Multivariate time-series classification with Deep Learning and Wavelet Transform

A Preprint

Anıl Özdemir
Department of Computer Science
Sabancı University
Istanbul, TR
aozdemir@sabanciuniv.edu

June 20, 2020

1 Introduction

Time series modeling has been always a challenging problem that pushes researchers to find better models. The main purpose of time series modeling is to collect and observe past observations to estimate future conditions. Time series modeling has attracted many researchers in the last few decades since the importance in various fields such as economics, business, finance, science, and engineering [1]. Proposed methods for time series modeling could be classified into two types, these are time-domain methods and frequency-domain methods [2]. In time-domain methods, algorithms analyze the correlations among the data points in time series which considered a sequence of ordered time points. On the other hand, frequency-domain methods benefit from the transformation methods such as Fourier transforms FT) [3], Short Time Fourier Transform [4] in order to transform time-series into a frequency-spectrum, which could be used as features to statistical, machine learning and deep learning algorithms [5].Recently, a large number of methods and algorithms have been introduced to time series analysis starting from traditional models including Auto-Regressive Integrated Moving Average (ARIMA) [6], support vector machines [7] and deep learning models including Recurrent Neural Networks (RNN)[8], LSTMs[9] and CNNs [10, 11]. However, most of these models under the category of time-domain methods without benefiting frequency information of a time series.

According to the literature, Fourier transform (FT) has long been a standard way to calculate the frequency-domain representation of a signal. However, FT assumes the underlying signal to be stationary in the time-domain which is not the case in real-life scenarios most of the time. This causes loss of frequency information that varies with time as discussed in [12]. Short-Time Fourier Transform (STFT) [13] was proposed in the literature to overcome this shortcoming of FT. STFT uses the idea of fixed time-window and iterates this window over time-domain and applies FT. In this way, it will capture the time information. However, the major limitation of STFT is assuming the signal in each time window is stationary with respect to time, similar to the FT. To overcome the limitations of FT and STFT, Wavelet Transform [14], which is a well-known method for capturing features of time series both in time and frequency domains is proposed. The wavelet transform analyzes the signal at different frequencies with different scales using multi-scale analysis also named as multi-resolution analysis. This type of analysis partitions the time-frequency plane in an adaptive manner by using short time windows at high frequencies and long time windows at low frequencies. This results both time and frequency resolutions to vary in the time-frequency domain.

In this work, we tried to emphasize the importance of two different topics. First, we examined the performance of machine learning and deep learning models on the UCI-HAR dataset without applying pre-processing on the data to solve the multi-class classification problem. In order to examine this, we applied 5 different machine learning methods including K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Gradient Boosting Machine (GB) and Support Vector Machine(SVM) and 3 different deep learning methods including Convolutional Neural Network (CNN), Long-Short Term Memory (LSTM) and CNN-LSTM Network on the UCI-HAR dataset. The conclusion from the study was that deep learning methods overperformed machine learning methods in terms of both accuracy and macro f1 score. The second issue was to measure whether feature extraction techniques such as Wavelet Decomposition contributes to the performance of deep learning models. For this purpose, we applied Wavelet transform to the data and tried the CNN and CNN-LSTM models, in order to compare this, we tried the LSTM model on the raw data. In other words, we solved the problem on both frequency and time domain and compared the results. As a result, we observed the obvious effect of wavelet transform on the performance and support our hypothesis.

2 Task Dataset Description

In this project, the Human Activity Recognition Dataset (UCI-HAR) [15] is used. (UCI-HAR) the dataset created from an experiment carried out with a group of 30 volunteers ranging in age from 19 to 48 years. Each person performed the following six activities while wearing a smartphone on the waist:

- Walking
- Walking upstairs
- Walking downstairs
- Sitting
- Standing
- Laying

With the embedded accelerometer and the gyroscope 3-axial linear acceleration and 3-axial angular velocity are captured at a constant rate of 50 Hz. Then, the signals were pre-processed by applying noise filters to be sampled in fixed-width sliding windows of 2.56 sec and an overlap rate of 0.5 which corresponds to 128 readings per window. The sensor acceleration signal, which consists of gravitational and body motion components separated by Butterworth low-pass filter into body acceleration and gravity. From the time and frequency domain, a vector of features achieved for each window. In conclusion, for each record the dataset features:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

The dataset has been randomly partitioned where 21 volunteers were selected for generating train data while 9 volunteers creating test data. Additional notes for the dataset is noted as followed:

- Features are normalized and bounded within [-1,1].
- The units used for the accelerations (total and body) are 'g's (gravity of earth -> 9.80665 m/seg2).
- The gyroscope units are rad/seg.

