Innovative Features for Atrial Fibrillation Detection

D. Johnston, Brigham Young University
Faculty Advisors: Dr. Yishi Wang and Dr. Cuixian Chen
2019 UNCW REU Program

ABSTRACT. In this report, we present a new approach to atrial fibrillation (AF) detection. First, we explore associated works detailing previous methods used to detect AF. We then provide summaries of Physionet's MIT-BIH Atrial Fibrillation Database and Computing in Cardiology Challenge 2017 Database and present results of our efforts to develop a robust model for detecting AF in electrocardiogram data using novel features based on RR intervals. Our contributions include a new feature measuring irregularity of RR intervals and innovative applications of previous work to generate novel features. Using a random forest classifier and 12 original features, we achieved accuracy of 0.963 and averaged F1-score of 0.962 with leave-one-person-out cross validation on the MIT-BIH data. The same model achieved accuracy of 0.949 and averaged F1-score of 0.813 with 5-fold cross validation on the 2017 Challenge data.

Keywords: Atrial Fibrillation, ECG, Electrocardiogram, R peak, RR Interval, AF Detection, Cardiac Arrhythmia

1. Introduction

Atrial Fibrillation (AF) is a quick, irregular heartbeat that occurs when the atria (the upper chamber of the heart) beats out of rhythm. It is currently the most common cardiac rhythm disorder and is known to greatly increase risk of heart failure and stroke (Shields and Lip, 2015). Normal heartbeats are initiated by an impulse from the Sinoatrial Node. Atrial Fibrillation can occur when other electrical signals interfere with the impulse from the Sinoatrial node, causing the atria to quiver. Wearable devices, such as smart watches, are equipped to measure heart rates and detect atrial fibrillation. We aim to develop a robust model for real-time detection with novel features.

There are various methods for detecting AF, such as disappearance of P waves in an electrocardiogram (ECG), but these are often unreliable due to the chaotic nature of AF (Moody and Mark, 1983). Using portable or wearable devices it can be especially difficult to detect such smaller changes to an ECG signal. The most reliable measure is that of RR intervals, the time passed between the peaks of two R waves in an ECG. R waves are the largest waves present in a heartbeat and thus the easiest to detect consistently. Several different methods of AF detection utilizing RR intervals have been explored (Moody and Mark, 1983; Lian et al., 2011; Tateno and Glass, 2000; Duverney et al., 2002; Ghodrati et al., 2008; Shouldice et al., 2007). We will present some results of literature on this subject and conduct an analysis of the MIT-BIH Atrial Fibrillation Database and the PhysioNet/Computing in Cardiology Challenge 2017 Database using new features for atrial fibrillation detection. Our contributions in this paper include:

- Innovative features based on transitions described by Mark and Moody (Moody and Mark, 1983)

Received by the editors May 8, 2020.

- A novel feature generalized from the Non-Empty Cell count described by Lian et al. (Lian et al., 2011)
- A powerful new feature for measuring irregular irregularity in RR intervals.

In our discussion and future work, we also propose a new method for preprocessing and measuring noise in an ECG signal.

2. Associated Work

It is difficult to detect AF based solely on RR interval length. The following papers propose some methods for extracting features from RR interval data to fit an effective model for AF detection. They do not all use the same dataset or the same ECG source (some use small, portable devices, while others use data from professional grade equipment), or even the same metric for model performance, so a comparison between the papers is difficult to make. In this section, a number of methodologies and reported results are presented.

2.1. Moody and Mark Paper

Moody and Mark apply their methodology to ECG data that became the MIT-BIH Atrial Fibrillation Dataset (Moody and Mark, 1983; Goldberger et al., 2000). They classify RR intervals as short, normal, or long by comparing each interval to a running mean at that point, given by

$$Rmean(i) = 0.75 * Rmean(i-1) + 0.25 * RR(i), \tag{2.1}$$

for each RR interval, $RR(i)$, that is shorter than 1.5 seconds. If the RR interval is within 15% of the running mean, that is,

$$0.85 * Rmean(i) \leq RR(i) \leq 1.15 * Rmean(i), \tag{2.2}$$

then the interval is classified as normal. If it is smaller than this lower bound, it is considered short, and if it is larger than the upper bound, it is considered long.

Moody and Mark then use these classes to construct a Markov model for the probabilities of transitions between these intervals of different lengths. For example, the probability that the next RR interval will be short given that the current RR interval is long. This would be considered a long-to-short transition (LS). They found that ECG periods where AF is present display different transition probabilities than ECG data of a normal heartbeat rhythm, or even ECG data where arrhythmia other than AF is present.

Using this Markov model, they were able to achieve sensitivity of 0.900 with positive predictivity of 0.801. After implementing interpolation to reduce quantization error and a first-order filter to remove noise from the signal, the model was able to achieve 0.961 sensitivity with 0.868 positive predictivity (Moody and Mark, 1983).

2.2. Tateno and Glass Paper

Tateno and Glass propose a new method of detecting AF (Tateno and Glass, 2000). They use the same MIT-BIH AF database to develop their models. Instead of extracting features based on the RR interval length relative to a running mean, they extract a new feature, ΔRR, the difference between RR interval lengths. This is given by:

$$\Delta RR(i) = RR(i) - RR(i-1). \tag{2.3}$$

They then segment the data where AF is present into blocks of 100 heartbeats and create density histograms for the average RR interval length within a block and for the average ΔRR value within a block. These histograms serve as standard density histograms for AF.

To classify a given block of 100 heartbeats as AF or non-AF, they compare the density histograms of RR and ΔRR for that block to the previously calculated standard density histograms using the Kolmogorov-Smirnov test. If the tests results in a statistically significant similarity between the histograms ($P = 0.01$), the block is classified as AF. Otherwise, it is considered non-AF.

