

Arrhythmia Detection in ECG Signals Using a Multilayer Perceptron Network

Gaurav Kumar, Urja Pawar and Ruairi O'Reilly

Cork Institute of Technology, Ireland,
gaurav.kumar@mycit.ie, urja.pawar@mycit.ie, ruairi.oreilly@cit.ie

Abstract. Electrocardiography (ECG) is a form of physiological data used to record the electrical activity of the heart. Numerous researchers have proposed and developed methods to extract features from the ECG signal (for example, R-R segment, P-R segment). These features can be used to analyse and classify various forms of heart arrhythmia.

In this work, a method for ECG classification that employs a generalised signal pre-processing technique and uses a Multi-Layer Perceptron network to classify arrhythmia per the AAMI EC57 standard accurately is presented. The method is trained and evaluated using PhysioNet's MIT-BIH dataset, and an average accuracy of 98.72% is achieved. The proposed methodology is comparable to state-of-the-art CNN models, both in terms of accuracy and efficiency.

Keywords: Arrhythmia Classification, Multi-Layer Perceptron, Convolutional Neural Networks

1 Introduction

An Electrocardiogram (ECG) is a time-series signal used for recording the electrical activity of the heart. ECG recordings require a cardiologist to interpret and detect cardiac abnormalities or arrhythmia. A typical heart beats in a steady rhythm. A heartbeat varies across individuals and within individuals depending on a variety of conditions. The segments of a standard ECG signal consist of waveforms like P, Q, R, S, T and U as depicted in Figure 1.

The QRS complex which represents ventricular depolarisation and contraction typically begins with a downward movement and is composed of Q, R and S waves where the Q wave is a larger upwards deflection, a peak at R and then a downwards S wave as depicted in Figure 1. The PR segment or interval indicates the time endured by the wave to travel from the sinus node to the ventricles. The RR interval represents the time between successive QRS complexes and is used to calculate heart rate.

Electrocardiography (ECG) monitoring is used in diagnosing and treating patients with heart disorders. In order to detect and precisely categorise abnormal beats in an ECG signal, high-level expertise in the domain are required. This requirement introduces several constraints concerning the expert analysis of ECG data: i) It is time-consuming and prone to human errors; ii) There are a

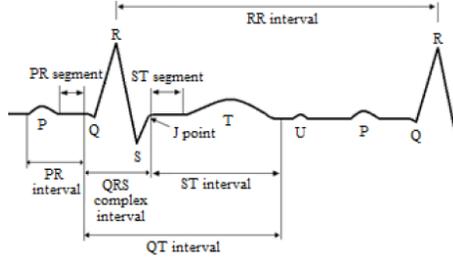


Fig. 1. ECG intervals and segments depicting QRS complex [1].

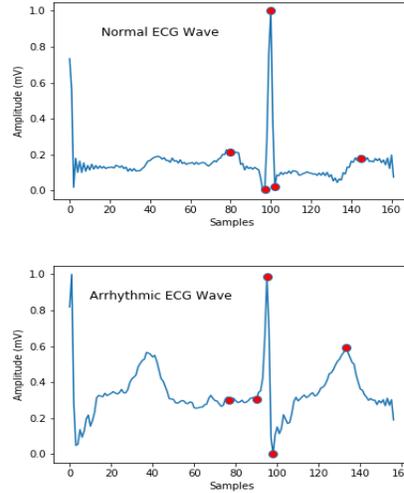


Fig. 2. Normal Vrs Arrhythmic ECG Wave.

limited number of expert cardiologists available to diagnose the millions of patients suffering from heart disorders; iii) The cost of diagnosis is expensive. These constraints highlight the need for a reliable and low-cost means of analysing and diagnosing an individual’s cardiac health [16].

In addressing this need, numerous researchers have investigated the application of machine learning techniques to ECG in order to automate the detection of abnormalities in ECG signals [6] [5]. The resultant models have demonstrated precision in identifying and classifying the wave morphologies of ECG signals which plays a significant role in the detection of abnormalities.

In this work, a Multi-Layer Perceptron (MLP) model is proposed which is trained on a pre-processed version of PhysioNet’s MIT-BIH dataset. The trained model has achieved an accuracy of 98.72%, which is comparable to state-of-the-art methods [10][2][12] in ECG classification. It is envisaged that this model will enable a sufficiently accurate analysis with a low computational cost such that analysis can be carried out in real-time on low-end devices. The model could, therefore, contribute to the incorporation of automated analysis solutions for conditions such as heart disorders into devices such as activity trackers with the intent of reducing the health impact on the general public.

