Multiclass classification of heart beats

Final project - Big Data Analysis 2018/2019

Buster Niels Pedersen - hwl395
Simon Nyrup - mkw755
Jesper Pedersen - xhn538
Helle Kogsbøll Leerberg - mdv971

(Everyone has participated equally)

UNIVERSITY OF COPENHAGEN

MIT-BIH Arrhythmia Dataset

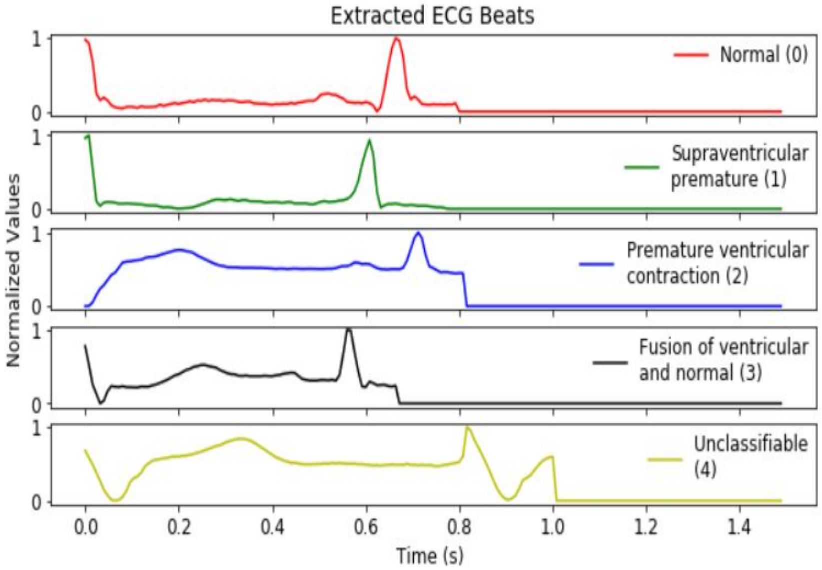
Two-channel ambulatory electrocardiogram (ECG) shapes of heartbeats, from 47 subjects, studied by the BIH Arrhythmia Laboratory (1975-1979).

Data from ECG Heartbeat Categorization Dataset on Kaggle, (https://www.kaggle.com/shayanfazeli/heartbeat), originally published by PhysioNet.org

Preprocessed when acquired. Long recordings have been cut in smaller bits. Zeros have been appended. Data has been normalized.

Data has been labeled by two doctors.

Number of samples: 109446
Number of categories: 5



Data acquired from: PhysioNet.org, cite: Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* **101**(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/content/101/23/e215.full]; 2000 (June 13).

MIT-BIH Arrhythmia Dataset

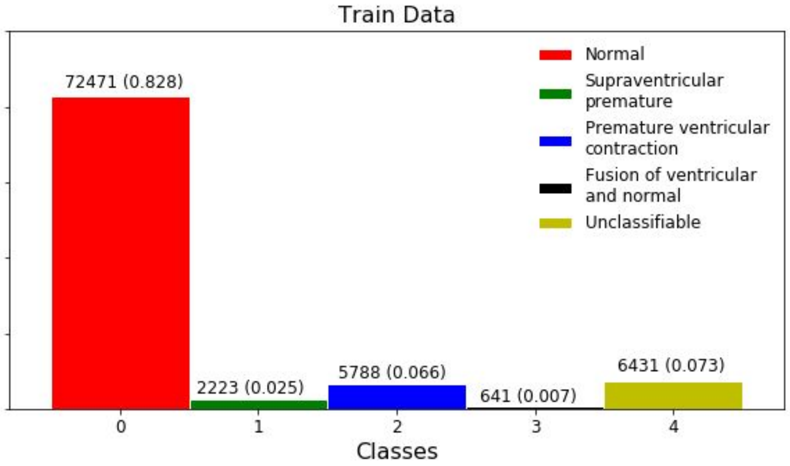
Two-channel ambulatory electrocardiogram (ECG) shapes of heartbeats, from 47 subjects, studied by the BIH Arrhythmia Laboratory (1975-1979).

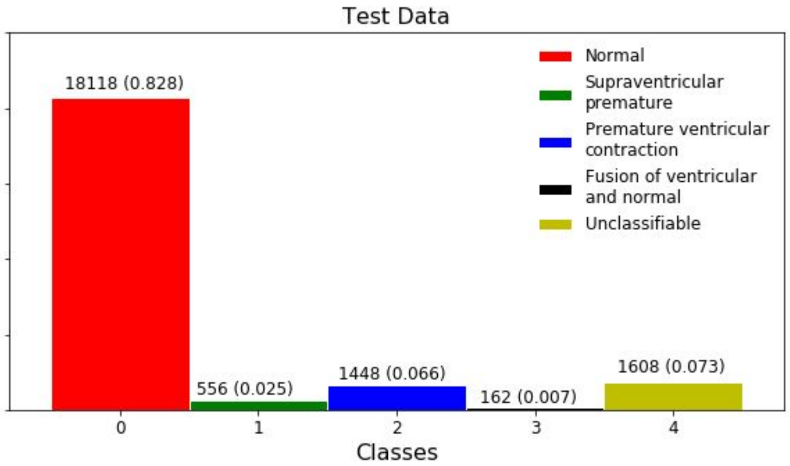
Data from ECG Heartbeat Categorization Dataset on Kaggle, (https://www.kaggle.com/shayanfazeli/heartbeat), originally published by PhysioNet.org

Preprocessed when acquired. Long recordings have been cut in smaller bits. Zeros have been appended. Data has been normalized.

Data has been labeled by two doctors.

Number of samples: 109446
Number of categories: 5





Evaluating data

To evaluate the data, we use:

- Accuracy
  Baseline is 82%

- Precision (Positive Predictive Value):
  `PPV = TP/(TP+FP)`
  Of all the patients we classified x how many does actually belong in label x

- Recall (True Positive Rate):
  `TPR = TP/(TP+FN)`
  Of all patients labeled x how many did we classify correctly.

- F1 score
  `F1 = 2(Recall*Precision)/(Recall + Precision)`
  Take both FP and FN into account.

- Averages - Micro and Macro
  Micro is weighted mean.
  Macro is harmonic average.

