Real-time ECG Heartbeat Classification

Trung Le Chi Phan [1,2†], Tuan Dat Nguyen [1,2†],
Thang Duc Nong [1,2†], Tai Huu Le [1,2†], Trong Hop Do [1,2*]

[1]Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam.
[2]Vietnam National University, Ho Chi Minh City, Vietnam.

*Corresponding author(s). E-mail(s): hopdt@uit.edu.vn;
Contributing authors: 21522725@gm.uit.edu.vn;
21522754@gm.uit.edu.vn; 21522593@gm.uit.edu.vn;
21522562@gm.uit.edu.vn;
†These authors contributed equally to this work.

Abstract

Electrocardiogram (ECG) analysis plays a crucial role in diagnosing and monitoring cardiovascular diseases. Nowadays, much attention has been given to the accurate and early detection of heartbeat anomalies in real-time to prevent complications and take necessary measures. In this project, we present an innovative approach to real-time ECG heartbeat categorization using deep learning techniques. We evaluate the system on a widely used dataset, MIT-BIH datasets from PhysioNet. Our testing shows that the CNN model has outstanding performance with up to 98.2% accuracy. The proposed system aims to enhance the accuracy and efficiency of heartbeat classification, enabling timely detection of abnormal cardiac patterns. Leveraging a comprehensive dataset of ECG signals, our model employs algorithms to categorize heartbeats into various classes, facilitating early identification of arrhythmias and other cardiac anomalies. The implementation of this system holds significant promise for improving patient outcomes through timely intervention and personalized healthcare.

Keywords: ECG, deep learning, heart beat, real-time

1 Introduction

ECG is widely used by cardiologists and medical practitioners for monitoring cardiac health. The main problem with the manual analysis of ECG signals, similar to many other time-series data, lies in the difficulty of detecting and categorizing different waveforms and morphologies in the signal. For a human, this task is both extensively time-consuming and prone to errors. Note that the proper diagnosis of cardiovascular diseases is of paramount importance since these are the cause of death for about one-third of all deaths around the globe. For instance, millions of people experience irregular heartbeats which can be lethal in some cases. Therefore, accurate and low-cost diagnosis of arrhythmic heartbeats is highly desirable. To address these challenges, the integration of machine learning techniques has emerged as a promising avenue. In this project, we focus on real-time ECG heartbeat categorization, leveraging a rich dataset of diverse cardiac rhythms. The objective is to develop a robust and efficient system capable of categorizing heartbeats into classes of heartbeat which were diagnosed by doctors. We employ cutting-edge machine learning algorithms, such as deep neural networks, to capture intricate patterns within the ECG signals. The utilization of a real-time dataset enhances the practical applicability of our model, allowing for instantaneous analysis and classification. The significance of our project lies in its potential to revolutionize cardiac healthcare by enabling the swift identification of arrhythmias and abnormalities. By providing healthcare professionals with a tool for real-time ECG analysis, we aim to contribute to early intervention strategies, ultimately improving patient outcomes and reducing the burden of cardiovascular diseases.

2 Dataset

We utilize the ECG heartbeat categorization dataset of ECG Heartbeat Classification: A Deep Transferable Representation paper (1) Kachuee, M., Fazeli, S., Sarrafzadeh, M. (2018, June). The dataset is a famous collection of heartbeat signals used in heartbeat classification, the MIT-BIH Arrhythmia Dataset . The dataset includes 109446 samples categorized into 5 classes, which is large enough for training a deep neural network.The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Twenty-three recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston's Beth Israel Hospital; the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that would not be well-represented in a small random sample. We used annotations in this dataset to create five different beat categories by the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard. See [Fig. 1] for a summary of mappings between beat annotations in each category.

Category	Annotations
N	• Normal • Left/Right bundle branch block • Atrial escape • Nodal escape
S	• Atrial premature • Aberrant atrial premature • Nodal premature • Supra-ventricular premature
V	• Premature ventricular contraction • Ventricular escape
F	• Fusion of ventricular and normal
Q	• Paced • Fusion of paced and normal • Unclassifiable

Fig. 1: Classes annotated in the dataset and their meanings

2.1 Data Preprocessing

The signals correspond to electrocardiogram (ECG) shapes of heartbeats for the normal case and the cases affected by different arrhythmias and myocardial infarctions. These signals are preprocessed and segmented, with each segment corresponding to a heartbeat.