2 Related Work

Several machine learning algorithms have been proposed and adapted for the accurate classification of ECG data, an excellent overview of which is presented in [15]. This section demonstrates the most relevant techniques and their applications in the field of ECG classification.

Application of Artificial Neural Networks (ANN) In this paper, [11], sufficient accuracy was achieved with a short learning time. A new arrhythmia classification algorithm was proposed, which had a fast learning speed and high accuracy by making use of Morphology Filter, Principle Component Analysis (PCA) and Extreme Learning Machine (ELM). The accuracy levels 98.00% in terms of average sensitivity, 97.95% in terms of average specificity was achieved. Additionally, a comparative study was performed in terms of learning rate using an ELM, comparing back propagation neural network (BPNN), radial basis function network (RBFN) and support vector machines (SVM). It was observed that

the learning time of the proposed algorithm using ELM was about 290, 70, and 3 times faster than an algorithm using a BPNN, RBFN and SVM, respectively.

Vishwa et al. [17] implemented an ANN (Artificial Neural Network) based classification system to detect heart disorders through ECG analysis using estimated feed-forward ANN and back-propagation learning algorithms. The approach was performed on a subset of arrhythmia classes in the MIT-BIH database resulting in an accuracy of 96.77%.

In order to classify ECG data, automatic extraction of both time interval and morphological features was carried out in [3]. Linear Discriminant Analysis (LDA) and Artificial Neural Networks (ANN) were used for classification. The ANN (in the form of an MLP) proved to be the more accurate of the two classifiers with a training accuracy of 85.07% and 70.15% on unseen data. Principal Component Analysis (PCA) was used for feature selection and dimensionality reduction.

Jadhav et al. [9] used Modular Neural Networks (MNN) to classify ECG signals into normal and abnormal classes. The UCI arrhythmia dataset was used for this experiment. The hidden layers in the network were varied, and the model was trained on different subsets of the training data. The model was capable of achieving an accuracy of 82.22% on the unseen or test dataset.

Application of Convolutional Neural Networks In [2], researchers used Convolutional Neural Network (CNN) to detect arrhythmia in ECG heartbeats. The CNN was trained using the MIT-BIH dataset. An accuracy of 93.5% was achieved. The dataset was pre-processed by: i) Removing noise from the ECG signals with the help of wavelet filters; ii) Segmenting the ECG signal into R-peak beats and; iii) Normalising each segment to scale the amplitude of the beat. The annotations in the dataset were divided into five categories namely: non-ectopic (N), supraventricular ectopic (S), ventricular ectopic (V), fusion (F), and unknown (Q) as per the Association for the Advancement of Medical Instrumentation (AAMI) standard [13].

Data Augmentation was practised to address the imbalance in the dataset. Synthetic data was generated for the minority classes to prevent the model from over-fitting on the majority class. The CNN model contained one input layer, one output layer and eight hidden layers of Convolutional, max-pooling and fully-connected layers.

In [10], researchers extracted R-R features from the MIT-BIH dataset and used these features as input to the model. The data is then subjected to a series of convolution layers applying 1-D convolution. The predictor network consists of five residual blocks, followed by two fully-connected layers and a softmax layer to predict output class probabilities. Each residual block contains two convolutional layers, two ReLU activation layers, a residual skip connection, and a pooling layer. In total, the resulting network is a deep network consisting of 13 weight layers.

In a residual block, a layer can either feed the data into the next layer or the layers 2-3 steps away. In other words, the model may train the layers in a residual block or may skip the training of those layers by using skip connection. This

ability makes the model more dynamic and overcomes some of the drawbacks of having extra layers (such as over-fitting and slow learning) in the network.

The TensorFlow computational library for model training and evaluation was employed. For the softmax layer, cross-entropy was used as the loss function. Adam optimiser was used to train the network, with the learning rate, beta-1 and beta-2 of 0.001, 0.9, and 0.999, respectively. Learning rate is decayed exponentially with the decay factor of 0.75 every 10000 iterations.

The performance of the arrhythmia classifier was tested on 4079 heartbeats (about 819 from each class) which were not used in the network training phase. Data augmentation was used to balance the number of beats in each category. The final accuracy which they were able to achieve was 93.4% on the MIT-BIH arrhythmia dataset.