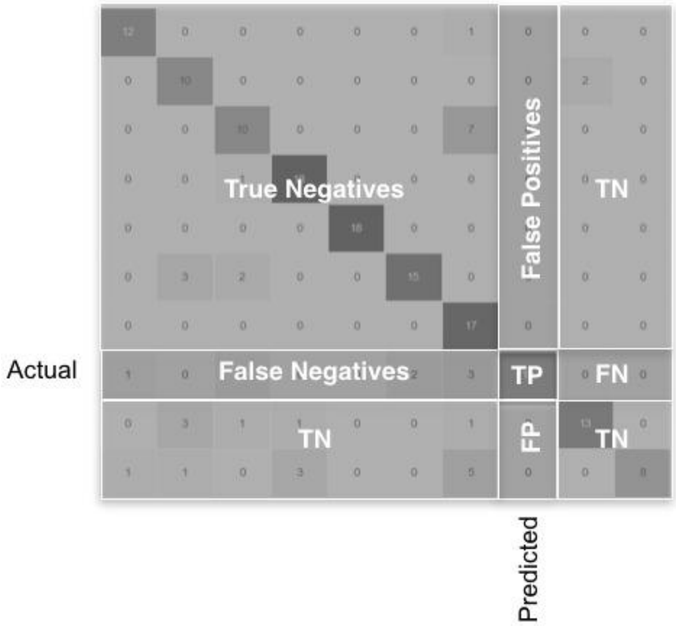


Image from:
https://stackoverflow.com/questions/50666091/true-positive-rate-and-false-positive-rate-tpr-fpr-for-multi-class-data-in-py

Overview of Machine Learning models

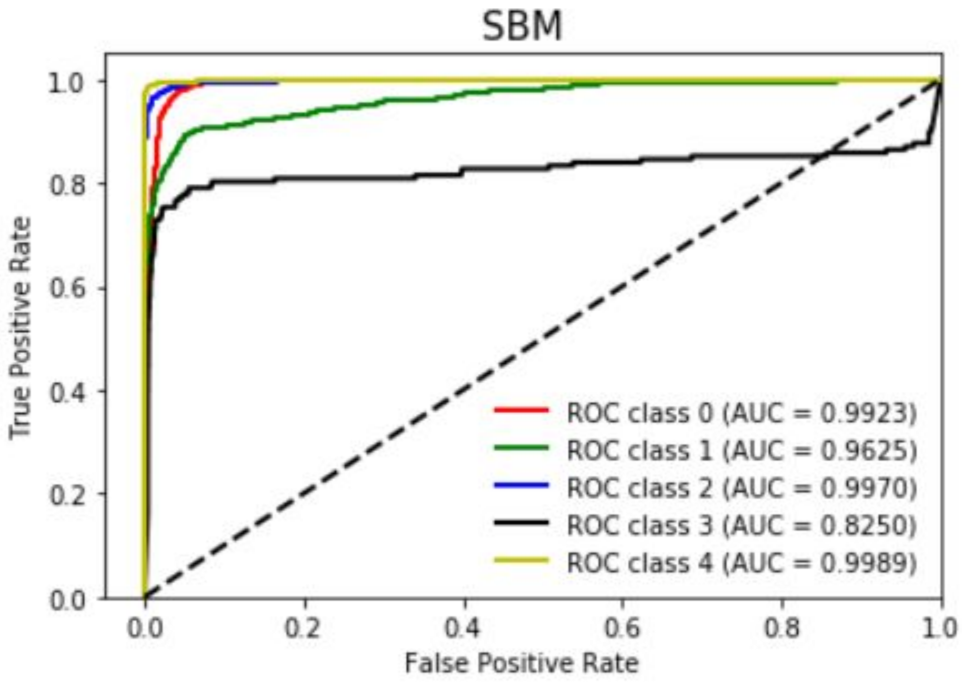
Type	Program	Justification
K-Nearest Neighbours (K=3)	sklearn	Easy to implement
GBM / SBM	sklearn	Good at handling imbalanced data
Neural Network	TensorFlow	Simple, fast, known
DWT preprocces with SBM	sklearn	Preprocessing to create time invariance
One vs. Rest (SBM)	sklearn	Default in multiclass classification
Recurrent Neural Network	PyTorch	Default recommendation for time sequences
1D Convolutional Neural Network	PyTorch	Convolutional filters for time invariance
Temporal Convolutional Network	PyTorch	Shown in paper to perform well on sequential data
Ensemble Learning	-	Fun

Imbalanced data - One vs. Rest (SBM)

Class transformation (class 3):

0	→	0
1		0
2		0
3		1
4		0

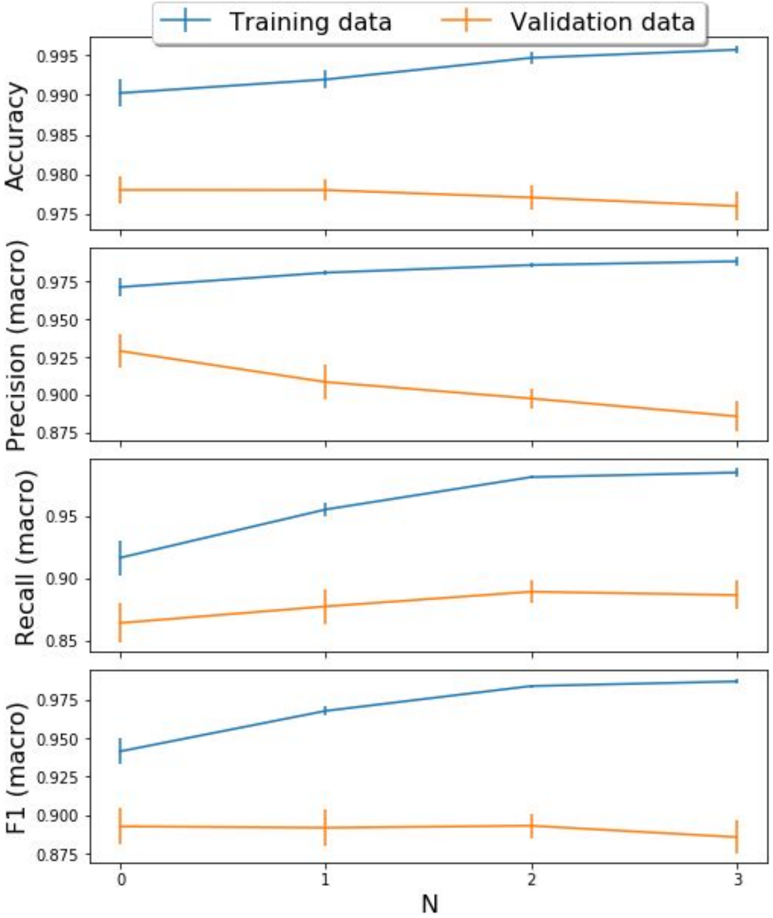
5 models specialized predicting single class



Imbalanced data - Neural Network

- Account for imbalanced data:
  Oversampling with 5-fold Cross Validation

- Undersampling and Synthetic data was not used

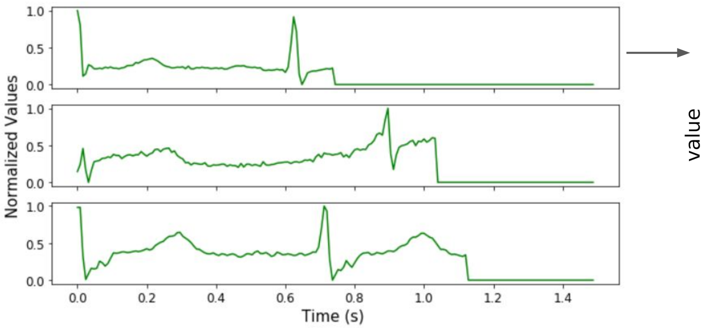


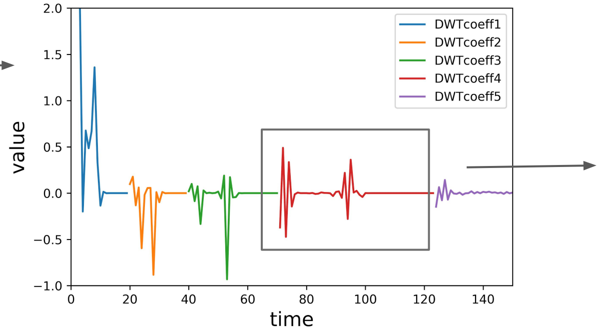
N: Times non-normal datapoints were added

Time dependence - Discrete wavelet transform(DWT) with SBM

Converts 1 heartbeat to 5 signals using DWT.

