2024Fall CS182 Final Project Report
Heartbeat Signal Classification

Group No.6: Yufei Liu 2022533037; Gubin Hu 2022533186; Yuanxuan Li 2022533167

ABSTRACT

This study explores advanced machine learning and deep learning techniques for automated ECG heartbeat classification into four categories, leveraging a large-scale Tianchi dataset. Preprocessing methods, such as batch normalization and stepwise dimensionality increase, were applied to enhance model performance. Models including classical machine learning approaches, LSTM, and transformer-based architectures were evaluated, with BERT excelling in capturing temporal dependencies. To address challenges like class imbalance and signal similarity, we proposed several strategies: improved feature extraction using multi-scale convolution and attention mechanisms, class imbalance mitigation through uneven sampling, and ensemble methods like voting to refine predictions. Eventually, our model achieved **98.92%** accuracy in ten-fold cross-validation on the train.csv provided by tianchi and **99.16%** on the Tianchi test set.

1 INTRODUCTION

Electrocardiogram (ECG) signal analysis plays a critical role in monitoring and diagnosing cardiovascular diseases, which remain one of the leading causes of mortality worldwide. With the rapid development of data science and machine learning techniques, automated classification of ECG heartbeat signals has garnered significant attention in both academic and clinical settings. Accurate identification of different heartbeat signal types can not only improve the efficiency and reliability of diagnosis but also facilitate real-time monitoring systems for the early detection of cardiac abnormalities.

Tianchi holds a competition to encourage participants to focus on this problem. This competition focuses on developing models to classify different types of heartbeat signals, leveraging a curated dataset sourced from a large-scale ECG data recording platform. The dataset comprises over 200,000 samples, each representing a standardized sequence of heartbeat signals. To ensure fairness, a subset of 100,000 samples has been designated for training, with two separate test sets, Test Set A and Test Set B, each containing 20,000 samples. The task requires participants to predict probabilities for four distinct heartbeat signal classes, evaluated by the absolute difference between predicted probabilities and ground truth labels.

1.1 Dataset Description

The dataset is anonymized to protect sensitive information and comprises three key fields:

- **`id`**: A unique identifier assigned to each heartbeat signal.
- **`heartbeat_signals`**: The sequence data representing the ECG signals for each sample.
- **`label`**: The class of the heartbeat signal, encoded as one of four categories (0, 1, 2, 3).

The Tianchi platform provides a training dataset, `"train.csv"`. The shape of the training dataset is:

- `heartbeat_signals: (100000, 205)`
- `label: (100000)`

Here, $n = 100000$ represents the number of training samples, and $t = 205$ corresponds to the time series data.

Tianchi also provides `"testA.csv"` for submission and scoring, which contains 20,000 `heartbeat_signals` and no `label`.

1.2 Evaluation and Submission

The evaluation metric measures the alignment between predicted probabilities and true labels. For a single sample, the true label vector $[y_1, y_2, y_3, y_4]$ is compared against the predicted probability vector $[a_1, a_2, a_3, a_4]$, and the metric **abs-sum** is calculated as follows:

$$\textbf{abs-sum} = \sum_{j=1}^{n} \sum_{i=1}^{4} |y_i - a_i| \tag{1}$$

The objective is obviously to minimize **abs-sum**.

1.3 Contribution

Classical methods for ECG heartbeat classification include decision trees[1], support vector machines (SVM) , Naive Bayes [2], multimodal feature and image fusion (MIF) [3], wavelet transform (WT), independent component analysis (ICA) , interval information (RR) [4], discrete cosine transform (DCT), and Fisher's linear discriminant analysis [5]. These algorithms focus on feature extraction from ECG signals, such as frequency, temporal, and statistical features, to classify heartbeats. Despite their simplicity, these methods require performance improvements,

To address these limitations, this study proposes several strategies to improve model performance and handle challenges like class imbalance and signal similarity:

(1) **Better Feature Extraction**
We can enhance the feature extraction process by using advanced techniques like multi-scale convolution and attention mechanisms. These can help the model capture both local and global patterns in the ECG signals.