The learned representations or weights (filters in the case of a CNN) were used to classify MI in the PTB Diagnostic dataset. This experiment involved freezing the learned weights till the last convolution layer of the CNN network and training only the last two fully-connected layers with 32 neurons each. The model achieved an accuracy, precision and recall of 95.9%, 95.2% and 95.1% in MI classification, respectively. The network was trained for approximately two hours on a GeForce GTX 1080Ti processor.

In summary, a variety of machine learning algorithms and their application to ECG data is evident in the literature. CNN's demonstrate state-of-the-art accuracy but are expensive in terms of the computation time required to train. To detect arrhythmias in real-time, the proposed model needs to be accurate as well as computationally inexpensive. As such, a generalised signal pre-processing approach, as described in [10], is adopted, and an MLP model is proposed to classify ECG heartbeats accurately.

3 Methods

This section presents the design and implementation of the proposed MLP network and its comparison with a state-of-the-art Convolutional Neural Network (CNN) for ECG beat classification. An overview of the proposed MLP network is depicted in Figure 3.

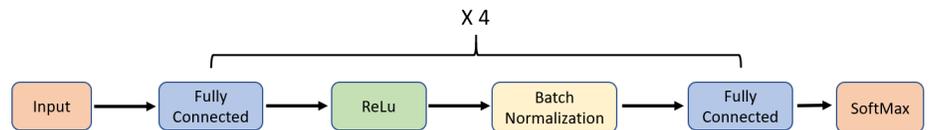


Fig. 3. Architecture of the proposed MLP network

The first layer is the input layer through which the data will be fed to the network. The number of neurons in the input layer is 187 (equal to the number of features/columns in the data). The following network consists of 4 sets of Fully-connected (Dense), Batch Normalization and ReLU activation layers. Number of neurons in each of the fully-connected layers are 50, 150, 900 and

400, respectively. The last layer of the network is the fully-connected output layer of 5 neurons (equal to the number of distinct classes in the dataset) with an activation function of SoftMax (as this is a multi-class classification problem).

3.1 MIT-BIH Dataset

The MIT-BIH dataset [14] [8] consists of ECG recordings of 49 distinct subjects recorded at the sampling rate of 360Hz. Each record contains a recording of 30 minutes from two leads namely modified limb lead II (MLII) and one out of the modified leads V1, V2, V3, V4 or V5. The dataset contains more than 109,000 beats annotated individually, belonging to one of possible 15 beat types.

The R-R interval of the ECG signal is widely used in the literature for the classification of ECG signals. M. Kachuee *etal.* [10] extracted the R-R intervals from the ECG signals of the MIT-BIH dataset. These features are then used for arrhythmia classification. The annotations available in the MIT-BIH dataset contains five different beat categories as denoted in Table 1.

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

Table 1. Mapping between heartbeat annotations and AAMI EC57 [13] categories.

The researchers in [10] used 47 recordings for the experiment and down-sampled the sampling frequency of the MIT-BIH dataset from 360Hz to 125Hz. The steps followed to extract ECG beats from the original signal are: i) Extracting the R-R intervals by splitting the original continuous ECG signal to windows of 10 seconds and selecting a 10-second window from the signal; ii) Normalising the amplitude of the extracted signal to a range between zero and one; iii) Extracted the set of local maximums with a threshold of 0.9 representing ECG R-peaks and; iv) Padding the extracted R-R interval with zeros making sure that all the extracted beats are of identical length.

The advantages of this pre-processing include: i) It is useful in extracting R-R intervals from signals with distinct morphologies (shapes); ii) No filter is applied to extract the beats that make an assumption about the signal morphology (for example, Fourier filter makes an assumption that the actual signal frequencies fall at low frequencies while noise at high); iii) All the extracted beats have identical length which is essential for being used as input to the successive processing parts. The pre-processed MIT-BIH dataset has been made available on Kaggle [7] by [10] and is used for training the proposed MLP network. It provides the extracted R-R features from the dataset along with a predefined 80:20 split for training and testing.

3.2 Pre-processing

The MIT-BIH training data is highly imbalanced i.e. the distribution of instances per class is not uniform. The number of instances for the “Normal” class is

82.77% of the whole training dataset. Therefore, there is a high probability for the model to over-fit or gets biased towards the majority class (“Normal”) and generalises the other classes to be Normal as well. To prevent the problem of over-fitting, Compute Class Weight function of Class Weight library with “balanced” as a parameter is used. This function calculates the weights per class by weighing classes inversely proportional to their frequency:

$$w_j = \frac{n}{kn_j}$$

Here, w_j is the weight to class j , n is the number of observations, n_j is the number of observations in class j , and k is the total number of classes.