Sensitivity and specificity are reported. These metrics are defined by $Sensitivity = \frac{TP}{TP+FN}$ and $Specificity = \frac{TN}{TN+FP}$, where TP is the number of true positive predictions, TN is the number of true negative predictions, FP is the number of false positive predictions, and FN is the number of false negative predictions. Using the RR histograms, they achieve 0.539 sensitivity and 0.989 specificity on the dataset they used to construct the standard density histograms. Using a new dataset, they attain 0.259 sensitivity and 0.932 specificity. Using the ΔRR histograms, they achieve 0.932 sensitivity and 0.967 specificity with the dataset used to construct the histograms. They attain sensitivity of 0.888 and specificity of 0.641 for the new dataset (Tateno and Glass, 2000). They mention some limitations of the paper; firstly, they do not optimize the number of beats in a block. There may be more effective numbers than 100. In addition, they express concerns about the rhythm assessment in the MIT-BIH database, as their analysis leads them to believe that some portions of the data may have been poorly classified.

2.3. Lian et al. Paper

Lian et al. use the MIT-BIH AF database, as well as 3 other PhysioNet heart rhythm databases, to evaluate model performance (Lian et al., 2011). In their 2011 paper, they endeavor to create a metric that incorporates information from both the RR interval lengths and the difference between consecutive RR interval lengths, ΔRR, as defined above by Tateno and Glass. They redefine this metric as dRR in their paper, and that is how we refer to it in this paper.

Their algorithm for AF detection is based on the scatter plot of RR versus dRR. Segmenting the data into intervals of 32, 64, or 128 RR intervals, they plot data points for one interval and measure how spread out the data points are by dividing the resulting map into a grid with resolution 25 ms and counting the number of non-empty cells in the grid.

If AF is present in the sample, the scatter plot is much more spread out due to the irregular relationship between RR and dRR during atrial fibrillation, resulting in a higher number of non-empty cells (NEC). Figure 2.1 displays an example illustration of the scatter plots. Clearly, if a grid is drawn over the plots, the one with AF has many more non-empty grid cells than the plot without AF present.

This metric divides the AF data from the non-AF data quite well. Over all the data, the best results are achieved using segments of 128 RR intervals. The model achieves sensitivity of 0.959 and specificity of 0.954. At 0.958 sensitivity and 0.943 specificity, the segments of 64 RR intervals perform almost as well.

2.4. Clifford et al. Paper

The PhysioNet/Computing in Cardiology Challenge 2017 focused on determining whether a short interval (9-61 seconds) of ECG recording is classified as atrial fibrillation, normal, other rhythm, or noise (Clifford et al., 2017). This paper presents an overview of this challenge and the

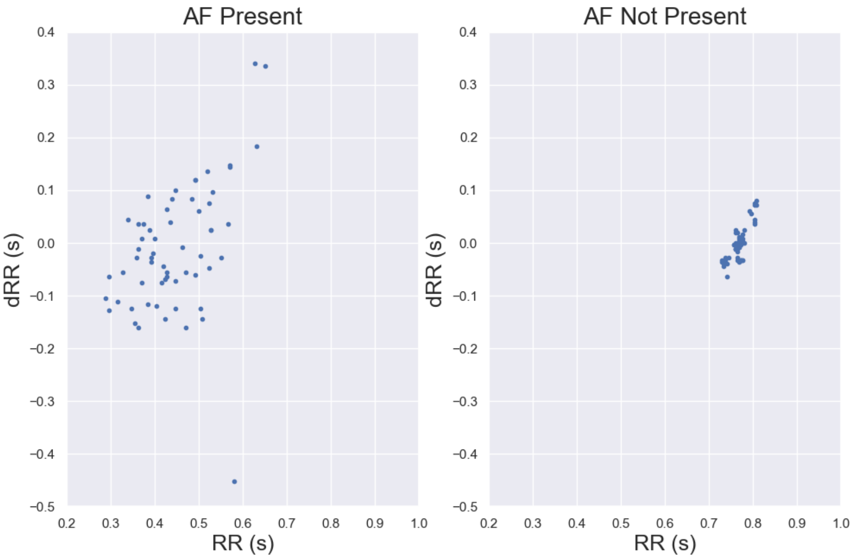

FIGURE 2.1. Scatter Plots of RR vs dRR in MIT-BIH Data

results, including results achieved through combining a number of the top models through voting techniques.

The models are scored by averaging the F1-scores for the normal, AF, and other classes. For each class, the F1-score is defined to be:

$$F1 = \frac{2pr}{p+r}, \tag{2.4}$$

where p is precision and r is recall. Note precision is defined by $p = \frac{TP}{TP+FP}$ and recall is defined by $r = \frac{TP}{TP+FN}$, which is equivalent to sensitivity. The top-performing models have averaged F1-scores that round to 0.83. The highest score of 0.868 is achieved by weighted voting of 45 algorithms.

2.5. Behar et al. Paper

In their publication based on their entry in the 2017 competition, Behar et al. describe their method and results (Behar et al., 2017). They extract a wide variety of features describing signal quality, predictability of the RR intervals, ECG morphology, and heart rate variability. They employ dozens of features, but the cascaded approach they use is the most interesting component of their paper.

They train three Support Vector Machines (SVM) with Radial Basis Function (RBF) kernels to be used for classification. The first distinguishes between normal rhythms and rhythms that were not normal. The second differentiates the not normal rhythms into AF rhythms and non-AF rhythms. The third determines if the remaining rhythms are other rhythms or simply noise.

This cascaded approach placed well in the competition, resulting in an F1-score of 0.80 on the final test set in the 2017 Challenge (Behar et al., 2017).

