Effectiveness of Curvature and Signal Derivatives in Fast Curve Segmentation

Gordon Dickers [1], John Rees [1], Tim Bashford [1] , Oluwakemi Ademoye [2]

[1] *School of Applied Computing*, University of Wales Trinity St David, Swansea, UK
Email: gordon.dickers@uwtsd.ac.uk, john.rees@uwtsd.ac.uk, tim.bashford@uwtsd.ac.uk

[2] *School of Computing*, University of Kent, Canterbury, Kent
Email:k.ademoye@kent.ac.uk

Abstract **- Segmentation of a plane curve or times series data is an important step in classification and region localisation. Here we investigate the effectiveness of curvature and derivatives of curves as features for a time series classifier. We evaluate their effect on the accuracy of a long short term memory recurrent neural network used to segment electrocardiogram signals and identify useful techniques to improve run-time efficiency. We compare our results with existing features pre-processors that have been used in the literature and present summary results of their relative speed and accuracy. We find that using curvature, first and second Gaussian derivatives can produce a significant speed up when pre-filtering of a sampled dataset is required. When used in combination, first and second derivatives improve classification accuracy by up to 24% when compared with the un-processed signal and, in terms of speed, they outperform by orders of magnitude the second best classifier's execution time.**

Keywords - curvature, Derivative of Gaussian, Laplacian of Gaussian, Curvature, LSTM, recurrent neural networks

I. INTRODUCTION

Segmentation of a region of interest (ROI) within a time series signal or plane curve is an important step in many modelling and simulation applications, for example in the field of computer vision it has a role in object classification and also object localisation and regression. It has been used in medical applications to identify salient waveform complexes within electro-cardio-gram (ECG) signals and can be used in identifying ROIs within photo-plethysmography (PPG) signals to estimate blood pressure. It plays an important role in speech recognition, stock market forecasting and is also used in fitness tracking applications.

In this paper we explore the relevance of feature preprocessing in a time-series dataset, concentrating on methods and techniques used to improve both the accuracy and runtime efficiency of time-series segmentation and classification. We focus on three features, curvature and the first two derivatives of a signal and identify efficient approaches to calculating these features. We evaluate them against some commonly used techniques and assess their run-time efficiency and effectiveness as feature inputs to a suitable classifier/segmenter, in this case a long short term memory (LSTM) recurrent neural network (RNN).

II. LITERATURE REVIEW

Segmentation can be defined as the process of subdividing an object into its constituent regions or parts and is commonly applied to two dimensional images in the field of computer vision [1] and to univariate or multivariate time series data.

Several approaches have been used to segment plane curves and times series signals. Typically the time-series data, whether uni-variate or multivariate, is first processed to generate salient features before being applied to a classifier or segmentation process. Common features used include statistical and distance based similarity measures that are used to estimate the similarity of the shape of a candidate curve with that of an archetype, either locally on a sub-sequence of the curve, or globally. Cross-correlation [2] is a simple approach that can work well on similar data sets, however it does not perform well when the series to be matched are at different scales and have been stretched or compressed in a non-linear fashion at a local scale.

Dynamic time warping (DTW) [3] attempts to solve this problem. It uses a mapping of two vectors found by minimizing the distance between them using linear programming methods and has been used successfully in time series analysis, for example in speech processing [4], electrocardiogram analysis [5] and other domains [6]. Auto-regressive methods [7] have also been used to segment multivariate time-series signals.

First and second derivatives of time series data and, more generally, plane curves can also act as effective features for segmentation and classification. Lippold [8] observes that the first and, particularly, the second derivatives of a face profile are helpful in identifying regions of interest in an image and is aware of the need to smooth sample contour curves prior to calculating these derivatives. In the medical fields, electrocardiogram (ECG) segmentation applications use the first derivative to segment ECG signals into their respective parts [9].

In other areas the curvature, κ, of a plane curve has been used in classifying profile images, sometimes with a priori knowledge of the landmarks and regions of interest [10]. Curvature has also been used to identify silhouette images from their profiles [11]. It is worthwhile pointing out that the curvature is dependent on both first and second derivatives of the sampled time series.

