Exploration of Sensor-Based Activity Recognition Based on Time Series Feature Extraction

Wen-Hui Chen^{a,*}, Ting-Hsi Chen^a, Cheng-Han Tsai^b

^aGraduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan

^bMechanical and Mechatronics Systems Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan

*Corresponding Author: whchen@ntut.edu.tw

Abstract

Sensor-based human activity recognition (HAR) has gained its momentum and become an active research topic due to the advance of machine learning (ML) algorithms and ubiquitous sensing devices in our daily life. Recent research trend in ML algorithms for HAR is deep learning-based approaches that have already developed state-of-the-art learning models in various tasks. However, complex deep learning models may not be the best choice when it comes to data sufficiency problems and model transparency. Exploratory data analysis (EDA) can benefit feature extraction, which is an important step in a machine learning pipeline. In this study, to explore sensor-based HAR, a widely used HAR dataset is adopted to examine the effectiveness of time series feature extraction together with conventional machine learning models. Experimental results show that EDA can be beneficial for obtaining data insights and determining better features for HAR classification.

Keywords: exploratory data analysis, human activity recognition, machine learning, time-series feature extraction.

1. Introduction

Human activity recognition (HAR) has many potential applications such as smart healthcare and is an important component to build a smart environment. Sensor-based HAR is considered a non-intrusive and less privacy-sensitive data collection way compared to the vision-based one. Therefore, sensor-based HAR is more acceptable to people who have privacy concerns. Recent

advances and success in deep learning have led to many studies on the use of convolutional neural networks (CNNs), recurrent neural networks (RNNs) with long short-time memory (LSTM), and mixed models such as ConvLSTM to sensor-based HAR research^(1,2).

Although deep learning-based approaches have been proven to generate better performance and provide an end-to-end training framework, it still faces some challenges such as preparation of sufficient labeled data, determination of model architecture, and many hyper-parameters tuning, which is time-consuming and needs efforts on various trials to obtain good results.

Hence, traditional machine learning models with appropriate feature representation may still be a good choice for sensor-based HAR as they provide more insights to model output over the black-box deep learning models⁽³⁾. Data collected from inertial measurement units (IMUs), including accelerometer and gyroscope sensors are in the format of time series. How to extract good features from sensor data needs efforts and domain knowledge. In this study, we investigated the advantage of exploratory data analysis and a time series feature extraction library (TSFEL) that can facilitate the process of feature extraction on sensor data⁽⁴⁾.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work on sensor-based activity recognition and discuss the dataset we adopted for the experiment in Section 3. The proposed approach in Section 4 explains what we did and why we adopted EDA. Experimental results and a comparison of recognition performance of common learning algorithms are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Work

Human activity recognition plays an important role in building smart environments. Numerous approaches to activity recognition for various applications have been proposed in past research⁽⁵⁻⁸⁾. Sensor-based HAR requires sensor deployment to obtain activity information.

Nowadays, smart wearable devices such as smartphones, and smartwatches have become popular and affordable. These wearable devices in our daily lives provide us with rich information in a mobile and real-time fashion. They have equipped with inertial sensors such as accelerometers, gyroscopes, and digital compasses. Making good use of the unobtrusive embedded sensors, an assistive system can be built for monitoring the activities and making further analysis through the use of information technology for context-aware applications to improve the quality of daily living.

Many activity recognition studies using smartphones have been conducted over the past few decades^(9,10). Most of the studies in sensor-based activity recognition relied on hand-crafted features. In recent years, the development of deep learning has achieved great success in various areas and has become a trending research direction in activity recognition⁽¹¹⁻¹³⁾. Deep learning requires sufficient data to train network parameters, which is time-consuming and needs labor-intensive labeling work. When the training data is limited, the performance of deep learning models is not quarantined performing well. Therefore, the majority of the previous works are done with sufficient datasets and claimed that deep learning outperforms many state-of-the-art approaches. However, not every task can easily provide sufficient data for deep learning in practice.

3. Dataset Description

The HAR dataset used in this study was adopted from⁽¹⁴⁾. It also can be downloaded from the UCI Machine Learning Repository⁽¹⁵⁾. This is a dataset built from the recordings of 30 subjects performing six activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The sensor signals (accelerometer and gyroscope) were pre-processed and sampled in a fixed-width sliding window of 2.56 seconds with 50% overlap (i.e., 128 readings per window in a record).

Each record consists of nine signals, namely: tri-axial

acceleration from the accelerometer (total acceleration), the estimated tri-axial body acceleration, and tri-axial angular velocity from the gyroscope. Those nine signals can be considered as nine components or channels at each sampling time and were stored separately in nine files.

