SUPPLEMENTARY MATERIALS FOR

DiaBeats: A MACHINE-LEARNING ALGORITHM TO NONINVASIVELY DETECT DIABETES AND PREDIABETES FROM ELECTROCARDIOGRAM

Anoop R. Kulkarni [1,2], Ashwini A. Patel [1], Kanchan V. Pipal [1], Sujeet G. Jaiswal [1], Manisha T. Jaisinghani [1], Vidya Thulkar [1], Lumbini Gajbhiye [1], Preeti Gondane [1], Archana Patel [1], Manju Mamtani [1,3], Hemant Kulkarni [1,3]

[1] Lata Medical Research Foundation, Nagpur, India

[2] Innotomy Consulting, Bengaluru, India

[3] M&H Research, LLC, San Antonio, Texas, USA

Corresponding author:

Dr. Hemant Kulkarni,
12023 Waterway Rdg
San Antonio, TX, USA 78249

Phone: (210) 602 5537
Fax: (210) 920 4434
Email: hemant.kulkarni@mnhresearch.com

Table of Contents

Supplementary Note 1: Choice of Deep Learning Architectures

Rationale: We aimed to compare the performance of the DiaBeats algorithm (XGBoost classifier) with deep learning architectures that are currently considered mainstream in the field of time series classification tasks. The chosen deep models represented a spectrum of deep learning methods (e.g. that included one-dimensional convolutional neural networks (1DCNN), long short-term memory (LSTM)), model complexity (number of layers in a model) and total number of model parameters.

Selected deep learning models: We selected the following five deep learning models for comparative performance evaluation:

- An LSTM-based model (hereinafter LSTM model),
- a 1DCNN time series classification model documented on the Keras platform (https://keras.io/examples/timeseries/timeseries_classification_from_scratch/, hereinafter 1DCNN-Keras model),
- the DeepECG model (the Conv1D.py model from https://github.com/ismorphism/DeepECG),
- the time series transformers model for classification (https://keras.io/examples/timeseries/timeseries_transformer_classification/, hereinafter Transformers model), and
- a deep, 16-layer 1D CNN-BatchNorm model (adapted from Kim and Pan[1], hereinafter CNN16 model)

Model architecture description: All the models used an input shape of a 100 x 12 matrix that represented a 12 lead ECG for a single heartbeat downsampled to a sampling frequency of 125 Hz (from the original recording at a sampling frequency of 1000 Hz). Also, the last layer of all the models was a fully connected layer with three neurons and a softmax activation function.

The backbone of the LSTM model was a total of 64 LSTM cells followed by batch normalization and a fully connected layer with 64 relu-activated neurons (Supplementary Figure 5). The 1DCNN-Keras architecture was as described by the original author and comprised of 1DCNN layers and batch normalizations, followed by global average pooling and the final fully connected layer (Supplementary Figure 6). The DeepECG model was also a 1D CNN model with the same architecture as described by the original author (with the exception of the sizes of the kernels for the convolutional layers as appropriate for the input shape used here). The details of this model architecture are shown in Supplementary Figure 7. The Transformers model has gained popularity for time series classification tasks in the clinical settings (Song et al[2], 2017; Wu et al[3], 2021) and consists of an encoder-decoder core that uses the concept of multi-head attention. The transformers model we used had 4 heads with a head size of 256, a series of 4 transformer blocks and a total of 128 multilayer perceptron units (Supplementary Figure 8). Lastly, the CNN16 model comprised of serial combination of 1D CNN layers (gradually increasing in number of kernels with the depth of the network) and batch normalization layers culminating into the final fully connected output layer. The detailed architecture is shown in Supplementary Figure 9.

Runtime model specifications: All the models used a batch size of 32, sparse categorical crossentropy loss function and the Adam optimizer with a learning rate of 0.0001. Model performance was monitored using sparse categorical accuracy in the validation set. The training, validation and test sets used for training and evaluation were the same as used by the DiaBeats classifier. Best model performance for each model was used for comparative purpose (Supplementary Table 1).