Frequency domain features are also used in time-series classification and image segmentation. For example, frequencies corresponding to the first 5 peaks in amplitude of the discrete Fourier transform (DFT) have been used as a multi-dimensional feature for classifying human activity [12]. Extracting time-frequency features allows a classifier to use related time and frequency information together. The Fourier synchro-squeezed transform (FSST) [13], for example, has been used to generate features to classify individual samples of ECG signals as part of the segmentation process [14].

Once the original signal or curve has been processed and features generated, they are then input to a classifier or segmentation process. The machine learning literature is replete with methods, support vector machines (SVMs) and decision forests being examples of two. Hidden Markov models (HMM) together with Bayesian methods were popular in the speech recognition community to segment and classify phonemes and other speech segments [15]. More recently recurrent neural networks, in particular the LSTM [16] and its variants [17] have been applied successfully to sequential data and have been responsible, in part, for the impressive advances seen in speech recognition and other time-series domains.

State of the art deep learning classifiers still use feature engineering to produce effective features for hand crafted classifiers, though Fawaz [18] notes that the increasing availability of large time series datasets makes the use of end-to-end deep learning classifiers an attractive option. By end-to-end we refer to the network learning the best features and feature transformations from the raw, annotated dataset. To do this it is not only suitably large and well annotated datasets that are required. Additionally, deeper architectures are needed, together with the requisite resources to train the data over possibly thousands of epochs. This can be a disadvantage when creating the network and can reduce inference speed. Where such resources are not available or a dataset is limited in size or poorly annotated, we believe there is still a place for feature engineering.

Our contribution here is twofold. First, we provide an evaluation of arc length curvature as a feature for segmenting a univariate time-series dataset and compare this with first and second derivatives. By applying existing but underused properties of Gaussian kernels we demonstrate improvements in run-time efficiency and place our results within the context of similar pre-processing activities.

Secondly, we show these features compare favourably with other commonly used pre-processing operations in terms of speed and accuracy of classification for a given classifier and dataset.

III. THEORY

A. Gaussian Derivatives

Typically a sampled time series signal or curve will initially need to be smoothed to remove noise. Here we could make use of the Gaussian function to smooth the signal. In one dimension it is given by,

$$G^{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x)^2}{2\sigma^2}} \tag{1}$$

where σ is the standard deviation and the mean μ is set to zero and so not shown. Smoothing is achieved by convolving the signal with the Gaussian using an appropriate σ. As the Gaussian approaches zero around ±3σ it suffices to convolve the signal with a kernel of size 6σ + 1 sampled on the Gaussian function at integer points. However we observe that in the case of calculating derivatives of a signal it is possible to combine both Gaussian smoothing and finding the signal derivative into a single operation. This is achieved by convolving the signal, f(x), with the derivative of the Gaussian (*DoG*). The *DoG* is,

$$\frac{dG^{\sigma}(x)}{dx} = \frac{x}{-\sigma^2\sqrt{2\pi\sigma^2}} e^{-\frac{(x)^2}{2\sigma^2}} \tag{2}$$

We make use of the associative and commutative properties of the convolution next and note that,

$$\frac{d[G^{\sigma}(x) * f(x)]}{dx} = f(x) * \frac{dG^{\sigma}(x)}{dx} \tag{3}$$

That is, we can calculate the derivative of the smoothed signal by convolving the signal with the derivative of the Gaussian. We achieve this by creating a suitable kernel of the sampled (*DoG*) and convolve this with a subsection of the sampled signal.

Similarly we can find the second derivative of the Gaussian, referred to in the literature as the Laplacian of the Gaussian (LoG), and convolve this with the signal to find its second derivative. The *LoG* is,

$$\frac{dG^{\sigma}(x)}{dx} = \frac{(x^2 - \sigma^2)}{\sigma^4\sqrt{2\pi\sigma^2}} e^{-\frac{(x)^2}{2\sigma^2}} \tag{4}$$

Both the DoG and LoG fall to zero at a slightly slower rate than the Gaussian, so we find that a kernel of size 8σ + 1 avoids any amplification of the signal during convolution. Combining smoothing and differentiation into one operation like this allows us to achieve some efficient speed ups. The runtime complexity is O(nm) where n is the number of samples of the signal of interest and m is the length of the kernel.

