Khoury College of Computer Science

CS 6220 - Data Mining

Using Language Model for Arrhythmia Detection: A Comparative Study

Author:

Dong, Qishu
Wu, Chenjie
Zhang, Haoran

04/25/2023

Table of Contents

1 Introduction

Our project is to explore the application of machine learning algorithms in automatically detecting arrhythmia from ECG data. The goal is to create a model capable of distinguishing abnormal ECG signals from normal signals, enhancing the diagnosis and treatment of arrhythmia.

2 Background

Arrhythmia is a common and potentially life-threatening condition that affects millions of people worldwide. It can be viewed as irregular heartbeat, affects millions of people worldwide and is a significant cause of morbidity and mortality. It can result in a wide range of symptoms, from mild palpitations to cardiac arrest. It is been detected within more than 1 in 50 Americans under the age of 65 and 1 in 10 Americans over age 65 in 2023. Therefore, Early detection and diagnosis are essential for effective treatment and prevention of complications.

3 Data Analysis and Preprocessing

Our dataset, MIT-BIH Arrhythmia Database, consists of ECG recordings from 47 patients, which includes a total of approximately 30,456,000 ECG signals. The database also provides reference annotations for each beat, totaling around 110,000 annotations.

The dataset comprises ECG data and annotation data, which throws out the difficulty in labeling heart rates with according annotations. That is to say, the Heart Rate Dataset has two columns: timestamp (sampled at 360HZ) and heart beat voltage at such timestamp, while the Annotation Dataset has columns for each heartbeat: the timestamp and the corresponding annotation. To label the heart rate, the first step is to segment heart rate signal into individual heart beat, and label every heart beat with N (normal heart beat) or not N. This pre-processing method is inspired by this paper *Symbols used in plots* 2016.

Although algorithms like the bandpass filter can be used to extract heartbeats from ECG waveforms, relying solely on this method may not be sufficient. This is because some heartbeats may fall outside the limitations of the algorithm, leading to inaccuracies in heartbeat detection. However, by utilizing the full annotation data for each heartbeat, we can accurately segment each heartbeat and ensure that no heartbeats are missed or misidentified. The approach we took is to normalize and resample heart rate onto annotations, so that we generate a series of heart beats, each with 128 coordinates, and we can assign classes to every heart beat.

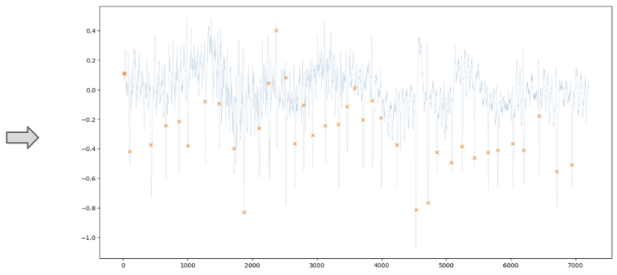
Signals

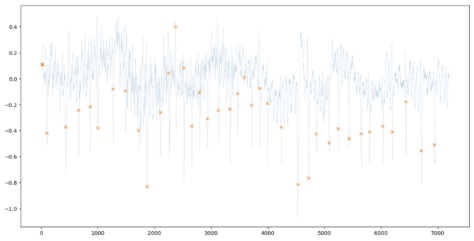
Time (elapsed)	MLII (mV)	V1 (mV)
00:28:30.000	-0.295	-0.060
00:28:30.003	-0.340	-0.120
00:28:30.006	-0.360	-0.170
00:28:30.008	-0.325	-0.240
00:28:30.011	-0.260	-0.345
00:28:30.014	-0.185	-0.420
00:28:30.017	-0.150	-0.540
00:28:30.019	-0.095	-0.680

Annotations

Annotator: atr

Time (elapsed)	Sample #	Type	Sub	Chan	Num	Aux
00:28:30.083	615630	N	0	0	0	
00:28:31.000	615960	N	0	0	0	
00:28:31.883	616278	N	0	0	0	
00:28:32.781	616601	N	0	0	0	
00:28:33.636	616909	N	0	0	0	
00:28:34.511	617224	N	0	0	0	
00:28:35.364	617531	N	0	0	0	
00:28:36.225	617841	N	0	0	0	
00:28:37.136	618169	N	0	0	0	
00:28:38.006	618482	N	0	0	0	
00:28:38.831	618779	N	0	0	0	
00:28:39.617	619062	N	0	0	0	





```
data[100]

{'beats': array([[-0.0275445 , -0.03476478, -0.04212791, ...,  0.0418236 ,
          0.02505303, -0.01262503],
        [-0.0049647 , -0.02726073, -0.02459804, ...,  0.00385776,
         -0.01406059, -0.01533878],
        [-0.01315528, -0.03954138, -0.02729585, ...,  0.04331073,
         -0.00846112, -0.02197514],
        ...,
        [-0.03968485, -0.03872239, -0.03920411, ..., -0.03969629,
         -0.03516211, -0.0505896 ],
        [-0.03610631, -0.04026123, -0.03522159, ..., -0.05704345,
         -0.05580564, -0.05910287],
        [-0.09061379, -0.04489519, -0.04724623, ..., -0.07697315,
         -0.04605489, -0.07292854]]),
 'class': array(['S', 'S', 'S', ..., 'N', 'N', 'N'], dtype='<U1'),
```