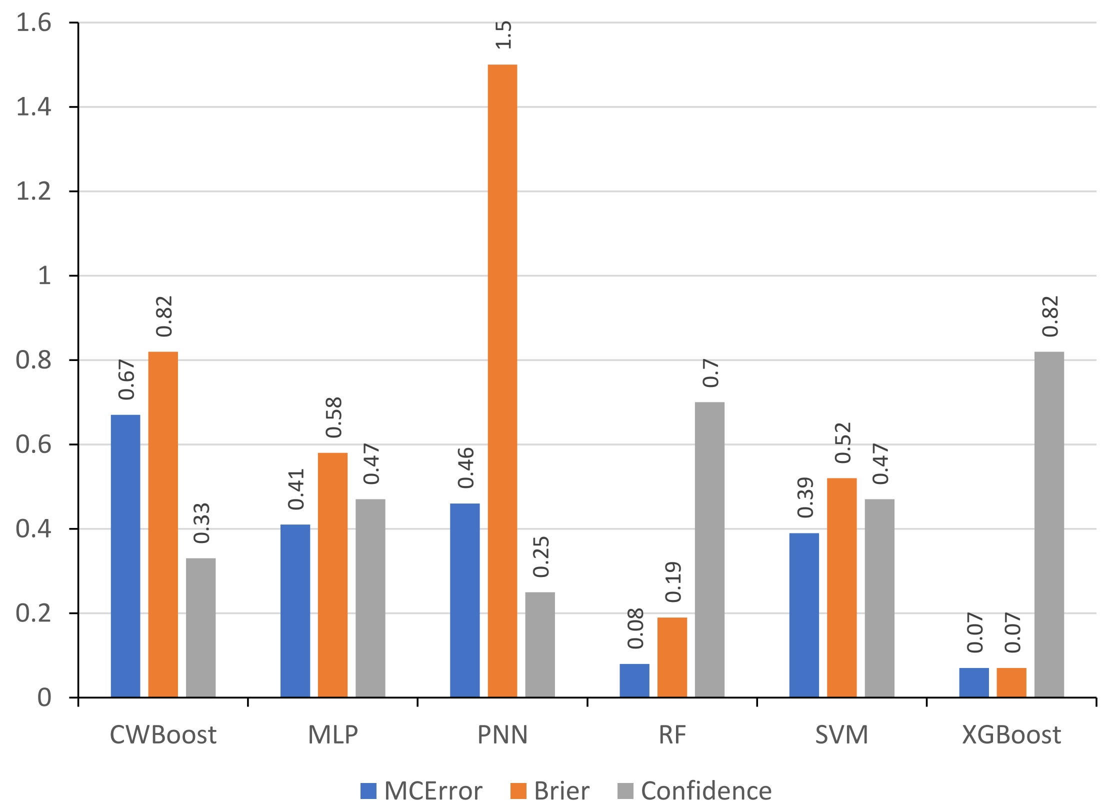
References:

1. Kim M-G, Pan SB. A study on user recognition using the generated synthetic electrocardiogram signal. Sensors 2021, 21(5), 1887.
2. Song H, Rajan D, Thiagarajan JJ, Spanias A. Attend and Diagnose: Clinical time series analysis using attention models. arXiv 2017:1711.0.905v2 [stat.ML]
3. Wu N, Green B, Ben X, O'Banion S. Deep transformer models for time series forecasting: the influenza prevalence case. arXiv 2020:2001.08317v1 [cs.LG]

Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

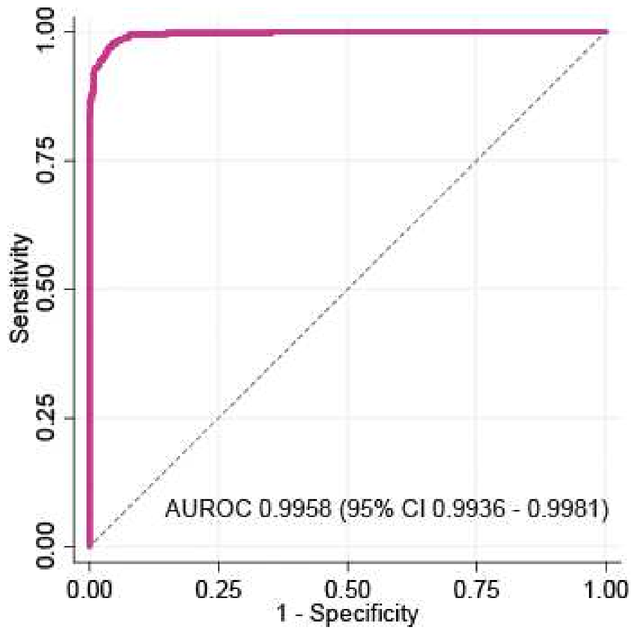
Supplementary Figure 1. Comparison of predictive performance of candidate machine learning techniques.

These analyses were restricted to the unbalanced training set (number of beats = 8,892) and used a 10-fold cross-section scenario. Best method was chosen as the one that yielded least misclassification error, lowest Brier score and highest classification confidence. The machine learning techniques used were component-wise boosting (CWBoost), multilayer perceptron (MLP), probabilistic neural networks (PNN), random forest classifier (RF), support vector machines (SVM) and extreme gradient boosting (XGBoost). Each method was evaluated with three performance metrics: misclassification error (MCError), Brier score (Brier) and confidence (average softmax probability when predicting correctly). We aimed to select a method that yielded lowest values of MCError and Brier scores and highest value of confidence.