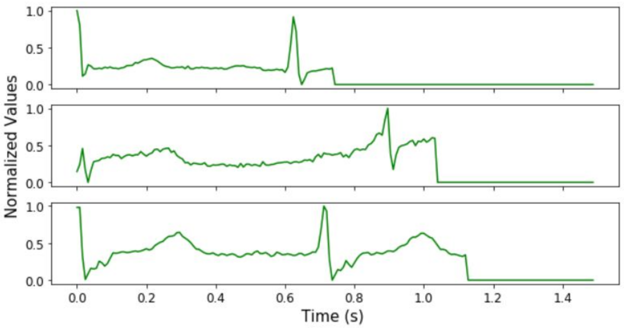
Converts every 1 one of the DWT to 12 statistics like mean, variance, zero-crossings etc.

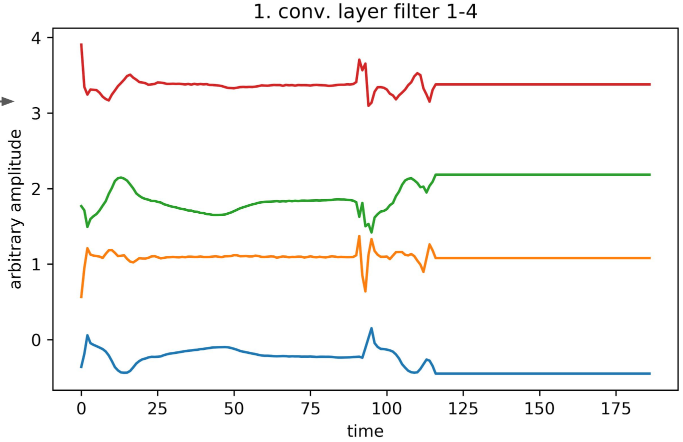




Parameter	Value
# zero-crossings	18
# mean-crossings	16
mean	0.0009
std	0.1379
entropy	2.69

Time dependence - 1D-Convolutional Neural Network





Time dependence - Models with “memory”

Recurrent Neural Network - RNN

- Memory in form of hidden nodes

Temporal Convolutional Network - TCN

- Causal 1D convolution
- Require that points in time only depends on previous points

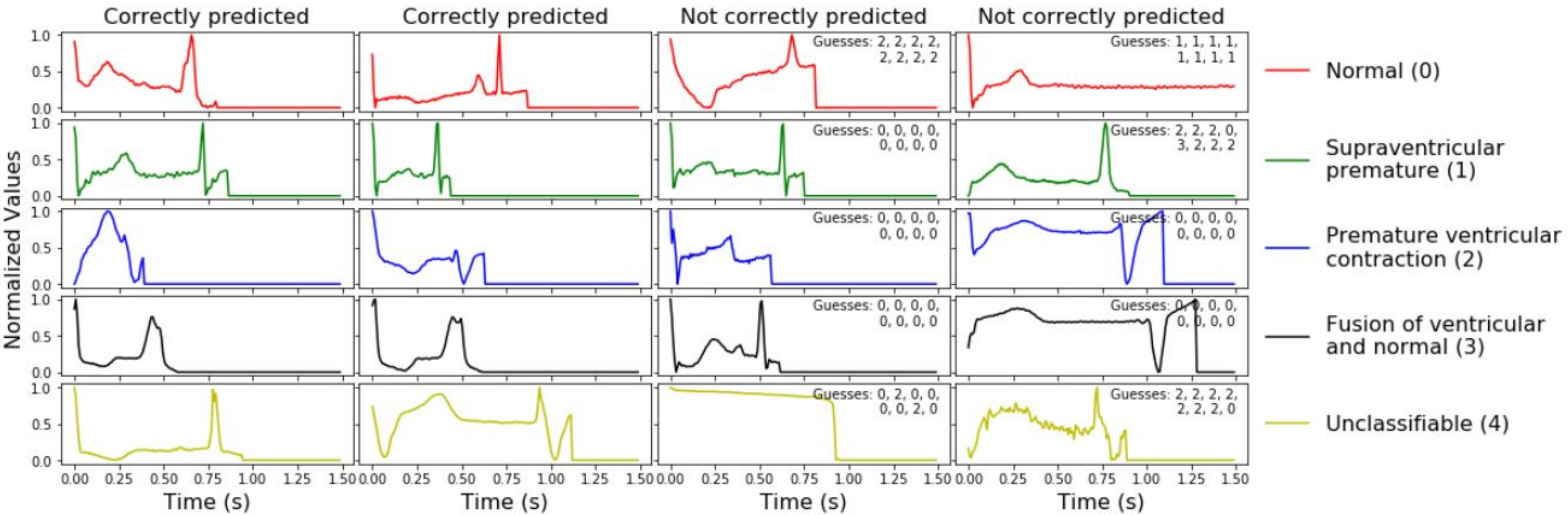
https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0

Machine Learning models - Test data results

Type	# params	Time	Acc.	Prec. (Macro)	Recall (Macro)	F1 (Macro)	Prec. Class 0
K-Nearest Neighbours (K=3)	1	2 s	0.9764	0.9196	0.8470	0.8799	0.9732
SBM	~ 1200	8 m	0.9767	**0.9610**	0.8270	0.8828	0.9920
Neural Network	16517	2 m **	0.9783	0.9192	0.8771	**0.8967**	0.9626
DWT preprocess with SBM	~ 6613	13 m	0.9713	0.9261	0.8124	0.8602	0.9753
One vs. Rest (SBM)	~ 6000	7 m	0.9714	0.9483	0.7755	0.8427	**0.9944**
Recurrent Neural Network	23705	26 m	0.9772	0.9133	0.8450	0.8756	0.9651
1D Convolutional Neural Network	82701	34 m	0.9798	0.9082	**0.8839**	0.8956	0.9574
Temporal Convolutional Network	40230	49 m *	**0.9819**	0.9363	0.8588	0.8930	0.9820
Ensemble (knn3, rnn, tcn, cnn)	-	-	**0.9848**	0.9482	0.8745	**0.9074**	**0.9891**