(2) **Extensive exploration**
We thoroughly analyzed the confusion matrix and conducted many explorations to address the existing issues. For example, we applied techniques like uneven sampling to mitigate the class imbalance problem and simplified the classification task to help the model better learn the features of Class 0.

(3) **Ensemble Methods**
We analyzed the predicted values of the misclassified instances and found that the predicted value of true label and

predicted label are very similar. Thus, we combine the predictions of multiple models using ensemble methods like simply adding or voting to help improve overall accuracy.

2 DATA

First, we examined the distribution of samples across the different classes in the training data (see Table 1):

Class 1	Class 2	Class 3	Class 4
64.33%	**3.56%**	**14.20%**	**17.91%**

Table 1: Distribution of samples across classes

As we can see, the samples across these four classes are not well balanced, with the highest class having 64.33% and the lowest class only 3.56%. This imbalance may lead to the model having varying capabilities in extracting features from ECG signals across different classes, potentially reducing classification accuracy. We will discuss this issue further in later sections.

Next, we took an observation at examples of the four classes of ECG signals (see Figure 1, 2, 3, 4).

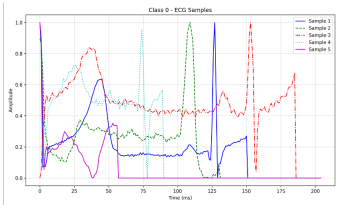

Figure 1: Class 0

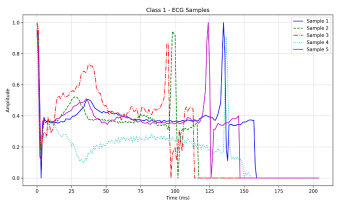

Figure 2: Class 1

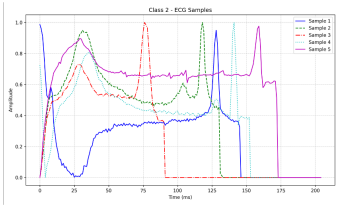

Figure 3: Class 2

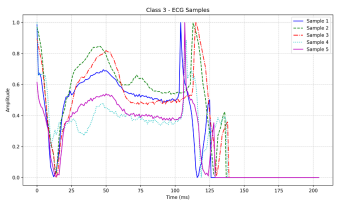

Figure 4: Class 3

We analyzed the maximum and minimum values of all ECG signals and found that regardless of the class, the minimum value is 0, and the maximum value is 1. Since we need to input the signal data into the neural network, we apply batch normalization to normalize the data for better model convergence (see Algorithm 1 [6] and Figure 5).

As observed in Figure 5, after applying batch normalization, the shape of our data remains unchanged, but the mean of the ECG signals has shifted to 0, which can help our model converge more quickly.

Algorithm 1 Batch Normalizing Transform, applied to activation x over a mini-batch.

Require: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Ensure: $\{y_i = \mathrm{BN}x_i\gamma, \beta\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{normalize}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv \mathrm{BN}x_i\gamma, \beta \qquad \text{scale and shift}$$

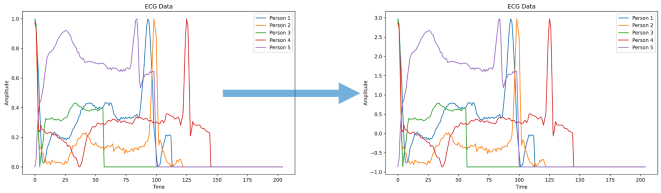

Figure 5: Batch Normalization

3 MODEL

3.1 Classical Machine Learning Methods

Intially, we tried to do the task with some simple models and observed the results, which served as our baselines. We only applied naive normalization instead of batch normalization when training baseline models.