3 Experiment

During the work, we wanted to emphasize two different comparisons throughout the project. The first is whether machine learning methods or deep learning methods work better in solving our problem. The second is to measure whether the wavelet transform contributes to performance. Thus, we present the different architectures, together with a comprehensive description of the designs of classifiers

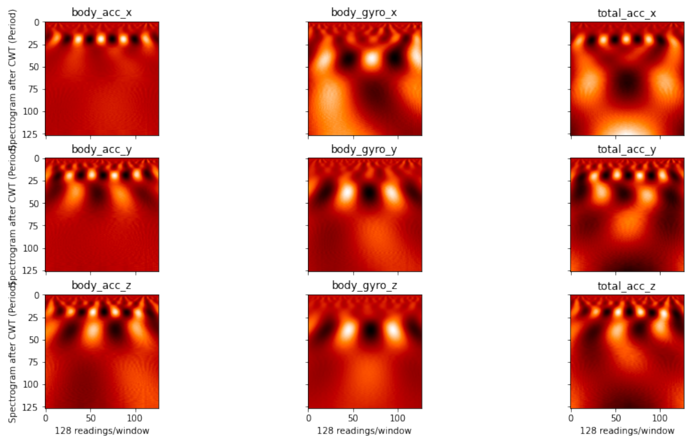
3.1 Designing Input Types and Feature Extraction

Throughout the project, we followed two different ways when preparing the features, these features are in the time domain and frequency domain.

Time Domain - Time-domain shows how a signal changes over time for a given instance. For a given instance, there is an vector of (128,9) unit. 9 corresponds number of component and 128 corresponds time point (see [2]). We used time-domain inputs in machine learning models in [3.2] and LSTM model in [3.3].

Frequecny Domain - In this work, we leveraged from the continuous Wavelet transform technique to obtain features from the time domain, then solve multi-class classification problem by using these attributes as inputs in CNN [3.3] and CNN-LSTM [3.3]. In the literature, examples of this include obtaining time-frequency representation called scalogram by using Wavelet Decomposition and use them for the input in the CNN's to capture localized time-frequency features and latent feature interactions from the scalogram in [12]. As discussed in [2], the signals in Human Activity Recognition dataset have 9 different components. This is why we applied continuous wavelet transform 9 times for each signal. As a result of this transformation, we obtained 9 different scalograms per signal. Scalogram is a visual representation of a wavelet transform, having axes for time, scale, and coefficient value. As a result, we obtained 3 rank tensor of (127,127,9) for 9 components. For designing the input, we placed the nine scalograms (see figure 1) on top of each other and create one single image with nine channels.

Figure 1: Example of 9 scalogram corresponds 9 component for single instance.



3.2 Machine Learning Models

All models have the following procedures:

1. For each Machine Learning model, corresponding hyper-parameters are determined by using a grid search with cross-validation. The hyper-parameters that are fine-tuned are listed as follows:
    - KNN: Number of neighbors
    - Decision Tree: Criterion, the minimum number of samples required to be at a leaf node
    - Random Forest: Number of trees in the forest, criterion, the number of features to consider when looking for the best split
    - SVM: Kernel type, regularization parameter, kernel coefficient
    - Gradient Boosting Machine: Learning rate, the number of boosting stages to perform, the number of features to consider when looking for the best split
2. For each set of parameters, the validation scores are recorded for each fold, and the best set of parameters according to the mean validation score is selected.
3. With the best set of parameters the model is built.
4. The performances of the algorithms are evaluated with respect to test Macro-F1 scores

3.3 Deep Learning Models

All models have the following procedures:

- Hyper-parameters including learning rate, epoch, batch-size, learning rate determined by manually tweaking.
- Validation Loss has been monitored and training is stopped at the patience level of 7.
- For optimizing the weights in the models, we used Adam optimizer that published recently by [16].
- Categorical cross-entropy loss is used as a loss function.
- Softmax is used as activation in the last fully connected layer.

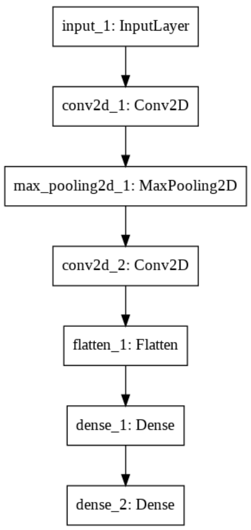


Figure 2: Architecture of CNN model used in project.

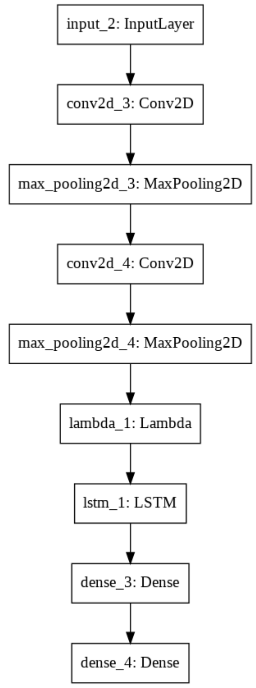


Figure 3: Architecture of CNN-LSTM model used in project.

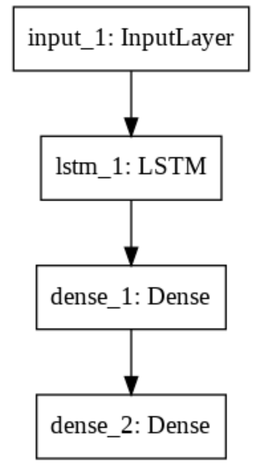


Figure 4: Architecture of LSTM model used in project.