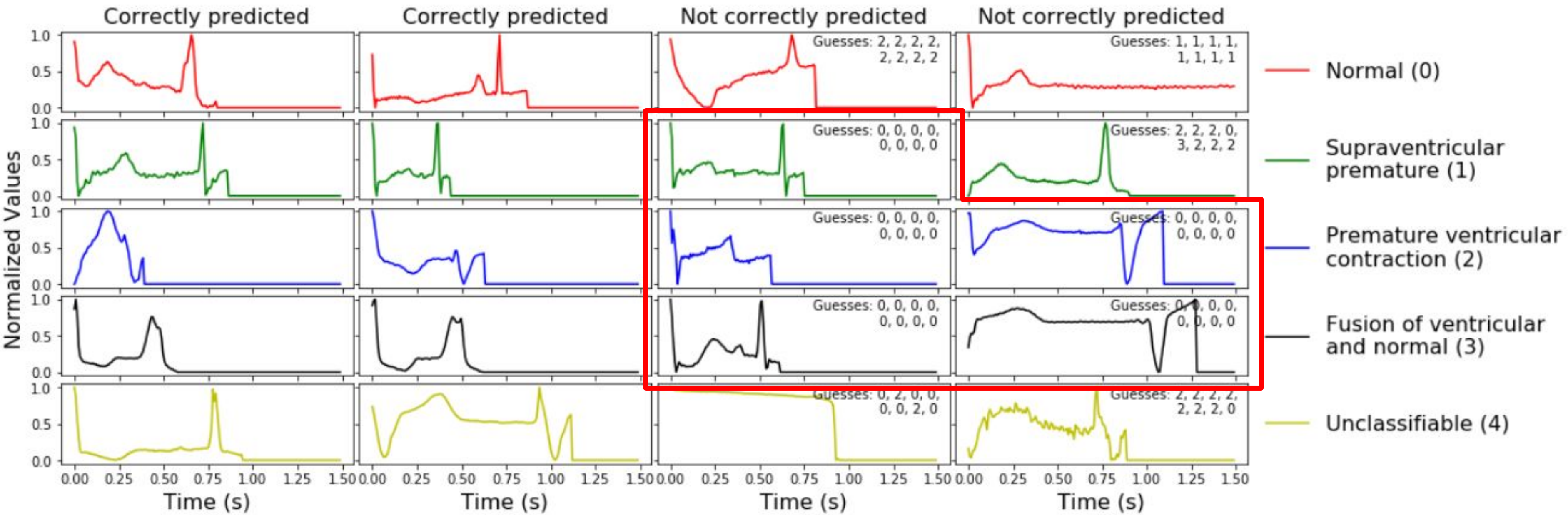
* Calculated using GPU, ** Calculated on another computer

Discussion


Correctly predicted
Correctly predicted
Not correctly predicted
Not correctly predicted
Guesses: 2, 2, 2, 2, 2, 2, 2, 2
Guesses: 1, 1, 1, 1, 1, 1, 1, 1
Guesses: 0, 0, 0, 0, 0, 0, 0, 0
Guesses: 2, 2, 2, 0, 3, 2, 2, 2
Guesses: 0, 0, 0, 0, 0, 0, 0, 0
Guesses: 0, 0, 0, 0, 0, 0, 0, 0
Guesses: 0, 0, 0, 0, 0, 0, 0, 0
Guesses: 0, 0, 0, 0, 0, 0, 0, 0
Guesses: 0, 2, 0, 0, 0, 0, 2, 0
Guesses: 2, 2, 2, 2, 2, 2, 2, 0
Normalized Values
Time (s)
Normal (0)
Supraventricular premature (1)
Premature ventricular contraction (2)
Fusion of ventricular and normal (3)
Unclassifiable (4)

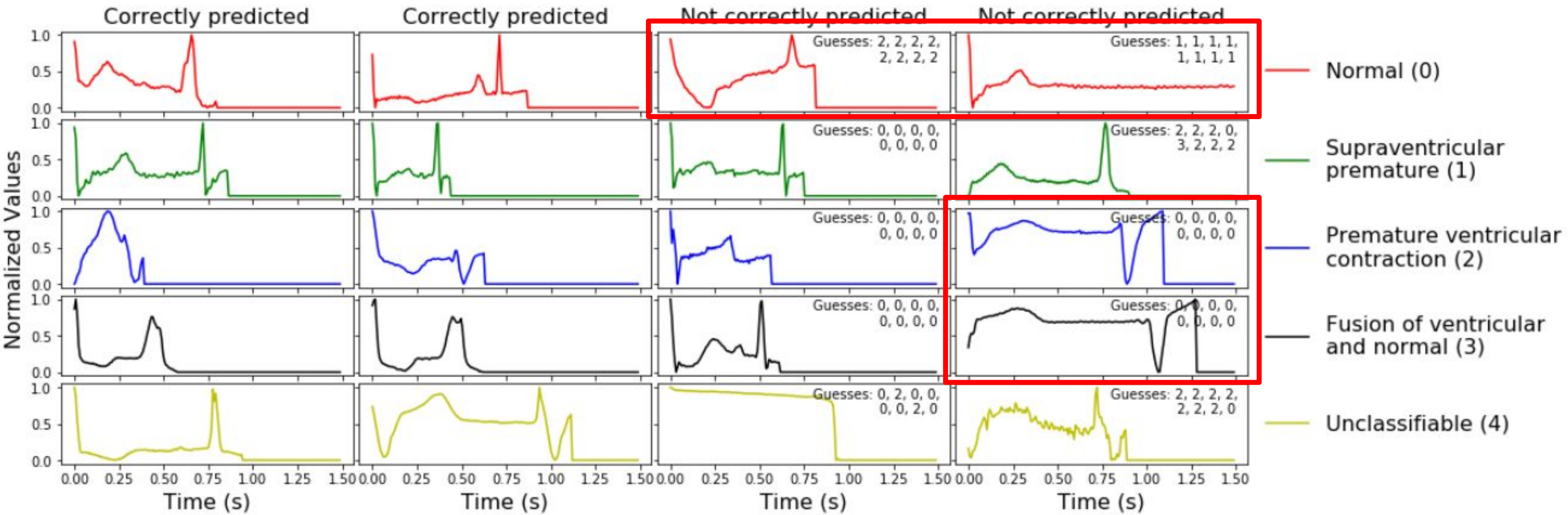
Discussion

Default label zero?



