompasses the determination of the minimum, maximum, and average duration values of all the data within the dataset. The minimum duration for the abnormal class is 6.61 seconds, while the normal class has a minimum duration of 5.31 seconds. The average recording duration for the abnormal class is 21.66 seconds, while the normal class has an average duration of 25.57 seconds. Given these values, a duration of 15 seconds was selected to prevent the deletion of voice recordings that were too brief.

2) *Denoising:*

The denoising process of PCG signals in this study employs one of the signal processing techniques, specifically filtering techniques, to reduce or eliminate noise that may be present in the signal [17].

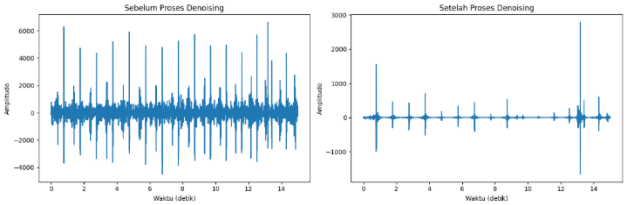

Image 3. Signal with normal class in the denoising process.

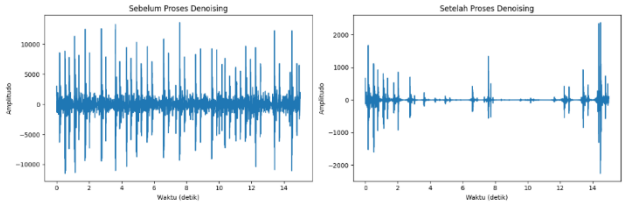

Image 4. Signal with abnormal class in the denoising process.

In this process, the signal containing noise is filtered first to retain the important information relevant for further analysis. Figure 3 shows the signal with normal class after denoising, while Figure 4 shows the signal with abnormal class in the same process. This denoising stage is an important part of preprocessing to improve the signal quality before it is applied to the model. After going through this process, the data becomes more meaningful, as shown in Figures 3 and 4. For both abnormal and normal signals, the difference can be seen more clearly than the condition before denoising, if visualized.

3) *Normalization:*

The main purpose of normalization is to guarantee that each feature or variable contributes equally to the analysis or modeling, regardless of initial scale differences [18]. This procedure helps prevent the dominance of variables that exhibit a wider range of values, thus allowing the model to see patterns with higher precision. In this study, the normalized data will be in the range of -1 to 1, thus

simplifying the analysis process and increasing the effectiveness of the algorithm.

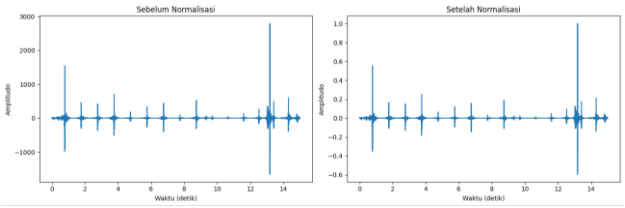

Image 5. Signal with normal class in the normalization process.

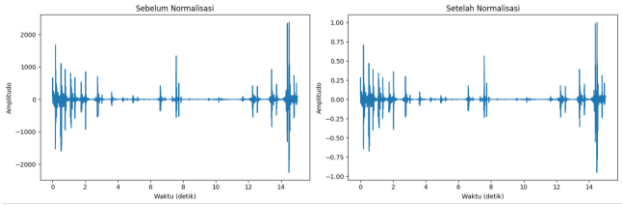

Image 6. Signal with abnormal class in the normalization process.

By normalizing the data, the model can better capture the relationship between variables, as each feature has a uniform scale. The normalization process also minimizes potential biases that can arise due to scale differences, which can affect model performance. This normalization ensures that all features contribute equally, allowing the model to learn from the data in a more efficient manner. Figure 5 shows signals with normal classes in the normalization process, while Figure 6 shows signals with abnormal classes in the normalization process. This process clarifies the differences between the classes of signals, so that relevant patterns can be better recognized.

4) *Fast Fourier Transform (FFT):*

The Fast Fourier Transform (FFT) is a method for converting signals from the time domain to the frequency domain. This transformation process enables the analysis of the frequency distribution of phonocardiogram (PCG) signals, which record heart sounds. The analysis of the frequency spectrum generated by the FFT provides deeper insight into the characteristics of heart sounds, including important information about the presence and strength of frequency peaks. Furthermore, the FFT can reveal the presence of murmurs, which are potential indications of abnormal conditions in the PCG signal [19]. Mathematically, the FFT or Fast Fourier Transform, can be represented as follows:

$$f_n = \frac{1}{N}\sum_{k=0}^{k=N-1} \hat{f}_k \,.\, e^{-i\frac{2\pi}{N}kn} \quad (1)$$

Formula 1 states that the signal value f_n is obtained by summing the product of the fourier transform values $\hat{f}_k$ and the weighting factor $e^{-i\frac{2\pi}{N}kn}$, where N is the total number of samples in the signal. This weighting factor is a complex exponential function that transforms each frequency component $\hat{f}_k$ in the frequency domain to the time domain. This process is performed for each value of n from 0 to $N-1$, resulting in a representation of the signal in the time domain based on the calculated frequency components [20].