3. Atrial Fibrillation Datasets

In this paper, we consider two data sets that have been applied for AF detection in our studies: The MIT-BIH Atrial Fibrillation Database, which we refer to as the “MIT-BIH Data,” and the PhysioNet/Computing in Cardiology (CinC) Challenge 2017 Data, which we refer to as the “Challenge Data.”

3.1. The MIT-BIH Data

The MIT-BIH Atrial Fibrillation Database contains 10 hours of dual-lead ECG recordings for 25 subjects (Moody and Mark, 1983; Goldberger et al., 2000). Two of these subjects do not have complete data files, so they are not used for our studies in this paper. These recordings are sampled at 250 Hz and contain annotations by medical experts indicating the heart rhythm being experienced. Rhythm types include Normal (N), Atrial Fibrillation (AF), Atrial Flutter (AFL), and Junctional Rhythm (J). For our purposes, we relabel these rhythms as AF or non-AF by grouping AFL, N, and J rhythms together in the non-AF group. The annotations also indicated the sample number of the peak of the R wave for each heartbeat. This information was used to extract the RR interval lengths for all of the data, which were then used to extract new features to be used in our studies.

3.2. The Challenge Data

The PhysioNet/Computing in Cadiology Challenge 2017 focused on classifying a short interval (9-61 seconds) of ECG recording as one of four cardiac rhythm types (Clifford et al., 2017). The dataset used in this challenged is composed of 12,186 ECGs and was collected using a small, portable device. Of the ECGs, 8,528 were made available as a public training set, while 3,658 were retained as a private, hidden test set. A subset of 300 ECGs from the public training set are included as a public validation set.

The main differences between the Challenge Data and the MIT-BIH Data include that the Challenge Data was generated with single-lead, portable devices, not professional-grade medical equipment, and that the Challenge Data was segmented beforehand and not given as 10 hours of constant ECG recording for each subject. These segments were annotated as AF (A), Normal (N), Other (O), or Noise ($\sim$), but there were no RR interval annotations given, unlike the MIT-BIH Data. Figure 3.1 displays the distribution of rhythms present in the Challenge data.

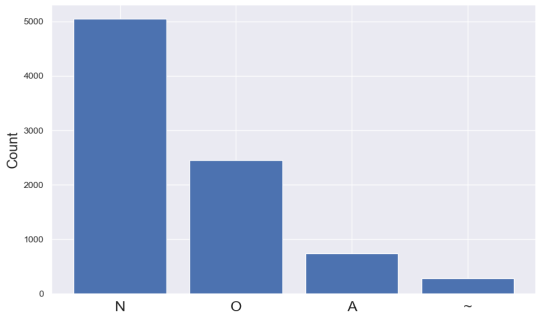

FIGURE 3.1. Challenge Data: Multiclass Rhythm Distribution

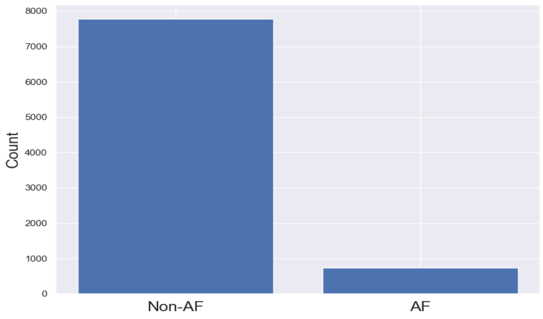

FIGURE 3.2. Challenge Data: Binary Rhythm Distribution

It is clear that there is a much more severe imbalance in the data here than was present in the MIT-BIH dataset. Normal rhythms are by far the most numerous, followed by Other Rhythms. Figure 3.2 shows the distribution when we only consider AF versus Non-AF rhythms. Each segment has been expertly labeled as one of the four rhythms indicated above; any rhythm that is not AF has been placed in the Non-AF group.

4. Data Preprocessing

Preliminary processing of the data was conducted using Python 3.6.8 (Python Software Foundation, 1991). This included formatting and segmenting the data, as well as testing peak-detection algorithms.

4.1. Preprocessing the MIT-BIH Data

By using Python's Waveform Database (WFDB) package, the ECG signals were extracted and visualized, and the R peak annotations were extracted for each of the 23 subjects of the MIT-BIH dataset (Python Software Foundation, 1991; Xie et al., 2016). This data is given as one 10-hour-long segment for each subject, but we wish to be able to identify AF in real time with as little as 30 seconds of data available. With that in mind, the data was segmented into 27,017 signals of length 30 seconds. Each segment was then classified as AF or Non-AF depending on the rhythm present in the majority of the segment. As there are expert annotations indicating the rhythm type present at every sample in the ECG, a majority vote is taken to determine if a segment is AF or Non-AF. If the majority of the samples in the segment are marked AF, the segment is marked AF. Otherwise it is marked Non-AF. Figure 4.1 displays the distribution of those classes across the whole dataset.

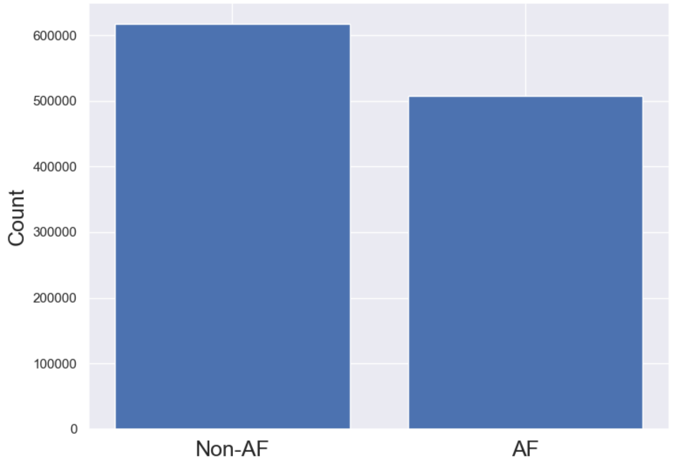

FIGURE 4.1. MIT-BIH Data: Rhythm Distribution

As the data has already annotated by medical professionals and we did not need to extract R-wave peaks from the signal, there was little preprocessing to be done regarding signal quality (Moody and Mark, 1983; Goldberger et al., 2000). At the beginning of a recording, there could be small amounts of noise present in the signal, so the first 90 seconds of each ECG signal were not used in the segmenting process. The MIT-BIH data was segmented into 27,017 segments from which to extract predictive features and other desired information.