B. Curvature

The curvature of a plane curve given by the Cartesian parametric equations x = x(t) and y = y(t) is,

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

Where $x' = \frac{dx}{dt}$ and $y' = \frac{dy}{dt}$ (5)

We can use the *DoG* and *LoG* discussed previously to calculate the curvature equation. If x(t) = constant as happens in a periodically sampled time series then the curvature equation simplifies to,

$$\kappa = \frac{y''}{(1 + y'^2)^{\frac{3}{2}}} \quad (6)$$

This version of the equation is more efficient when working with a univariate, regularly sampled time series. Intuitively and equivalently the curvature is the reciprocal of the radius of the osculating circle to a curve at any point and is also given by κ = 1/radius.

In this paper we z-normalize the calculated curvature.

IV. METHOD

We aim to assess the efficiency and efficacy of features engineered from a sampled curve. First we focus on smoothing a curve, then creating the first and second derivatives of the curve and finally the curvature. We then assess their run-time efficiency and suitability as feature inputs for a classifier, comparing them with commonly used features from the literature. We duplicate the experiment detailed in [14] to generate the comparative results and, additionally, we perform a run-time analysis of the feature processors used. Subsequently we apply our feature processors using the same classifier and finally perform a run-time analysis of these.

We need a suitable dataset to make best use of these features. We use the dataset based upon the publicly available Research Resource for Complex Physiologic Signals QT Database ECG dataset [19]. The derived dataset consists of 210 ECG recordings sampled at 250Hz and segmented by an automated expert system [74]. Recordings are taken from 105 separate subjects and each is approximately a quarter of an hour in length. This dataset has properties useful for analysing the effectiveness of curvature and plane curve derivatives since it consists of areas of sudden change, has varying gradients and is un-smoothed.

Our focus is on the features used, however we need to assess these in conjunction with a suitable classifier. We assess the effectiveness of each feature by calculating the overall accuracy of the classifier, the recall, sensitivity and F1 score, measures that are well understood in the field. We also measure the time taken to engineer the feature from the raw signal. We chose to use the same LSTM RNN classifier for this purpose since this network was used to generate the classifications in the experiment referenced above, is repeatable and its accuracy with this dataset has already been documented using the features derived from the raw signals.

We train it over 10 epochs using mini-batches. We stop training the network after 10 epochs since, for each feature or combination of features used, the testing accuracy has plateaued.

We do not adjust the LSTM RNN architecture except to alter the number of inputs when combining features. The are 200 hidden units; a fully connected output layer with 4 outputs corresponding to the P segment, the QRS complex, the T segment and a neutral, none of the above, classification; and a Softmax layer. It is trained with an Adam optimiser and a minibatch size of 45.

The features we assess are:-
Raw ECG signal,
Band pass filtered signal,
Normalized, curvature of Gaussian filtered signal, σ = 1,
First order derivative of Gaussian filtered signal,
Second order derivative of Gaussian filtered signal.
Curvature and first derivative,
Curvature and second derivative,
First and second order derivatives,
Curvature, first and second derivatives,
FSST of raw ECG signal.

For each feature or feature combination we train and test the LSTM RNN using a 70:30 train:test dataset ratio. The entire dataset comprises of approximately 46 million samples.

V. RESULTS AND DISCUSSION

A. Accuracy

The section details the results when using a feature or feature combination and evaluates the performance using precision, recall, F1 score and overall accuracy.

The accuracy of the Raw ECG signal forms the benchmark against which others are compared. Table I summarizes the accuracy of the LSTM network with the raw ECG signal as input.

TABLE I. EVALUATION OF LSTM NETWORK WITH RAW ECG SIGNAL AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	39.26	74.56	51.44
QRS	60.76	79.75	68.97
T	57.77	78.68	66.62
N/A	86.97	65.75	74.89
Overall Accuracy	**70.30%**		

The Bandpass filter used here is an IIR bandpass elliptic filter with 60dB roll-off, 0.1dB ripple, with a pass band between 0.5Hz and 40Hz. The sample rate of the filtered signal is 250Hz. Table II shows an improvement in each class F1 score. This is probably due to the removal of base line, low frequency movement due to breathing and a reduction in sampled noise.