Figure 1 illustrates a randomly selected record (128 points) that shows the variations of the nine signal components for six activities. From Fig. 1, we can observe that the variations of sensor signals measured from different activities reflect the body change and movement. The activities used in the UCI-HAR dataset, include *walking*, *sitting*, *standing*, *laying*, *walking upstairs*, and *walking downstairs*. Note, the authors used “*laying*” in the dataset instead of “*lying*.” In compliance with the dataset, “*laying*” was used in the context.

The distinction of activities between dynamic and static activities (inter-class variation) is significant but less obvious among dynamic activities or static activities (intra-class variation). HAR aims to correctly identify each activity according to the received sensor signal patterns.

Fig. 1. Sensor signal variations capturing from nine components for six different activities.

Note that the UCI-HAR dataset used in this study also provided a pre-processed 561 engineered features but it does not fit our research goal. As we plan to extract

time-series features of our own from the raw sensor data, we do not use those pre-defined engineered features.

The distribution of the raw dataset has already split into a training set (7,352 samples or records) and a test set (2,947 samples or records). The number of samples for each activity in both sets is not well-balanced as shown in Fig. 2. It can be observed that the activity of the *walking downstairs* activity has the least number of samples while *laying* has the largest sample size.

(a) Training data distribution

(b) Test data distribution

Fig. 2. Distribution of the UCI-HAR dataset.

4. The Proposed Approach

Exploratory data analysis is a way of analyzing datasets that aim to obtain the main facts or ideas about their meaning and characteristics using statistical and graphical visualization tools. EDA can provide better insights behind formal modeling and can be beneficial to the subsequent machine learning model.

In this study, we use EDA in complement with time series feature extraction to visualize if the extracted features can be used to separate activities effectively and test the proposed approach on various machine learning models for HAR.

The workflow of a standard machine learning pipeline consists of multiple stages starting at data preprocessing, feature extraction to algorithm deployment, as shown in Fig. 3. Feature extraction is to find a good representation for data and is used for model training, which plays an important role and highly impacts the model performance at the subsequent stage.

Fig. 3. A standard machine learning pipeline.

The dataset used in this study is embedded IMUs as described in Section 3. IMUs are types of commonly used wearable sensors for motion tracking and also a good choice for HAR. The data collected from IMUs are in their time-series format. When the data preparation is complete, feature extraction is performed and the extracted features are fed into the machine learning models as inputs.

The UCI-HAR dataset has already been partitioned into a training and a test set, with 7,352 and 2,947 records, respectively. The loaded training and test set running in the program were *Numpy ndarray* in the size of (7352, 128, 9) and (2947, 128, 9), respectively.

There are several performance metrics used in classification tasks, such as confusion matrix, classification accuracy, precision, recall, and F-measure. We adopted the F-measure to evaluate the recognition performance. Defined in (1), the F-measure or F1-score is a measure of test accuracy by combining the measure that assesses the precision and recall scores in (2) and (3), respectively.

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1)$$

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (2)$$

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i} \quad (3)$$

where

N is the total number of class instances

TP is true positive

FN is false negative

FP is false positive

5. Experimental Results

In this section, we empirically examine three scenarios to evaluate the performance of HAR: feature extraction by TSFEL with and without correlation removal and normalization; feature extraction by Fast Fourier Transform (FFT), power spectral density (PSD), autocorrelation (AC) function, and peaks.

5.1 Feature extraction by FFT, PSD, AC, and Peaks (Case 1)

In this study case, we use FFT, PSD, and AC function in each of the three transformations (i.e., FFT, PSD, AC) to extract signal features. In addition, for each transformation, we take the first peak in the signal for both the x and y values. Therefore, we can obtain 54 transformed features to represent the original 128 sampling points in each 2.56-second signal record. Fig. 4 shows the signal variations in terms of the first feature. As we can see that the first transformed feature can differentiate dynamic activities from static activities. The activity “*laying*” is mostly separable from other activities.

Fig. 4. Signal variations of different activities in terms of the first transformed feature.

From Fig. 4, we can see that *standing* and *sitting* have similar features, same as *walking downstairs* and *walking upstairs*. The six activities in the dataset can be roughly categorized into dynamic activities (sitting, standing, laying) and static activities (walking, walking upstairs, walking

downstairs).

After completing the feature extraction, we examine the performance of nine machine learning algorithms from *sklearn* packages, namely Gradient Boosting, Random Forest, Logistic Regression, Multilayer Perceptron, Decision Tree, Linear Support Vector Machines (SVM), Naïve Bayes, Nearest Neighbor, and AdaBoost to evaluate which learning algorithm performed well.