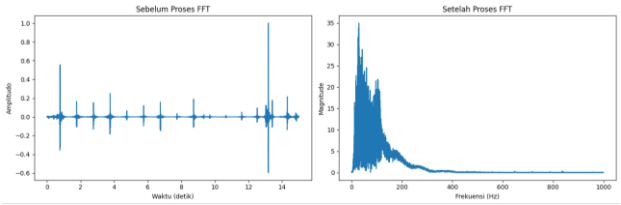

Image 7. Signal with normal class in the fft process.

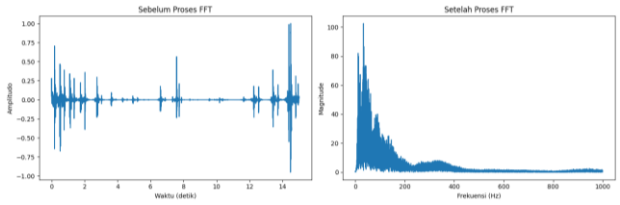

Image 8. Signal with abnormal class in the fft process.

By performing FFT, a more in-depth analysis of the heart sounds can be performed, allowing for more accurate identification of heart conditions. FFT provides a spectral overview of the PCG signal which helps in detecting the various frequency and amplitude components of heart sounds. Through spectrum analysis, it can provide an indication of abnormalities or pathological conditions, such as heart murmurs or other abnormal heart sounds.

Figure 7 shows a normal-grade signal in the FFT process, where the frequency spectrum displays a pattern consistent with healthy heart sounds. In contrast, Figure 8 shows a signal with an abnormal class, which may exhibit additional frequencies or unusual peaks, which can be a sign of heart abnormalities. This FFT process is essential for interpreting changes in heart sound characteristics, as shifts or additions of certain frequencies can provide critical information regarding the patient's heart condition.

5) *Augmentation:*

The objective of data augmentation techniques is to enhance model performance by introducing additional variation into a dataset. This is achieved by manipulating, transforming, or duplicating existing data. The aim is to ensure that the model can accurately recognize each class, without exhibiting bias towards classes that have more data. This phenomenon is known as the imbalance class condition.

TABEL II
THE AMOUNT OF DATA BY CLASS DURING THE AUGMENTATION PROCESS

Class Name	Before process	After process
abnormal	2464	1656
normal	848	1656

In order to address the issue of imbalance class, the researcher will employ a technique known as SMOTE (Synthetic Minority Over-sampling Technique). Synthetic

Minority Over-sampling Technique (SMOTE) is a method used to augment the number of instances in a minority class by generating a new synthetic sample based on existing data [21][22]. In this study, the researcher determines the mean value of the sum of the data in the dataset to be the final value of the dataset prior to the training process with the machine learning model and then divides it into two. This approach aims to ensure that the data in the class with the least amount is not too different from the data in the class with the most amount, thus balancing the distribution of data between classes (normal and abnormal).

D. Base Model Training

During the model training process, we will use several callback functions to streamline the training time of each model, such as the 'ReduceLROnPlateau', 'EarlyStopping', and 'ModelCheckpoint' callbacks. The ReduceLROnPlateau callback serves to dynamically reduce the learning rate if the evaluation metric does not show improvement within a certain number of epochs. The EarlyStopping callback is used to stop training early if the evaluation metric does not improve in a certain number of consecutive epochs, thus preventing overfitting. Meanwhile, the ModelCheckpoint callback serves to save the best model weights during training based on the specified evaluation metric.

After the data preprocessing stage in the previous step, where the data has been prepared to be optimized as model input, researchers at this stage determine the model architecture configuration. This determination includes the number of layers and the number of neurons in each layer that will be used in the model.

1) *Long Short Term Memory (LSTM):*

LSTM or Long Short-Term Memory is a type of artificial neural network architecture designed to overcome challenges in long and short-term memory in the analysis and prediction of sequential data. First introduced by Hochreiter and Schmidhuber in 1997, LSTM is able to handle vanishing gradient and exploding gradient problems that often occur in conventional neural networks when processing long sequences of data [23][24]. LSTMs use specialized memory units called 'cells', which have the ability to store, delete, and access information over long periods of time. This ability allows LSTM to capture long-term dependencies in data sequences, making it very effective in modeling complex patterns and temporal relationships [24].

After the data preprocessing process in the previous stage, where the data has been prepared to be optimized as model input, researchers at this stage determine the configuration of the model architecture. This determination includes the number of layers and the number of neurons in each layer used in the LSTM model. Parameters such as the number of hidden layers and the number of neurons per layer are optimized to achieve the best performance.