TABLE II. EVALUATION OF LSTM NETWORK WITH IIR BANDPASS FILTERED ECG AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	50.46	65.99	57.19
QRS	72.74	77.50	75.04
T	75.30	79.60	77.40
N/A	80.67	73.27	76.79
Overall Accuracy	**74.81%**		

The normalized curvature feature is calculated directly from the raw ECG signal data. σ values of 1, 2 and 3 were used. A σ of 1 gave best results and is shown in table III, improving upon the bandpass filtered signal. Curvature is calculated using a kernel of length 19 compared with the bandpass filter that has 46 coefficients. Note that the curvature calculation also includes an implicit filtering operation from the use of the *LoG* and the *DoG*.

TABLE III. EVALUATION OF LSTM NETWORK WITH Z-NORMALIZED CURVATURE AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	53.84	72.22	61.69
QRS	71.65	76.18	73.85
T	73.88	82.19	77.81
N/A	85.01	75.22	79.82
Overall Accuracy	**76.75%**		

Since the 1st derivative has been used in previous work as part of a hand-crafted expert system used to segment ECG signals, then the *DoG* feature was expected to perform well. It achieved a surprisingly good accuracy of 85.28%. The full results are shown in table IV. A significant jump from the baseline ECG raw signal accuracy of 70.3% and, also, it improves upon the curvature feature. The *DoG* both filters and calculates the first derivative in one pass. Here, setting σ = 2 gave the best accuracy.

TABLE IV. EVALUATION OF LSTM NETWORK WITH FIRST DERIVATIVE (*DoG*) AS INPUT FEATURE (σ = 2)

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	74.15	83.31	78.46
QRS	86.27	92.61	89.33
T	82.63	84.88	83.74
N/A	88.93	83.91	86.34
Overall Accuracy	**85.28%**		

Interestingly, the second derivative (*LoG*) feature performs equally as well as the first derivative (*DoG*) in terms of its overall accuracy as detailed in table V. Once again the *LoG* both filters and calculates the first derivative in one pass.

TABLE V. EVALUATION OF LSTM NETWORK WITH SECOND DERIVATIVE (LoG) AS INPUT FEATURE (σ = 2)

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	78.20	82.71	80.39
QRS	88.06	93.28	90.60
T	80.24	83.92	82.04
N/A	88.62	84.17	86.34
Overall Accuracy	**85.20%**		

Using the *LoG* and the *DoG* as a 2-dimensional input feature to the network produces the overall best result (table VI), improving upon the FSSTs overall accuracy by 2% (table VIII). This combination of features also outperforms the FSST feature vector in three out of the four, per class F1 scores.

TABLE VI. EVALUATION OF LSTM NETWORK WITH FIRST DERIVATIVE (*DoG*) AND SECOND DERIVATIVE (*LoG*) AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	79.23	84.96	81.99
QRS	89.99	94.01	91.95
T	85.24	85.02	85.13
N/A	89.19	86.87	88.01
Overall Accuracy	**87.18%**		

The FSST improves on the P class F1 score, alone, by 0.69%. The LoG and the DoG feature also marginally outperforms the network trained with the curvature combined with the first and second derivatives (see table VII).

TABLE VII. EVALUATION OF LSTM NETWORK WITH FIRST DERIVATIVE (DoG), SECOND DERIVATIVE LoG), AND CURVATURE AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	76.78	85.30	80.81
QRS	89.38	94.14	91.70
T	84.58	85.24	84.90
N/A	89.83	86.21	87.98
Overall Accuracy	**86.96%**		

TABLE VIII. EVALUATION OF LSTM NETWORK WITH 40 DIMENSIONAL SFFT VECTOR AS INPUT FEATURE

ECG Class	Recall (%)	Precision (%)	F1 Score (%)
P	82.21	83.16	82.68
QRS	90.45	91.90	91.17
T	82.09	84.43	83.25
N/A	86.57	84.67	85.06
Overall Accuracy	85.48%		

Figure 1 summarises the overall accuracy and macro F1-score of each feature used in the classification and segmentation process. This shows performance increasing significantly when *DoG* and *LoG* are used as features.