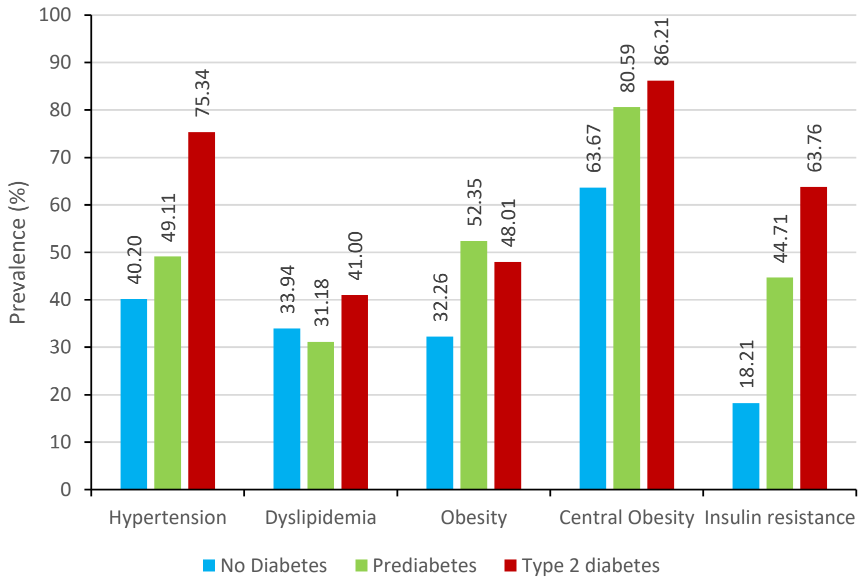


Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

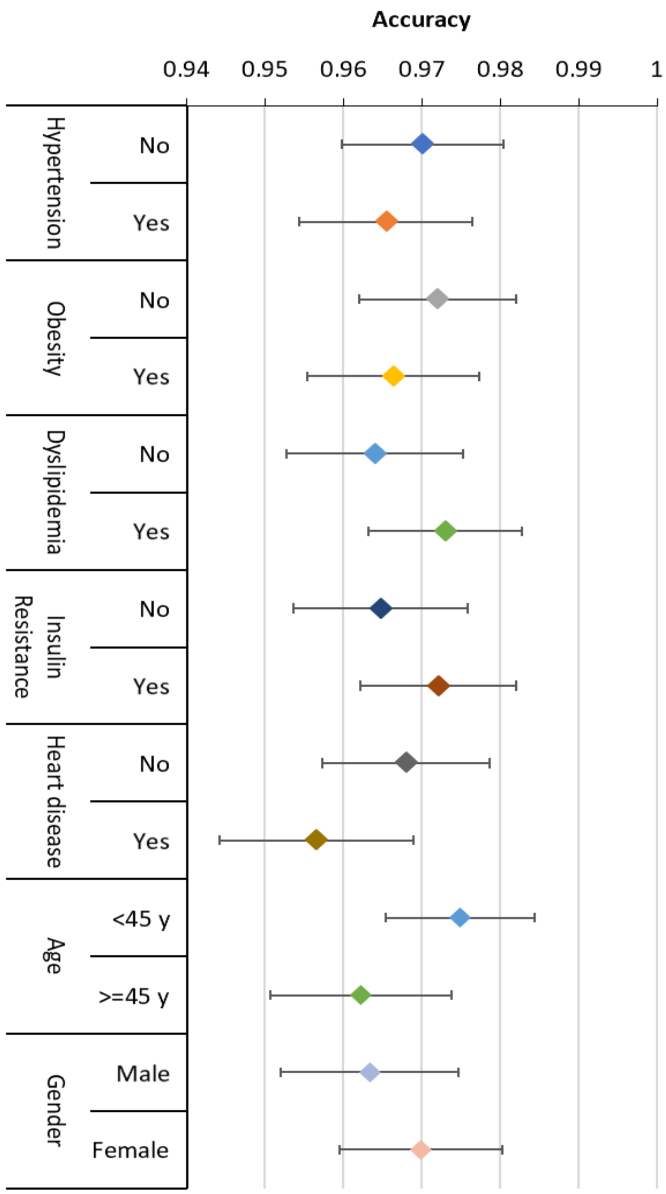
Supplementary Figure 2. Receiver operating characteristic curve to detect type 2 diabetes/prediabetes based on DiaBeats model. The target variable was a composite of type 2 diabetes and/or prediabetes. The predictor was a maximized softmax probability for the target class based on predictions given by the DiaBeats model. These analyses were restricted to the independent test set (1,046 beats).



Supplementary Figure 3. Coexistence of other metabolic conditions along with prediabetes and diabetes in the DISFIN study

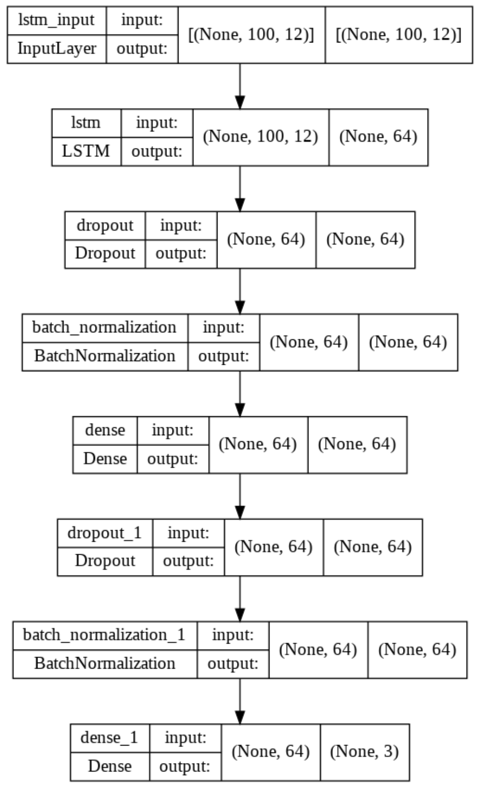


Supplementary Figure 4. Sensitivity analyses of the accuracy of the DiaBeats algorithm in the presence and absence of comorbidities and risk factors. Diamonds and error bars represent the point and 95% confidence interval estimates for the predictive accuracy.