In order to transform the dataset from the MIT-BIH Arrhythmia Database into a more trainable and suitable data, we apply the transformation suggested in the paper ECG Heartbeat Classification: A Deep Transferable Representation [1] .The suggested beat extraction method is simple and effective in extracting R-R intervals from signals with different morphologies. The steps of extracting heartbeats include the following steps:

1. Splitting the continuous ECG signal to 10s windows and select a 10s window from an ECG signal.
2. Normalizing the amplitude values to the range of between zero and one.
3. Finding the set of all local maximums based on zero-crossings of the first derivative.
4. Finding the set of ECG R-peak candidates by applying a threshold of 0.9 on the normalized value of the local maximums.
5. Finding the median of R-R time intervals as the nominal heartbeat period of that window (T).
6. For each R-peak, selecting a signal part with the lengthequal to 1.2T.

7. Padding each selected part with zeros to make its length equal to a predefined fixed length.

We visualize one sample of each classes after perform the heartbeat extraction method to the dataset. As presented in [Fig. 2] we could see the difference between each classes at the same window of time.

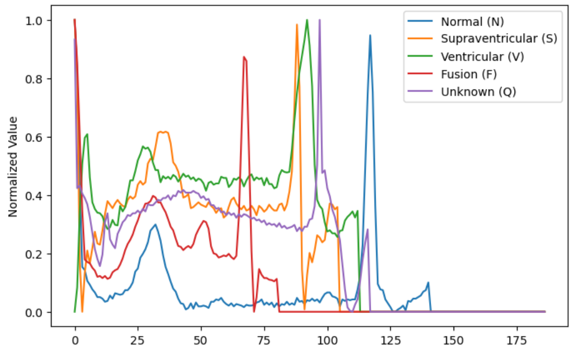

Fig. 2: Visualize a sample of each classes

2.2 Dataset Analysis

As we are doing a classification task in the medical domain, it is important to consider about the ratio of the classes in the datasets. We can see in [Fig.2] that the class N, which stands for normal cases have a very high ratio in the dataset(82,8%). However, other classes like class F, only plays a small part in the whole dataset with 0.7% and the class with second highest rate only plays 7,3%. This means that the dataset we are using are not balanced and might affect the results when training the model to be biased into one class. Due to the imbalanced dataset issue, we decided to perform a resample method(Over-Resampling) before doing the training, which involves randomly duplicating samples of low ratio classes.

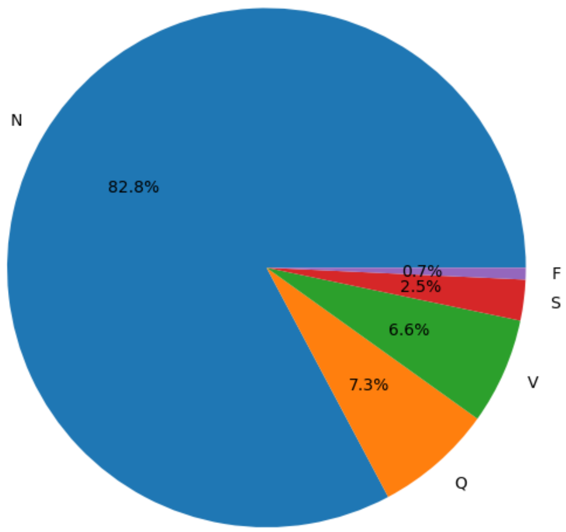

Fig. 3: Ratio of the dataset's classes

Next, we want to visualize the difference between the original data and the data after extracted heartbeats. As you can see in the [Fig. 4], which is the visualization of a sample in the original dataset, the result are recorded by the time and only show a small difference between values. This could be a challenge for even professional human to recognize if an anomaly is there or not.

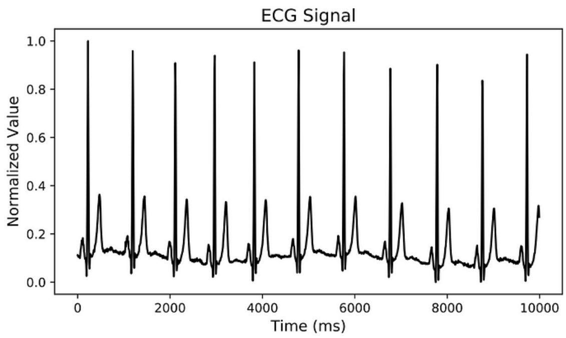

Fig. 4: A sample of the original ECG signal records

After applying the heartbeat extraction methods, we transformed the original sample into a padding with fixed length of 188 [Fig. 5]. The data now can visualize the more visible trends. With being padded into fixed length, the data will be more suitable to be the input of the training models we proposed. And it is also more convenient for users to observe and analyze the heartbeat signals.

