Application of Deep Machine Learning for Classification of Heart Diseases

Shalin Savalia SSAVALIA@MAIL.STMARYTX.EDU
Vahid Emamian VEMAMIAN@STMARYTX.EDU
Electrical and Computer Engineering,
St. Mary's University, Texas

Abstract

Healthcare professionals commonly use Electrocardiogram (ECG) as a low-cost diagnostic tool for monitoring heart electrical signals. Arrhythmia, or an abnormal cardiac signal, can be life-threatening and even fatal. Arrhythmia can be categorized into various types, such as, tachycardia, bradycardia, supraventricular arrhythmias, and ventricular. The automated monitoring of arrhythmia and classification with ECG is beneficial for doctors. In this research, we use deep machine learning for automated arrhythmia classification focusing on the recent trends in arrhythmia classification. Using St. Mary's University Deep Learning Platform, we conducted heavy and complex simulations to measure the performance of the various arrhythmia classification and detection models. Finally, we present the proposed deep learning algorithms' accuracy, which surpasses the existing algorithms' performance in precision and sensitivity.

Keywords: Electrocardiogram (ECG), Arrhythmia Classification, Deep Machine Learning

1 Introduction

ECG is often used to evaluate the heart's electric activity by simply placing the number of electrodes on various skin parts and has been broadly used to identify heart diseases due to its simplicity and non-invasive nature. By examining the electrical activity of each heartbeat, i.e., the mixture of action instinct waveforms produced by different cardiac tissues found in the heart, it is possible to identify some of its heart abnormalities. Features like P waves, T waves, QRS complex can be extracted from ECG, and studying and classifying them is crucial in diagnosing various heart diseases (Artis, Mark & Moody, 1991). An ECG signal with its features is shown in Figure 2. Detecting and classifying various arrhythmias is possible by studying such features, including abnormal heart rate or abnormal features of the signal. Irregular heartbeats, known as arrhythmias, have their unique pattern; therefore, it is possible to classify and detect the type (Melo, Caloba & Nadal, 2000). Two main categories are considered for arrhythmias. The first category consists of arrhythmias shaped by an abnormal heartbeat, a.k.a as morphological arrhythmia. The other type is arrhythmias formed by a set of irregular heartbeats, known as rhythmic arrhythmias. Abnormal heartbeats form alterations in the wave frequency or morphology, and the ECG tests may detect all these alterations. This research focuses on recognizing heart diseases by using ECG feature extraction and deep machine learning. This is feasible by classifying regular and irregular ECGs using deep neural network techniques and extracting the ECG signals' features. We use the state-logic machine algorithm to identify heart diseases, such as bradycardia, tachycardia, and first and second-degree Atrioventricular (AV) block (Moody & Mark, 2001). Other arrhythmia types are described in this research, mainly ventricular tachycardia, atrial fibrillation, malignant ventricular, atrial flutter, and ventricular bigeminy. We can detect these with deep machine learning algorithms (Salam & Srilakshmi, 2001).

Normally, researchers strive to automatically distinguish between regular and irregular heartbeats. Signal processing, feature extraction, and classification are the three stages that these researchers generally follow. The heartbeats are extracted from the continuous ECG waveform into individual heartbeats by signal processing raw ECG signals (Perez, Marques & Mohammadi, 2014). Following that, feature extractions transform the variable-length time-domain heartbeat waveforms into fixed-length feature vectors that encode the heartbeat's features (Palreddy, Tompkins & Hu, 1995). Several features have been extracted from ECG signals to identify the heartbeats, such as morphological features, Hermite coefficients, wavelet transform features, heartbeat interval features, and sparse decomposition. On the other hand, for classification, various deep machine learning algorithms have been used here, including artificial neural networks (ANNs), deep neural networks (DNNs), support vector machines (SVMs), convolutional neural networks (CNNs), and multi-layer perceptron (MCP). When various datasets are available, machine learning techniques are good to consider and often exceed human agreement rates (Zebardast, Ghaffari & Masdari, 2013). As shown in Figure 1, In traditional Machine learning methods, most of the features need to be extracted by a domain expert to reduce the ease of the data and make patterns more noticeable to learning algorithms to be successful. The significant improvement of Deep Learning algorithms is that they aim to learn high-level features from data incrementally. This reduces the need for domain expertise and hard-core feature extraction. Hence DNN has far better performance when we compare it with traditional methods.

