Machine Learning for Health care
Detection of MSNA bursts using ECG and MSNA signals

Surabhi Sunil
5889453
sunil022@umn.edu
University of Minnesota

Abstract: Detection of bursts in the physiological signal like MSNA is important for understanding autonomic nervous system function, evaluating cardiovascular reflexes, and diagnosing conditions such as hypertension and heart failure. This study uses a machine learning pipeline that is designed for detecting the bursts in physiological signals, using ECG and Integrated MSNA signals (Muscle Sympathetic Nerve Activity). The problem is seen as a binary classification task, where a layer of Convolutional Neural Network (CNN) along with Long Short-Term Memory (LSTM) neural network is used to capture the temporal dependencies in the data. The method incorporates feature engineering, windowing approach and a re-mapping process to ensure precise burst localization within the temporal signal. The model achieves an F1 score of 0.79 across patients.

I. Introduction

Muscle Sympathetic Nerve Activity (MSNA) is a critical physiological marker of autonomic nervous function, reflecting the outflow of sympathetic nerve impulses to the vasculature. It is tightly linked to blood pressure regulation through its influence on total peripheral resistance (TPR). The sympathetic branch of the autonomic nervous system plays a vital role in cardiovascular function, with MSNA being a proxy for measuring TPR and total autonomic activity. Therefore, Burst detection is important for understanding the physiological and pathological mechanisms underlying conditions such as hypertension, heart failure and orthostatic hypotension.

Usually, the MSNA signals are recorded using microbiology, an invasive technique that captures the electrical activity from postganglionic sympathetic nerves, often from the fibular nerve. To generate a neurogram that is suitable for analysis, the signals are processed using amplification, band-pass filtering, rectification, and integration. The bursts of MSNA are characterized by a spike and have tight temporal association (~1.2s) with the ECG signals.

Traditional methods of detecting bursts relies heavily on manual identification, which can be prone to error and time-consuming. Therefore, this study introduces a CNN-LSTM model for burst identification. The task is framed as a binary classification problem, which uses ECG and Integrated MSNA signal to identify bursts with high precision. The model uses advanced feature engineering such as normalization of the MSNA signals, introduction of ECG knowledge into features by shifting them by 1.2 seconds, windowing approach and a re-mapping to localize bursts temporally. The F1 score achieved by this model is 0.79 which out-performs the traditional peak finding algorithm.

II. Methodology

The burst detection is treated as a binary classification task, where each window is classified as either 0, not containing a burst or 1, containing a burst.

a. Data Preprocessing

i. **Signal windowing:** The time series data is segmented to fixed-length windows with the window size of 100 for this particular problem. The target values in each window are determined based on the presence of burst in the window. If there is even one row of the window with burst, then the window is labeled as burst window (1). Otherwise, the window is labeled as non-burst window (0)

ii. **Feature Engineering:** The features used for this model are ECG and Integrated MSNA. A shifted ECG is also incorporated to account for ~1.2s delay between the MSNA burst and corresponding ECG peak. The MSNA signal is the normalized using Butterworth bandpass filter to retain frequencies between 0.1 Hz and 50 Hz to retain relevant physiological components of the signal, while surpassing the high frequency noise. The normalized signal was then scaled to enhance the signal-to-noise ratio.

iii. **Handle Missing values:** NaN values can occur due to signal shifts or edge cases which are replaced with 0 to ensure model compatibility. (the dataset itself had no NaN values).

b. Model Architecture

A CNN network is used as it can find patterns or local features from the data along with a simple LSTM neural network for the binary classification as it is well-structured with time-series data and has the ability to model long-term relationships within sequences. It actively captures the temporal dependencies, and the fully connected layer enables classification into binary outputs.

Components of the model:

1. CNN layer: is used for feature extraction from the signal data, identifying local patterns like spikes in the MSNA signal.

$$y_i^{(l)} = \mathrm{ReLU}\left(\sum_{j=0}^{k-1} w_j x_{i+j} + b\right)$$

Where k is the kernel size, w are the convolutional weights, x is the input and b is the bias.

Max pooling:

$$y_i^{(l)} = \max(x_{i:i+p})$$

P = pooling size

2. LSTM Layer: Captures the sequential dependencies.

Forget gate: determines the information that can be discarded.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input gate: Selects the information to update the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell state update: Updated the internal memory cell

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Output Gate: Determines the output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

F_t, i_t, o_t, and C_t forget input, output and candidate cell gates. W and b are the weights and biases during training. H_t is the hidden state at time t

$$h_t, C_t = \text{LSTM}(x_t, (h_{t-1}, C_{t-1}))$$

3. Fully connected Layer: the hidden state from last time step is passed through fully connected layer to produce:

$$y = W \cdot h_{\text{last}} + b$$

Y: output for classification
W: weights for fully connected layer
b: Bias of fully connected layer