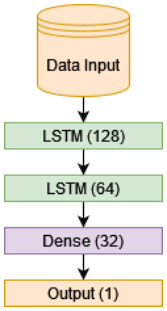

Image 9. Architecture of LSTM algorithm.

2) *Gated Recurrent Unit (GRU):*

GRU or Gated Recurrent Unit is one type of network architecture in Recurrent Neural Network designed to overcome some of the constraints in the LSTM model while still maintaining the ability to capture temporal dependencies on sequential data. Compared to LSTM, GRU has a simpler structure with only two main gates, namely reset gate and update gate [23][25]. GRU can adaptively learn and store relevant information from the past, so it can overcome vanishing gradient or exploding gradient problems. In addition, GRU is effectively able to capture temporal dependencies in sequential data. During training, GRU parameters are updated through a gradient-based learning process to improve the model's ability to understand patterns and relationships in temporal data.

The GRU model, which is a simpler variant of the LSTM, is also used in this study. The model configuration includes adjustments to the number of layers and neurons to ensure efficiency and effectiveness in processing heart signal sequence data.

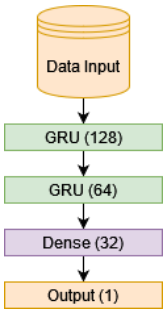

Image 10. Architecture of GRU algorithm.

3) *Temporal Convolutional Network (TCN):*

TCN or Temporal Convolutional Network is an artificial neural network architecture specifically designed to handle sequential data efficiently. Unlike the Convolutional Neural Network (CNN) that is commonly applied to image data, TCN is designed to capture patterns and temporal dependencies in sequential data. One of the key components in TCN is the dilated convolution layer, which allows the model to extend its temporal range without sacrificing computational efficiency [26]. The advantages of TCN make it a highly efficient and effective choice in a variety of tasks, including

time series prediction, time series data analysis, and other tasks involving sequential data [27]. With its ability to capture complex temporal information, TCN makes an important contribution in the development of models that can understand and process time series data with high accuracy.

TCN is used to process sequential data with a convolutional approach. The number of layers and filters are determined and optimized to handle long-term dependencies efficiently.

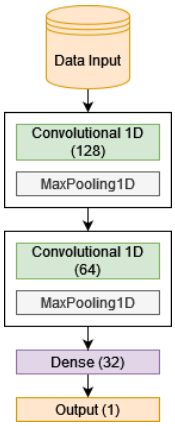

Image 11. Architecture of TCN algorithm.

E. Ensemble Model Training

The ensemble learning method is a machine learning approach in which a number of models or base learners are combined to produce more accurate predictions [28]. By combining the results of several models, ensemble learning has the potential to reduce overfitting on data [28][29]. One method that is often used in ensemble learning, especially in the stage of combining prediction results, is soft voting. In soft voting, each model provides predictions by generating probability values for each class in the classification [30]. The final prediction is taken by calculating the weighted average of the class probabilities generated by each model. Mathematically, soft voting can be represented as follows:

$$P(c_i) = \frac{\sum_{j=1}^{M} w_j \cdot P_j(c_i)}{\sum_{j=1}^{M} w_j} \quad (2)$$

The experiment was conducted twice with different weights: 1:1:1 and 2:1:1, where weight 2 is given to the best performing model based on the evaluation metric. The predicted probability of each model is weighted according to its contribution to the ensemble. These weights are normalized so that they total to 1. The predicted probability of each model is then multiplied by the corresponding weights and summed to get the ensemble prediction. This combined probability is compared with a threshold of 0.5 to determine the final class. This approach is expected to improve prediction accuracy by utilizing the strengths of each underlying model contributing to the ensemble.

F. Model Evaluation

This research utilizes one of the popular methods for assessing the performance of models that have been designed and trained, namely the confusion matrix. By analyzing the values in the confusion matrix, several commonly used evaluation metrics can be calculated to provide a comprehensive overview of model performance, including accuracy, precision, recall, f1-score, and ROC-AUC. Mathematically, it can be written as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{F1-score} = \frac{2\times precision\times recall}{precision+recall} \quad (6)$$

$$\text{ROC-AUC} = \sum_{i=1}^{n-1} \left(\frac{TPR_i+TPR_{i+1}}{2}\right).(FPR_{i+1}-FPR_i) \quad (7)$$

In the performance evaluation of classification models, some key terms from the confusion matrix are often used to calculate metrics that provide deep insights into the model's performance. True Positive (TP) refers to the number of cases where the model correctly predicts a positive class, while True Negative (TN) refers to the number of cases where the model accurately predicts a negative class. False Positive (FP) is when the model incorrectly predicts a positive class when it is actually negative, while False Negative (FN) occurs when the model misclassifies a negative class when it should be positive [31].

Two additional metrics that are often calculated from the confusion matrix are True Positive Rate (TPR) and False Positive Rate (FPR). TPR, also known as Recall, describes the model's ability to detect positive cases, calculated as the ratio of TP to the total number of positive cases (TP + FN). On the other hand, FPR measures the proportion of negative cases that are misclassified as positive, which is calculated as the ratio of FP to the total number of negative cases (FP + TN). These metrics are essential in providing a more comprehensive picture of the model's effectiveness in correctly classifying the data, especially in contexts involving class imbalance or the disparate impact of misclassification.