4.2. Preprocessing the Challenge Data

The WFDB library that has been developed in a variety of programming languages provides multiple methods for detecting peaks of R waves (Xie et al., 2016). In the Python package, we made use of the GQRS and XQRS detectors to extract this data from each segment (Xie et al., 2016). Each algorithm has strengths and weaknesses, and we ran into some problems attempting to extract these peaks of R waves. For example, on one normal segment, we see the results of both algorithms in Figure 4.2. Each red × marks where the algorithm labels an R wave peak. It is clear from the figure that the XQRS algorithm has labeled this segment much more precisely than the GQRS algorithm.

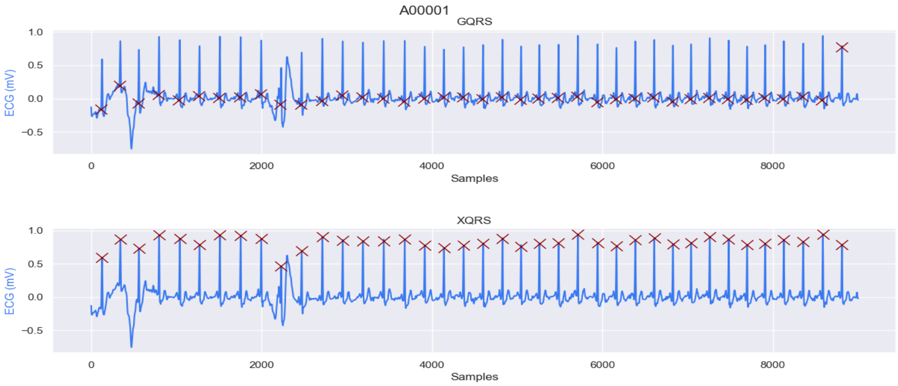

FIGURE 4.2. Labeling R Wave Peaks: GQRS vs XQRS

Initially, it appears that the XQRS algorithm is a much better choice for peak detection. However, when dealing with noise or spikes in the ECG data, the GQRS algorithm is much more robust than the XQRS algorithm. In Figure 4.3 we can see the results of the two algorithms when given a segment that is labeled as normal, but has one brief spike in the data.

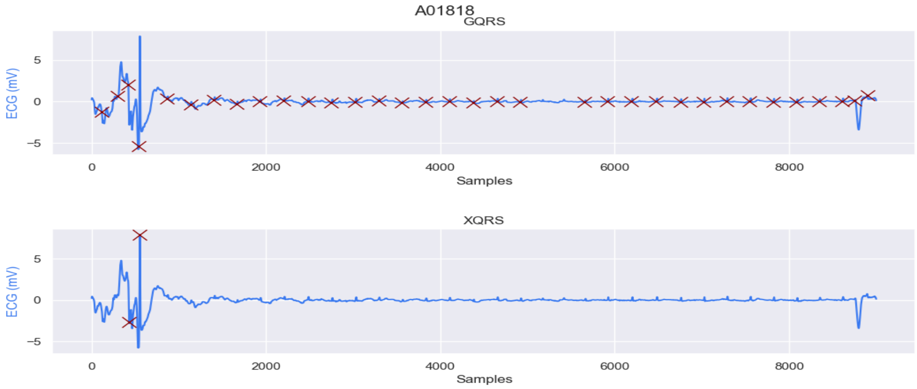

FIGURE 4.3. GQRS Outperforms XQRS with Spikes in ECG

The beginning of the reading has some noise in it, but the last 8000 samples are read properly and show regular heart rhythms. The GQRS algorithm detects this fairly well, while the XQRS algorithm performs horrendously, failing to pick up any peaks after the very large spike at the beginning of the data. This is because the XQRS algorithm is an adaptation of the Pan-Tompkins algorithm and depends on a running mean of the peak altitudes, so a large spike can throw off peak detection for the rest of the signal (Pan and Tompkins, 1985).

Many other segments marked as noise did not yield any results from the algorithms, making it difficult to extract features as there were no RR intervals given by the data. Figure 4.4 shows what these segments of “noise” can look like and what the algorithms attempt to classify as R wave peaks.

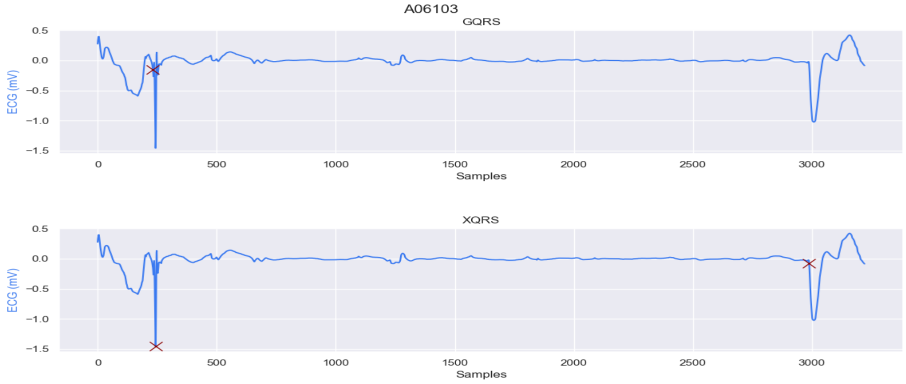

FIGURE 4.4. Example of a Segment of Noise

Because the GQRS algorithm performed more consistently in the presence of noise, it was selected to produce initial results. If fewer than 5 heartbeats were detected using this algorithm, the segment was not included in the training of the model as no reliable features based on R peaks could be extracted from it.