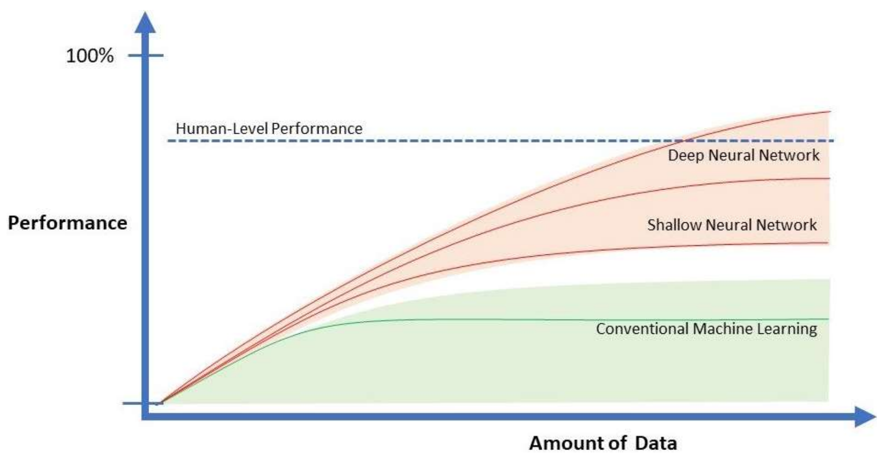
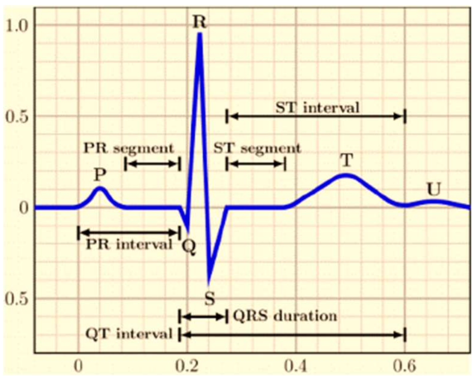


Figure 1. Performance graph versus the Amount of data

CNN was employed for automated coronary artery disease diagnosis, and it was found that CNN remains resilient despite shifting and scaling invariance, which makes it a better choice (Acharya et al., 2017). In this research, we propose robust methods for heart disease diagnosis using CNN and multi-layer perceptron (MLP). We also use CNN to differentiate between normal and abnormal heart sound recordings with an accuracy of 82% (Acharya et al., 2017). The deep machine learning method for single-image super-resolution (SR) was also tested using a CNN algorithm with better performance than the state-of-the-art method (Nilanon et al., 2016). In the 2017 PhysioBank competition, Fernando et al. (2017) proposed a method with an accuracy of 83% on PhysioBank data, which applies CNN to identify four different arrhythmias from short segments of ECG recordings. Ghiasi et al. [15] proposed algorithms to detect atrial fibrillation using a feature-based algorithm and CNN with 80% accuracy on training datasets in the same competition.



P wave	Depolarization of the atria
Q wave	Activation of the anterioseptal region of the ventricular myocardium
R wave	Depolarization of the ventricular myocardium
S wave	Activation of the posteriobasal portion of the ventricles
T wave	Rapid ventricular repolarization

(a) **(b)**

Figure 2. Ideal electrocardiogram (ECG) signal with key features indicated; (a) P wave, QRS complex, and T wave which play important roles in diagnosis abnormality of heart signal; (b) Features of an ECG signal; how and which part of the heart is used to generate each feature (Fernando et al. 2017).

The problem-solving approach is the critical distinction between the simple and deep neural network (DNN) method. DNN has been shown to solve the problem end to end, whereas simple neural network methods need problem statements to break down into multiple parts to be solved in the beginning, and then the results will be merge at the final stage. Commonly, when there are more than three layers of neurons, including input and output, the method is referred to as "deep learning" (Shadi et al., 2017). Figure 3 showed differences between simple NN and deep NN. Usually, a deep neural network algorithm takes longer to train than a simple neural network due to the large number of parameters it contains. However, utilizing state of the art HPC deep learning platform at St. Mary's University, we were able to execute deep learning algorithms very quickly. The main advantage of DNN is that it can detect more complex features than a simple neural network because of the number of hidden layers. This function of DNN makes it able to handle a large amount of data that contains many features. Deep learning neural networks often end in an output layer: a logistic, Softmax, or classifier that assigns a chance to a particular outcome (Shadi et al., 2017).

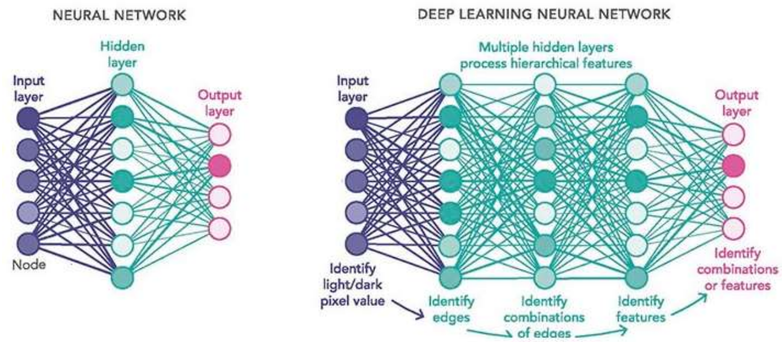


Figure 3. Comparison between simple neural network (left) and deep NN (right) (Shadi et al., 2017).

We used two PhysioBank datasets for our proposed algorithms, i.e., normal sinus rhythm database (NSR-DB) and MIT/BIH arrhythmia database, to classify regular and irregular ECG signals using the multilayer-perceptron technique. Another technique uses a four-layer of CNN to detect several arrhythmias in arbitrary length ECG dataset features. The dataset employed in this study contains various heart diseases, such as normal sinus arrhythmia, second degree AV block, first degree AV block, atrial flutter, malignant ventricular, atrial fibrillation, ventricular tachycardia, and ventricular bigeminy. The data were obtained from kaggle.com. The NN algorithms were trained using the Google TensorFlow library, a free and open-source software library for machine and deep learning. The algorithm can be used across a range of tasks but focuses on training for deep neural networks. Once both algorithms had been trained on the downloaded dataset, they were trained using another dataset with distinct characteristics from the training dataset.