(1) **Unsupervised Models**

We tried K-means Clustering and EM for GMM (Expectation-Maximization for Gussian Mixture Model), whose performance results were very bad. In Table 2, the accuracy is calculated on `"train.csv"`, since the chance of submission on Tianchi is limited.

Model	Accuracy
K-means	83.76%
EM for GMM	64.33%

Table 2: Unsupervised Models

(2) **SVM**

We also tried SVM (Support Vector Machine) with different kernels. The accuracy is calculated on `"train.csv"`, except for the last row (see Table 3).

As we can see in Table 3, the polynomial kernels overfit soon as the degree increased, yet its best score on degree 3 was still comparatively poor. The gussian kernel performed the best with 97.07% accuracy, so we submitted it onto Tianchi to see what score we would get on `"testA.csv"`.

SVM Kernel	Accuracy
polynomial	
degree=2	94.75%
degree=3	95.04%
degree=4	93.74%
degree=5	90.84%
sigmoid	69.45%
guassian	**97.07%**
guassian (submit)	**97.31%**

Table 3: SVM

The result is shown in Figure 6. The score of 1078 means there's still much to be improved, so we would then turn to the better approaches from deep learning.

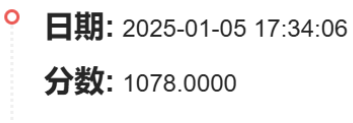

Figure 6: Score of Guassian-Kernel SVM on Tianchi

3.2 Deep Learning Methods

When utilizing deep learning, the first challenge we faced was that the features of the training data is actually 1-D, i.e. `(100000, 205, 1)`. We need to increase its dimensionality so as to extract the features properly.

Considering that ECG signals exhibit some local similarity, we can use convolution (see Figure 7) to increase the dimensionality of the data. However, a clear issue arises: we are directly increasing the feature dimension from 1 to d (where $d \gg 1$), which could lead to poor quality in feature extraction. To address this, we can use multiple convolutional kernels to incrementally increase the feature dimension. For instance, we can start with a convolutional kernel of size 3 to raise the feature dimension to $d/2$, and then apply a kernel of size 5 to further increase the feature dimension to d. This gradual dimensionality increase allows us to extract better features, facilitating improved processing of the ECG signals.

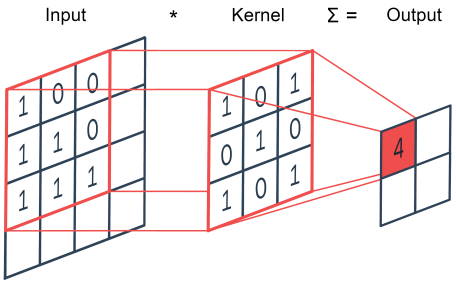

Figure 7: An Example of Convolution

After feature extraction, we can choose our main model to perform this multi-class classification task. Since ECG signals are temporal data, we can chooose common models for processing time series signals including LSTM (Long Short-Term Memory) [7] as shown in Figure 8.

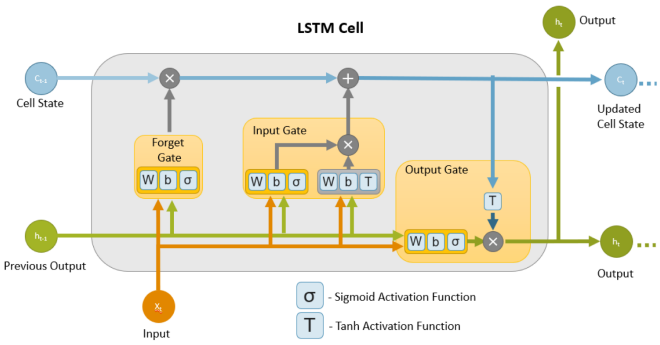

Figure 8: LSTM

We directly utilized the last hidden state of the LSTM, denoted as h_{205}, for classification. Specifically, we added a learnable matrix $\boldsymbol{W}$ to use $\boldsymbol{W}h_{205}$ as the predicted values for the four classes of signals, selecting the highest value as the predicted category.

In addition to LSTM, the transformer model (see Figure 9), which has gained popularity since 2017 [8], can also be applied to this problem.

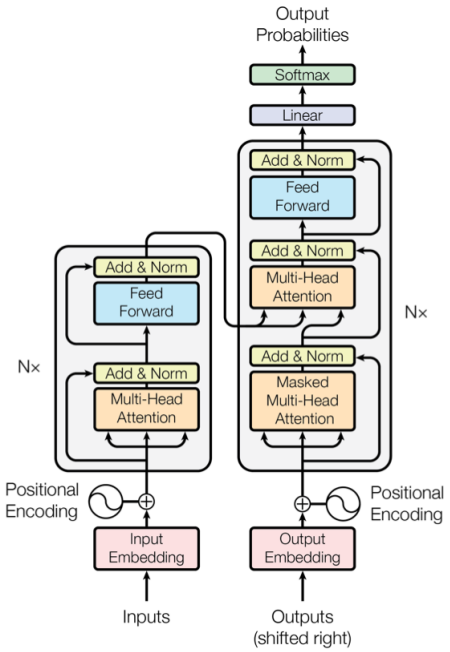

Figure 9: Transformer model

It is important to note that we did not directly use the encoder-decoder architecture of the transformer, but rather employed only

the encoder or decoder, which can also be referred to as BERT (Bidirectional Encoder Representations from Transformers) [9] and GPT-2 (Generative Pre-trained Transformer - 2) [10]. To adapt the transformer for classification tasks, we need to add an additional special learnable token, `[CLS]`, at the end of the signal. Similar to the LSTM approach, we added a learnable matrix $\boldsymbol{W}$, using $\boldsymbol{W}h_{206}$ as the predicted values for the four classes of signals, and selected the highest value as the predicted category.

4 EXPERIMENTS

4.1 Batch Normalization

We utilized the cross-entropy loss function:

$$E(\{\boldsymbol{w}_j, w_{j0}\}_j \mid \mathcal{X}) = -\sum_{i=1}^{N}\sum_{j=1}^{K} r_{ij} \log y_{ij} \tag{2}$$

where $\mathcal{X} = \{\boldsymbol{x}_i, r_i\}_{i=1}^{N}$ is the dataset containg N samples indexed by i.

First, we validated the improvement in convergence due to batch normalization (see Figure 10). It is clear that after adding batch normalization, our model's convergence speed improved significantly, indicating that data normalization is highly beneficial for our training process.

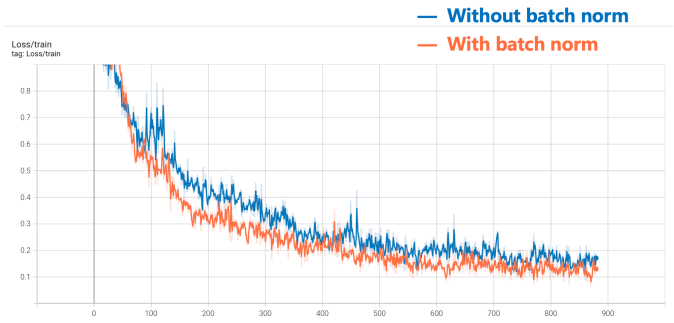

Figure 10: The Effect of Batch Normalization

4.2 Dimensionality Increase

Next, we performed a simple test using the BERT model to assess the effectiveness of the aforementioned two-step dimensionality increase, discovering that the accuracy improved from 93.06% to 95.86%. This demonstrates that a gradual increase in dimensionality indeed facilitates better feature extraction.

4.3 Main Model Comparison

To select the final main model from three alternatives, we conducted ten-fold cross-validation to evaluate the accuracy of LSTM, GPT-2, and BERT models. The results are summarized in Table 4 (note that this accuracy is for comparison purposes and not the final result).

It is evident that BERT performed the best, followed by GPT-2, while LSTM showed the weakest performance. This outcome may be due to BERT's bidirectional attention allowing more interaction

BERT	**95.86%**
GPT-2	94.35%
LSTM	75.15%

Table 4: Performance of the three Main Model

of information compared to the unidirectional attention of GPT-2, which did not fully leverage the advantages of autoregressive modeling. Consequently, BERT outperformed GPT-2. For LSTM, its gating mechanism does not directly access previous information as effectively as attention mechanisms do. Thus, the sequence length of 205 may be too long for LSTM, as earlier information may not be preserved well, leading to its inferior performance.

Therefore, in future experiments, we will adopt BERT, specifically the transformer encoder, as our main model to further extract temporal information from the convolutionally processed features.

The main parameters used in our model are: (see Table **??**)

Layers	Heads	Hidden Size
6	4	64

Table 5: Parameters of BERT

The dataset is `"train.csv"` provided by Tianchi, and through ten-fold cross-validation, we achieved an average accuracy of **98.92%**.

Confusion Matrix Analysis

We generated the confusion matrix for BERT (see Figure 11).

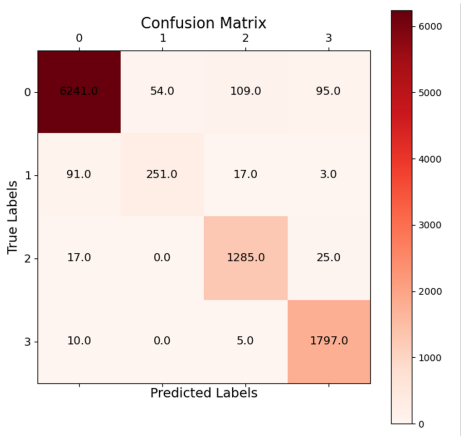

Figure 11: Confusion Matrix of BERT

It is apparent that Class 0 is frequently confused with the other three classes, especially Class 0 and Class 1, while confusion among the latter three classes is quite rare. We proposed two hypotheses to explain why this occurred:

(1) The data imbalance leads to confusion between Class 0 and the other three classes, particularly between Class 0 and

Class 1. Referring back to the Data section, the proportion of the four classes in the training dataset indicates that Class 0 comprises 64.33% of the total data, whereas Class 1 only accounts for 3.56%. We hypothesize that this imbalance results in insufficient feature extraction for the class with fewer samples, causing the model to more readily overlook the minority class during the generation process.

(2) There are clear differences among the three ECG signal classes, while the differences between Class 0 and the other three classes are not as pronounced.

To test these two hypotheses, we designed experiments to verify and attempt to resolve these issues.

4.4 Sample Imbalance

For the first hypothesis, a simple and effective approach is to adjust the weights assigned to different data samples, allowing the data loader to perform uneven sampling.

Sample Weights	0.5	2	1	1
Before Weighted	**64.33%**	**3.56%**	14.20%	17.91%
After Weighted	**45.05%**	**9.97%**	19.88%	25.08%

Table 6: Proportions of Different Classes

We did not make drastic adjustments to the sample proportions, as the inherent probabilities of the samples themselves should also be part of the modeling process. Additionally, we experimented with equalizing the sample ratios among the four classes, but these results were significantly poor. Therefore, after balancing various factors, we opted for this set of weights. The final experimental results are as follows:

Without Weighted	**98.92%**
Weighted	98.48%

Table 7: Model accuracy with and without weighted samples

We found that weighting the data did not improve the accuracy, and upon examining the confusion matrix, the previous issues were still present. This indicates that merely changing the sample ratios poses challenges in improving model accuracy.

4.5 Similarity Between Different Classes

The second hypothesis suggests that the similarity between the two classes prone to misclassification is relatively high. We computed the Mean Squared Error (MSE) between each pair of the four ECG signal classes (see Figure 12).

It can be observed that while the similarities between Class 0 and Classes 2 and 3 are not very high, Class 0 shows a significantly higher similarity with the most commonly confused Class 1 compared to the others. Correlating this with the earlier information from the confusion matrix, we can deduce that the bottleneck in model accuracy lies in distinguishing whether a signal belongs to Class 0, as there is very little confusion among the other three

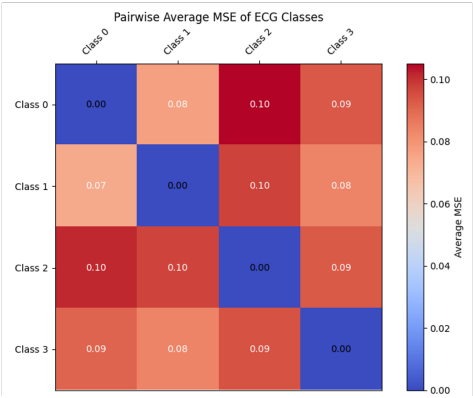

Figure 12: Between-class Similarity Measured by MSE

classes. Consequently, we can design a new classification task aimed at determining whether a signal belongs to Class 0.

We hypothesize that such a binary classification task could simplify the classification difficulty, allowing the model to better learn the unique features of Class 0 and further enhance the model's accuracy. We combined all samples from Classes 1, 2, and 3 into one class, while samples from Class 0 formed another class. The results of our experiment are as follows:

Original Four-Class Task	**98.92%**
New Binary Classification Task	98.69%

Table 8: Model accuracy in original and new binary classification tasks

Even after simplifying the classification task, our model's accuracy did not show significant improvement. This suggests that Class 0 indeed shares some similarity with the other three classes, and it's hard to separate them better.

4.6 Model Ensemble

We were curious about the misclassifications our model made, specifically how far off the predicted values for the output labels were from the true labels. Were they very close, or significantly different? Therefore, we analyzed the distribution of predicted values for the incorrectly classified results.

```
tensor([
1.9963, -1.7708, 2.2101, -3.9106, ), device='cuda:0'),
 predicted: 2, labels: 0

tensor([
1.8058, -1.6514, 1.8395, -2.4592, ), device='cuda:0',
 predicted: 2, labels: 0

tensor([
1.2364, -0.6873, 0.9485, -1.0386, ), device='cuda:0',
 predicted: 0, labels: 2
```

```
tensor([
3.6414, 2.3367, -2.2693, -4.3214, ), device='cuda:0',
 predicted: 0, labels: 1
```

To our surprise, we found that the predicted values for the incorrect labels were very close to those of the true labels, whereas the predicted values for the other two classes diverged considerably from the correct labels. This insight led us to consider whether training multiple models and aggregating their predicted values could potentially improve the situation and enhance our model's accuracy. To achieve this goal, we implemented two straightforward approaches:

(1) **Directly Suming the Predicted Values**:
We trained three different models, summed their predicted values, and selected the label with the highest total as the final prediction.
(2) **Voting**:
We trained three models, each providing a predicted label, and the label with the majority of votes was chosen as the final prediction. In the event of a tie for the most votes, one label was selected at random.

For this part, we used the test data "`testA.csv`" provided by Tianchi (consisting of 20,000 test samples) and submitted the output results for evaluation on Tianchi to obtain the final score (see Figure 13) calculated by **abs-sum** (see Subsection 1.2). We also converted it to accuracy (see Table 9).

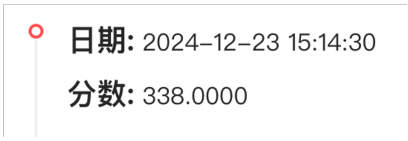

Figure 13: Score on Tianchi

Single Model	99.03%
Sum of Predicted Values	**99.16%**
Voting	**99.16%**

Table 9: Accuracy for Single and Ensemble Models

This indicates that both ensemble approaches improved the overall accuracy of the model.

5 CONCLUSION

In this project, we successfully applied various simple models (maily SVM) and more importantly deep learning techniques to the four-class classification task of electrocardiogram (ECG) signals. Our deep learning techniques includes Convolutional Neural Networks (CNNs), LSTM, and Transformer models (particularly BERT), and we ultimately selected BERT as the primary model. In terms of data preprocessing, we significantly enhanced the model's convergence speed through Batch Normalization. During feature extraction, we employed a stepwise dimensionality increase convolution strategy, which effectively improved the model's performance.

The experimental results demonstrate that the BERT model exhibited exceptional performance in handling sequential data. Compared to LSTM and GPT-2, BERT's bidirectional attention mechanism allows it to better capture the temporal relationships within ECG signals. Ultimately, the BERT-based model achieved an accuracy of **98.92%** in ten-fold cross-validation on '`train.csv`', showcasing its superiority in this task.

Given the significant class imbalance present in the dataset, we attempted to balance the class samples through weighted sampling methods. However, these approaches did not yield significant improvements in accuracy. Additionally, through the analysis of the confusion matrix, we identified that the root of the problem lies in the high misclassification rate of Class 0 compared to the other classes. As a result, we considered using a binary classification task, focusing solely on determining whether a signal belongs to Class 0 to simplify the model's learning objective, but the results indicated no improvement in accuracy. Future research could explore more refined feature engineering or the integration of generative models for data augmentation.

Furthermore, we explored model ensemble methods, attempting to aggregate or vote on the predictions from multiple models. The results showed that ensemble methods could further enhance model performance, ultimately achieving an accuracy of **99.16%** on the Tianchi test set '`testA.csv`', surpassing the single model's accuracy of **99.03%**. This indicates that ensemble approaches can effectively mitigate the limitations of individual models.

Lastly, although we employed relatively complex deep learning methods, the results indicate that our model is quite lightweight, with a final size of only **1.5MB**, allowing it to run on many machines while maintaining strong transferability and scalability. Additionally, due to time and computational resource constraints, we performed only limited tuning of the model parameters. Future work could involve more systematic hyperparameter searches to further enhance model performance.

In summary, this research provides an efficient and practical deep learning solution for the ECG classification problem, demonstrating strong practical applicability. In future work, we can further optimize the model, address the class imbalance issue, and explore additional deep learning techniques.

REFERENCES

[1] Kumari, L. and Sai, Y.P. Classification of ECG beats using optimized decision tree and adaptive boosted optimized decision tree. *Signal Image Video Process.* 2022;16:695–703.
[2] Ahmad, Z., Tabassum, A., Guan, L., Khan, N.M. ECG heartbeat classification using multimodal fusion. *IEEE Access.* 2021;9:100615–100626. doi: 10.1109/ACCESS.2021.3097614.
[3] Martis, R.J., Acharya, U.R., Tan, J.H., Petznick, A., Yanti, R., Chua, C.K., Ng, E.K., Tong, L. Application of empirical mode decomposition (EMD) for automated detection of epilepsy using EEG signals. *Int. J. Neural Syst.* 2012;22:1250027. doi: 10.1142/S012906571250027X.
[4] Ye, C., Kumar, B.V., Coimbra, M.T. Heartbeat classification using morphological and dynamic features of ECG signals. *IEEE Trans. Biomed. Eng.* 2012;59:2930–2941. doi: 10.1109/TBME.2012.2213253.

[5] Pathoumvanh, S., Hamamoto, K., Indahak, P. Arrhythmias detection and classification base on single beat ECG analysis; Proceedings of the 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE); Chiang Rai, Thailand. 5–8 March 2014; pp. 1–4.
[6] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, abs/1502.03167.
[7] Vennerød, C.B., Kjærran, A., & Bugge, E.S. (2021). Long Short-term Memory RNN. ArXiv, abs/2105.06756.
[8] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Neural Information Processing Systems.
[9] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics.
[10] Radford, A., Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training.