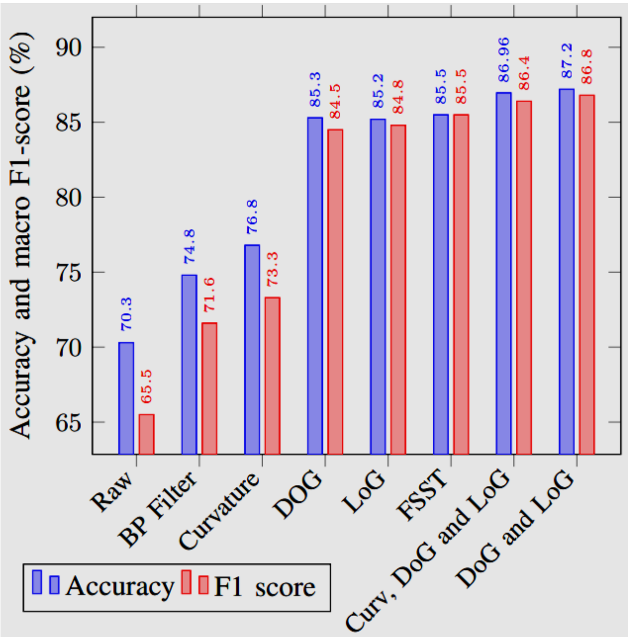

Figure 1. Overall classification accuracy and macro F1-score

B. Runtime Results and Comparisons

The focus of this section is the execution time of the preprocessing algorithms used to generate the input features for the LSTM network. The training times for the network are presented here also. For these timing tests σ was set to 3 to provide a worst case scenario. Both the first and second derivative's kernel size was set to 27 (cutoff = 4∗σ+1, kernel length = cutoff ∗2 + 1), there were 48 filter coefficients and the datasets used were of size 112, 000, 56, 000 and 28, 000 samples. The results of these tests are shown in table VI. Tests were run 1, 000 times and the average taken.

TABLE IX. ALGORITHM EXECUTION TIMES TO PROCESS DATASET OF SIZE N.

Dataset Size, N	112,000	56,000	28,000
1^{st} Derivative (DoG)	188.4μs	103.6μs	57.8μs
2^{nd} Derivative (LoG)	186.0μs	100.7μs	57.1μs
Filter	2.12ms	1.08ms	559.8μs
Curvature	3.72ms	1.92ms	973.6μs
FSST	1.37s	684.7ms	340.9ms

From the table and the discussion of these algorithms the time complexity of the 1st and 2nd derivative features is *O(nm)* where *n* is the size of the dataset and m is the kernel size which is a constant in the tests. The curvature algorithm calculates both the first and second derivatives and uses both square roots and 2nd and 3rd powers in the calculation which accounts for a slower run-time. The timings show an approximate complexity of *O(n)* for the derivatives and curvature.

C. Estimating the Effectiveness of the Feature Pre-Processing

Choosing a feature engineering algorithm for fast and accurate classification is often a trade-off. The fastest algorithm is not the most accurate and vice versa. As a guide, here we summarise the timing and overall accuracy results as a plot in Figure 2.

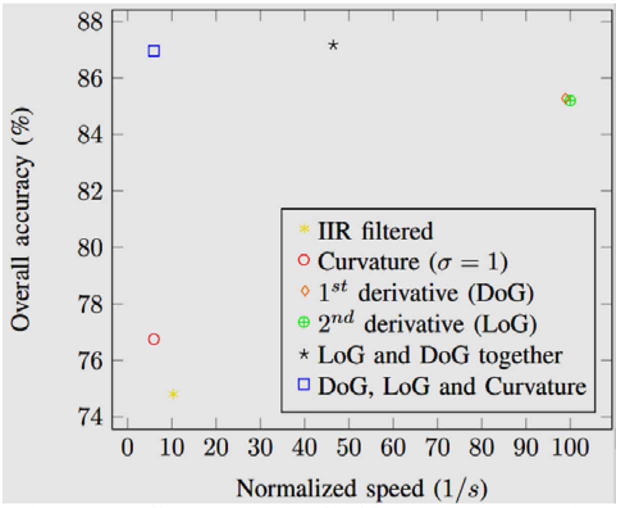

Figure 2. Overall accuracy vs normalized feature processor execution speed.