III. Result and Discussion

After describing the methods used in this research, the results of the research can be seen as follows

A. Basic Model Training Process Results

This study conducted the base model training process four times for each model architecture. The experiments were differentiated based on varying learning rate values: 0.01, 0.001, 0.0001, and one additional experiment that utilized

automatic learning rate adjustment through the Keras library's ReduceLROnPlateau callback which will be abbreviated to RLRO, which automatically determines the most optimal learning rate for the training process. The performance indicator used as a parameter for the EarlyStopping and RLRO callbacks is based on the validation loss (val_loss) value, with the patience value set to 5 for EarlyStopping and 2 for RLRO. The minimum learning rate in RLRO is set at 1.0000e-04. In addition, the class writing in the table will be abbreviated to optimize the placement of data in the table, the normal class will be abbreviated as 'N', while the abnormal class will be abbreviated as 'AN'.

1) Long Short Term Memory (LSTM):

Based on the results given in Table III regarding the training of the LSTM model, it can be concluded that the hyperparameter settings, especially the learning rate and the number of epochs, have a significant influence on the performance of the model. The model with a learning rate of 0.01 shows good results with 83% accuracy after 8 epochs, where precision and recall reach 0.85 and 0.81 respectively, with an F1-score of 0.83. Increasing the learning rate to 0.001 resulted in better performance, with accuracy reaching 87% and improvements in precision and recall, to 0.86 and 0.89, respectively, and F1-score 0.87. The best performance was achieved with a learning rate of 0.0001 after 15 epochs, where the model achieved 91% accuracy, precision 0.90, and recall 0.94, resulting in an F1-score of 0.91. Although using a learning rate of 1.0000e-04 gave slightly lower results compared to 0.0001, the model still performed well with 87% accuracy, precision 0.86, and recall 0.88. In general, there is a corresponding increase in precision, recall, and F1-score as the learning rate decreases and the number of epochs increases, which indicates that the model can better identify and classify classes more precisely.

TABEL III
LSTM MODEL TRAINING RESULTS

learning rate	Class	epoch	accu-racy	preci-sion	recall	F1-score
0.01	N	8	0.83	0.85	0.81	0.83
	AN			0.82	0.85	0.84
0.001	N	8	0.87	0.86	0.89	0.87
	AN			0.88	0.85	0.87
0.0001	N	15	0.91	0.90	0.94	0.91
	AN			0.93	0.89	0.91
RLRO 1.0000e-04	N	96	0.87	0.86	0.88	0.87
	AN			0.88	0.86	0.87

2) Gated Recurrent Unit (GRU):

Based on the results presented in Table IV regarding the GRU model training, it can be seen that the hyperparameter settings, especially the learning rate and the number of epochs, also significantly affect the model performance. At a learning rate of 0.01, the model shows an accuracy of 82% after 18 epochs, with precision and recall values reaching 0.87 and 0.77, respectively, and an F1-score of 0.81. When the learning rate was reduced to 0.001, the model performance improved with accuracy reaching 87% after 16 epochs, where precision and recall increased to 0.85 and 0.90, and F1-score 0.88. A smaller learning rate setting of 0.0001 gave the best results with 92% accuracy after 14 epochs. At this point, the model obtained a precision of 0.90 and recall of 0.95, resulting in an excellent F1-score of 0.92. Additionally, the use of a learning rate of 1.0000e-04 showed competitive performance, albeit with a slightly lower accuracy of 89% after 48 epochs, and with a precision of 0.87 and recall of 0.93, resulting in an F1-score of 0.90. Overall, there was a clear improvement in all evaluation metrics as the learning rate decreased and the number of epochs was adjusted, reflecting the model's ability to better recognize and classify the classes.

TABEL IV
GRU MODEL TRAINING RESULTS

learning rate	Class	epoch	accu-racy	preci-sion	recall	F1-score
0.01	N	18	0.82	0.87	0.77	0.81
	AN			0.79	0.88	0.83
0.001	N	16	0.87	0.85	0.90	0.88
	AN			0.89	0.84	0.87
0.0001	N	14	0.92	0.90	0.95	0.92
	AN			0.95	0.89	0.92
RLRO 1.0000e-04	N	48	0.89	0.87	0.93	0.90
	AN			0.92	0.86	0.89

3) Temporal Convolutional Network (TCN):

Based on the results presented in Table V regarding the training of the TCN (Temporal Convolutional Network) model, it can be seen that the hyperparameter settings, including the learning rate and number of epochs, have a significant influence on the performance of the model. At a learning rate of 0.01, the model achieved 89% accuracy after 10 epochs, with precision and recall values reaching 0.91 and 0.86 respectively, and F1-score of 0.88. When the learning rate was reduced to 0.001, the model performance improved slightly with 90% accuracy after 12 epochs, and precision and recall reaching 0.89 and 0.91, resulting in an F1-score of 0.90.