Performance comparison of different machine learning models is listed in Table 1. All the results are computed using default settings of *sklearn* classifiers. The results show that *GradientBoostingClassifier*, the Gradient Boosting classifier performs best with 87.58% accuracy on the test set. Figure 5 shows the confusion matrix derived from Gradient Boosting for case 1.

Table 1. Performance comparisons of different machine learning algorithms (case 1).

algorithms	train_score	test_score
Gradient Boosting	0.976605	0.875806
Random Forest	0.997688	0.849678
Logistic Regression	0.826986	0.786563
Multilayer Perceptron	0.847933	0.776043
Decision Tree	1.000000	0.775025
Linear SVM	0.856366	0.706481
Naive Bayes	0.724565	0.701391
Nearest Neighbor	0.850789	0.671870
AdaBoost	0.533868	0.507974

Fig. 5. Confusion matrix derived from Gradient Boosting for case 1.

5.2 Feature Extraction by TSFEL without Correlation Removal and Normalization (Case 2)

In this study case, we adopted TSFEL to compute features across temporal, statistical, and spectral domains to expedite the feature extraction process. The pre-defined feature configuration file in TSFEL was retrieved to extract all available features using `tsfel.get_features_by_domain()` and `tsfel.time_series_features_extractor()`. The obtained training set and test set are in the shape of (7352, 1791) and (2947, 1791), respectively.

Performance comparison of different machine learning models is listed in Table 2. All the results are computed using default settings of *sklearn* classifiers. The results show that the Gradient Boosting classifier performs best with 94.71% accuracy on the test set. Figure 6 shows the confusion matrix derived from Gradient Boosting for case 2.

Table 2. Performance comparisons of different machine learning algorithms (case 2).

algorithms	train_score	test_score
Gradient Boosting	1.000000	0.947065
Logistic Regression	0.996736	0.918222
Random Forest	0.999320	0.900916
Multilayer Perceptron	0.981910	0.880556
Decision Tree	1.000000	0.873770
Naive Bayes	0.861534	0.808958
Nearest Neighbor	0.975653	0.797421
AdaBoost	0.545022	0.531049
Linear SVM	1.000000	0.182219

Fig. 6. Confusion matrix derived from Gradient Boosting for case 2.

To visualize the high dimensional data, we used *t*-distributed stochastic neighbor embedding (*t*-SNE) to

create a low dimensional map that discovers the underlying data structure. *t*-SNE is a technique for non-linear dimensionality reduction⁽¹⁶⁾. So, it is well-suited for the visualization of this study case that transforms high-dimensional features in a low-dimensional embedding. The transformation starts by computing the similarity of points in high-dimensional space and the similarity of points in the corresponding low-dimensional space and tries to minimize the difference between the similarities in high-dimensional and low-dimensional data.

The Kullback-Leibler (KL) divergence is a measure for the similarity of two probability distributions to see how one probability distribution diverges from the other. Hence, the *t*-SNE transformation process tries to minimize the sum of K-L divergence of overall data points for a perfect representation of data points in lower-dimensional space.

t-SNE requires the user to choose a perplexity, a tunable parameter whose typical values are between 5 and 50 to adjust the widths of its high-dimensional Gaussian neighborhoods. Fig. 7 (a)-(d) are the results at various perplexity values. The learning rate and the number of iterations are set to 200 and 1000, respectively. From Fig. 7, we can see that the activities are clustered in their own group.

(a) Perplexity=5, iteration=1000

(b) Perplexity=10, iteration=1000

(c) Perplexity=30, iteration=1000

(d) Perplexity=50, iteration=1000

Fig. 7. Data visualization with different perplexities.

5.3 Feature extraction by TSFEL with Correlation Removal and Normalization (Case 3)

In this study case, we adopted TSFEL to compute features across temporal, statistical, and spectral domains and correlation removal and normalization for sensor data the same as in case 2. Additionally, we applied correlation removal and data normalization to reduce the number of dimensions, which helps suppress some noise and can speed up the computation. The obtained training set and test set are in the shape of (7352, 969) and (2947, 969), respectively.

Performance comparison of different machine learning models is listed in Table 3. All the results are computed using default settings of *sklearn* classifiers. The results show that *MLPClassifier*, the Multilayer Perceptron classifier performs best with 96.1% accuracy on the test set. Figure 8 shows the confusion matrix derived from Multilayer Perceptron for case 3.