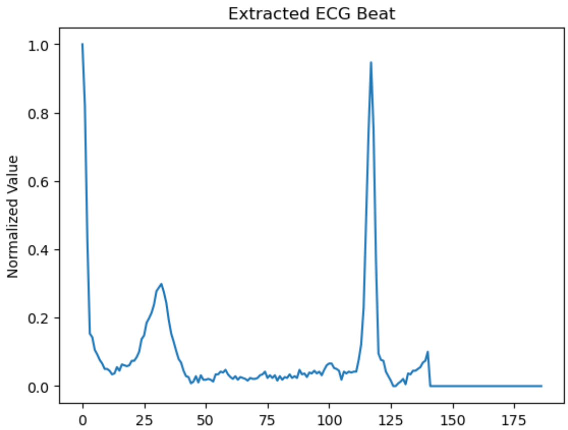

Fig. 5: A sample after performed heartbeat extraction methods

3 Our approach

In order to perform a project which is practical and can continue to develop into a real life applicable task, we proposed our detail approach method. The approach method includes two main parts. The first part is the real life process which will be used to perform the real life application. Secondly is the proposed models we used to approach with the training steps, which will directly affect to the performance of the application.

3.1 Our Process

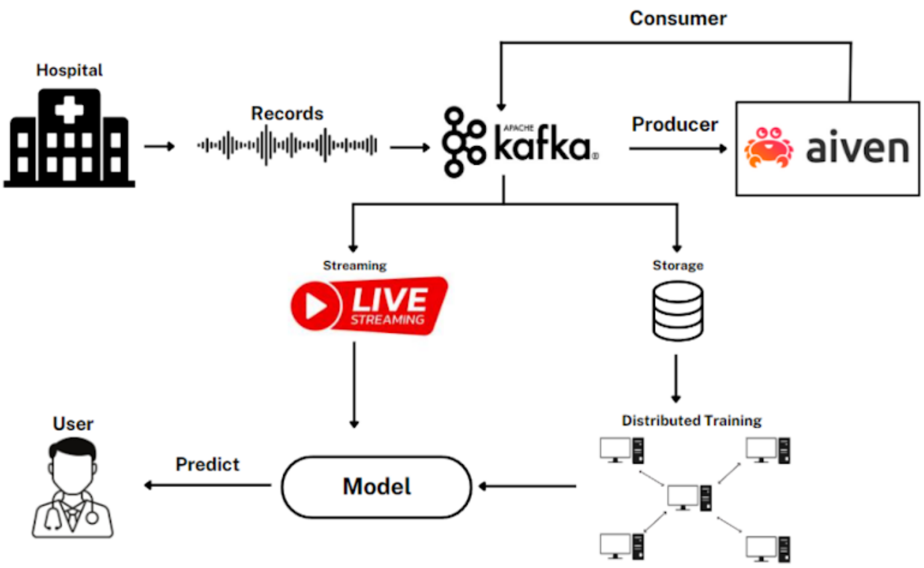

Fig. 6: Our Proposed Process

3.1.1 Data source

As we can observe in the [Fig. 6], we will start with the data collected and preprocessed from the hospital. The hospital symbol here is just a representation for the data source collection. The data can be collected from any different data sources from anywhere in the world. For example: A hospital in country A collaborate with hospital in country B to send the data to a database center at country C. We decided to approach at the distributed levels of the data to be suitable for the real life.

3.1.2 Data transporting

After being collected and performed the transformation, the data will be transported to the database center. At this step we will use Kafka as the transportation. As mention above, our goal is to use data from many data sources from different places. So, we rent a server in HongKong on Aiven to simulate a global transporting system. the data from each data source will then be sent on this server by the Kafka Producer and will be collected at the data center using Kafka Consumer. All the data will be archived at the data center for two purpose: training models and simulating the real life streaming system.

3.1.3 Distributed training

The archived data will be split into three sets: train, dev, test. Those sets will then be used for training the models in the distributed system. We used 5 different models approaching to train for the results. After the training finished, the models will be evaluated. The best peformance model will be saved and used. To ensure the updating of the models with the speed of the data scaling, the model will be trained again every time the data reach a required amount to update itself.

3.1.4 Real-time predicting

After trained and chosen the best model. We will use this model to predict the classes of the data and present the results to the end users. The end users we mentioned at this step could be anyone who have the needs and the authorization to use the data. For example: doctors at hospital, data analyst at data centers, etc. We will stimulate this step with the streaming from Kafka and visualize the result on a web created by Streamlit. And to leverage the heart rate ECG dataset mentioned above, we use the "faker" module to generate patient information synchronously with each data stream. This helps us simulate the data set so that it is diverse and reflects reality in many different hospitals