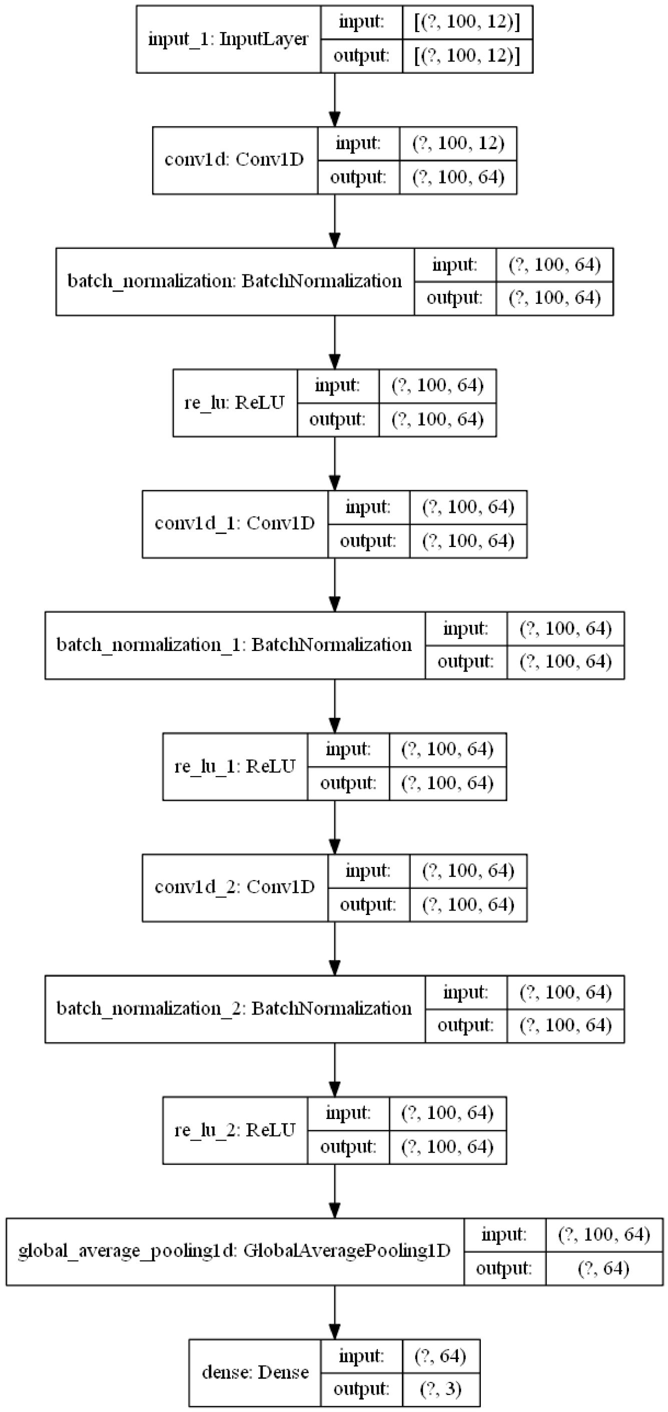
Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 5: Model architecture for the LSTM model (# parameters = 24,579)