3.3 Modelling

To build the network Keras (a high-level neural network API), Sequential Model is used. It is a linear stack of layers. Various layers can be added by specifying a list of layer instances to the model. The input layer in Keras Sequential model is defined by specifying the dimension of input data which in this case is 187 (the number of columns in the dataset). The number of rows is not specified because it may vary for the training and test dataset.

The first hidden layer of the network is a Dense layer with 50 neurons. The weights of this layer are initialised with an Identity matrix with a multiplicative factor (gain) of 1. A Batch Normalisation layer follows this. The Batch Normalisation layer is responsible for normalising the output of the hidden layer and increasing the learning speed of the model. Finally, these normalised values are passed to a ReLU activation layer which will decide based on polarity (negative or positive) of the value whether the individual neuron is activated or not.

The activated output from the first hidden layer is then rendered to the next three sets of Dense, Batch Normalisation and ReLU activation layers (successively) of the MLP network. The activated output from the last hidden layer is assigned to the output SoftMax activation layer with five neurons to classify the input data amongst one of the five distinct classes of the MIT-BIH dataset.

During the training phase of the network, the classifications on input data made by the model are compared with the actual labels (classes) to compute the training loss in each iteration. The function used to calculate the loss in the proposed network is Sparse Categorical Cross-Entropy, as it is a multi-class classification problem and only one label or class is applicable per instance.

Weights of hidden layers are updated or tuned by an Adam optimiser after the training loss has been calculated. Tuning helps to decrease the overall training loss in the next iteration or epoch of training. The learning rate (magnitude by which the weights are updated) used to update the weights is 0.001.

Apart from calculating loss and optimising weights, the model also evaluates training performance in each epoch, i.e. determining the number of correctly classified instances. The metric used for assessing the model’s performance is Accuracy.

Loss, Optimizer and Metric are specified while compiling the model. The compiled model is then trained on the training set (containing both the feature

and label data) of 87,554 instances of ECG beats for 100 epochs. At each epoch predictions made on the validation set (also known as the test set) are evaluated; this validates the performance of the model. The validation set contains 21,892 ECG beats and is not used in the training phase of the model.

3.4 Architectural Design of the Proposed MLP and the CNN

The proposed MLP network was then compared with a state-of-the-art CNN as presented in [10]. This CNN was chosen for comparison as it uses the generalised signal pre-processing technique without any form of filters applied. Therefore, both networks use the same pre-processed MIT-BIH dataset for training and validation of the models and so a comparative analysis can be derived.

The significant difference between the two networks is in their architectural design. The proposed MLP network employs dense or fully-connected layers to process the input data while CNN makes use of convolutional layers. There are in total six layers in the MLP network (one input layer, one output layer and four hidden weighted layers - see Figure 3), whereas the CNN architecture contains 15 layers (including input and output layer) of which 13 are weighted (11 convolutional and two fully-connected).

All convolutional layers in the CNN network apply 1-D convolution, and each layer has 32 kernels or filters of size five, and the two fully-connected layers have 32 neurons each. However, the four dense hidden layers of the proposed MLP network have 50, 150, 900 and 400 neurons, respectively. ReLU (Rectified Linear Unit) activation function is utilised to activate the neurons or filters in both networks. The weights of the first dense hidden layer of the MLP network are explicitly initialised by an identity matrix with a multiplicative factor of 1, whereas the Kernels or filters initialiser in the CNN model is not specified¹.

A batch normalisation layer (BNL) is employed after each hidden layer to normalise the weighted sum output of each dense hidden layer in the proposed network. The BNL helped the network to train faster and prevent over-fitting. No such standardisation technique appears to have been applied in the CNN network rather 5 Max-Pooling layers one in each residual block is practised.

The output layer of both the networks contains five fully-connected neurons with SoftMax activation function to classify the given instance amongst one of the possible five classes. Both the proposed MLP and the CNN is compiled by employing Accuracy, Adam and Categorical Cross Entropy as the metric, optimiser and the loss function respectively.

4 Results

This section details the evaluation and testing of the proposed MLP network and its comparison to a state-of-the-art CNN network.