3.2 Training models

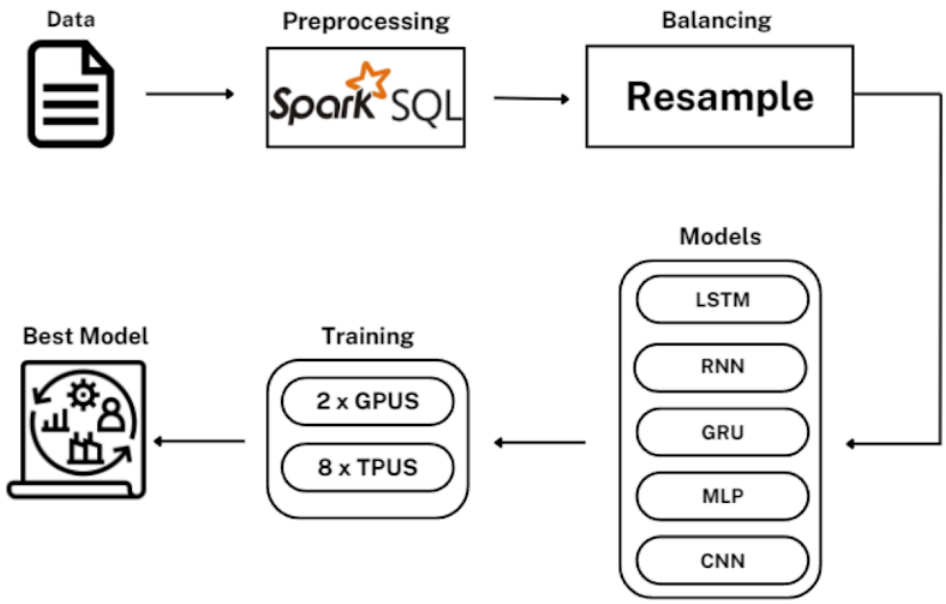

Fig. 7: Training models process

3.2.1 Distributed Deep learning

With the continuing scaling of the data size, we decide to perform the distributed training on our neural networks in order to increase the performance of the models due to the long time taking of each epoch and also the big size of the data. As presented in [Fig. 7], we trained the models on two different types of device: distributed on 2 GPUs of Kaggle and 8 TPUs of Google Colaboratory.However, distributed training involves training models across multiple machine in parallel. This means that our two types of device we used is just only a small number of machines. This could explain for the situation why there is only a small difference on the effectiveness, the performance and the training time between our distributed training and the normal training technique. That could be a way for us to develop the distributed training for this task more into real-life application where we would increase the numbers of parallel machine to really light up the efficiency of distributed training.

3.2.2 Models

As the amount of training data is large, we decided to apply the neural network methods for training models. As the record of the ECG signal is changing chronologically we proposed 3 traditional neural network that considered the changes over time: LSTM, GRU, RNN. Besides that, we also propose two more models using the MLP and CNN architecture to compare and evaluate the results and the performance.

RNN stands for "Recurrent Neural Network" (5) (Elman, 1990), which is a type of neural network architecture in the field of machine learning. What is special about RNN is its ability to process data in series or time series, where previous information can influence current information. This makes RNN powerful in tasks involving time series, such as series prediction, natural language processing, and many other applications. The RNN model consists of small units called "cells" (which operate in the same principle as a neuron), and these cells are connected to form a chain. Each cell receives input from input data and the previous state of the sequence. This allows the RNN to retain information from previous steps and use it to influence the prediction at the current step.

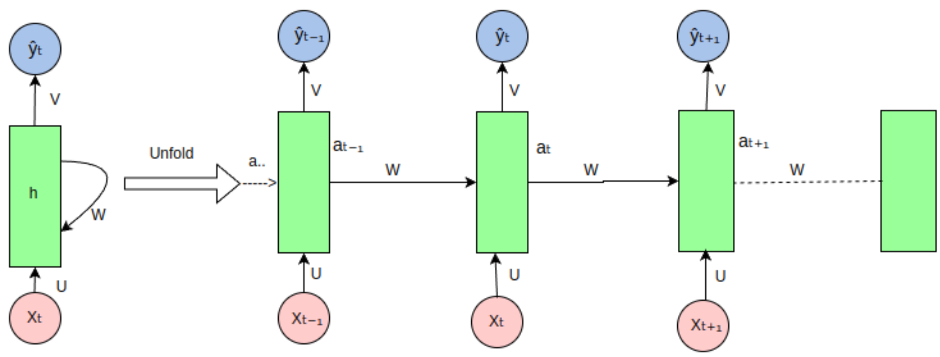

Fig. 8: RNN Architecture

LSTM stands for "Long Short-Term Memory" (6)(Hochreiter and Schmidhuber, 1997) a type of neural network architecture designed to solve the problem of vanishing gradients in the Recurrent Neural Network (RNN) model. The main problem that LSTM solves is the RNN model's ability to maintain information from the past over long sequences. In traditional RNNs, when the sequence becomes long, the gradient may disappear or explode, leading to weak learning ability. LSTM helps overcome this problem by using "gates" to control the flow of information through the network. Each LSTM unit, often called a "cell," consists of three main gates: Forget Gate, Input Gate, Output Gate. Thanks to this gating mechanism, LSTM is capable of storing and transmitting information over multiple time steps without losing gradients. This makes LSTM a popular choice in applications involving time series and natural language processing, where long-term retention of information is important.