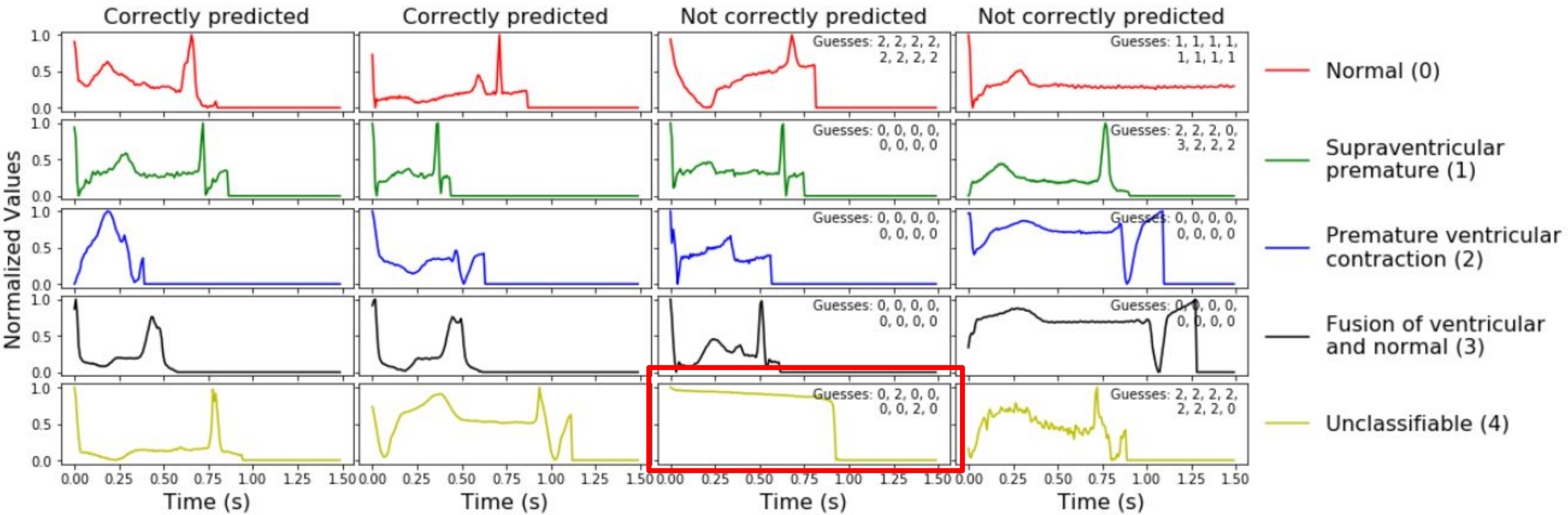
Discussion

Mislabelling?



Discussion

What?

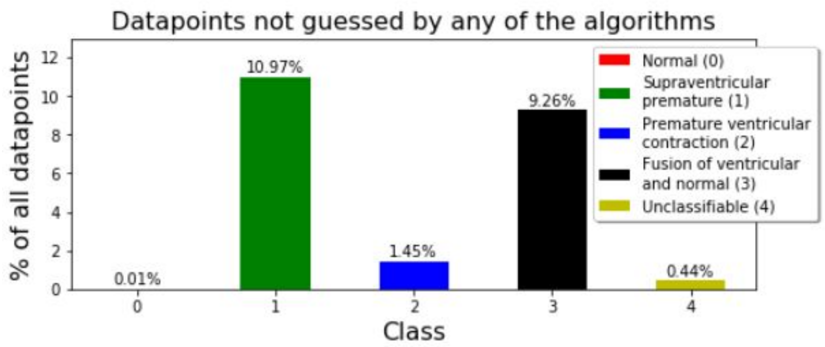


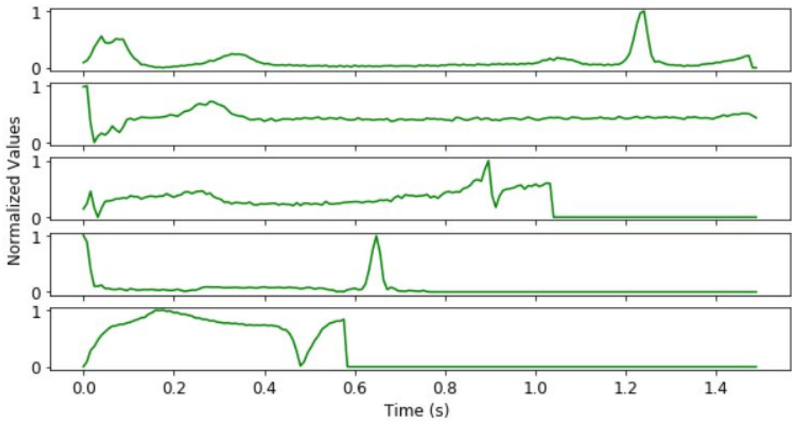
Discussion

- Difficulty in classifying ill represented data

- Classify by subclasses:

Category	Annotation
N	- Normal - Left/Right bundle branch block - Artial secape - 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 - Unclassified

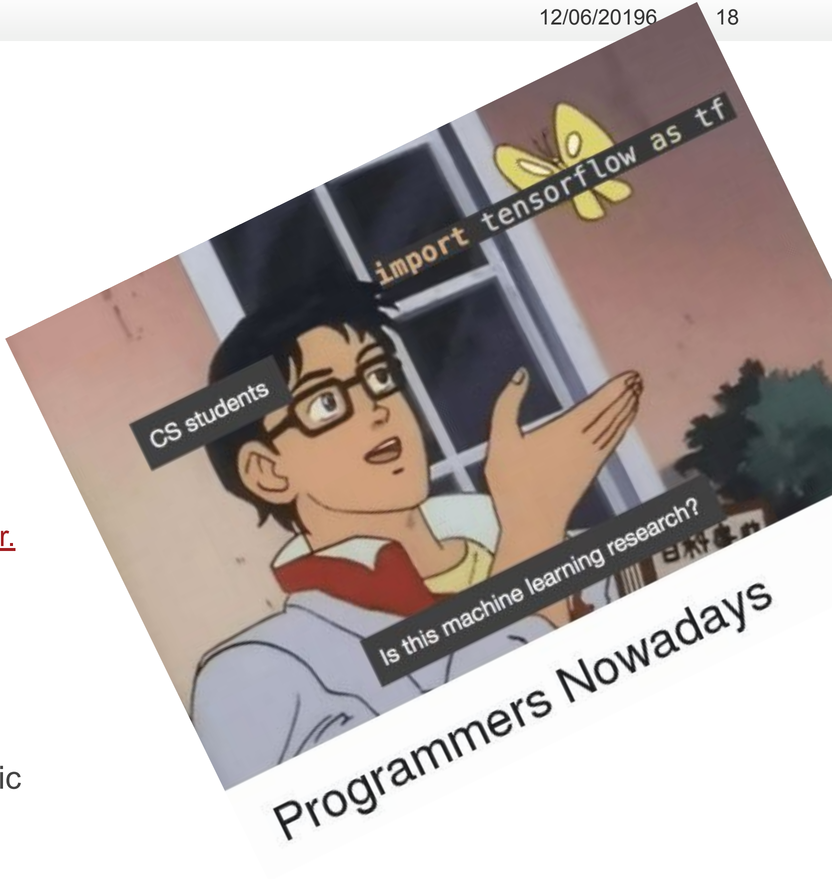




Outlook

- Calculate uncertainties using K-fold.
- Chose a specific metric and promising models.
- Combining DWT with the preprocessed data.
- Do our own preprocessing of data.
- Submit code to Kaggle → Get a job at NASA

Thank you for your attention



References:

- https://www.kaggle.com/shayanfazeli/heartbeat
- ECG Heartbeat Classification: A Deep Transferable Representation arXiv:1805.00794 [cs.CY]
- https://www.physionet.org/physiobank/database/mitdb/
- https://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm

Acknowledgement: Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* **101**(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/content/101/23/e215.full]; 2000 (June 13).