2 Approach

2.1 Problem Formulation

The proposed algorithm for detecting and classifying ECG arrhythmias is a sequence-to-sequence task that takes an input, i.e., ECG signal, S = [s1, …, sk]. It gives labels as an output in the form of r = [r1, …, rn], where r_i can take any of the labels. Here, for a multi-layer perceptron algorithm, we used $m = 2$, and for the CNN algorithm, we used $m = 9$. The individual output label refers to a segment of the input, and output labels cover the whole sequence (Chow, Marine & Fleg, 2012).

For an example in the training set, we enhance the cross-entropy function below

$$\mathrm{L}(S, \mathrm{r}) = \frac{1}{n}\sum_{i=1}^{n} \log p\ (R = r_i\ \mid S) \tag{1}$$

where p is the probability the network assigns to the i_{th} output, taking on the value r_i.

2.2 Convolutional Neural Network (CNN)

CNN here had a significant advantage over other neural networks methods by their superior performance with input signals and images. They are feed-forward ANN inspired by biological processes and intended to identify patterns directly from data or images by integrating feature extraction and classification (Fleg, 2012). A CNN involves three significant layers: convolutional layer, pooling layer, and fully connected layer. The convolutional layer is the most important block of a CNN, where most of the computation occurs. This layer normally requires the following components: i) input data, ii) a filter, and iii) a feature map (Fleg, 2012). A non-linear activation layer follows convolutional layers. This helps to capture more complex parameters of the input signal possible. Pooling layers are employed to subsample the last layer by mixing small rectangular subsets of values. This layer, also known as the down sampling layer, performs dimension reduction, reducing the number of parameters in the input 18]. Like the convolutional layer, the pooling layer sweeps a filter across the whole input; however, the difference is that this filter does not have weights. There are two types of pooling: max pooling and average pooling. Max and average pooling are applied by replacing the input values with the maximum or the average values, respectively (Acharya et al., 2016). On the other end, a significant amount of data is lost in the pooling layer. It also has several benefits to the CNN as this layer helps lower complications, increase efficiency and limit the risk of overfitting. A fully connected layer (FCL) performs classification based on the features extracted from the past layers and their different filters (Acharya et al., 2016). While convolutional and pooling layers employed the ReLU functions, FCL typically leverages a Softmax activation function that classifies inputs appropriately, generating a probability from 0 to 1.

2.3 Multi-layer Perceptron (MLP)

Deep neural networks are made of multiple layers and multi-layer perceptron (MLP), which indicates that it is made of more than one perceptron. A single layer perceptron can solve linearly separable problems, but when one or more layers are added in a single layer perceptron, it is known as MLP (Acharya et al., 2016). MLP network is recognized as a feed-forward neural network that consist of one or several hidden layers. It is usually used to classify input patterns, pattern recognition, prediction based on the input data, and approximation (Acharya et al., 2016). MLP is typically made up of an input layer that receives the data, an output layer that decides or predicts the input data, and between those two, any number of hidden layers that are the computational portion of the MLP (Acharya et al., 2016; Desai et al., 2016).

MLPs are often used to supervise learning. They train on a set of input-output data and learn to model the correlation or dependencies between input and output data sets. Training requires modifying the parameters, i.e., the weights and biases of the model, to minimize the error (Desai et al., 2016). Backpropagation is also applied to make those weights and bias adjustments relative to the error. Here the error can be calculated in various ways, for an instant, by using root mean squared error.

2.4 Model Architecture

CNN and MLP with several hidden layers are used for sequence-to-sequence learning algorithms. CNN is one of the central branches of deep, feed-forward machine learning neural networks that can handle extensive data. Like any other neural network, CNN has input, output, and several hidden layers. The hidden layers include mainly convolutional, pooling layers, normalization layers, fully connected layers, and finally, Softmax layers. Our proposed CNN algorithm has a convolutional layer with a Softmax function that provides the output for the trained network. It uses the ReLU, a.k.a rectifier linear unit and activation tool in all convolution layers. The max-pooling layer functions independently for each column and row of the input and resizes them spatially (Zubair, Kim & Yoon, 2016). We used the max-pooling layer with a stride size of 2 by 2 in the function because it gave improved accuracy than the original 3 by 3 pooling layer. The use of a 3 by 3 stride layer yields higher info loss. The pooling layer in the CNN lowers the overfitting problem by reducing the input size by half of the actual input. Flowchart diagrams of both algorithms are depicted in Figure 4. Both models take features of an ECG signal as the network's input and predict the output as labels of the data. Initially, ECG datasets are pre-processed, and for that purpose, the first network reads the datasets and identifies their features and labels. For the MLP, the labels will be arrhythmia and regular sinus. In contrast, in the CNN, the labels are normal sinus arrhythmia, first degree AV block, second degree AV block, atrial fibrillation, atrial flutter, malignant ventricular, ventricular bigeminy, and ventricular tachycardia (Oquab et al., 2015). Figure 5 illustrates our proposed architecture of the CNN in the algorithm where the first and last convolutional layers are not the same as the middle three convolutional layers.