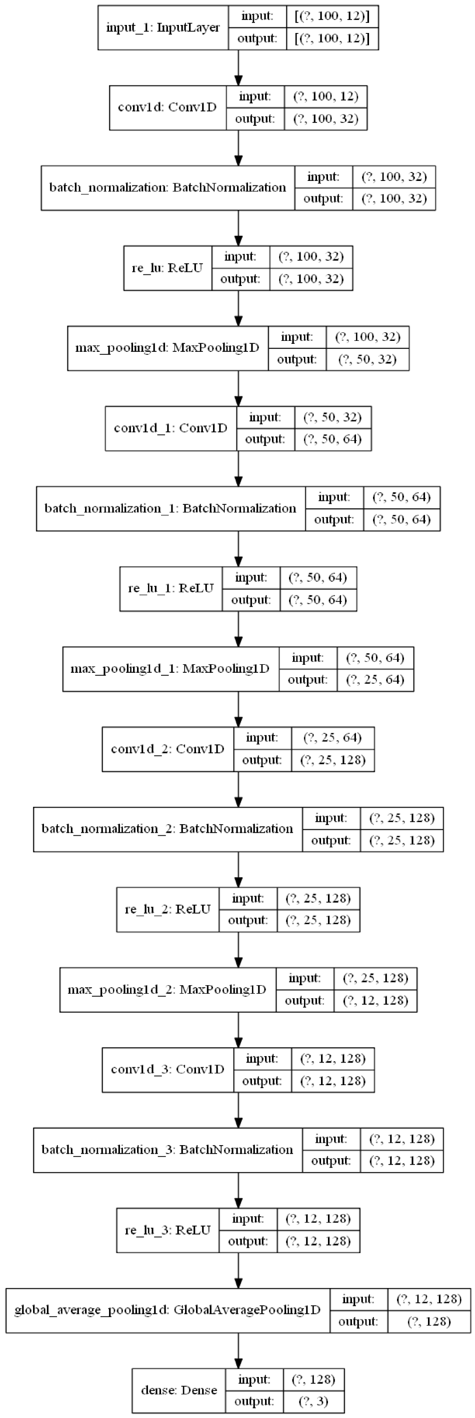
Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 6: Model architecture for the 1DCNN-Keras model (# parameters = 28,035)



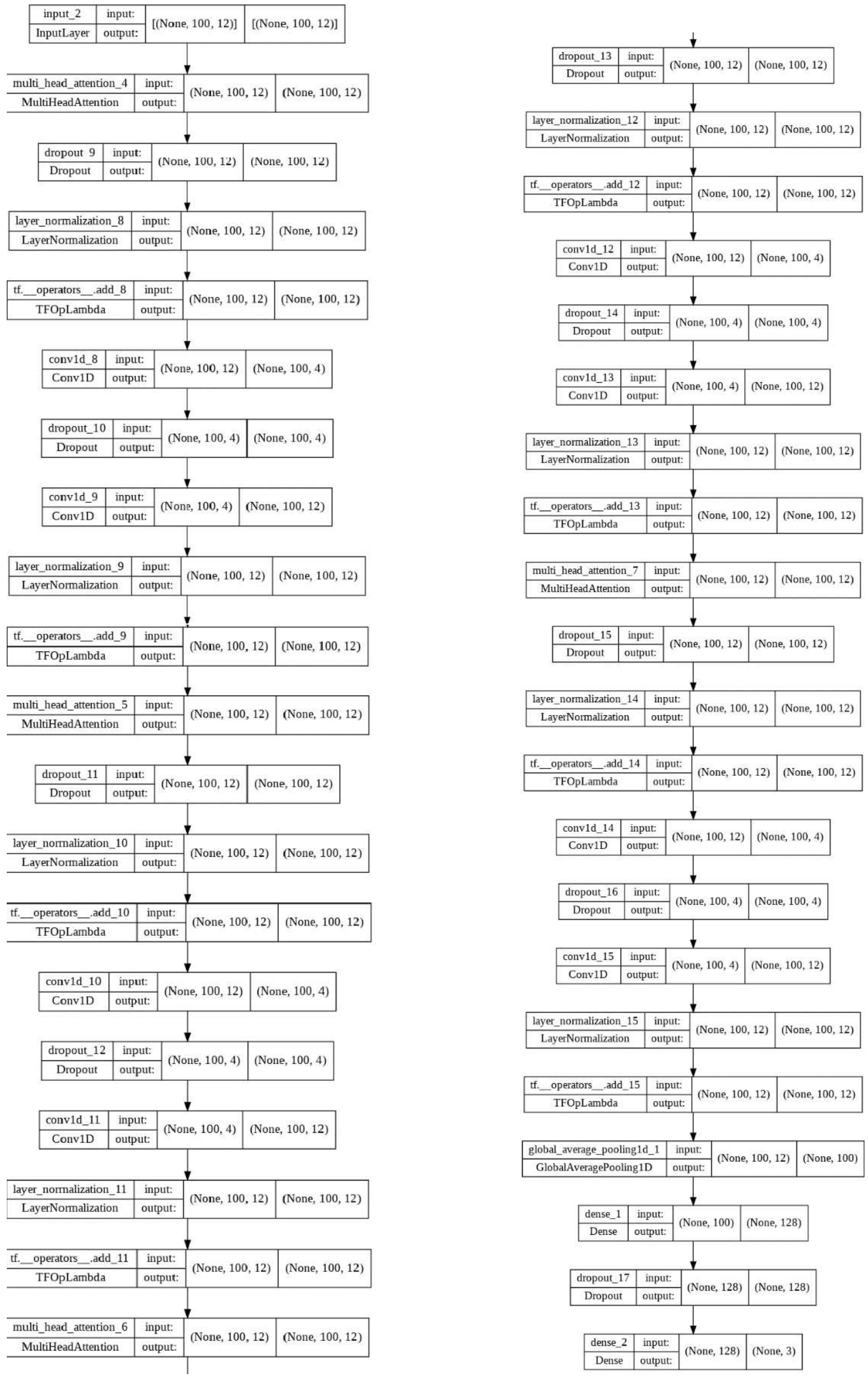
Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 7: Model architecture for the DeepECG model (# parameters = 112,355)