Appendix

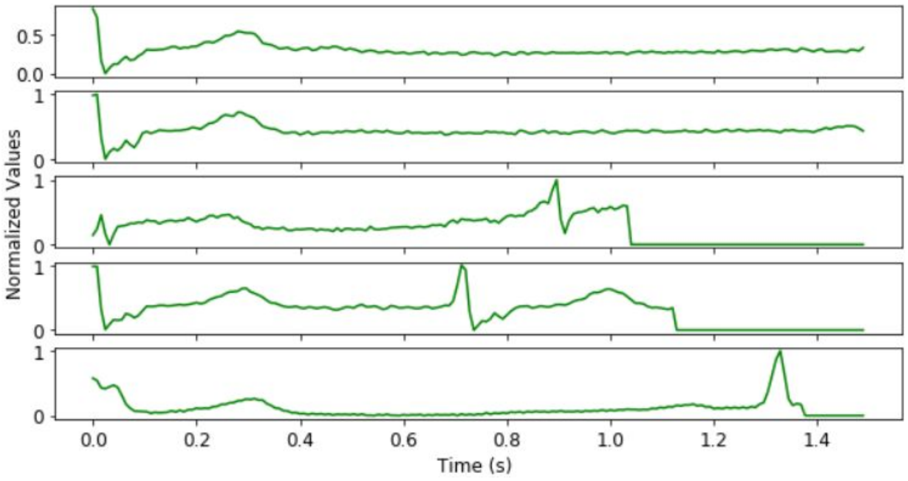
Machine Learning models - Validation data results

Type	# params	Time	Acc.	Prec. (Macro)	Recall (Macro)	F1 (Macro)	Prec. Class 0
K-Nearest Neighbours (K=3)	1	2 s	0.9740	0.92274	0.8549	0.8857	0.9596
SBM	~ 1200	8 m	0.9746	0.9066	0.8045	0.8481	0.9777
Neural Network	16517	2 m **	0.9785	0.9177	**0.8822**	0.8989	0.9565
DWT preprocess with SBM	~ 6613	13 m	0.9729	0.9461	0.8264	0.8768	0.9732
One vs. Rest (SBM)	~ 6000	7 m	0.9779	**0.9496**	0.8383	0.8858	**0.9857**
Recurrent Neural Network	23705	26 m	0.9798	0.9319	0.8673	0.8963	0.9650
1D Convolutional Neural Network	82701	34 m	0.9817	0.9437	0.8460	0.8866	0.9777
Temporal Convolutional Network	40230	49 m *	**0.9832**	0.9456	0.8784	**0.9086**	0.9783
Ensemble (knn3, rnn, tcn, cnn)	-	-	**0.9852**	**0.9640**	0.8782	**0.9158**	0.9856

* Calculated using GPU, ** Calculated on another computer

MIT-BIH Arrhythmia Dataset

Difference in category 1
- Supraventricular Premature



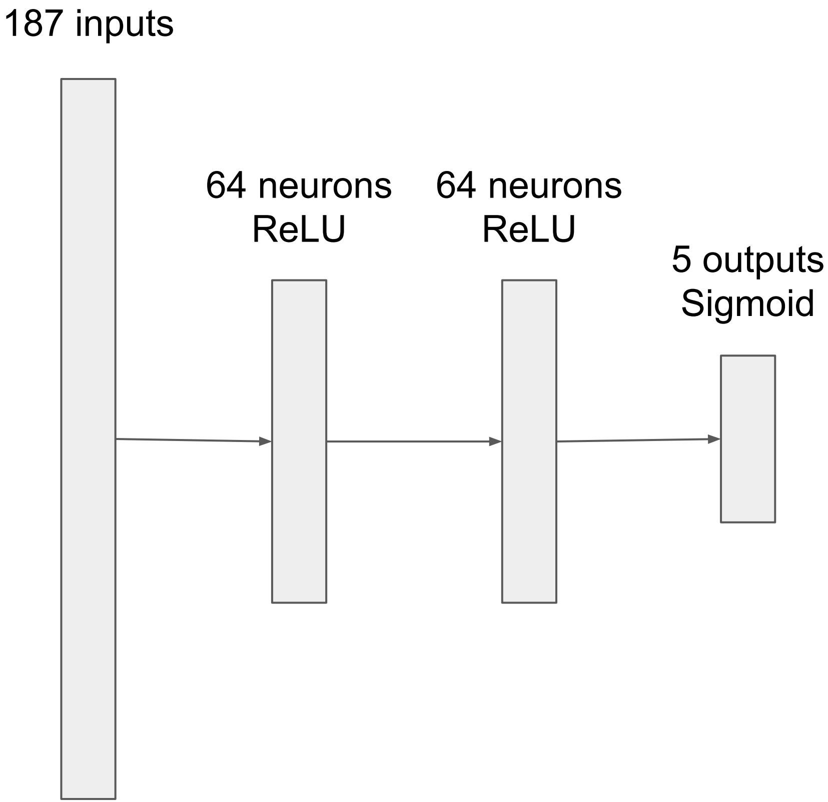
Preprocessing of original data

1) Splitting the continuous ECG signal to $10s$ windows and select a $10s$ window from an ECG signal.
2) Normalizing the amplitude values to the range of between zero and one.
3) Finding the set of all local maximums based on zero-crossings of the first derivative.
4) Finding the set of ECG R-peak candidates by applying a threshold of 0.9 on the normalized value of the local maximums.
5) Finding the median of R-R time intervals as the nominal heartbeat period of that window (T).
6) For each R-peak, selecting a signal part with the length equal to $1.2T$.
7) Padding each selected part with zeros to make its length equal to a predefined fixed length.