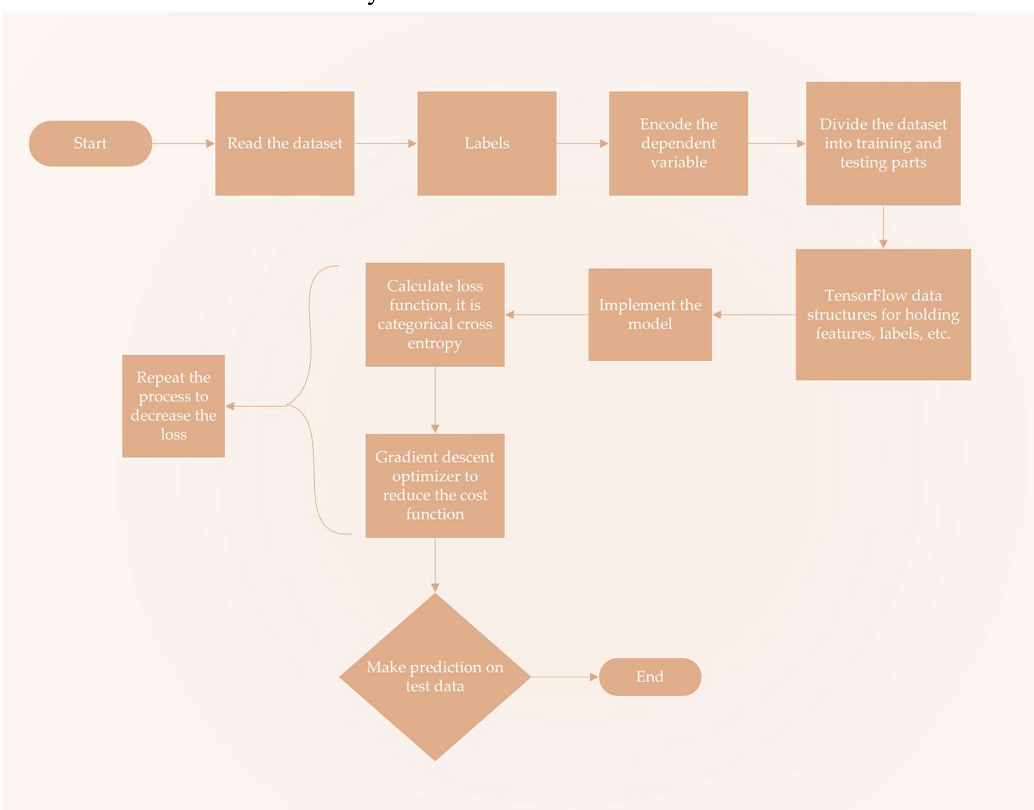


Figure 4. System process flowchart of Multilayer Perceptron (MLP) and Convolution NN. To define features and labels in the dataset, two TensorFlow variables were determined. One hot encoder was used to encode the dataset.

The next step is to encode the dependent variable—the dataset labels—for the deep network. As the data is classified as containing different arrhythmia names, it is mandatory to encode the information because the labels are not numerical, and it is not possible to read them directly by the algorithm (Krizhevsky, Sutskever & Hinton, 2012). There are two statistical ways of encoding data. The first is integer encoding, and the second is one-hot encoding. Integer encoding assigns an integer value typically to each unique category. For example, "red," "green," and "blue" are 1, 2, and 3, respectively (Krizhevsky, Sutskever & Hinton, 2012). For categorical variables where no ordinal correlation is recognized, integer encoding is not enough. In one-hot encoding, the integer encoded variable is discarded, and new binary variables are added for each unique integer value. There are three classes in the "color" variable, and consequently, three binary variables are required. A "1" value is inserted in the binary variable for the color, and "0" values are placed for the other colors. In our proposed machine learning algorithm, we used one-hot encoding to distinct integer encoding. This was followed by dividing the data into training, testing, and validation sets (Krizhevsky, Sutskever & Hinton, 2012).

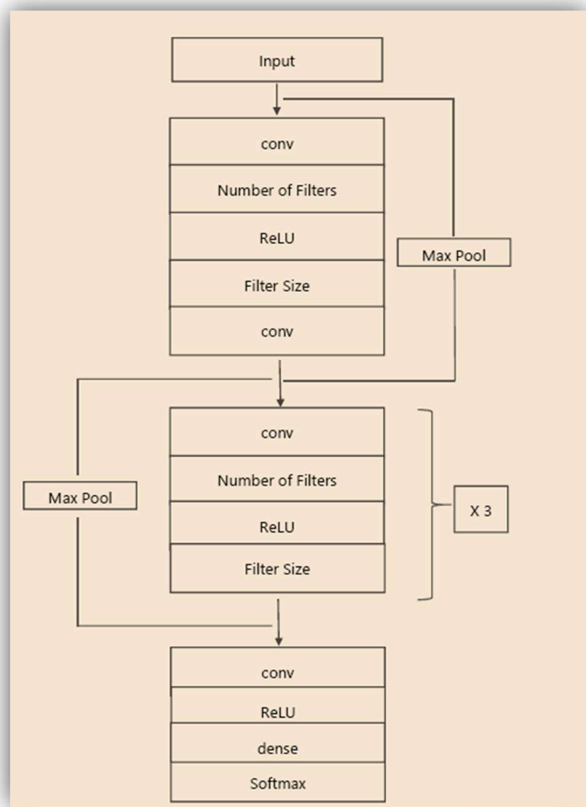


Figure 5. Proposed Algorithm for CNN

TensorFlow data structures were defined for holding features and labels, including determining weights, hidden layers, biases, activation tools, placeholders for inputs, filters, filter size, and desired output. Another Tensor is defined to store a trained output model. This was then followed by training of the proposed model with the training dataset. After training, the network will compute how different the trained model's output is from the actual output. After reaching the minimum value, the trained model will yield testing accuracy by training it using the test data (Kiranyaz, Ince & Gabbouj, 2015). The cross-entropy function is also used to reduce the error to a minimum level.