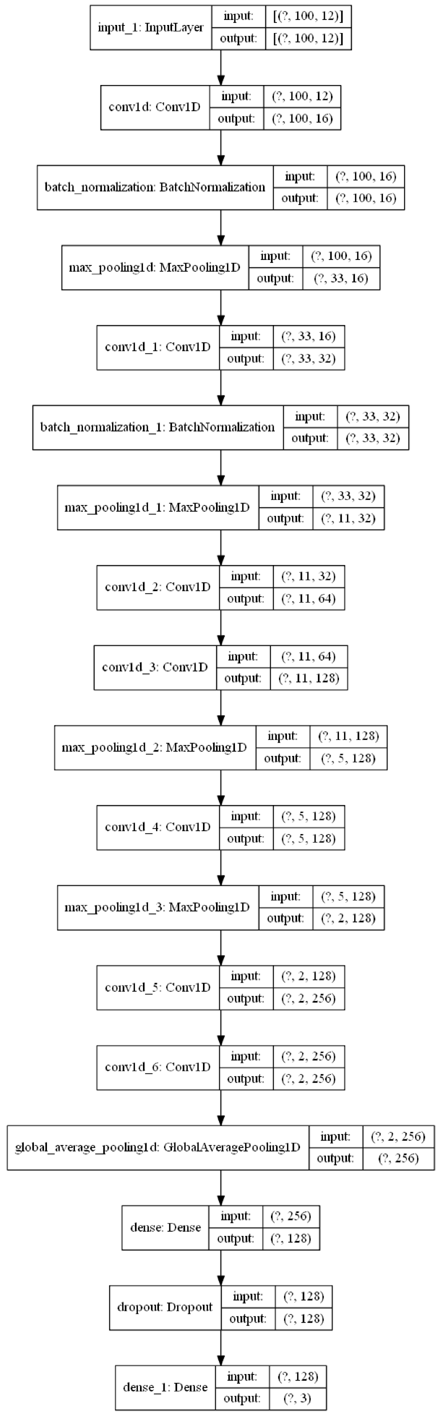
Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 8: Model architecture for the Transformers model (# parameters = 222,899)