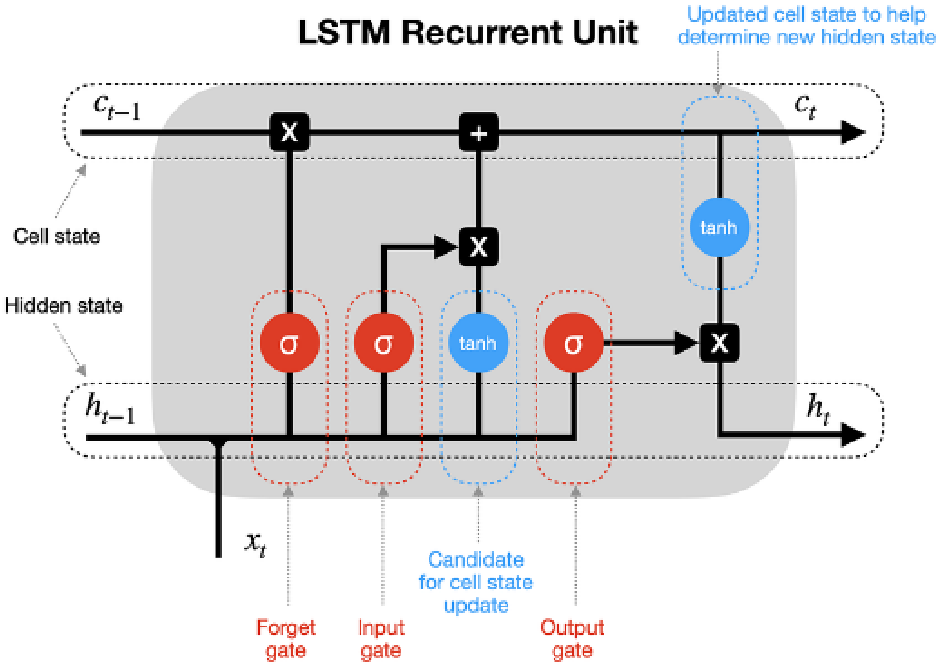

Fig. 9: LSTM Architecture

GRU stands for "Gated Recurrent Unit" (8)(Cho et al., 2014) which is a type of neural network architecture in the field of machine learning, specifically designed to solve the problem of vanishing gradients in the Recurrent Neural Network (RNN) model. GRU is like LSTM (Long Short-Term Memory), another model that solves the same problem, but it has a simpler structure. The main goal of GRU is to retain important information from the past and propagate it through time without the problem of vanishing gradients. Each GRU unit includes two main ports: Reset Gate, Update Gate. These gates help GRU maintain and use important information during the learning process, reduce the problem of vanishing gradients, and help the model learn long-term dependencies in time series data.

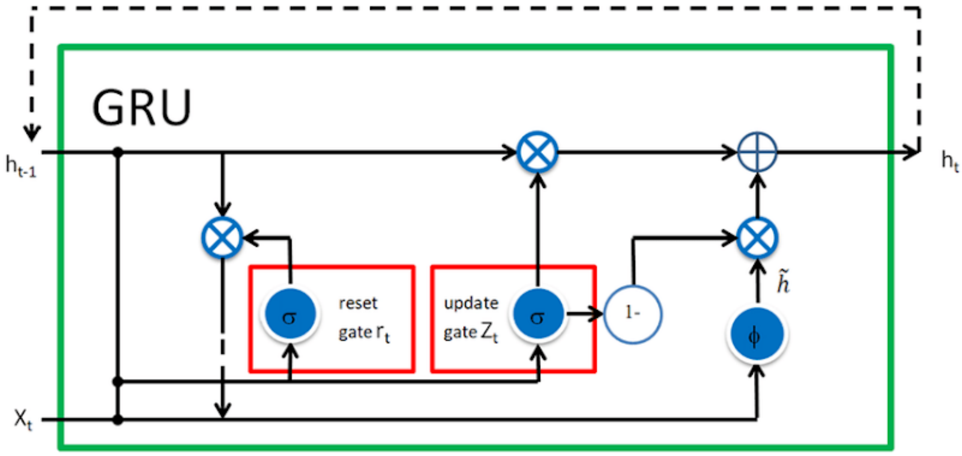

Fig. 10: GRU Architecture

MLP stands for "Multi-Layer Perceptron," which is a basic form of neural network in the field of machine learning. It is a type of feedforward model, which means that data moves through the network in one direction from the input layer to the output layer without iterations (no connections between nodes of the same layer). Each node in MLP is called a "neuron" and is organized into layers, including Input Layer, Hidden Layer, and Output Layer. Each edge in the MLP has a weight, and model training is the process of optimizing these weights so that the model can make accurate predictions on new data.