TABEL V
TCN MODEL TRAINING RESULTS

learning rate	Class	epoch	accu-racy	preci-sion	recall	F1-score
0.01	N	10	0.89	0.91	0.86	0.88
	AN			0.87	0.91	0.89
0.001	N	12	0.90	0.89	0.91	0.90
	AN			0.91	0.89	0.90
0.0001	N	11	0.92	0.90	0.95	0.92
	AN			0.94	0.90	0.92
RLRO 1.0000e-04	N	9	0.85	0.83	0.90	0.86
	AN			0.89	0.81	0.85

Furthermore, with a learning rate of 0.0001, the model showed the best results with 92% accuracy after 11 epochs, where precision reached 0.90 and recall 0.95, resulting in an F1-score of 0.92. However, using a learning rate of 1.0000e-

04 resulted in a lower accuracy (85%) after 9 epochs, with a precision of 0.83 and recall of 0.90, resulting in an F1-score of 0.86. This shows that although a smaller learning rate can give better results in the previous settings, not all settings give optimal results, and too small a learning rate can lead to suboptimal performance. From this analysis, it can be seen that the TCN model shows consistent performance improvement as the learning rate decreases up to a certain point, with accuracy and other evaluation metrics increasing.

B. Performance Evaluation of the Base Model

Based on Tables III, IV, and V, it can be seen that the effect of hyperparameter settings, especially learning rate and number of epochs, is very significant on the performance of LSTM, GRU, and TCN models in training. All models show improved performance as the learning rate decreases, with the best results generally obtained at a learning rate of 0.0001. The LSTM model achieved the highest accuracy of 91% after 15 epochs with a learning rate of 0.0001, showing a precision value of 0.90 and recall of 0.94, and an F1-score of 0.91. Although using a learning rate of 1.0000e-04 gave lower results, the model still showed an accuracy of 87%. The GRU model showed good performance, with 92% accuracy after 14 epochs at a learning rate of 0.0001, where precision and recall reached 0.90 and 0.95, resulting in an F1-score of 0.92. The use of a learning rate of 1.0000e-04 also resulted in 89% accuracy after 48 epochs, which still showed competitive performance. Meanwhile, the TCN model showed consistent results, with 92% accuracy after 11 epochs using a learning rate of 0.0001, as well as precision of 0.90 and recall of 0.95, resulting in an F1-score of 0.92. Although the learning rate of 1.0000e-04 showed lower results (85%), the model still showed an overall improvement in performance.

From this analysis, it can be seen that all three models show positive results with proper hyperparameter adjustment, but GRU and TCN tend to perform slightly better than LSTM in some settings. All models show that reducing the learning rate helps improve accuracy and other evaluation metrics, reflecting that the models can learn better with finer settings. In general, optimal hyperparameter settings, including learning rate and number of epochs, are key to achieving the best performance in all three models. These results confirm that GRU and TCN models may be more effective in solving classifier tasks compared to LSTM in certain settings. Therefore, it is recommended to apply further cross-validation and tuning strategies to these models to ensure generalization and optimal performance in real applications.

C. Results of Ensemble Model Training Process

After completing the training process on each model individually and obtaining the performance results for each trial, we then proceeded to the ensemble model training process. In this case, the three models that previously decided the classification results separately will be combined to produce one final decision. This integration process uses an ensemble learning technique with a soft voting approach.

The ensemble model training was conducted through several experiments, with groupings based on the number of epochs and variations in the weights given to each base model. In addition, at this stage, researchers also added a new metric to assess the performance of the ensemble model, namely ROC AUC. The addition of the ROC AUC parameter is expected to provide a more thorough assessment of the ensemble model performance, because ROC AUC calculates the area under the ROC curve. The ROC curve is a plot that illustrates the trade-off between true positive rate and false positive rate at various thresholds. By using ROC AUC, researchers can get a comprehensive view of the model's ability to distinguish between positive and negative classes, and help identify the model's performance under various conditions. This provides a more complete view of the effectiveness of the ensemble model built.

TABEL VI
ENSEMBLE MODEL TRAINING RESULTS

learning rate base model	class	accu-racy	preci-sion	recall	F1-score	ROC AUC
0.01	N	0.86	0.87	0.86	0.86	0.9356
	AN		0.86	0.87	0.87	
0.001	N	0.91	0.89	0.93	0.91	0.9533
	AN		0.93	0.88	0.90	
0.0001	N	0.92	0.90	0.95	0.92	0.9636
	AN		0.94	0.89	0.92	
RLRO	N	0.89	0.87	0.93	0.90	0.9496
	AN		0.92	0.85	0.89	

Table VI presents the training results of the ensemble model, which displays the performance metrics of accuracy, precision, recall, F1-score, and ROC AUC across different learning levels and base models. At a learning rate of 0.01, the model achieved an accuracy of 0.86, with an ROC AUC of 0.9356. With a learning rate of 0.001, the performance improved, resulting in an accuracy of 0.91, with a ROC AUC of 0.9533. The highest performance was observed at a learning rate of 0.0001, where accuracy reached 0.92, and ROC AUC peaked at 0.9636. The RLRO model also performed well, achieving an accuracy of 0.89 and an ROC AUC of 0.9496. Overall, these results show that optimizing the learning rate significantly improves the classification performance of the ensemble model.