The top right of the plot is where we would find the ideal feature pre-processor. In general, the higher the point is located, the better the accuracy and the further to the right the faster the algorithm runs. To achieve this we plot accuracy against the reciprocal of the run-time, which we have normalised.

Note the FSST algorithm is not shown as its runtime is several orders of magnitude greater than the other algorithms.

VI. CONCLUSION

The best segmentation results are achieved using the derivatives of ECG signals as inputs to the classifier. Using both the first and second derivatives as input features together in an LSTM RNN segmenter-classifier produces the best classification accuracy results of all and outperforms, by orders of magnitude, the second-best classifiers execution time.

The choice of classifier would change the results and no-doubt, adjusting the architecture and tuning the LSTM RNN attributes will also improve the accuracy of the network for all the pre-processed features. Bear in mind, the purpose here is to assess the suitability of the various features as inputs to a given classifier.

Additionally, this study is based on just one existing dataset. Consequently, more work using a range of datasets is needed to assess the effectiveness of curvature, DoG and LoG derivatives used in time series segmentation.

REFERENCES

[1] R. E. Woods and R. C. Gonzalez, Digital Image Processing, 4th ed. Pearson, 2017.

[2] S. Chandaka, A. Chatterjee, and S. Munshi, "Cross-correlation aided support vector machine classifier for classification of eeg signals," Expert Systems with Applications, vol. 36, no. 2, Part 1, pp. 1329–1336, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417407005696

[3] H. Sakoe, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," Tech. Rep. 1, 1978. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1163055/

[4] C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 6, pp. 623–635, 1980.

[5] B. Huang and W. Kinsner, "ECG frame classification using dynamic time warping," in Canadian Conference on Electrical and Computer Engineering, vol. 2, 2002, pp. 1105–1110.

[6] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, 2005, pp. 506–510.

[7] A. M. Khan, Y. K. Lee, and T.-S. Kim, "Accelerometer signalbased human activity recognition using augmented autoregressive model coefficients and artificial neural nets," in 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 5172–5175.

[8] C. Lippold, X. Liu, K. Wangdo, B. Drerup, K. Schreiber, C. Kirschneck, T. Moiseenko, and G. Danesh, "Facial landmark localization by curvature maps and profile analysis," Head and Face Medicine, vol. 10, no. 1, dec 2014.

[9] P. Laguna, R. Jane, and P. Caminal, "Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database," Computers and Biomedical Research, vol. 27, no. 1, pp.45–60, 1994.

[10] M. Pantic, I. Patras, and L. Rothkruntz, "Facial gensture recognition in face profile image sequences," Proceedings - 2002 IEEE International Conference on Multimedia and Expo, ICME 2002, vol. 1, no. March, pp. 37–40, 2002.

[11] T. Adamek and N. E. O'connor, "A Multiscale Representation Method for Nonrigid Shapes With a Single Closed Contour," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 14, no. 5, 2004.

[12] K. Altun, B. Barshan, and O. Tunel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," Pattern Recognition, vol. 43, no. 10, pp. 3605–3620, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320310001950

[13] F. Auger, P. Flandrin, Y.-T. Lin, S. McLaughlin, S. Meignen, T. Oberlin, and H.-T. Wu, "Time-frequency reassignment and synchrosqueezing: An overview," IEEE Signal Processing Magazine, vol. 30, no. 6, pp. 32–41, 2013.

[14] "Waveform Segmentation Using Deep Learning - MATLAB & Simulink - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/signal/ug/waveformsegmentation-using-deep-learning.html

[15] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[17] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares,¨ H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. Association for Computational Linguistics (ACL), jun 2014, pp. 1724–1734. [Online]. Available: https://arxiv.org/abs/1406.1078v3

[18] H. I. Fawaz, ·. G. Forestier, J. Weber, ·. Lhassane Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, pp. 917–963, 2019. [Online]. Available: https://doi.org/10.1007/s10618-019-00619-1

[19] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." Circulation, vol. 101, no. 23, jun 2000. [Online]. Available: http://www.physionet.org.