5. Feature Extraction

A variety of features were extracted from the R peaks of the ECG data. This section describes a feature used to measure signal quality, one original feature created to measure irregularity in the heartbeat, and innovative ideas to apply the findings of the some of the authors mentioned in the Associated Work section to generate new features.

5.1. Innovative Features Based on RR Interval Transitions

Moody and Mark developed classifications for transitions between different RR interval lengths and used these transitions to build a Markov model (Moody and Mark, 1983). Instead of building a Markov model, we propose an innovative idea to apply these transitions as predictive features. For each segment of ECG data, the proportion of the transitions that belong to each class are calculated and used as features. These features are labeled SL for short to long transition, LN for long to normal transition, etc.

Figure 5.1 presents an illustration of the average transition proportions in a segment using all 23 subjects of the MIT-BIH data. Table 11.1 in the appendix provides a numerical summary. Note

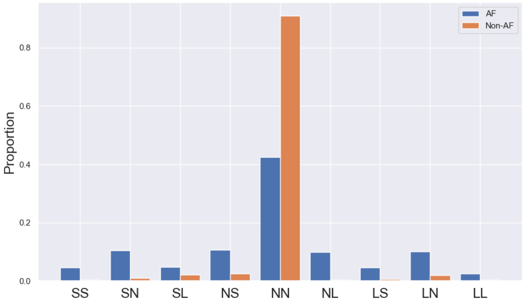

FIGURE 5.1. Average Proportions of Transitions in MIT-BIH Data

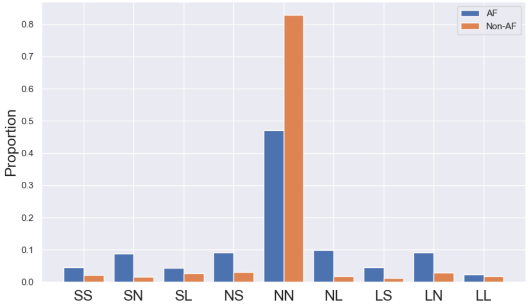

FIGURE 5.2. Average Proportions of Transitions in Challenge Data

the vastly different transition proportions in AF and Non-AF segments. As expected, the irregular AF data has a considerably smaller proportion of NN transitions. This holds true for the Challenge Data also, as shown in Figure 5.2 and Table 11.2 in the appendix. These transitions are consistent across databases in the differences between AF and Non-AF segments.

5.2. NEC Rate

The non-empty cell (NEC) feature has been shown in Lian et al., 2011 to be an effective measure of irregularity to be used in AF detection (Lian et al., 2011). However, this feature requires a fixed number of heartbeats in each segment. We wish to develop a feature that can be used for a fixed amount of time instead of a fixed amount of heartbeats. We extend NEC to a new feature, NEC Rate, which is calculated by simply dividing the NEC feature by the number of heartbeats in the segment. The histograms presented in Figure 5.3 demonstrate how well the NEC Rate feature separates the AF segments from the Non-AF segments.

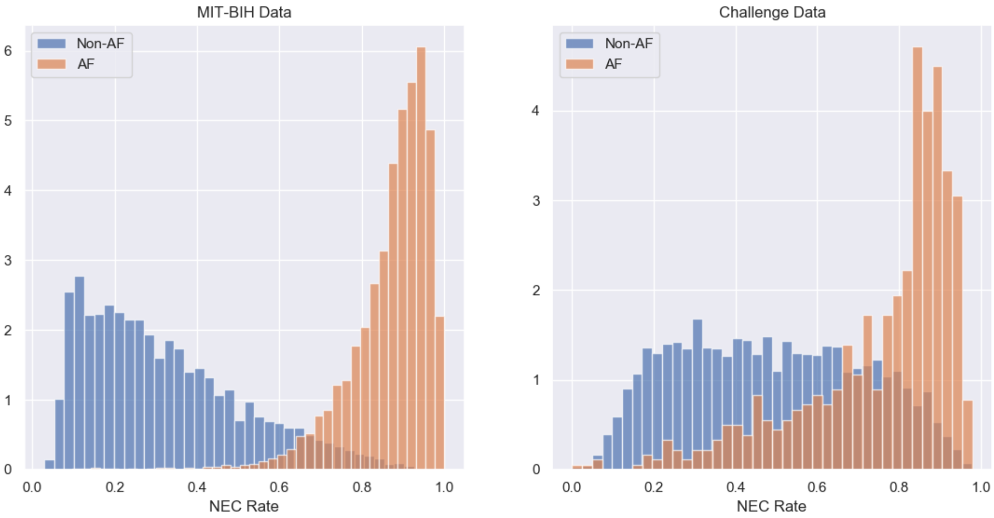

FIGURE 5.3. Histogram of NEC Rate

It is imperfect, but there is a clear trend evident in the histograms. When atrial fibrillation is present, the NEC Rate is more likely to be high. The trend is much clearer with the MIT-BIH data because it is predominantly AF and normal rhythms. There are many other rhythms types in the Challenge Data that may present different challenges in AF detection. We see a larger overlap in the histograms of the challenge data, which reflects similarities between R-wave patterns in AF rhythms and Other rhythms that are not present in normal rhythms. The histograms are normalized to account for the imbalance in the dataset.