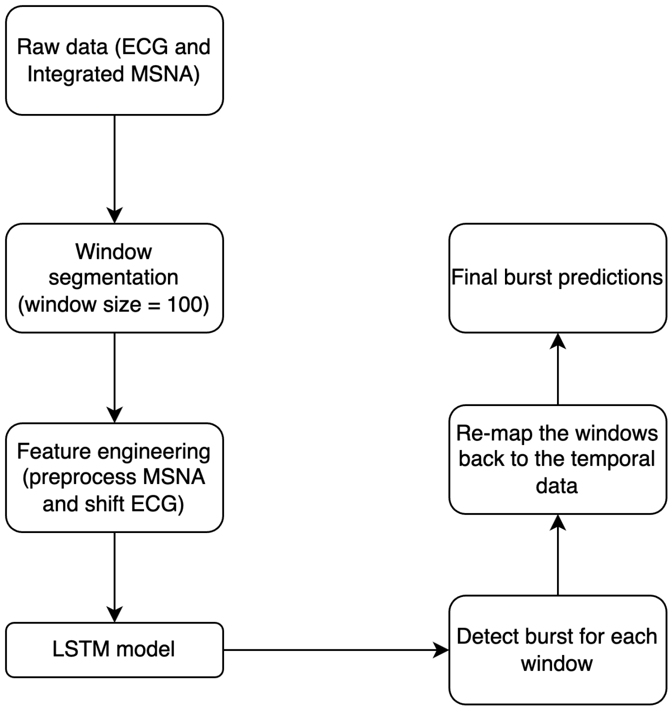
Input and Output: The input to the model is a sequence of features.

$$X \in R^{\text{batch size x sequence length x hidden size}}$$

Finally, a softmax function is applied to compute the class probabilities.

$$p = \text{softmax}(y) = \frac{\exp(y_i)}{\sum_{j=1}^{2} \exp(y_j)}$$

Which means for each sequence of length 100, the output will be 1 value.


Raw data (ECG and Integrated MSNA)
Window segmentation (window size = 100)
Feature engineering (preprocess MSNA and shift ECG)
LSTM model
Detect burst for each window
Re-map the windows back to the temporal data
Final burst predictions

c. Training and Evaluation

A 5 k-fold cross validation is used to access the model performance across files of different patients. Weighted cross entropy is used to address the imbalance between the number of burst and non-burst windows. The Adam optimiser with a learning rate of 0.001 is used for gradient updates. The F1 score that is provided in the metrics.py is used and an out of fold mean, std, min and max f1 scores are reported.

d. Post prediction remapping:

Once the window level predictions are done, a remapping is applied to align the predictions to the time-series test data.

For each window, if a prediction is 0, it indicates the absence of burst in that window. Target variables for all the rows in that window are set to 0. However, if the predicted value is 1, the MSNA value above the threshold within the window is identified as the peak and the rest of the rows are set to 0.

III. Hyper- parameter Tuning

Hyper-parameter tuning was done manually to optimize the performance of the model. Vital parameters such as number of epochs, folds for cross validation, layers in the neural networks, hidden layer sizes were adjusted iteratively. The learning rate was also fine tuned to enable convergence during training. The configuration of 15 epochs, 5 -fold cross validation, 1 CNN and 1 LSTM layer, 64 hidden units, learning rate of 0.001 and sequence length of 0.001 was chosen as was the best for the f1 score.

IV. Results

Prior to considering Neural networks, I used Tree based models such as Random forest and XGBoost which gave a very low accuracy score of 0.23.

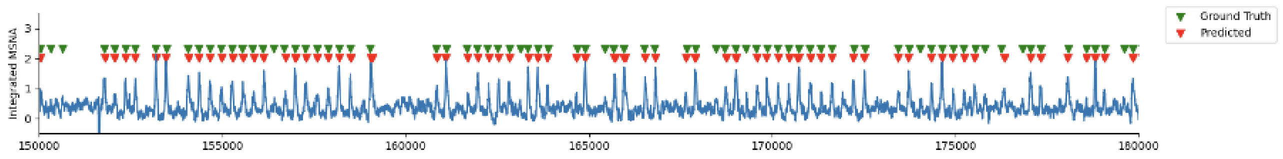
After further analysis and discussions with the TA, I recognized that tree-based models were not well-suited for the nature of my data.

Therefore, I use neural networks.

Three models of Neural Networks were analyzed using different types of hyper parameter tuning.

1. LSTM model - a simple LSTM was used with 1 layer. It produced an F1 score of 0.77
2. CNN - LSTM model - one layer of CNN and 1 layer of LSTM was used. The f1 score reported was 0.79
3. RNN model - a simple RNN model produced a f1 score of 0.66.

F1 score	**Mean**	**Min**	**Max**	**Std.**
LSTM	0.77	0.63	0.88	0.08
CNN - LSTM	0.79	0.72	0.84	0.04
RNN	0.66	0.56	0.9	0.12



CNN - LSTM model that has the best f1 score

V. Conclusion

This paper shows the efficacy of an CNN-LSTM-based machine learning approach for detecting MSNA bursts using ECG and MSNA signals. The model achieves an F1 score of 0.78, out-performing the traditional peak-finding algorithms.

VI. Future improvements

Explore adding more convolutional and LSTM layers to increase the capacity of the model. Add more features like rolling mean, standard deviation and other statistics for Integrated MSNA to add more features for the model to learn from.
Maybe use another model that is more suited for temporal data like GNN, or transformers models.