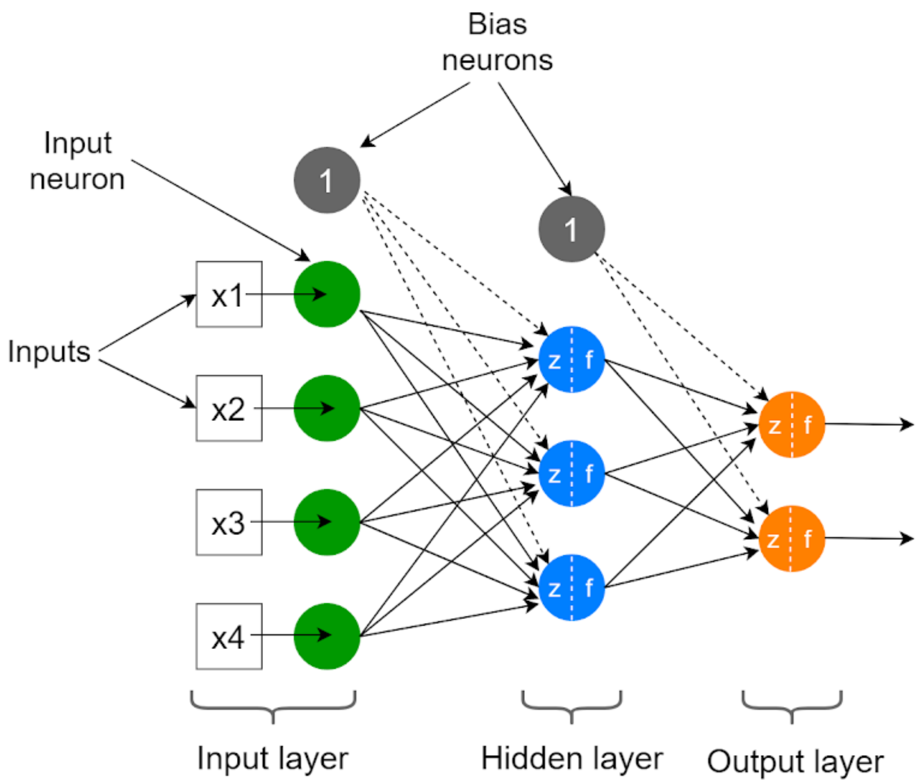

Fig. 11: MLP Architecture

Conv1D is a layer in machine learning, often used in neural network models using convolutional neural network (CNN) architecture (7)(LeCun et al., 1998). "Conv1D" stands for "1D Convolution," and it is a form of convolution performed on one-dimensional data, typically time series or numeric data. At a basic level, the Conv1D layer works like regular convolution. It uses a filter that moves through the input data to create features. This layer helps the model learn spatial representations of the input data. Specifically, in the context of time series data, Conv1D can be used to learn important features in the series, such as volatility, trends, or cyclical patterns. This is often useful in applications such as natural language processing, time series prediction, or audio processing.

4 Evaluation

4.1 Metrics

As our task is performed in the medical domain, how to evaluate the performance of models and what metric to use are affected by many factors. However the most reasoning factor is the morality and the ethical issues in medic as one wrong decision could ruin somebody's life. So to ensure that we would satisfy those reasoning, we would use 4 metrics are: Precision, Recall, F1-score, Accuracy.

Because we are working with an imbalanced dataset where all classes are equally important, using the macro average would be a good choice as it treats all classes equally.

4.1.1 Accuracy Score

The accuracy score is calculated as the ratio of correct predictions (both true positives and true negatives) to the total number of cases examined. Mathematically, it can be expressed as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- True Positives (TP): The number of cases correctly identified as having the condition.
- True Negatives (TN): The number of cases correctly identified as not having the condition.
- False Positives (FP): The number of cases incorrectly identified as having the condition (also known as Type I error).
- False Negatives (FN): The number of cases incorrectly identified as not having the condition (also known as Type II error).