5.3. Utilizing Signal Quality Index as a Feature

There are a variety of Signal Quality Indices (SQIs) that have been developed to measure the quality of a signal. One SQI that can be implemented with only information on R peaks is called bSQI (Liu et al., 2018). As discussed previously, the XQRS and GQRS detectors perform very differently from one another. The bSQI metric compares the results from both detectors to establish a measure of signal quality. It calculates the proportion of R peaks that both detectors identify. If N_1 is the number of R peaks that the first detector identifies, and N_2 is the number of R peaks that the second detector identifies, bSQI is given by

$$bSQI = \frac{min(N_1, N_2)}{max(N_1, N_2)}. \tag{5.1}$$

Because the MIT-BIH data has been annotated by experts, bSQI was only used on the Challenge data. bSQI is used as a measure of how well the algorithms agree on where the R peaks are, so it cannot be used on the MIT-BIH data because the location of the R peaks is already provided. Employing bSQI as a feature improved model performance.

5.4. A Novel Feature for Measuring Irregular Irregularity

In an effort to measure irregularity in the RR intervals, we developed a novel feature called ddRR, which makes use of the dRR values that Lian et al. used. For any given segment of ECG data, ddRR is calculated to be the mean absolute value of the differences between consecutive dRR values. For a segment with n dRR values, this can be written as

$$ddRR = \frac{1}{n-1} \sum_{i=2}^{n} |dRR_i - dRR_{i-1}|. \tag{5.2}$$

This new value shows promise as a predictive feature. In fact, ddRR gave the highest feature importance of any feature used on the Challenge Data, as shown in Table 6.2.

6. Experiment Design and Methodology

In this section, we present important elements of our methodology, including model selection and the use of feature importance to reduce dimensionality for additional testing. For model performance, we employ cross validation to get results. Because the MIT-BIH data is taken from 23 individual subjects, leave-one-out cross validation is used. 5-fold cross validation to get results on the Challenge data as it has unique subjects for each segment.

6.1. Model Selection

After extracting the features described above, we consider a variety of statistical learning methods in our preliminary studies. We compared performances of SVMs with linear and RBF kernels, linear discriminant analysis (LDA), logistic regression, gradient boosting, and random forest models. The predominant measures of model performance were accuracy and averaged F1-score. These metrics were chosen because the 2017 Physionet Challenge used an averaged F1-score as their metric for performance. They only averaged the F1-scores for the normal, other, and AF classes, ignoring the F1-score for noise. Preliminary results showed comparable performance across these models, as shown in Table 6.1.

TABLE 6.1. Preliminary Model Results with MIT-BIH Data

	Accuracy	F1-Score
Random Forest	0.964	0.963
Gradient Boosting	0.963	0.962
SVM with RBF Kernel	0.955	0.953
SVM with Linear Kernel	0.953	0.951
LDA	0.952	0.950
Logistic Regression	0.951	0.949

Random forest and gradient boosting slightly outperformed the other models. The random forest model was chosen for the purposes of this paper as it does not require normalization of data. Futhermore, it can be easily modified to account for imbalanced data through weights, and it can be easily generalized to multiple classes. Its utilization of bootstrap aggregation helps to reduce variance, and it provides a method to compare selected features through Gini Importance while avoiding exorbitant computation cost.

6.2. Feature Importance

Through fitting a random forest model to both the MIT-BIH data and the Challenge data, we were able to gather information on feature importance for each dataset. Table 6.2 displays the results. These feature importances are calculated as “Gini Importance,” or Mean Decrease in Impurity. This is defined as the total decrease in impurity at each node weighted by the probability of reaching that node and averaged over every tree in the random forest (Breiman et al., 1984).

These feature importances were helpful in determining which features had the greatest impact on the performance of the model. This knowledge can be useful for dimension reduction. With the MIT-BIH data, we observe that the NL feature has a much higher importance than the other transitions. This led us to experiment with using only the NL feature in place of all 9 transition features, which drastically reduces complexity of the model while sacrificing minimal information and performance.

7. Results

In this paper, we present final performance results using both the MIT-BIH dataset and the Challenge dataset. Table 7.1 displays the results for each dataset using all features, while Table 7.2 displays results using only the NL transition, NEC Rate, ddRR, and bSQI. Binary Challenge Data

TABLE 6.2. Feature Importances

Feature	MIT-BIH Data	2017 Challenge Data
SS	0.008	0.064
NS	0.020	0.089
LS	0.030	0.029
SN	0.129	0.059
NN	0.070	0.095
LN	0.096	0.069
SL	0.014	0.055
NL	0.308	0.068
LL	0.007	0.042
NEC Rate	0.270	0.166
ddRR	0.047	0.259

refers to only considering AF versus Non-AF, while Multiclass Challenge Data includes all four original classes: N, O, A, and $\sim$.

TABLE 7.1. Performance Results Using All Features

	Accuracy	F1-Score
Binary MIT-BIH Data	0.964	0.963
Binary Challenge Data	0.949	0.813
Multiclass Challenge Data	0.752	0.704

We learned from studying the feature importances that the NL transition has greater impact on model performance than the other transitions. We tested the model using only this transition instead of all 9 transition features and saw only a small drop of 0.002 in accuracy and F1-Score for the MIT-BIH data. There was a larger drop in performance when using the Challenge data. For binary Challenge data, accuracy decreased by 0.015 while F1-Score decreased by 0.050. For multiclass Challenge data, accuracy decreased by 0.047 and F1-Score decreased by 0.062.

TABLE 7.2. Performance Results with 4 Features

	Accuracy	F1-Score
Binary MIT-BIH Data	0.962	0.961
Binary Challenge Data	0.934	0.763
Multiclass Challenge Data	0.705	0.642

This dimension reduction was particularly useful with the MIT-BIH dataset. Accuracy and F1-Score only dropped by 0.002, while computational complexity was significantly reduced. The run time for the model across the MIT-BIH data was reduced from 6 minutes and 11 second, to 4 minutes and 9 seconds.