4 Approach

4.1 Related Work: BERT

The pre-trained deep learning model for natural language processing tasks, Bidirectional Encoder Representations from Transformers (BERT), has been able to achieve state-of-the-art performance on diverse benchmarks, according to Devlin et al. (2018). Since its introduction, BERT has been extensively employed in various NLP applications, such as sentiment analysis (Devlin et al., 2019), text classification, question answering, and machine translation. Furthermore, there has been an increasing interest in using BERT for non-NLP tasks, such as image classification and speech recognition, as recent studies have revealed that BERT can be fine-tuned for specific tasks by integrating task-specific input and output layers.

4.2 Related Work: Random Forest

Random Forest is a popular machine learning algorithm used for classification, regression and feature selection tasks. It is an ensemble learning method that combines multiple decision trees, where each decision tree is built using a subset of the available features and training data. During training, the algorithm randomly selects a subset of the available features for each decision tree, and then uses a random subset of the training data to train each tree. This randomness helps to reduce overfitting and increase the generalizability of the model.

4.3 Modeling

In our approach, BERT is fine-tuned for the arrhythmia detection. This involves adding a classification layer on top of the pretrained BERT model and training it on labeled ECG data from the MIT-BIH database. The classification layer is trained to predict the presence or absence of arrhythmia in each ECG recording. Here is a more detailed explanation of the different components of the classification layer and the training process:

Dropout layer: A dropout layer is a regularization technique that helps prevent overfitting by randomly dropping out (setting to zero) some of the activations in the preceding layer during training. 10% of the activations will be randomly dropped out during each training iteration.

In contrast to the approach used for the BERT model, it seems that for the Random Forest Classifier, the numerical values of the ECG voltage signals are directly used as input features.

Linear layer: The linear layer is the final layer in the classification layer, which takes the output of the BERT model and maps it to a binary classification (arrhythmia or no arrhythmia). This layer is typically initialized with random weights and trained using backpropagation to minimize the cross-entropy loss.

Cross-entropy loss: Cross-entropy loss is a commonly used loss function for binary classification problems. It measures the difference between the predicted probabilities and the true labels, and is optimized during training to improve the accuracy of the model.

Adam optimizer: Adam is a popular optimization algorithm for training deep learning models. It uses a combination of adaptive learning rates and momentum to accelerate the training process and improve the convergence of the model.

During training, the labeled heart beat voltages, extracted from the MIT-BIH ECG database, is used to update the weights of the classification layer and fine-tune the BERT model for arrhythmia detection. It is important to note that each heart beat is resampled, normalized, and concatenated into a long sequence of voltages, which can be fed into the pretrained BERT model as input. The training process involves 10 epochs, where each epoch consists of feeding batches of ECG data into the model, computing the loss and gradients, and updating the weights using the optimizer.

In contrast to the approach used for the BERT model, for the Random Forest Classifier, the numerical values of the ECG voltage signals are directly used as input features. The number of decision trees in a Random Forest Classifier is a hyperparameter that can be tuned during the model training process. Generally, a larger number of trees will increase the accuracy of the model, but also increase the computational complexity and training time. 100 decision trees are used in our approach.

5 Implementation Specifics

5.1 Sampling strategy

According to section 3, "Data Analysis and Preprocessing", the ECG data is divided into individual heart beats based on the annotations provided in the MIT-BIH dataset. Each heart beat is then resampled to 128 data points with respect to time and normalized to align with the x-axis. In addition, for BERT model, data is concatenated into a single long sequence of voltages for each heart beat.

5.2 Hyperparameter specifications

5.2.1 BERT

classification layer:
a dropout layer with 0.1 dropout rate and a linear layer
loss function: Cross Entropy
optimizer: Adam
learning rate: 2^{-5}
epsilon: 1^{-8}
epoch: 10
batch size: 16
shuffle data: True

5.2.2 Random Forest

100 decision trees
The function to measure the quality of a split: Gini impurity
Unconstrained maximum depth of decision tree, maximum leaf nodes
The minimum number of samples required to split an internal node: 2
The minimum number of samples required to be at a leaf node: 1
The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node: 0.0

5.3 Instruction on running code

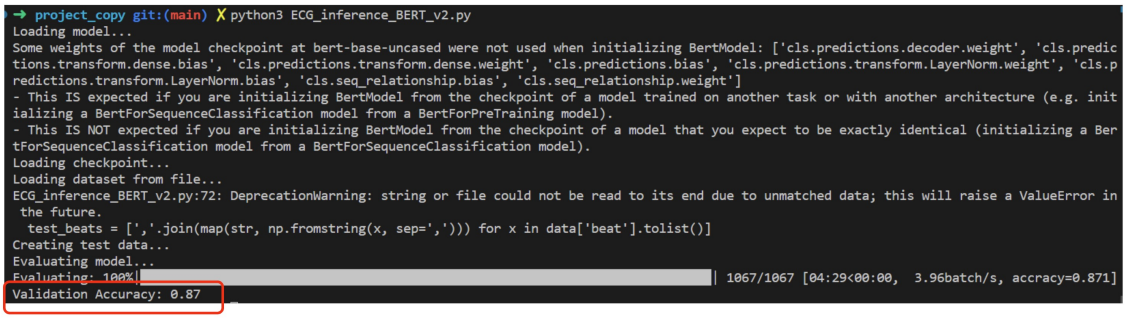
The feature engineering part is ready to run without loading local files. We have included wfdb API to retrieve data automatically.
The "Train with Random Forest" part is ready to run on Colab.
The "Trained with Pre-Trained BERT" part was rewrote and ran on local machine.