4.1 MLP: Mini-Batch Training

Propagating the whole training set in a neural network in each iteration (epoch) is referred to as batch training. It typically increases memory consumption and

¹ Authors of [10] e-mailed querying implementation details. No reply to date.

the time to train the model. To address this, mini-batch training is practised. Mini-batch training propagates fixed subsets of the training data through the network one by one. For instance, if the training data has 32,000 instances, and the mini-batch size is 32, then there will be 1000 mini-batches.

The advantages of using mini-batch training are: i) Reduced memory consumption as only one batch of training data is loaded in the memory at a time; ii) Reduced training time. The network weights are updated with each propagation of a mini-batch while the weights are updated only once per epoch in batch training. The default batch size of the Sequential Model API is 32. Depending on the size of the training data, the batch size can be altered.

Mini-Batch testing on MLP Network

The proposed MLP network is trained with various batch sizes for 50 epochs. The performance of the model is evaluated based on the time taken to train and accuracy achieved on the validation or test dataset. Using the batch size of 512 yielded the best performance of the MLP model in terms of accuracy and speed. The accuracy achieved on validation dataset is 98.28% and the model took 3.4 minutes to train. Table 2 denotes the performance of the MLP network for different batch sizes.

Batch Size	Training time	Accuracy	F1 score
32	39.2 min	97.69%	90.64%
64	20.8 min	97.92%	91.46%
128	11.2 min	98.13%	91.73%
256	6.5 min	98.21%	92.05%
512	3.4 min	98.28%	92.24%

Table 2. Results for the MLP network over different batch sizes with regard to training time accuracy and F1 score.

4.2 MLP: Kernel Initialiser testing on the MLP network

Kernel initialiser (also known as weights initialiser) is a technique that helps to assign initial values to the weights of hidden layers in the network. By default, the weights initialiser for a Dense layer is Glorot-Uniform. This technique draws samples from a uniform distribution within -limit to +limit where the limit is defined as:

$$limit = \sqrt{\frac{6}{fan - in + fan - out}}$$

Here, fan-in is the number of neurons in the previous layer and fan-out is the number of neurons in the current later. The MLP network is tested on various kernel initialiser techniques like glorot-uniform, glorot-normal, identity, orthogonal, and random-uniform.

1. **Glorot-Normal:** This technique draws samples from a truncated normal distribution centered on 0 with standard deviation defined as:

$$stddev = \sqrt{\frac{2}{fan - in + fan - out}}$$

Here, fan-in is the number of neurons in the previous layer and fan-out is the number of neurons in the current later.

2. **Identity:** This technique generates an identity matrix of weights. It is used only for 2-Dimensional matrices. If the resulting matrix is not square, it pads the additional rows/columns with zeros.
3. **Orthogonal:** This technique generates a random orthogonal matrix of weights.
4. **Random-Uniform:** This technique initialises the weights with a uniform distribution. It takes three arguments: (minval: the lower bound of the range of random values to generate, maxval: the upper bound of the range of random values to generate and seed: A seed is a python integer used to seed the random generator. Mainly used for the similar production of values.)

Initializer	Accuracy	F1 score
Glorot Normal	97.80%	90.11%
Orthogonal	98.22%	91.90%
Random Uniform	97.86%	90.54%
Identity	98.43%	92.46%

Table 3. Evaluating the performance of the MLP network on different weight initializers.

Out of all the above-mentioned kernel initialisers, identity initialisation produced the best results. An accuracy of 98.43% is achieved on the validation dataset. The batch size used for this testing is 512. Table 3 denotes the performance of the different initializers evaluated on the MLP network.

4.3 MLP: Gradual Decay in Learning Rate

This technique is used to reduce the learning rate of the model while training. It monitors a metric or quantity, and if no improvement is seen for X number of epochs, the learning rate is reduced. Validation accuracy is monitored during the training of MLP network, and if no improvement in the validation accuracy is observed for five epochs, the learning rate is decreased by a factor of 1. Reducing the learning rate by introducing a gradual decay improved the overall performance of the MLP network (With Gradual Decay — Accuracy: 98.72% F1 score: 93.20%, Without Gradual Decay — Accuracy: 98.43% F1 score: 92.46%).

4.4 Comparative Analysis of the MLP and CNN

To evaluate the performance of the proposed MLP network a state-of-the-art CNN [10] was selected. The CNN was trained on the pre-processed MIT-BIH dataset for 50 epochs and the associated validation dataset used to evaluate the performance of the model. The model took approximately 28 minutes to train and achieved an accuracy of 98.6% on the validation dataset. In [10], an accuracy of 93.4% on the MIT-BIH dataset is reported. The increase of 5.4% in the validation accuracy observed in this experimental is probably due to the split used between the training and validation dataset.

A slight variant of the replicated CNN was also trained to enable a more transparent comparison. The techniques utilised by the MLP model to yield better performance such as Batch-Normalization layer, mini-batch learning, kernel initialisation, class-weight computation, and reducing learning rate were also utilised by this instance of the CNN, referred to as CNN-REP*.

Method	Training time	Accuracy	F1 score	Precision	Recall
Proposed MLP	3.4 min	98.72%	93.20%	94.6%	91.0%
CNN-Rep	28.4 min	98.64%	92.58%	94%	91.4%
CNN-Rep*	15.9 min	98.79%	93.01%	94.4%	91.8%

Table 4. Evaluating the performance of the MLP and CNN networks in the classification of heartbeats. Note: CNN-Rep as replicated from [10] CNN-Rep* with the performance enhancing techniques applied.

Table 4 denotes the results of the three networks, the proposed MLP, the replicated CNN and the improved replicated CNN. It is evident that: i) The MLP network is less computationally expensive when compared to both CNNs and demonstrates a reduced training time; ii) The proposed MLP outperformed the replicated CNN network in terms of validation accuracy and F1 score while it did not outperform CNN-Rep*. Figure 4 depicts the validation loss and accuracy graph for the MLP and CNN-Rep network. iii) The performance enhancement techniques utilised by CNN-Rep* resulted in a 44% decreased in training time when compared to CNN-Rep and a .15% improvement in accuracy. While the CNN-Rep* demonstrated an accuracy .07% better than the proposed MLP, it took 4.68 times longer to train.

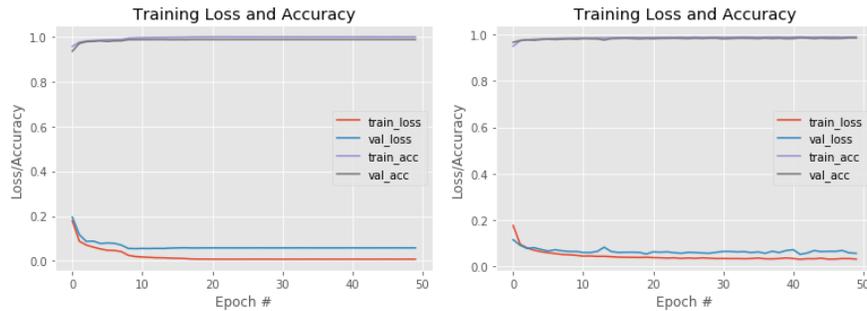


Fig. 4. Training Loss, Accuracy and Validation Loss and Accuracy visualised for MLP (Left) and the CNN (Right).

4.5 Classification of ECG data in real-time

This experiment intended to evaluate the classification of ECG data in real-time. The results are indicative of a models suitability for real-time analysis. For this experiment, five ECG beats are extracted from the validation dataset and joined

to make a continuous stream of ECG data. A window of size 187 is constructed, and a sliding window is used to analyse the ECG signal by one frame (column) width from left to right. At each interval, the window contains 187 bits of ECG data. The data obtained on sliding the window is analysed and classified by the network. The proposed MLP network slightly outperformed the CNN network in terms of average prediction time. The average prediction time for the MLP Network was 3.12ms and 4.3ms for the CNN Network [10]. Low-end devices, such as activity trackers and smartwatches, have access to limited compute, memory and storage. A model running on these devices will be competing for scarce resources with the applications and services being utilised by the device. As such, it is envisaged that the prediction time may vary, and this warrants the computational complexity of the underlying model being considered as part of the evaluation. A thorough assessment of trained models for arrhythmia detection running on low-end and edge devices will be the subject of future work.

Note: All experiments were performed using Google Co-laboratory (RAM: 12GB, Disk: 358GB, GPU: Tesla K80) and Keras computational library [4] for model training and evaluation.