4.1.2 Precision Score

Precision measures the proportion of true positive predictions among all the positive predictions made by the model. It is calculated by dividing the number of true positives by the sum of true positives and false positives.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

4.1.3 Recall Score

Recall, also known as sensitivity, measures the proportion of true positive predictions among all the actual positive samples in the dataset. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4.1.4 F1-Score

F1-Score is a weighted average of Precision and Recall, where the weights are equal. It is used to balance the trade-off between precision and recall.

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4}$$

4.2 Result

Model	**Precison**	**Recall**	**F1-Score**	**Accuracy**
LSTM	0.8699	0.6047	0.6497	0.8386
RNN	0.5575	0.3136	0.2291	0.2340
GRU	**0.9344**	0.7744	0.8317	0.9563
MLP	0.9072	0.7479	0.8058	0.9471
CNN	0.9199	**0.9080**	**0.9138**	**0.9820**

Table 1: Our models results

As we trained all five proposed models on two type of device, the results showed in [Table. 1] is the best performance of each model on both device they were trained on. Notably, the RNN model achieved quite poor performance compared to the other 4 models when it only achieved an accuracy score of 55.75%, and the remaining metrics only achieved less than 0.35. This implies that the RNN model has difficulty processing time series data. It is possible that RNN is not effective in capturing complex and time-varying relationships in time series data. Besides that, we could also see a very high performance in GRU and CNN. GRU reach the highest precision score of 93.44% while CNN achieved the best result in the three other metrics with 90.8% on Recall, 91.38% on F1 and 98.2% on Accuracy. After all, we chose the CNN to become our main model to use in the application as it performs better one most of the important metrics.

Work	**Accuracy(%)**	**Precision(%)**	**Recall(%)**
Our Model	**98.2**	91.99	90.8
Archarya et al.	93.5	92.8	93.7
Safdarian et al.	94.7	-	-
Kojuri et al.	95.6	97.9	93.3
Sun et al.	-	82.4	92.6
Liu et al.	94.4	-	-
Sharma et al.	96	**99**	93
Kachuee et al.	95.9	95.2	**95.1**

Table 2: Performance Comparison

However when we compare our best model with the models over the world [Table. 2] on this task. We can see that our model has the highest accuracy whilst the precision and recall score is not coming together with the high accuracy. This result can be explained due to the imbalanced of the dataset. The imbalanced dataset caused our model to predict more correctly on one class. As the ethical problems concerns we mention above, though our model has the highest accuracy, it is still not satisfy our requirement of precision and recall, which will heavily be affected by the ethical requirements. As we can observe in the [Fig. 12] below, whilst the model seems to perform really good on three classes "N", "V", "Q", it seems to predict not very well on the remaining classes "S", "F". The main wrong predictions on classes are class "S" wrongly predicted to class "Q" and class "F" wrongly predicted to class "Q" and "V". It means that a heartbeat problem of a patient could be predicted as "Normal" with the percentage of 15%. That number is too high and too risky to be applied into real-life problems. So this could be considered as a huge problem in our model due to the high requirements in medical domain.