Performance was considerably lower on the Challenge data for a variety of reasons. For binary classification, the Challenge Data was much more imbalanced. Furthermore, there were more mistakes in R-wave detection with the Challenge data because it was not validated and annotated by medical professionals. Also, in the MIT-BIH data, the ECG signals were almost entirely normal rhythms or AF rhythms, while a variety of other, unlabeled arrhythmia are present in the Challenge data along with noise. Any type of atrial arrhythmia, junctional arrhythmia, or ventricular arrhythmia that is not AF is grouped into one class of "Other" rhythms. Some of these other rhythms have similar R-wave patterns to the AF rhythms, while some have similar R-wave patterns to normal rhythms, making classification methods based on RR intervals alone less effective. The multiclass classification problem is a much more challenging problem to solve, as is evident from the much lower performance measures. Effectively solving this problem will require more than features based on RR interval alone. Frequency-based features and features extracted from information from P, Q, S, and T waves of ECG signals may contribute to a more accurate model (Behar et al., 2017; Datta et al., 2017).

8. Conclusion

The MIT-BIH data provided a well-controlled environment in which to tests the efficacy of features for distinguishing atrial fibrillation from normal heart rhythms. Using the novel features defined in this paper as well as bSQI for the Challenge data, a random forest model performed well at identifying both normal and AF rhythms. Three well-chosen features based only on RR intervals are sufficient to distinguish between normal and AF ECG signals as small as 30 seconds long with over 0.960 accuracy.

However, as the model transitioned to the Challenge data, performance dropped. With other types of cardiac rhythms present it became much more difficult to identify the AF rhythms. Other forms of arrhythmia share similar patterns of RR intervals with AF. While the features defined in this paper are effective at identifying AF from normal rhythms, it will likely require features based on more than RR intervals to effectively classify between a variety of rhythms types, such as those in the Challenge data.

9. Discussion and Future Work

While using the 2017 Challenge data it became clear that much more work in preprocessing would be required to produce excellent results. A brief foray into signal processing helped shed light on possible methods to classify noisy signals, reduce noise within a signal, and identify further features to be used in modeling.

As discussed previously, the WFDB package provides two QRS detector algorithms. As the XQRS detector did not seem to be robust to brief instances of noise, the GQRS detector was used to detect R peaks. However, Behar et al. and Datta et al. achieved much better results using variations on the Pan-Tompkins algorithm for R peak detection, which is what the XQRS detector is based on (Behar et al., 2017; Datta et al., 2017). We found an open-source implementation of the Pan-Tompkins algorithm, translated it to Python, and adapted it for our needs. This helped to gain a greater understanding of what the algorithm was doing to identify R peaks. Figure 9.1 displays different steps of the algorithm performed on a Normal heart rhythm of length 10 seconds. In the top left, the raw ECG signal is presented, followed by its derivative in the top right for comparison. The first step is to impose a band pass filter on the raw signal, as shown in the middle left subplot. A derivative filter is then performed by convolution on the resulting data, displayed in the middle right plot. The bottom left plot shows the resulting signal squared. The squared signal is then averaged with a moving window of length 30 samples, which is shown in the bottom right plot.

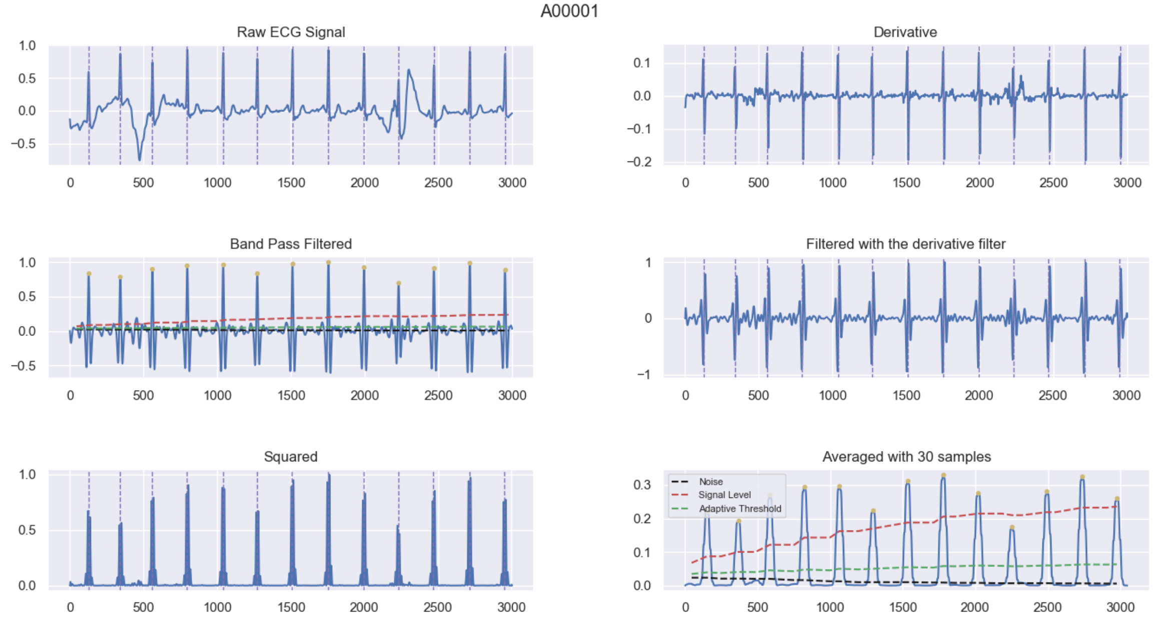

FIGURE 9.1. Example of Pan-Tompkins on Normal Rhythm

From here, a peak detector is used on the final signal, and a process based on iteratively updating thresholds is performed to determine if the peaks detected are, in fact, R peaks. Outlining this process visually and testing it with outlier signals helped in the development of ideas for detecting and removing noise in a signal during this R peak detection process.