2.5 ECG Data

We download ECG datasets from PhysioBank.com and kaggle.com for training and testing the proposed CNN and MLP algorithms. The MLP dataset size was 208x61, where total ECG signals were 208, and the total number of features and labels were 61. The first 60 columns contained features, and the last column contained the disease label for each data. However, the CNN dataset had dimensions of 26x543x60, following the same configuration as the MLP dataset, but this dataset consisted of 9 labels. Both algorithms consisted of 80% of the total data for training and 20% for testing. The training dataset was separated into 70% for actual training and 30% for validation. Each ECG vector in the dataset was 10 seconds long and contained only one rhythm class. A demonstration of the distribution for the ECG signals used for the training, testing, and validation process is shown in Figure 6 below.

2.6 Training of Data

We used batch sizes of 50 for the training stage with the standard backpropagation for stochastic learning. To update the weights, we used the following formula (Sufi & Khalil, 2010):

$$w_l = \left(1 - \frac{n_\lambda}{ts}\right) w_{l-1} - \frac{n}{x}\frac{\partial c}{\partial w} \tag{2}$$

where
w = weights
l = layer number
n = learning rate
c = cost function
ts = total number of training samples
x = batch size
λ = regulation parameter

Additionally, the biases in the formula were updated through,

$$b_l = b_{l-1} - \frac{n}{x}\frac{\partial c}{\partial w} \tag{3}$$

The deep neural learning rate was set to 0.002 for MLP and to 0.003 for CNN.

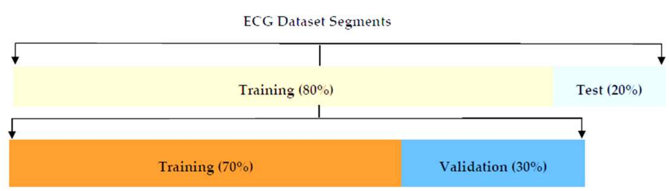


Figure 6. ECG segments distribution in training and testing data set.

2.7 Testing of Data

After running each training epoch, the algorithms are expected to test the CNN and MLP algorithms to give test accuracy. The CNN and MLP algorithms have 500 and 1000 epochs, respectively. We applied 30% of the total training data, 80% of the original dataset, to the validation part after completing every epoch for better accuracy. Testing and validation dataset consisted of 20% of the total data used for testing, as described in previous sections (Ciresan et al., 2011).

3 Results

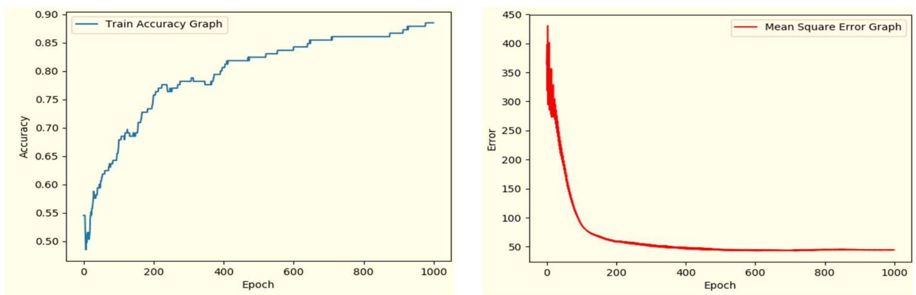
We concluded that CNN networks have a remarkable ability to extract all differentiable features invariant to local temporal and spectral variations. This has resulted in significant breakthroughs in higher accuracy results. The proposed CNN algorithm contained the following stages: i) data pre-processing; processing of ECG signals (so that automatic algorithm can understand different diseases), ii) stacking of max-pooling layers and convolution layers to extract the known features, iii) activation of the Softmax function and layering of a fully connected layer to predict the disease (Fernando et al., 2017). Table 1 shows the CNN layers' parameters, the filter's size, and output layer neuron size. To distinguish between regular sinus rhythm and irregular rhythm, we used MLP. To accomplish this, we used four hidden layers, each consisting of 60 neurons. The ReLU was then used to activate the first and last hidden layers, while two hidden middle layers employed a sigmoidal activation function. At the output layer, these were followed by the linear activation function. Additionally, a gradient descent optimizer was employed to lower the actual and trained network output error. When the parameters cannot be calculated analytically or by linear algebra, we realized that it was advantageous to implement a gradient descent optimizer. Figure 7 depicts the accuracy and mean square error graphs for the MLP algorithm.

After training the network with 1000 epochs yielded an accuracy of about 89% for the dataset from PhysioBank.net. Figure 8 depicts the visual confusion matrix for the training dataset. The confusion graph demonstrates true label vs. predicted label, where 0 stands for periodic ECG signal and 1 represents regular sinus rhythm. The dataset used here consisted of 208 ECG data recordings, 97 of which are irregular (arrhythmia), and 111 represent a regular sinus rhythm. The 80% of the data applied for training, consisting of 165 ECG signals, 72 describe arrhythmia, and 93 represent regular sinus rhythm. From this training dataset, 63 arrhythmia and 81 regular sinus signals were rightfully classified by the algorithm, demonstrating a significant improvement in the accuracy of the MLP algorithm.

Layers	**Type**	**Size of Neurons (Output Layer)**	**Filter Size of Each Layer**
0–1	Convolution	(None, 1, 60, 1)	32
1–2	Max Pooling	(None, 1, 30, 1)	2
2–3	Convolution	(None, 1, 30, 1)	32

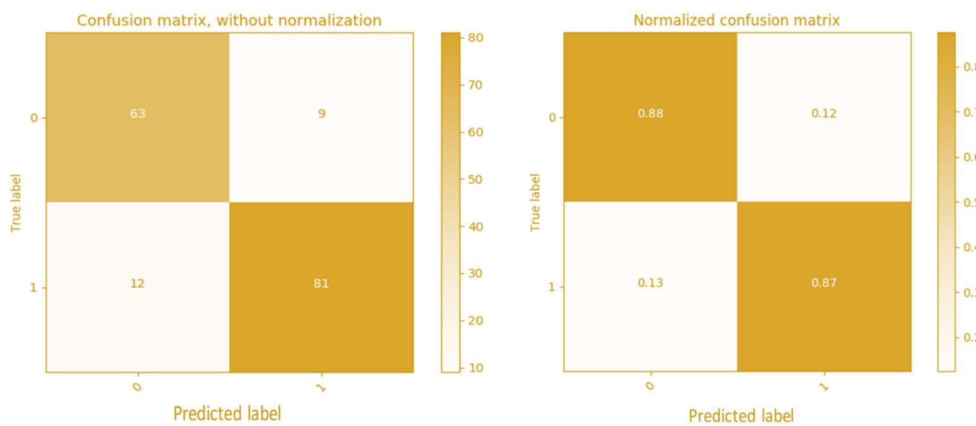
3–4	Max Pooling	(None, 1, 15, 1)	2
4–5	Convolution	(None, 1, 15, 1)	32
5–6	Max Pooling	(None, 1, 8, 1)	2
6–7	Convolution	(None, 1, 8, 1)	32
5–6	Fully connected layer	2048	-

Table 1. Parameters for the proposed CNN algorithm



(a) **(b)**

Figure 7. Accuracy and mean square error for the MLP algorithm; (a) accuracy increases as the number of epochs increases; (b) error reduces with every epoch reaches the lowest after 1000 epochs.



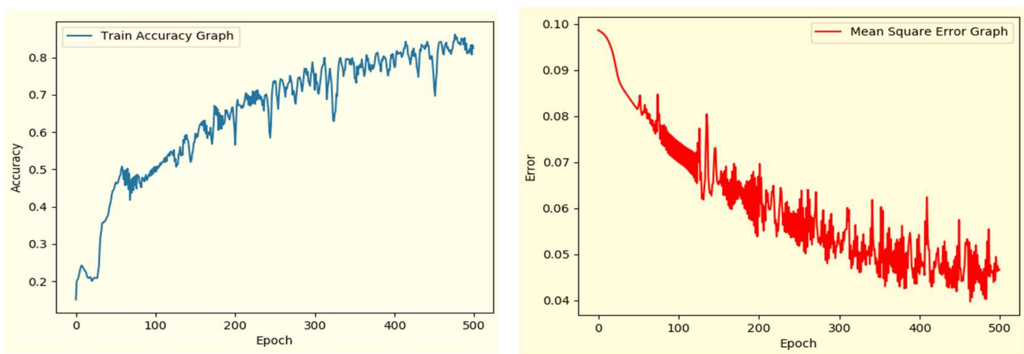
(a) **(b)**

Figure 8. Confusion matrix (CM)

We then used the deep machine learning technique to create CNN to identify different cardiovascular diseases. We used the ReLU non-linear activation tool to activate the CNN alongside the gradient descent optimizer to minimize the error. This method proved to work great when the parameters could not be obtained analytically (Sainath, Mohamed & Kingsbury, 2015). The architecture of the CNN for each convolution layer had 32 filters, and each filter had a size of 5 by 5. Figure 9 shows the accuracy and MSE error. On average, the accuracy improves with every epoch and, after about 500 epochs, reaches the max of 83.5%. The error reduces continuously with each epoch and gradually reaches a minimum.

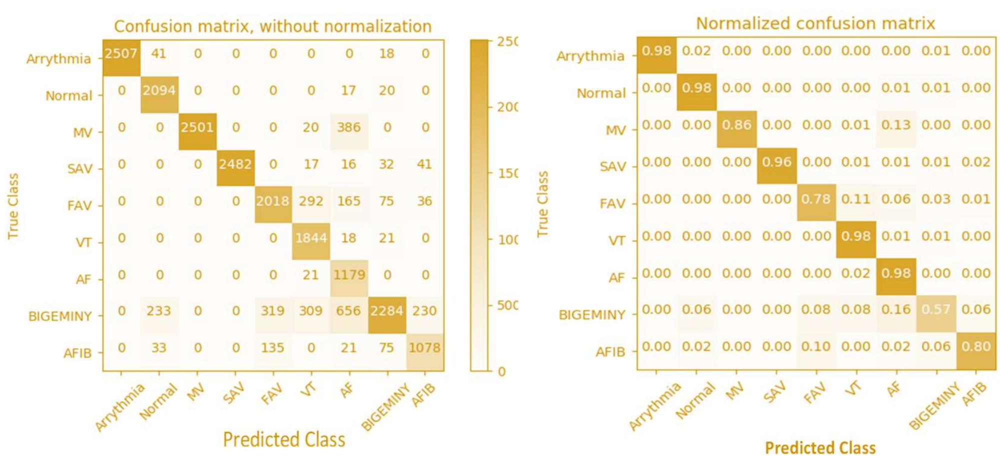
We defined two variables as features and labels for the datasets. The proposed algorithm reshaped dimensions of features 1 by 4 because the convolution layer here only accepts 4-