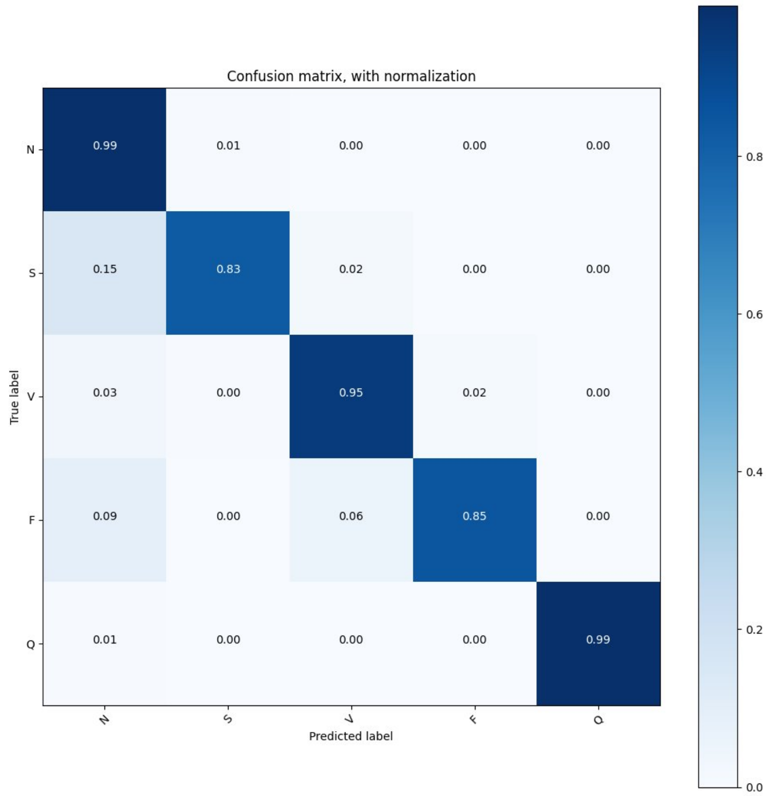

Fig. 12: Predition Confusion Matrix

5 Conclusion

In summary, the research has achieved important results in heart rate ECG classification by taking advantage of available datasets and simulating them into real-time data, delivered through the Kafka system and trained distributedly on two different types of device. Specifically, we have trained 5 models including LSTM, RNN, GRU, MLP, and CNN. According to the results, the GRU model achieved the highest performance on the precision score with 93.44, while the CNN model achieved the highest performance on all three remaining evaluation forms: recall 90.08, f1-score 91.38, and accuracy 98.20. However there are still important problems inside our researching project that must be improved if we want to apply into real life application.

Besides that, there are still some aspects that we haven't tested and taken experiments on.The first thing I can mention about is the data source collection step which we haven't tested on the time the data is transported. Another very important aspect as this step is the data security. The patients information are private data and needed to be secured. While that, we just only approach the transporting steps by renting an online web page with provide us the cloud server locating at HongKong without any information about the securities, etc. And we also didn't test on if the data will be lost or not when the hospitals sending data are far from the HongKong server. Next up, when there are many hospitals sending data at the same time, will there be a synchronization appear at any places. Finally is that we still not have a chance to connect to a real ECG to set up the program to collect and send data, that is our most failure and will need to be develop a lot more to really bring the knowledge into applications.

Overall, the research opened up positive prospects and a new knowledge for us to apply artificial intelligence with deep learning in the healthcare field and contributes a way for the future development of practical applications. We will continue to research on this project to improve the aspects we didn't consider to. This demonstrates that the combination of data simulation, distribution via Kafka, and multi-model training is an approach with great potential for supporting the medical field in the future, especially in the ECG analysis domain.

References

[1] Kachuee, M., Fazeli, S., Sarrafzadeh, M.: Ecg heartbeat classification: A deep transferable representation. In: 2018 IEEE International Conference on Healthcare Informatics (ICHI), pp. 443–444 (2018). IEEE

[2] Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. circulation **101**(23), 215–220 (2000)

[3] Moody, G.B., Mark, R.G.: The impact of the mit-bih arrhythmia database. IEEE engineering in medicine and biology magazine **20**(3), 45–50 (2001)

[4] Bousseljot, R., Kreiseler, D., Schnabel, A.: Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet (1995)

[5] Elman, J.L.: Finding structure in time. Cognitive science **14**(2), 179–211 (1990)

[6] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

[7] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[8] Cho, K.: Bart van merri enboer, dzmitry bahdanau, and yoshua bengio. On the properties of neural machinetranslation: Encoder-decoder approaches. arXiv preprintarXiv **1409**, 1259 (2014)

[9] Sun, L., Lu, Y., Yang, K., Li, S.: Ecg analysis using multiple instance learning for myocardial infarction detection. IEEE Transactions on Biomedical Engineering **59**(12), 3348–3356 (2012)

[10] Kojuri, J., Boostani, R., Dehghani, P., Nowroozipour, F., Saki, N.: Prediction of acute myocardial infarction with artificial neural networks in patients with nondiagnostic electrocardiogram. Journal of Cardiovascular Disease Research **6**, 51–59 (2015) https://doi.org/10.5530/jcdr.2015.2.2

[11] Safdarian, N., Jafarnia Dabanloo, N., Attarodi, G.: A new pattern recognition method for detection and localization of myocardial infarction using t-wave integral and total integral as extracted features from one cycle of ecg signal. Journal of Biomedical Science and Engineering **07**, 818–824 (2014) https://doi.org/10.4236/jbise.2014.710081

[12] Sharma, L.N., Tripathy, R., Dandapat, S.: Multiscale energy and eigenspace approach to detection and localization of myocardial infarction. IEEE Transactions on Biomedical Engineering **62**, 1827–1837 (2015) https://doi.org/10.1109/TBME.2015.2405134

[13] Acharya, U.R., Fujita, H., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M.: Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. Information Sciences **415-416**, 190–198 (2017) https://doi.org/10.1016/j.ins.2017.06.027

[14] Sun, K., Lu, Y., Yang, K., Li, S.: Ecg analysis using multiple instance learning for myocardial infarction detection. IEEE transactions on bio-medical engineering **59** (2012) https://doi.org/10.1109/TBME.2012.2213597

[15] Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patient-specific ecg classification by 1d convolutional neural networks. IEEE transactions on bio-medical engineering **63** (2015) https://doi.org/10.1109/TBME.2015.2468589