Neural Network - Structure

Optimizer: Adam
Loss function: categorical_crossentropy

Batch Size:	30
Number of epochs:	46
Learning rate:	0.0046
Decay:	0.00014

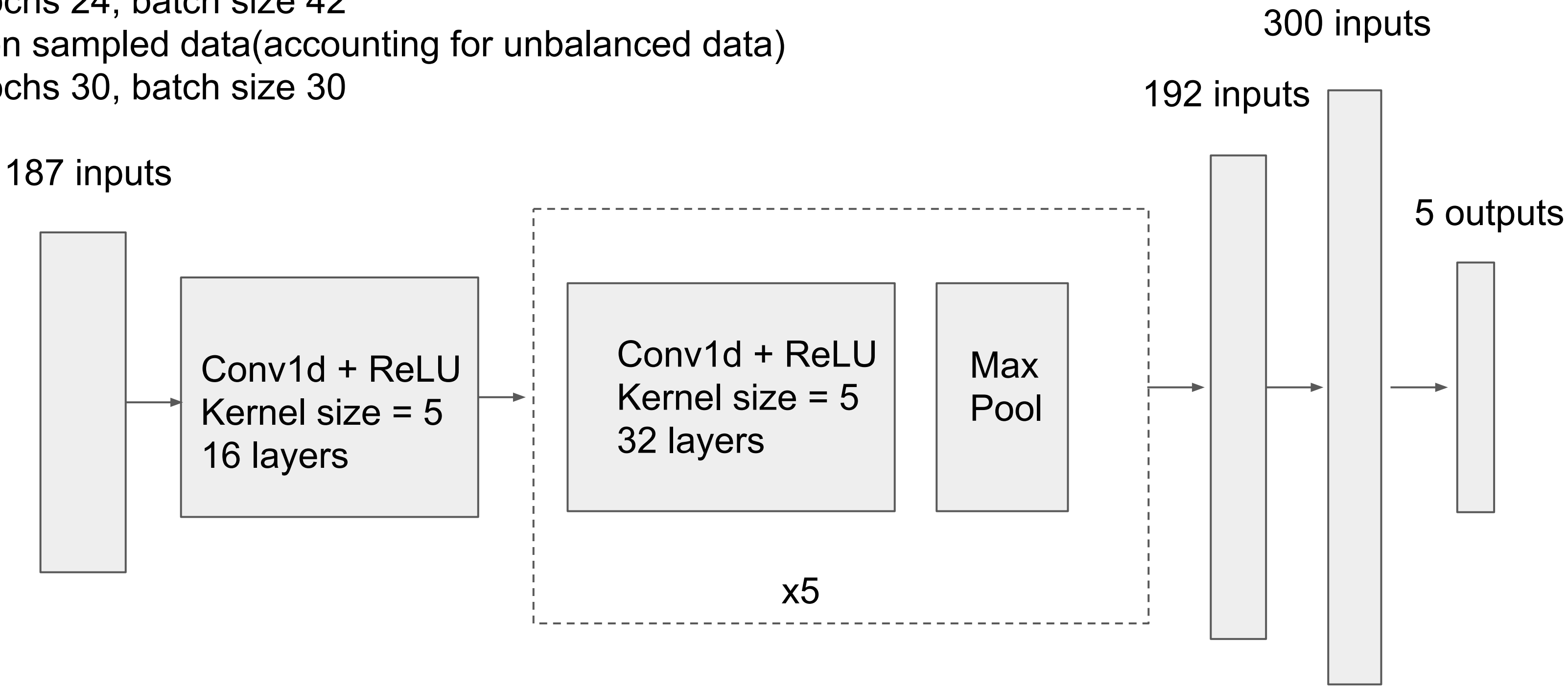


CNN - Structure

1D convolutions

Trained on unsampled data(not accounting for unbalanced data)
- epochs 24, batch size 42

Trained on sampled data(accounting for unbalanced data)
- epochs 30, batch size 30

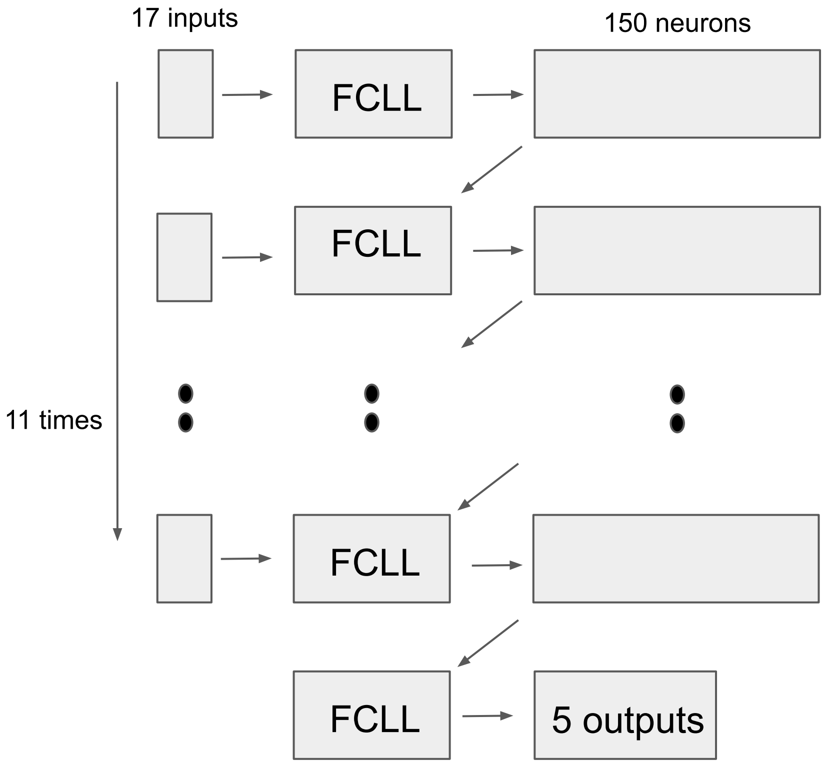


Recurrent Neural Network (RNN) - Structure

Optimizer: Adam
Loss function: nn.CrossEntropyLoss()

Batch Size:	64
Number of epochs:	16
Learning rate:	0.001

At epochs 15 & 25: lr /= 5

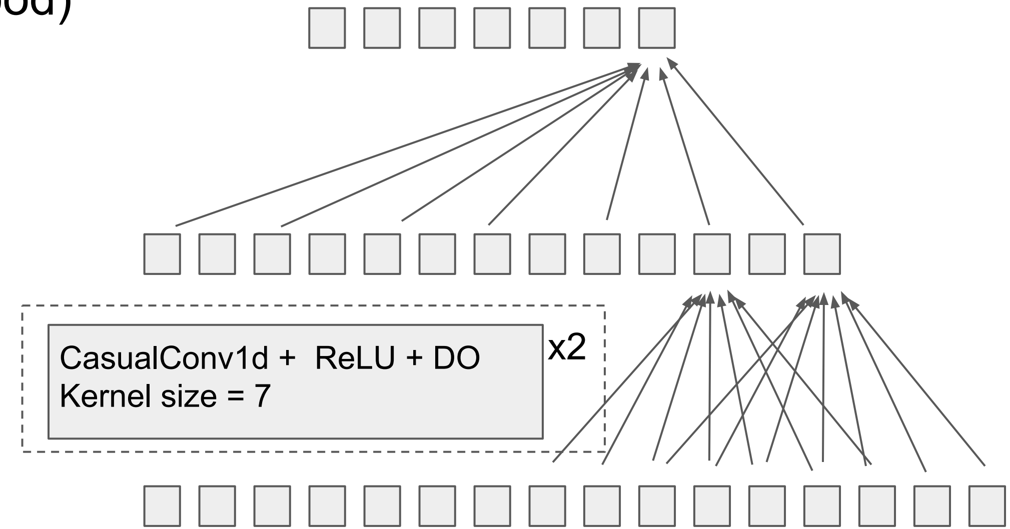


https://medium.com/dair-ai/building-rnns-is-fun-with-pytorch-and-google-colab-3903ea9a3a79