D. Ensemble Model Performance Evaluation

After conducting the ensemble model training and collecting the performance metrics, the next step is to evaluate the effectiveness of the ensemble model in a detailed manner. The evaluation focuses on understanding the model's performance not only through traditional metrics but also by examining the behavior of the model under various conditions.

The ensemble model's performance metrics, as indicated in Table VI, provide a clear picture of how the model performs at different learning rates. At a learning rate of 0.01, the model achieved an accuracy of 0.86, with a precision of 0.87, recall

of 0.86, F1-score of 0.86, and ROC AUC of 0.9356. Although the model demonstrated decent performance at this learning rate, there was still room for improvement, particularly in recall. With a learning rate of 0.001, a notable improvement was observed in all metrics: accuracy rose to 0.91, precision to 0.89, recall to 0.93, F1-score to 0.91, and ROC AUC to 0.9533. This indicates an enhanced ability to identify true positive instances. The highest performance was recorded at a learning rate of 0.0001, where the accuracy reached 0.92, precision improved to 0.90, recall increased to 0.95, F1-score was 0.92, and ROC AUC peaked at 0.9636. These results show that finer adjustments in the learning rate can lead to substantial gains in classification capabilities. The RLRO model also showed respectable performance, achieving an accuracy of 0.89, precision of 0.87, recall of 0.93, F1-score of 0.90, and ROC AUC of 0.9496, though it did not outperform the other configurations in terms of accuracy and ROC AUC.

This research investigates the performance of three different deep learning architectures, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Temporal Convolutional Network (TCN), with a focus on how hyperparameter tuning, specifically learning rate and number of training epochs, impacts their classification capabilities on heart signal fft feature objects. Through rigorous experiments with various learning rates, including the use of automatic learning rate adjustment (ReduceLROnPlateau), this research provides a comprehensive insight into the effectiveness of each model for classification tasks. The results consistently show that fine-tuning the learning rate has a significant impact on model performance across all architectures.

Specifically, decreasing the learning rate generally improves accuracy, precision, recall, and F1-score. Among each model, the GRU and TCN architectures showed superior performance compared to LSTM in certain settings, achieving the highest accuracy of 92% at a learning rate of 0.0001. These findings suggest that GRU and TCN are potentially more effective in classification tasks, especially when the learning rate is fine-tuned. However, LSTM remains a competitive choice, achieving 91% accuracy under optimal conditions.

The ensemble learning approach further strengthens these findings. By integrating the predictions of the LSTM, GRU, and TCN models through soft voting techniques, the ensemble model achieved strong performance, with a peak accuracy of 92% and a maximum ROC AUC of 0.9636 at a learning rate of 0.0001. The inclusion of ROC AUC as a metric provides a more nuanced understanding of the model's performance, especially in evaluating the model's ability to balance the correct positive and negative rates across various thresholds. This metric confirms that the ensemble approach, with proper tuning of the learning rate, offers a reliable and effective classification solution.

The evaluation of ensemble models shows that combining the strengths of multiple architectures has not been able to significantly improve performance compared to individual models. The results also underscore the potential of GRU and TCN models, which slightly outperform LSTM in this study, suggesting that these architectures are more adept at handling the complexity of time series data classification.

E. Discussion

While this study provides important insights into the effectiveness of various deep learning architectures in classifying PCG signals, there are some limitations that need to be noted. Firstly, the size of the dataset used in this study is limited, which may affect the generalizability of the model. With a small amount of data, the model is at risk of overfitting, which is a state where the model learns very well on training data but cannot adapt to new data. Therefore, it is recommended to collect and use a larger and more diverse dataset to increase the validity of the research results. In addition, the quality of the existing dataset is not optimal enough to distinguish between normal and abnormal PCG signals. The unevenness in audio duration and less than ideal sampling caused the preprocessing applied to be ineffective, especially due to the high level of noise.

Secondly, the challenges in PCG signal processing also need to be taken into account. Although preprocessing steps such as denoising, normalization and windowing have been applied to improve signal quality, the diverse complexity of PCG signals can make pattern identification difficult. Noise and artifacts resulting from the recording device or environmental conditions can affect classification results, thus demanding the use of more robust preprocessing methods.

Thirdly, it is also worth noting the limitations of the chosen deep learning model architecture. Although LSTM, GRU, and TCN perform well, each has its own weaknesses. For example, LSTM is often slower in the training process compared to GRU and TCN due to the complexity of its structure. Conversely, although GRU is more efficient, it may not always be able to capture long-term dependencies in the data as well as LSTM. While TCN, although performing well, may be less effective in dealing with very long sequences without additional strategies such as residual connection or dilation.