5.4 Team Contribution

In this project, we had a team consisting of three members, Haoran, Qishu, and Chenjie, who all made significant contributions towards achieving our research goals. Haoran was mainly responsible for preprocessing the data, which involved cleaning, formatting, and preparing the dataset for analysis. He utilized various techniques as mentioned in part 4.3 modeling, which helped in improving the accuracy of the models. Qishu, on the other hand, was mainly responsible for implementing BERT, a pre-trained deep learning model, for NLP tasks. She fine-tuned the model to suit the specific needs of the project and utilized it to extract relevant information from the text data. Finally, Chenjie was mainly responsible for implementing a random forest model. He fine-tuned both models and implemented various hyperparameter specifications to achieve the best possible accuracy. Together, the contributions of Haoran, Qishu, and Chenjie were instrumental in the success of the project.

6 Results

The BERT model is trained on RTX-3090 for 2 hours and resulted in 87 percent validation accuracy.



The Random Forest model is trained on Colab for 15 minutes and resulted in 98.45 percent validation accuracy.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("Accuracy:", accuracy_score(test_labels, y_pred))
print("Precision:", precision_score(test_labels, y_pred, average='macro'))
print("Recall:", recall_score(test_labels, y_pred, average='macro'))
print("F1 score:", f1_score(test_labels, y_pred, average='macro'))
```

```
Accuracy: 0.9845152381891705
Precision: 0.9834132918204657
Recall: 0.933672684858976
F1 score: 0.9568010848692545
```

7 Conclusion

This project focused on the development of a machine learning model capable of distinguishing cardiac arrhythmias from normal electrocardiogram (ECG) signals. The primary aim was to build a reliable tool that could assist healthcare professionals in the diagnosis of arrhythmias, potentially leading to improved patient outcomes.

This project has successfully trained, compared two models and creatively identified popular deep learning model, Bidirectional Encoder Representations from Transformers (BERT), as a powerful tool. Models were trained and tested using the widely-used MIT-BIH Arrhythmia Database, which contains ECG recordings from a diverse range of patients with various cardiac conditions. A model was optimized using a random forest algorithm, which resulted in a high accuracy rate of 98.45% in identifying arrhythmias. And the other model was optimized using BERT model, with 87% accuracy rate.

The BERT model is typically pre-trained on vast amounts of unlabeled text data to acquire comprehensive language representations. As a future work for this project, the model could be to adapt for pre-training to predict the next value in a sequence of ECG data, similar to its training on text data, before it is fine-tuned with a classification layer. We expect this approach would greatly improve the accuracy of the BERT model.

Another approach that might also improve the result for BERT model is to convert each heart beat into a sequence of fixed-length numerical vectors, which can be fed into the BERT model as input. This approach is known as sequence embedding and might have several advantages over the resampling of voltage against time, such as preserving more information about the signal and reducing the amount of preprocessing required. It can be done using techniques like sliding window segmentation, where the ECG signal is divided into overlapping windows of fixed length, and each window is converted into a vector using features like the amplitude (voltage), duration, and shape of the waveform.

The findings of this project have important implications for the field of cardiology and healthcare in general. The development of an accurate and reliable machine learning model for the classification of arrhythmia has the potential to significantly improve patient outcomes. It can also help healthcare professionals to make more informed decisions when it comes to treatment plans and improve the overall quality of care.

Bibliography

Devlin, Jacob et al. (2019). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: URL: https://arxiv.org/abs/1810.04805 (visited on 15th Mar. 2023).

Goldberger, A. et al. (2000). 'PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals [Online]'. In: *Circulation* 101.23, E215–E220. DOI: 10.1161/01.cir.101.23.e215.

Moody, George B and Roger G Mark (2001). 'The impact of the MIT-BIH Arrhythmia Database'. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3, pp. 45–50. DOI: 10.1109/51.932724. URL: https://physionet.org/content/mitdb/1.0.0/ (visited on 15th Mar. 2023).

Symbols used in plots (2016). URL: https://archive.physionet.org/physiobank/annotations.shtml (visited on 15th Mar. 2023).

Yamaç, Mehmet et al. (2022). 'A Personalized Zero-Shot ECG Arrhythmia Monitoring System: From Sparse Representation Based Domain Adaption to Energy Efficient Abnormal Beat Detection for Practical ECG Surveillance'. In: URL: https://arxiv.org/abs/2207.07089 (visited on 15th Mar. 2023).