Temporal Convolutional Neural Network (TCN) - Structure

Optimizer: Adam
Loss function: F.nll_loss()
(negative log likelihood)

Batch Size: 64
Number of epochs: 30
Learning rate: 0.002
Every 10 epoch lr /= 10

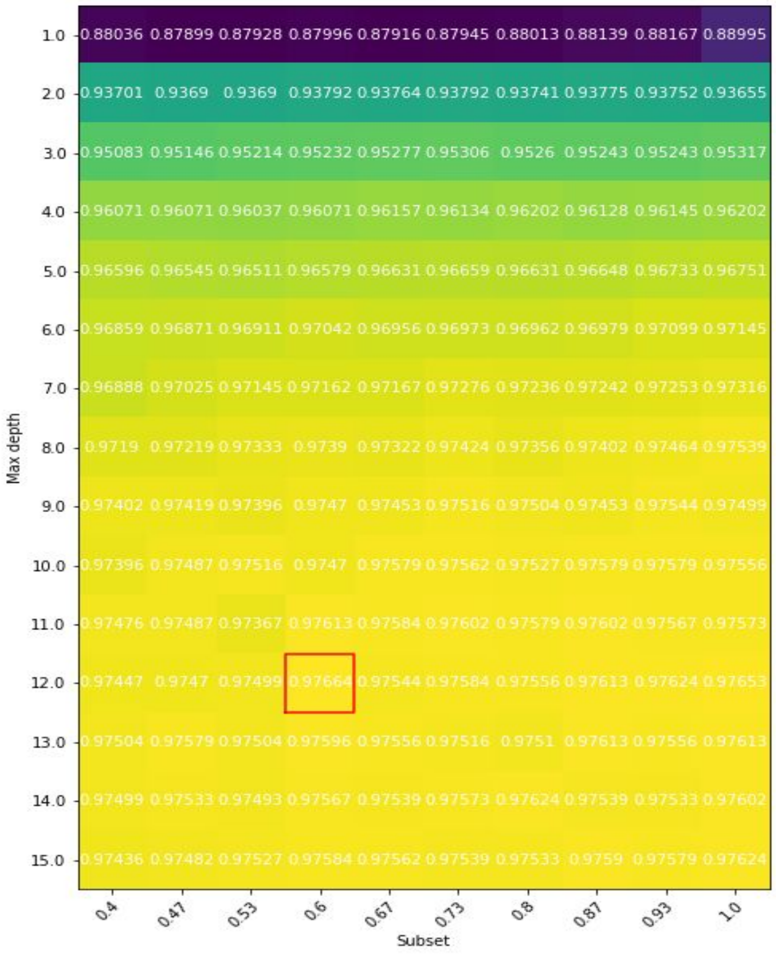


https://github.com/locuslab/TCN
arXiv:1803.01271 [cs.LG]

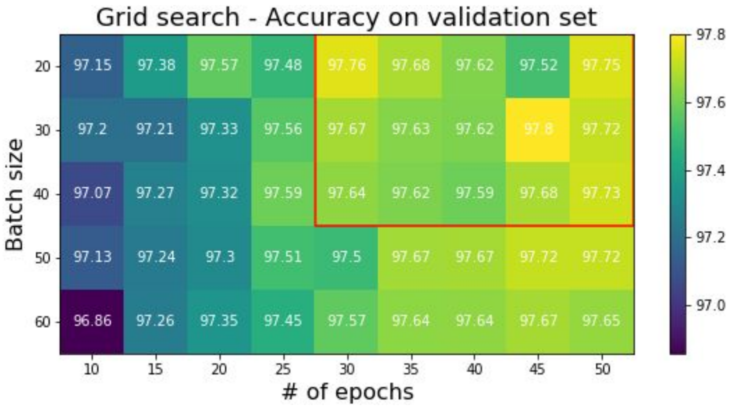
GBM - Hyperparameter optimization

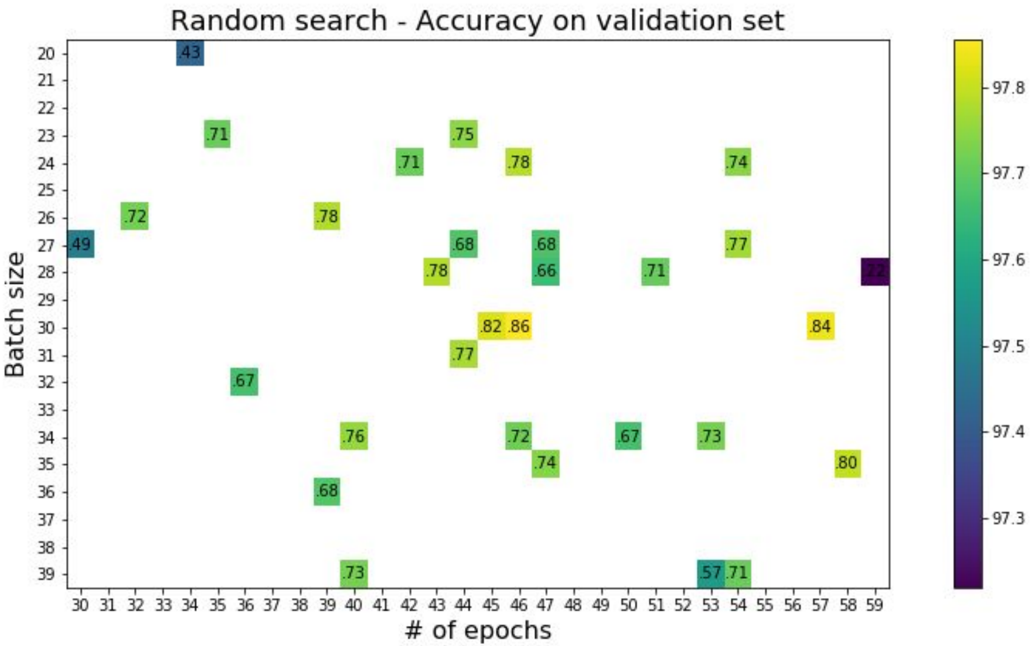
```
GradientBoostingClassifier(learning_rate=0.1, n_estimators=100,
                           max_depth=12, min_samples_split=2,
                           min_samples_leaf=2, subsample=0.6,
                            random_state=42, max_features=12)
```

Subsample < 1 : Stochastic Gradient Boosting
- reduced variance
- increased bias



Neural Network - Hyperparameter Optimization





Neural Network - Hyperparameter Optimization