The Pan-Tompkins algorithm performs well in the presence of baseline wander or low-amplitude noise within a signal, but is not robust to large spikes in the signal. With that in mind, the raw ECG signal can be analyzed before running the peak detection algorithm in an attempt to recognize and ignore large spikes in the data. An ECG recording rarely as QRS complex amplitude greater than 1 mV, with the maximum for a human heartbeat at around 3 mV. In this section, a method for discounting large spikes due to noise is proposed.

First, using a simple peak finder with a moving window of 100 samples, identify the number of peaks present in a segment and the absolute value of their amplitudes. If there are fewer than 10 peaks identified, this is insufficient and the segment is discarded. If the mean of the peaks is greater than 2, perform the Pan-Tompkins algorithm without removing any noise. If the mean is lower than 2, then we will treat any peak higher than 2 as an anomaly and consider the region around it to be noise and flatten it before performing the Pan-Tompkins algorithm. Then peaks can be properly identified without disturbance from large spikes due to noise. The median RR interval length is calculated from this peak data and the areas that were flattened are imputed with RR intervals of the median RR interval length.

These peak locations can then be used to segment the signal into individual heartbeats. By comparing each heartbeat to every other heartbeat in the signal, each heartbeat will receive a value indicating how noisy it is. The heartbeats that are most different from the rest of the heartbeats in the signal are considered the most noisy.

We look forward to implementing this idea to test its effectiveness in noise detection and accurate peak identification.

10. Acknowledgements

Thank you to Dr. Yishi Wang and Dr. Cuixian Chen at UNCW for their mentorship throughout this project. Additional thanks to Christian Austin, Georgia Smith, and Jericho Lawson for their discussions and support, and to the National Science Foundation for funding this program (Grant #1659288).

References

Behar, J. A., Rosenberg, A. A., Yaniv, Y., and Oster, J. (2017). Rhythm and quality classification from short ecgs recorded using a mobile device. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Clifford, G. D., Liu, C., Moody, B., Li-wei, H. L., Silva, I., Li, Q., Johnson, A., and Mark, R. G. (2017). Af classification from a short single lead ecg recording: the physionet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE.

Datta, S., Puri, C., Mukherjee, A., Banerjee, R., Choudhury, A. D., Singh, R., Ukil, A., Bandyopadhyay, S., Pal, A., and Khandelwal, S. (2017). Identifying normal, af and other abnormal ecg rhythms using a cascaded binary classifier. In *2017 Computing in Cardiology (CinC)*, pages 1–4. IEEE.

Duverney, D., GASPOZ, J.-M., Pichot, V., Roche, F., Brion, R., Antoniadis, A., and BARTHÉLÉMY, J.-C. (2002). High accuracy of automatic detection of atrial fibrillation using wavelet transform of heart rate intervals. *Pacing and clinical electrophysiology*, 25(4):457–462.

Ghodrati, A., Murray, B., and Marinello, S. (2008). Rr interval analysis for detection of atrial fibrillation in ecg monitors. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 601–604. IEEE.

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., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220.

Lian, J., Wang, L., and Muessig, D. (2011). A simple method to detect atrial fibrillation using rr intervals. *The American journal of cardiology*, 107(10):1494–1497.

Liu, F., Liu, C., Zhao, L., Jiang, X., Zhang, Z., Li, J., Wei, S., and Zhang, Y. (2018). Dynamic ecg signal quality evaluation based on the generalized bsqi index. *IEEE Access*, 6:41892–41902.

Moody, G. and Mark, R. (1983). A new method for detecting atrial fibrillation using rr intervals. *Computers in Cardiology*, pages 227–230.

Pan, J. and Tompkins, W. J. (1985). A real-time qrs detection algorithm. *IEEE Trans. Biomed. Eng*, 32(3):230–236.

Python Software Foundation (1991). Python language reference, version 3.6.8. Available at `https://www.python.org`.

Shields, A. and Lip, G. Y. (2015). Choosing the right drug to fit the patient when selecting oral anticoagulation for stroke prevention in atrial fibrillation. *Journal of internal medicine*, 278(1):1–18.

Shouldice, R. B., Heneghan, C., and de Chazal, P. (2007). Automated detection of paroxysmal atrial fibrillation from inter-heartbeat intervals. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 686–689. IEEE.

Tateno, K. and Glass, L. (2000). A method for detection of atrial fibrillation using rr intervals. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 391–394. IEEE.

Xie, C., Dubiel, J., et al. (2016). Python waveform-database (wfdb) package. Available at `https://github.com/MIT-LCP/wfdb-python`.

11. Appendix

TABLE 11.1. Average Proportions of Transitions per segment in MIT-BIH Dataset

	SS	SN	SL	NS	NN	NL	LS	LN	LL
AF	0.046	0.104	0.048	0.106	0.425	0.099	0.045	0.101	0.025
Non-AF	0.004	0.009	0.021	0.025	0.909	0.004	0.005	0.020	0.003

TABLE 11.2. Average Proportions of Transitions per segment in Challenge Dataset

	SS	SN	SL	NS	NN	NL	LS	LN	LL
AF	0.045	0.088	0.044	0.091	0.471	0.099	0.046	0.092	0.024
Non-AF	0.022	0.016	0.026	0.030	0.829	0.018	0.013	0.028	0.018

(D. Johnston) DEPARTMENT OF MATHEMATICS AND STATISTICS, THE UNIVERSITY OF NORTH CAROLINA WILMINGTON, WILMINGTON, NC 28403, USA

E-mail address, Corresponding author: `drew.johnston13@gmail.com`