Given these limitations, future research should include larger datasets, more advanced preprocessing methods, as well as exploration of more diverse model architectures. Future research could also consider using more complex ensemble techniques to harness the power of different models.

IV. CONCLUSIONS

This research confirms that hyperparameter optimization, especially the learning rate, is key in maximizing the performance of LSTM, GRU, and TCN models for classification tasks. Experimental results show that GRU and TCN models consistently outperform LSTM; however, all models exhibit significant performance improvements when the learning rate is set appropriately.

Although the ensemble approach can enhance stability and overall classification performance, no significant improvement over the base model (single model), particularly in terms of accuracy, was observed. These findings provide practical implications for future research and applications, particularly in the field of PCG signal classification. Future work should focus on hyperparameter adjustment, more thorough cross-validation, and exploration of more diverse model architectures and ensemble methods to create more robust and generalized models.

However, in the context of this study, refinements to the base model remain more effective in improving performance than the use of ensemble learning. Thus, the optimal approach depends on specific model settings and the particular case at hand. Additionally, the limitations identified in this study, such as the size and quality of the dataset, highlight the need for further investigation with larger and more diverse datasets to validate and enhance the findings.

REFERENCES

[1] S. Handayani, *Analogi dan Fisiologi Tubuh Manusia*. Media Sains Indonesia dan Penulis, 2021. Accessed: Jul. 01, 2024. [Online]. Available: http://repository.stikes-yogyakarta.ac.id/id/eprint/24/3/Buku%20Anatomi%20dan%20Fisiologi%20Tubuh%20Manusia.pdf

[2] O. Maria Pujiastuti, I. Hizkia, W. Munthe, and S. Santa Elisabeth Medan, "Gambaran Tekanan Darah Pada Masyarakat Yang Mengikuti Senam Jantung Sehat Di Rambung Merah Tahun 2022," 2023. [Online]. Available: http://bajangjournal.com/index.php/JCI

[3] WHO and Q. Mattingly, "Cardiovascular diseases." Accessed: Jul. 03, 2024. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases

[4] M. Alwi, A. Amal, D. Zulherman, R. Widadi, and P. Korespondensi, "Klasifikasi Sinyal Phonocardiogram Menggunakan Short Time Fourier Transform Dan Convolutional Neural Network," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 2, pp. 237–244, Apr. 2023, doi: 10.25126/jtiik.2023105424.

[5] Kemenkes, "Satu dari Tiga Kematian Disebabkan oleh Jantung, Ayo Cegah serangan jantung." Accessed: Jul. 03, 2024. [Online]. Available: https://upk.kemkes.go.id/new/satu-dari-tiga-kematian-disebabkan-oleh-jantung-ayo-cegah-serangan-jantung

[6] J. Prince *et al.*, "Deep Learning Algorithms to Detect Murmurs Associated With Structural Heart Disease," *J Am Heart Assoc*, vol. 12, no. 20, Oct. 2023, doi: 10.1161/JAHA.123.030377.

[7] M. Wang, B. Guo, Y. Hu, Z. Zhao, C. Liu, and H. Tang, "Transfer Learning Models for Detecting Six Categories of Phonocardiogram Recordings," *J Cardiovasc Dev Dis*, vol. 9, no. 3, Mar. 2022, doi: 10.3390/jcdd9030086.

[8] H. Li *et al.*, "A fusion framework based on multi-domain features and deep learning features of phonocardiogram for coronary artery disease detection," *Comput Biol Med*, vol. 120, May 2020, doi: 10.1016/j.compbiomed.2020.103733.

[9] Y. Triyani, W. Khabzli, N. Harpawi, and W. Styorini, "Computer Aided Diagnosis (CAD) untuk Phonocardiogram (PCG) Berbasis Fast Fourier Tranform," *Jurnal ELEMENTER*, vol. 7, pp. 66–75, May 2021. Available: https://jurnal.pcr.ac.id/index.php/elementer/

[10] S. Y. Lee, P. W. Huang, J. R. Chiou, C. Tsou, Y. Y. Liao, and J. Y. Chen, "Electrocardiogram and Phonocardiogram Monitoring System for Cardiac Auscultation," *IEEE Trans Biomed Circuits Syst*, vol. 13, no. 6, pp. 1471–1482, Dec. 2019, doi: 10.1109/TBCAS.2019.2947694.

[11] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," Apr. 01, 2020, *MDPI AG*. doi: 10.3390/info11040193.

[12] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," Oct. 01, 2022, *Elsevier Ltd*. doi: 10.1016/j.engappai.2022.105151.

[13] A. R. Barzani, P. Pahlavani, and O. Ghorbanzadeh, "Ensembling Of Decision Trees, KNN, And Logistic Regression With Soft-Voting Method For Wildfire Susceptibility Mapping," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus Publications, Jan. 2023, pp. 647–652. doi: 10.5194/isprs-annals-X-4-W1-2022-647-2023.