dimension vectors [28-31]. Upon completing the simulation using the deep learning platform HPC, the first three convolution layers were defined. The output of the first layer was fed into the max pooling layer to reduce the dimension of the vector to make the network faster and avoid overfitting. The same approach was followed for the second and third convolution layers as well. The result of the third pooling layer was then fed into the fully connected layer, followed by the Softmax layer so that the algorithm predicts the diseases [32]. Same as before, we showed the classification results of the algorithm by using a confusion matrix. Each cell consisted of the raw exemplars classified for the matching combination of expected and actual outputs. Figure 10 below shows a visual representation of the confusion matrix for the proposed CNN algorithm. We noticed many arrhythmias were confused with first-degree AV Block and ventricular bigeminy, but overall, the network gives a good prediction accuracy for the other diseases. We anticipate part of this is due to the ambiguous location of the exact onset and offset of the arrhythmia in the ECG vectors [33-35].



(a) **(b)**

Figure 9. Accuracy and error of the proposed CNN



(a) **(b)**

Figure 10. CM of the CNN algorithm

4 Conclusions

In this article, we used deep machine learning for automated arrhythmia classification, focusing on the recent trends in arrhythmia classification. Using St. Mary's University Deep Learning Platform, we conducted heavy and complex simulations to measure the performance of the various arrhythmia classification and detection models. The proposed algorithms in this research were tested on ECG signals obtained from Physio.net and keggar.com and succeeded in detecting abnormal states in each signal with significant accuracy using MLP and CNN models. Our results showed that the proposed algorithms can accurately diagnose various heart diseases with 89% and 83% accuracy for the proposed MLP and CNN algorithms, respectively.

Acknowledgements

This research is supported by the DURIP Grant W911NF-19-1-0159 (73980-NS-RIP)

References

A. K. Salam and G. Srilakshmi. An algorithm for ECG analysis of arrhythmia detection. In *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*: 1-6. IEEE, 2015.

A. D. Jigar. Swift single image super resolution using deep convolution neural network. In *2016 International Conference on Communication and Electronics Systems (ICCES)*, 1-6. IEEE, 2016.

A. Fernando, C. Oliver, A. F. P. Marco, M. Adam and V. de Maarten. Comparing feature based classifiers and convolutional neural networks to detect arrhythmia from short segments of ECG. In *2017 Computing in Cardiology (CinC)*, 1-4. IEEE, 2017.

A. Krizhevsky, I. Sutskever and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25: 1097-1105, 2012.

Alaa M. Elsayad. Classification of ECG arrhythmia using learning vector quantization neural networks. In *2009 International Conference on Computer Engineering & Systems*, 139-144. IEEE, 2009.

B. Zebardast, A. Ghaffari and M. Masdari. A new generalized regression artificial neural networks approach for diagnosing heart disease. *International Journal of Innovation and Applied Studies*, 4(4), 679, 2013.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121-167, 1998.

D. C. Ciresan, U. Meier, L. M. Gambardella and J. Schumacher. Convolutional neural network committees for handwritten character classification. In *2011 International Conference on Document Analysis and Recognition*, 1135-1139. IEEE, 2011.

F. Sufi and I. Khalil. Diagnosis of cardiovascular abnormalities from compressed ECG: a data mining-based approach. *IEEE Transactions on Information Technology in Biomedicine*, 15(1), 33-39, 2010.

G. B. Moody and R. G. Mark. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, *20*(3), 45-50, 2001.

G. D. Clifford, C. Y. Liu, B. Moody, B. L. Lehman, I. Silva, Q. Li, A. E. W. Johnson and R. G. Mark. AF classification from a short single lead ECG recording: The PhysioNet computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, 1-4. IEEE, 2017.

G. Shadi, A. Mostafa, M. Nasimalsadat, K. Kamran and G. Ali. Atrial fibrillation detection using feature-based algorithm and deep conventional neural network. In *2017 Computing in Cardiology (CinC)*, 1-4. IEEE, 2017.

G. V. Chow, J. E. Marine and J. L. Fleg. Epidemiology of arrhythmias and conduction disorders in older adults. *Clinics in Geriatric Medicine* 28(4): 539–553, 2012.

J. S. Sonawane and D. R. Patil. Prediction of heart disease using multi-layer perceptron neural network. *International Conference on Information Communication and Embedded Systems (ICICES)*, 1-6, 2014.

K. He, X. Zhang, S. Ren and J. Sun. Identity mappings in deep residual networks. *IEEE Transactions on Medical Imaging*, 35(5): 1332-1343, 2016.

K. Najarian and R. Splinter. *Biomedical Signal and Image Processing*, 2nd ed. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, 2012.