LSTM Network - Forward pass of LSTM model is depicted in Figure 3.3.Our model contains a single LSTM layer that has 64 units and a fully connected layer that has 128 input and 2 output units. Input for an LSTM network is a vector of (128,9) in the time domain. LSTM approaches the 128 as a time step and 9 as a feature.

CNN Network -. The forward pass of the CNN model is depicted in Figure 3.3 and consists of two 2-D convolutional layers, two max-pooling layers, and a fully connected layer. Input for a CNN model is a rank 3 tensor of size (127,127,9) and this tensor going to be input for remaining layers. For the first and second convolutional layers, 32 and 64 filters with sizes (5,5) are used respectively. After both convolution layers, the max-pooling layer with the receptive field of (2,2) was placed. The output from the last max-pooling layer is flattened and sent to the dense layer that has 512 input and 6 output units.

CNN-LSTM Network - Using convolutional layers and long short-term memories in the same network is a common concept in the literature [17]. As stated by [18], while CNNs focus on learning local features, LSTMs are trying to learn long term dependencies from the inputs. This is why they provide complementary utility for most of the tasks. Input for a CNN model is a rank 3 tensor of size (127,127,9) and this tensor going to be input for remaining layers. For the first and second convolutional layers, 32 and 64 filters with sizes (5,5) and Relu as an activation are used respectively. After both convolution layers, the max-pooling layer with the receptive field of (2,2) was placed. In addition, there is a single abstract lambda layer placed between the max-pooling and LSTM layer. This is due to the reshaping the tensors and made them available for the LSTM operation. Finally, 128 input and 2 output units are used in the fully connected layer for classification.

4 Results

Based on the results in Table 1, we observed two important outputs. First of all, when we do not apply any transformation on UCI-HAR data and give the data as input to the models over time domain, the proposed solutions for non-linear and ensemble machine learning algorithms are under-performing compared to proposed LSTM algorithm. Secondly, by taking wavelet transformation of the time series and inputting the attributes in the frequency domain as input to the CNN and CNN-LSTM models, the performance significantly exceeded the performance of the rest of the models. In other words, the proposed CNN-LSTM Network outperformed all the other proposed solutions with 94.60% test accuracy. Our literature research shows that state-of-the-art algorithm achieves 97.90% test accuracy on (UCI-HAR) dataset [1].

Model	Preprocessing	Validation Acc	**Test Acc**	Test Macro F1
kNN	None	0.8301	0.6450	0.6376
Decision Tree	None	0.8142	0.7305	0.7215
Random Forest	None	0.9623	0.8554	0.8538
SVM	None	0.9406	0.8832	0.8859
Gradient Boosting Machine	None	0.9604	0.8863	0.8891
LSTM	None	0.9538	0.8989	0.8992
CNN	Wavelet Transform	0.9783	0.9230	0.9219
CNN-LSTM	Wavelet Transform	0.9918	**0.9460**	0.9448

Table 1: Performances of Models on UCI-HAR Dataset

5 Conclusion

To conclude, it can be said that deep learning algorithms are outperforming machine learning algorithms for the specified multivariate, time-series classification task. Additionally, using wavelet transformation on the dataset is increasing the learning ability of a designed network since this transformation converts the dataset into a function of frequency and time.

References

[1] David R Brillinger. *Time series: data analysis and theory*, volume 36. Siam, 1981.

[2] Anthony Bagnall, Jason Lines, Jon Hills, and Aaron Bostrom. Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535, 2015.

[3] Ian Naismith Sneddon. *Fourier transforms*. Courier Corporation, 1995.

[4] Shuhei Okamura. The short time fourier transform and local signals. *Department of Statistics Carnegie Mellon university, Dissertations*, 2011.

[5] Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. Multilevel wavelet decomposition network for interpretable time series analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2437–2446, 2018.

[6] Steven Craig Hillmer and George C Tiao. An arima-model-based approach to seasonal adjustment. *Journal of the American Statistical Association*, 77(377):63–70, 1982.

[7] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281, 2004.

[8] Paul Rodriguez, Janet Wiles, and Jeffrey L Elman. A recurrent neural network that learns to count. *Connection Science*, 11(1):5–40, 1999.

[9] Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE international conference on big data (big data)*, pages 2823–2824. IEEE, 2015.

[10] Ehsan Hoseinzade and Saman Haratizadeh. Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129:273–285, 2019.

[11] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. *Frontiers of Computer Science*, 10(1):96–112, 2016.

[12] Yi Zhao, Yanyan Shen, Yanmin Zhu, and Junjie Yao. Forecasting wavelet transformed time series with attentive neural networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1452–1457. IEEE, 2018.

[13] Karlheinz Gröchenig. *Foundations of time-frequency analysis*. Springer Science & Business Media, 2013.

[14] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[1] `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6412893/`

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Curtis Wigington, Seth Stewart, Brian Davis, Bill Barrett, Brian Price, and Scott Cohen. Data augmentation for recognition of handwritten words and lines using a cnn-lstm network. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 639–645. IEEE, 2017.

[18] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.