[14] M. Alkhodari and L. Fraiwan, "Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings," *Comput Methods Programs Biomed*, vol. 200, Mar. 2021, doi: 10.1016/j.cmpb.2021.105940.

[15] G. D. Clifford *et al.*, "Classification of normal/abnormal heart sound recordings: The PhysioNet/Computing in Cardiology Challenge 2016," in *Computing in Cardiology*, IEEE Computer Society, Mar. 2016, pp. 609–612. doi: 10.22489/cinc.2016.179-154.

[16] M. Boulares, R. Alotaibi, A. Almansour, and A. Barnawi, "Cardiovascular disease recognition based on heartbeat segmentation and selection process," *Int J Environ Res Public Health*, vol. 18, no. 20, Oct. 2021, doi: 10.3390/ijerph182010952.

[17] T. H. Chowdhury, K. N. Poudel, and Y. Hu, "Time-Frequency Analysis, Denoising, Compression, Segmentation, and Classification of PCG Signals," *IEEE Access*, vol. 8, pp. 160882–160890, 2020, doi: 10.1109/ACCESS.2020.3020806.

[18] A. Meliboyev, J. Alikhanov, W. Kim, M. Azizjon, and A. Jumabek, "1D CNN Based Network Intrusion Detection with Normalization on Imbalanced Data," 2020. [Online]. Available: https://www.researchgate.net/publication/339641880

[19] A. Hasan and Z. Bahri, "Comparative Study on Heart Anomalies Early Detection Using Phonocardiography (PCG) Signals," *International Journal of Computing and Digital Systems*, vol. 14, no. 1, pp. 643–655, Oct. 2023, doi: 10.12785/ijcds/140180.

[20] S. Lucarini, M. V. Upadhyay, and J. Segurado, "FFT based approaches in micromechanics: Fundamentals, methods and applications," Mar. 01, 2022, *IOP Publishing Ltd*. doi: 10.1088/1361-651X/ac34e1.

[21] T. G.S., Y. Hariprasad, S. S. Iyengar, N. R. Sunitha, P. Badrinath, and S. Chennupati, "An extension of Synthetic Minority Oversampling Technique based on Kalman filter for imbalanced datasets," *Machine Learning with Applications*, vol. 8, p. 100267, Jun. 2022, doi: 10.1016/j.mlwa.2022.100267.

[22] S. Wang, Y. Dai, J. Shen, and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-03430-5.

[23] A. Rehmer and A. Kroll, "On the vanishing and exploding gradient problem in Gated Recurrent Units," Berlin, Germany, Jul. 2020.

[24] J. Wang, X. Qiang, Z. Ren, H. Wang, Y. Wang, and S. Wang, "Time-Series Well Performance Prediction Based on Convolutional and Long Short-Term Memory Neural Network Model," *Energies (Basel)*, vol. 16, no. 1, Jan. 2023, doi: 10.3390/en16010499.

[25] Z. Wu *et al.*, "Predicting Groundwater Level Based on Machine Learning: A Case Study of the Hebei Plain," *Water (Switzerland)*, vol. 15, no. 4, Feb. 2023, doi: 10.3390/w15040823.

[26] A. K. Shaikh, A. Nazir, N. Khalique, A. S. Shah, and N. Adhikari, "A new approach to seasonal energy consumption forecasting using temporal convolutional networks," *Results in Engineering*, vol. 19, Sep. 2023, doi: 10.1016/j.rineng.2023.101296.

[27] S. Li, W. Zhang, and P. Wang, "TS2ARCformer: A Multi-Dimensional Time Series Forecasting Framework for Short-Term Load Prediction," *Energies (Basel)*, vol. 16, no. 15, Aug. 2023, doi: 10.3390/en16155825.

[28] J. M. A. S. Dachi and P. Sitompul, "Analisis Perbandingan Algoritma XGBoost dan Algoritma Random Forest Ensemble Learning pada Klasifikasi Keputusan Kredit," *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam*, vol. 2, no. 2, pp. 87–103, Oct. 2023, doi: 10.55606/jurrimipa.v2i2.1336.

[29] I. Saluza and H. Hartati, "Neural Network Optimization Using Ensemble Method In Forecasting Financial Data," *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 10, no. 4, p. 381, Dec. 2022, doi: 10.26418/justin.v10i4.50771.

[30] Y. Mahendra Awaludin and F. Budiman, "Optimasi Analisis Kesuburan Tanah Dengan Pendekatan Soft Voting Ensemble," *Jurnal SIMETRIS*, vol. 14, no. 2, 2023.

[31] A. Z. Noorizki and W. I. Kusumawati, "Perbandingan Performa Algoritma VGG16 Dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras," *Journal of Computer, Electronic, and Telecommunication*, vol. 4, no. 2, Dec. 2023, doi: 10.52435/complete.v4i2.387.