5 Conclusion

In this paper, an MLP for the real-time detection of arrhythmia in ECG data is presented. In order to enhance the performance of the proposed model techniques including mini-batch training, gradual reduction in learning rate, batch-normalisation, kernel initialisation, and class-weight computation, are implemented. The performance of the resultant model is compared with a state-of-the-art CNN. Classification of the ECG data in real-time is performed to compute the average prediction time of the models.

The training and validation of the models were carried out using the MIT-BIH dataset. The proposed MLP outperformed a replicated state-of-the-art CNN in ECG beat classification. An average accuracy of 98.72% was achieved with an average time of 3.12 milliseconds to classify an ECG beat in real-time .

The rationale for the comparative accuracy gains experienced by the MLP is due to a combination of the pre-processing and/or implementation details omitted from the CNN network. Another instance of the CNN was implemented (CNN-Rep*) with the same hyper-parameters as those used by the MLP to enable a more transparent comparison. CNN-Rep* outperformed the MLP concerning accuracy ($\sim 0.07\%$) but also required approximately 4.6 times the amount of time to train.

The MLP network demonstrated itself as relatively computationally inexpensive approach and while this naturally implies it would take less time than a CNN to classify an ECG beat in real-time it also highlights its appropriateness for low-end devices, particularly with the level of accuracy demonstrated.

Acknowledgement: This material is based upon works supported by Science Foundation Ireland under Grant No. SFI CRT 18/CRT/6222

References

1. A real time ECG signal processing application for arrhythmia detection on portable devices - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/ECG-intervals-and-segments_fig1_321455361, accessed: 05-2019
2. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A., San Tan, R.: A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine* 89, 389–396 (2017)
3. Alexakis, C., Nyongesa, H., Saatchi, R., Harris, N., Davies, C., Emery, C., Ireland, R., Heller, S.: Feature extraction and classification of electrocardiogram (ecg) signals related to hypoglycaemia. In: *Computers in Cardiology, 2003*. pp. 537–540. IEEE (2003)
4. Chollet, F., et al.: *Keras* (2015)
5. Dastjerdi, A.E., Kachuee, M., Shabany, M.: Non-invasive blood pressure estimation using phonocardiogram. In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. pp. 1–4. IEEE (2017)
6. Esmaili, A., Kachuee, M., Shabany, M.: Nonlinear cuffless blood pressure estimation of healthy subjects using pulse transit time and arrival time. *IEEE Transactions on Instrumentation and Measurement* 66(12), 3299–3308 (2017)
7. Fazeli, S.: ECG Heartbeat Categorization Dataset. <https://www.kaggle.com/shayanfazeli/heartbeat>, accessed: 05-2019
8. Goldberger AL, Amaral LAN, G.L.H.J.I.P.M.R.M.J.M.G.P.C.K.S.H.: Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *IEEE Engineering in Medicine and Biology Magazine* 101(23), 215–220 (2003)
9. Jadhav, S.M., Nalbalwar, S.L., Ghatol, A.A.: Modular neural network based arrhythmia classification system using ecg signal data. *International Journal of Information Technology and Knowledge Management* 4(1), 205–209 (2011)
10. Kachuee, M., Fazeli, S., Sarrafzadeh, M.: Ecg heartbeat classification: A deep transferable representation. In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. pp. 443–444. IEEE (2018)
11. Kim, J., Shin, H.S., Shin, K., Lee, M.: Robust algorithm for arrhythmia classification in ecg using extreme learning machine. *Biomedical engineering online* 8(1), 31 (2009)
12. Martis, R.J., Acharya, U.R., Lim, C.M., Mandana, K., Ray, A.K., Chakraborty, C.: Application of higher order cumulant features for cardiac health diagnosis using ecg signals. *International journal of neural systems* 23(04), 1350014 (2013)
13. for the Advancement of Medical Instrumentation, A., et al.: Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms. *ANSI/AAMI EC38* 1998 (1998)
14. Moody, G.B., Mark, R.G.: The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20(3), 45–50 (2001)
15. Roopa, C., Harish, B.: A survey on various machine learning approaches for ecg analysis. *International Journal of Computer Applications* 163(9), 25–33 (2017)
16. Society, H.R.: Heart diseases and disorders. <https://www.hrsonline.org/Patient-Resources/Heart-Diseases-Disorders>, accessed: 05-2019
17. Vishwa, A., Lal, M.K., Dixit, S., Vardwaj, P.: Clasification of arrhythmic ecg data using machine learning techniques. *IJIMAI* 1(4), 67–70 (2011)