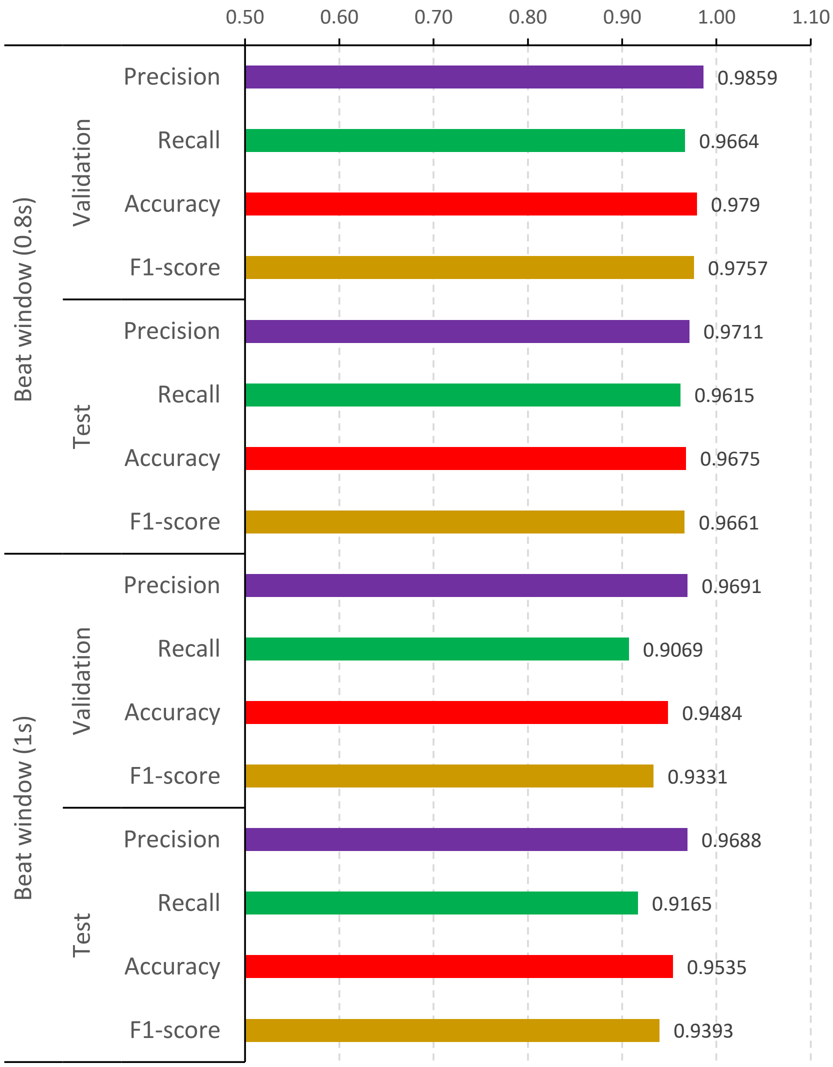
Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 9: Model architecture for the CNN16 model (# parameters = 433,523)



Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Figure 10: Comparison of differing beat widow size on classification performance of DiaBeats



Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759

Supplementary Table 1. Comparison of classification performance of deep learning architectures to predict diabetes and prediabetes.

Model (#Parameters)	**Metric**	**Dataset**		
		Training	***Validation***	***Test***
LSTM (24,579)	Loss	0.0855	0.2428	0.1663
	Accuracy	0.9737	0.9465	0.9426
1DCNN-Keras (28,035)	Loss	0.0878	0.1941	0.1602
	Accuracy	0.9701	0.9350	0.9503
DeepECG (112,365)	Loss	0.1134	0.1604	0.1438
	Accuracy	0.9574	0.9426	0.9474
Transformers (222,899)	Loss	0.4932	0.6508	0.6044
	Accuracy	0.7982	0.7467	0.7467
CNN16 (433,523)	Loss	0.0678	0.2044	0.1355
	Accuracy	0.9768	0.9407	0.9465

Kulkarni AR, *et al. BMJ Innov* 2022;0:1–11. doi: 10.1136/bmjinnov-2021-000759