M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe and S. Mougiakakou. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. In *IEEE Transactions on Medical Imaging*, 35(5): 1207-1216, 2016.

M. K. Gautam and V. K. Giri. A neural network approach and wavelet analysis for ECG classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, 1136-1141. IEEE, 2016.

M. Oquab, L. Bottou, I. Laptev and J. Sivic. Is object localization for free? —Weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 685-694, 2015.

M. Zubair, J. Kim and C. W. Yoon. An automated ECG beat classification system using convolutional neural networks. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*, 1-5. IEEE, 2016.

R. J. Martis, U. R. Acharya, H. Adeli, H. Prasad, J. H., Tan, K. C. Chua, C. L. Too, S. Yeo and L. Tong. Computer aided diagnosis of atrial arrhythmia using dimensionality reduction methods on transform domain representation. *Biomedical Signal Processing and Control*, 13, 295-305, 2014.

R. J. Martis, U. R. Acharya, H. Prasad, K. C. Chua, C. M. Lim and J. S. Suri. Application of higher-order statistics for atrial arrhythmia classification. *Biomedical Signal Processing and Control*, 8: 888–900, 2013.

R. R. Perez, A. Marques and F. Mohammadi. The application of supervised learning through feed-forward neural networks for ECG signal classification. In *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1-4, 2016.

S. G. Artis, R. G. Mark and G. B. Moody. *Detection of Atrial Fibrillation using Artificial*

Neural Networks. MS thesis. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1991.

S. L. Melo, L.P. Caloba and J. Nadal. Arrhythmia analysis using artificial neural network and decimated electrocardiographic data. In *Computers in Cardiology 2000.* 27: 73-76. IEEE, 2000.

S. Kiranyaz, T. Ince and M. Gabbouj. Real-time patient-specific ECG classification by 1-D convolutional neural network. *IEEE Transactions on Biomedical Engineering*, 63(3), 664-675, 2015.

S. M. Debbal. Model of differentiation between normal and abnormal heart sounds in using the discrete wavelet transform. *Journal of Medical and Bioengineering*, 3(1): 5–11, 2014.

S. Palreddy, W. J. Tompkins and Y. H. Hu. Customization of ECG beat classifiers developed using SOM and LVQ. In *Proceedings of 17th International Conference of the Engineering in Medicine and Biology Society*, 1: 813-814. IEEE, 1995.

S. S. Xu, M. W. Mark and C. Cheung. Towards end-to-end ECG classification with raw signal extraction and deep neural networks. In *IEEE Journal of Biomedical and Health Informatics*, 23(4):1574-1584 2019.

S. Savalia and V. Emamian. Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks. *Bioengineering*, 5(2): 35, 2018

S. Savalia, E. Acosta and V. Emamian, Classification of cardiovascular disease using feature extraction and artificial neural networks, *Journal of Biosciences and Medicines*, 5: 64, 2017.

T. N. Sainath, A. R. Mohamed and B. Kingsbury. Ramabhadran, B. Deep convolutional neural networks for LVCSR. *Neural Networks*, 64: 39-48, 2015.

T. Nilanon, J. Yao, J. Hao, S. Purushotham and Y. Liu. Normal/abnormal heart sound recordings classification using convolutional neural network. In *2016 Computing in Cardiology Conference (CinC)*, 585-588. IEEE, 2016.

U. Desai, R. J. Martis, U. R. Acharya, C. G. Nayak, G. Seshikala and R. Shettyk. Diagnosis of multiclass tachycardia beats using recurrence quantification analysis and ensemble classifiers. *Journal of Mechanics in Medicine and Biology*, 16(01), 1640005, 2016.

U. R. Acharya, H. Fujita, M. Adam, S. L. Oh, J. H. Tan, V. K. Sudarshan and J. E. W. Koh. Automated characterization of arrhythmias using non-linear features from tachycardia ECG beats. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 000533-000538. IEEE, 2016.

U. R. Acharya, H. Fujita, O. S. Lih, M., Adam, J. H. Tan and C. K. Chua. Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network. *Knowledge-Based Systems*, 132: 62-71, 2017.

V. Golkov, A. Dosovitskiy, J. I. Sperl, M. I. Menzel, M. Czisch, P. Sämann, T. Brox and D. Cremers. q-Space deep learning: Twelve-fold shorter and model-free diffusion MRI scans. I*EEE Transactions on Medical Imaging* 35(5): 1344-1351, 2016.

Y. Pan, M. Fu, B. Cheng, X. Tao and J. Guo. Enhanced deep learning assisted convolutional neural network for heart disease prediction on the internet of medical things platform. In *IEEE Access,* 8: 189503-189512, 2020.

V. Vapnik. *The Nature of Statistical Learning Theory*; Springer Verlag: New York, NY, USA, 2013.

Y. Wang, Y. S. Zhu, N. V. Thakor and Y. H. Xu. A short-time multifractal approach for arrhythmia detection based on a fuzzy neural network. *IEEE Transactions on Biomedical Engineering*, 48(9), 989-995, 2001.

Z. Yan, Y. Zhan, Z. Peng, S. Liao, Y. Shinagawa, S. Zhang, D. N. Metaxas and X. S. Zhou. Multi-instance deep learning: Discover discriminative local anatomies for body part recognition. *IEEE Transactions on Medical Imaging*, 35(5), 1332-1343, 2016.