The results are much better than⁽¹⁷⁾ that reaches 91.65% accuracy on the test set using LSTM deep learning model⁽¹⁷⁾.

Table 3. Performance comparisons of different machine learning algorithms (case 3).

algorithms	train_score	test_score
Multilayer Perceptron	0.999184	0.960977
Logistic Regression	0.999320	0.955548
Linear SVM	0.990615	0.952494
Gradient Boosting	1.000000	0.944690
Random Forest	0.999184	0.912453
Nearest Neighbor	0.983542	0.880217
Decision Tree	1.000000	0.857143
Naive Bayes	0.747688	0.740075
AdaBoost	0.738575	0.709196

True label	WALKING	492	0	4	0	0	0
	WALKING_UPSTAIRS	2	467	2	0	0	0
	WALKING_DOWNSTAIRS	7	9	404	0	0	0
	SITTING	0	1	0	436	54	0
	STANDING	0	0	0	25	507	0
	LAYING	0	9	0	0	2	526
		WALKING	WALKING_UPSTAIRS	WALKING_DOWNSTAIRS	SITTING	STANDING	LAYING
		Predicted label					

Fig. 8. Confusion matrix derived from Multilayer Perceptron for case 3.

6. Conclusions

Deep learning-based approaches have become the mainstream for many tasks. However, there still exist some limitations behind deep learning models. It turns out that not all problems are suitable for them. In this study, we have conducted experiments on sensor-based human activity recognition. By utilizing TSFEL, a time series feature extraction library in complement with exploratory data analysis, we can easily design a traditional machine learning model that achieves a high classification accuracy, even better than LSTM, which is a popular deep learning model for time series problem.

Acknowledgment

This work is supported by the Taiwan Ministry of Science and Technology (MOST) under Grant No. MOST 109-2221-E-027-049 and Industrial Technology Research Institute (ITRI) under Grant No. 3000657053.

References

- (1) J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: a survey," *Pattern Recognition Letters*, Dec. 2017.
- (2) C. Betancourt, W. H. Chen, C. W. Kuan, "Self-attention networks for human activity recognition using wearable devices," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, Canada, Oct. 11-14, 2020.
- (3) W. H. Chen, T. C. Chiu, and W. H. Chen, "Implementation of in-home activity recognition based on web service and machine learning approaches," *Proc. of the 6th International Conference on Intelligent Systems and Image Processing*, Matsue, Japan, Sept. 10-14, 2018.
- (4) M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "TSFEL: Time series feature extraction library," *SoftwareX*, vol. 11, Jan. 2020.
- (5) W. H. Chen, Y. C. Lin and W. H. Chen, "Comparisons of machine learning algorithms for driving behavior recognition using in-vehicle can bus data," *Proc. of the 9th International Conference on Informatics, Electronics & Vision*, Spokane, USA, May 30-June 2, 2019.
- (6) J. P. Wherton and A. F. Monk, "Problems people with dementia have with kitchen tasks: The challenge for pervasive computing," *Interacting with Computers*, vol. 22, no. 4, pp. 253-266, July 2010.
- (7) S. Lindley, D. Randall, W. Sharrock, M. Glancy, N. Smyth and R. Harper, "Narrative, memory and practice: tensions and choices in the use of a digital artefact," In *Proceedings of the 23rd British HCI Group Annual Conference on Human Computer Interaction*, pp.1-9, 2009.
- (8) J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: a review," *ACM Computing Surveys*, vol. 43, no. 3, pp. 1-47, April 2011.
- (9) V. R. Jakkula and D. J. Cook, "Detecting anomalous sensor events in smart home data for enhancing the living experience," *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Smarter Living*, pp. 33-37, 2011.
- (10) W. H. Chen, Y. H. Lin, and S. J. Yang, "A generic framework for the design of visual-based gesture control interface," *the 5th IEEE Conference on Industrial Electronics and Applications*, pp. 1522-1525, June 2010.
- (11) S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, and S. Gupta, "Deep ConvLSTM with self-attention for human activity decoding using wearable sensors," *IEEE Sensors Journal*, vol. 21, no. 6, March 2021.
- (12) S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A unified deep learning framework for time-series mobile sensing data processing," *arXiv:1611.01942v2*, July 2017.
- (13) F. J. Ordoñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 115, pp.1-25, 2016.
- (14) D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, April 24-26, 2013.
- (15) <https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>
- (16) L. van der Maaten and C. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research* vol. 9, pp. 2579-2605, 2008.
- (17) G. Chevalier, "LSTMs for Human Activity Recognition," 2016. Available at

<https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition>