A COMPREHENSIVE COMPARISON OF DATA PROCESSING METHODS FOR HUMAN ACTIVITY RECOGNITION FROM ACCELEROMETER-BASED DATA

Guo Feng, Lu Jiahao, Yam Gui Peng David, Zhao Yazhi

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

Prediction of human activity has become more valuable in daily life. Providing accurate information based on people's activities and behaviours can help in health, medical, security, entertainment and many scenes that are closely related to people's lives. Human activity recognition systems have been an active field in the past with the relevant features extracted from wearable sensors.

With the emergence of deep learning and increased computation power, machine learning and deep learning are becoming more and more popular in classifying human activities. Our human activity recognition model uses Long Short-Term Memory (LSTM) networks approach. The method was tested based on wearable sensors from Movement Analysis in Real-world Environments using Accelerometers (MAREA) [1]. It can predict user activity based on the historical data to further improve the activity classification. The performance is further compared with algorithms like XGBoost and Convolutional Neural Network (CNN) to see the difference.

Index Terms— Human activity reccognition, Machine Learning, Deep learning, Long short-term memory, Inertial sensors

1. INTRODUCTION

Back in the late 90s [2], some researchers have started working on human activity recognition. Limited by the techniques of the time, there are many issues and accuracy requirements to be improved. In the last two decades, more and more studies have focused on human activities recognition. Different devices, sensors and data were utilised for activity detection. Some examples were video-based sensors [3], force sensors [4], and inertial sensors [5]. Many data acquisition systems are mainly utilized for research, which can be expensive and hard to use in daily life [4]. Some studies even require external sensors in a specific room to recognize activities [6].

Generally, wearable based sensors for human activity identification are cheaper than other sensors. Especially since many people now use mobile phones and smartwatches, which can provide alternatives and reduce unnecessary additional cost from sensors. With the popularity of mobile phones and smartwatches, that are carried all the time, lots of data can be collected in a realistic condition. Our aim is to accurately discover the activity pattern based on wearable sensors. To further understand whether the activity pattern can still be correctly recognized, we also limit the sensor and axis to see how much accuracy would be decreased.

In this project, we proposed a Long Short-Term Memory (LSTM) model approach and evaluate its performance (i) with different data preprocessing methods, (ii) with different sensors and (iii) against performance on the machine learning algorithm XGBoost and another popular deep learning method Convolutional Neural Network (CNN).

2. RELATED WORK

Researchers have utilized different methods to detect human activities from full bodied motions like walking, running, falling to specific ones like Heel-Strike and Toe-Off events. Convolutional Neural Networks were used to achieve 99% accuracy for fall detection in [7]. Whilst in RapidHARe a dynamic Bayesian Network was created to achieve continuous activity recognition [8]. In both cases multiple sensors placed at strategic locations were utilized to obtain the best accuracy and readings. Researchers also managed to obtain high accuracy on Heel-Strike events by utilizing a modified LSTM model [9].

Fourier Transform is a common method to reduce noise and extract information from signal processing data. The formula of Fourier Transform is the dot product between real signal and various frequency of sine wave. The basis functions are simply the complex oscillations:

$$b_\omega(t) := \exp(i\omega t) \tag{1}$$

where t is the time axis of the signal and ω is the single frequency parameter that determines the basis function in the family.

There is one basis function for every ω. The Fourier Transform of the signal $s(t)$ is then simply the inner product written as an integral:

$$\begin{aligned} \mathcal{F}\{s(t)\}(\omega) &= \langle b(\omega, t), s(t)\rangle \\ &= \int_{-\infty}^{\infty} \exp(-i\omega\tau)s(\tau)d\tau \end{aligned} \tag{2}$$

From Fourier Transform, we can get a frequency spectrum of the real signal without the index that the 'frequency' happened. So we lose the time resolution of the real signal. In order to get both frequency and time resolution, we can apply Short Time Fourier Transform to divide the original signal into several parts. Short Time Fourier Tranformation adds a time dimension to the base function parameters by multiplying the infinitely long complex exponential with a window to localize it. The base functions are formulated by

$$b_{(\omega,t_0)}(t) := w\left(t-t_0\right)\exp(i\omega t) \tag{3}$$

where $w(t)$ is the window functions that vanishes outside some interval and (w, t_0) are the time-frequency coordinates of the base function in the family. The inner product then becomes:

$$\mathcal{S}\{s(t)\}\left(\omega,t_0\right) = \int_{-\infty}^{\infty} w(\tau)\exp(-i\omega\tau)s(\tau)d\tau \tag{4}$$

However, Short Time Fourier Transform can only catch $n/2$ frequency where n is the length of the partial signal. To avoid this issue, we need to apply Wavelet that uses a bigger time window to catch low frequency and smaller window for higher frequency. The family of base functions for wavelet is

$$b_{(\sigma,t_0)}(t) = w\left(\frac{t-t_0}{\sigma}\right) \tag{5}$$

where ω is the original wavelet and τ the scale parameter. The inner product becomes then [10]

$$\mathcal{W}\{s(t)\}\left(\sigma,t_0\right) = \int_{-\infty}^{\infty} w\left(\frac{t-t_0}{\sigma}\right)s(t) \tag{6}$$

As one of the simpler to implement but accurate and robust machine learning algorithms, XGBoost has been used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. In Kaggle competitions, XGB is used in almost every winning solution, XGB is generally the first model to try and is the best performing single model [11].

On the other hand, traditional machine learning algorithms are limited by their ability to process raw data. Features need to be manually extracted into a presentation by expertise so the system can make inferences. However, deep learning algorithms can learn the features and use the appropriate identifier for regression, classification and prediction [12]. Deep learning can deal with data types such as images, audios, videos and text.

CNN is a type of deep learning network designed to process data in multiple arrays. For example, signal information or text can be arranged as 1D arrays; audio spectrograms or images as 2D arrays; and video as 3D arrays [12]. There are two main layers in CNN, one is convolutional and the other is pooling. The features are extracted by convolutional layers and pooling layers merge semantically similar features [12].

Unlike other data types, signal data add the complexity of a sequence dependence among the input variables. While CNNs are designed to deal with data arranged in multiple arrays, Recurrent Neural Networks (RNNs) is a powerful type of neural network designed to handle time series data. When dealing with long time dependencies, RNNs suffer from a problem called vanish gradient [13]. LSTM is a variation of traditional RNN to solve this issue. In LSTM networks, the recurrent neuron is replaced by a memory cell and its operation is controlled by three gates: input gate, forget gate, and output gate [14]. Data flow through the cells, the gradient is trapped within the cell, preventing the vanish gradient problem [15].

3. PROPOSED APPROACH

3.1. Datasets

Our raw data is from Human Gait Datasets by Halmstad University [1]. The datasets can be divided into 3 parts: (i) activity timing (5 indoor and 2 outdoor activities), (ii) accelerometer signal magnitude of 20 users in 3 axes. and (iii) timing of the Heel-Strike and Toe-Off events. For the $4th$ user (Sub4), only 3 sensors' data are available but the rest subjects have 4 sensors data, thus we decided to remove this dataset.

In timing file, that uses time period (separated by columns) to indicate the type of activities, we used it to label the respective activities for each data row in the sensor data. In the final data frame, we have incorporated the subject name as another column. As such our intermediary dataset has the following structure: (accX_LF, accY_LF, ..., accZ_Wrist, label, subject).

From Fig.1 and Fig.2, we can see the data is not highly skewed among the activities. Although the ourdoor activity subjects data points are fewer than indoors', the ourdoor activity data points are similar to other individual indoor activities.

From Fig.3, we can see the mean values of magnitude of different activities are quite close, which is hard to be linearly separated from activities with motion. However, the maximun and minimun magnitude is different from each other. And such distinction degree is various among different axes.

Fig.4 shows the original signals from left foot in 3 axes for 'tread flat walk'. It can be observed some axis has higher maximum and minimum magnitude, which proves the result from Fig.3.

Among different finding peak methods, the best result is given by distance and height with 100 and 10 (5). However, the result is not stable if the same parameter sets are put for different activities and axes, which means cutting the samples by peaks may not a good approach.

Different wavelets are tested to see which works better in our signals. In Fig.6, 'db4' is used from wavelet families with level 2 refinement and thresholded coefficient 0.8.

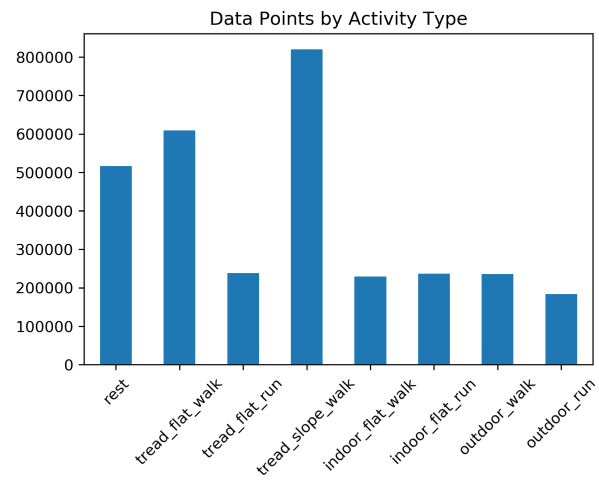


Fig. 1: Data points distribution (activity).

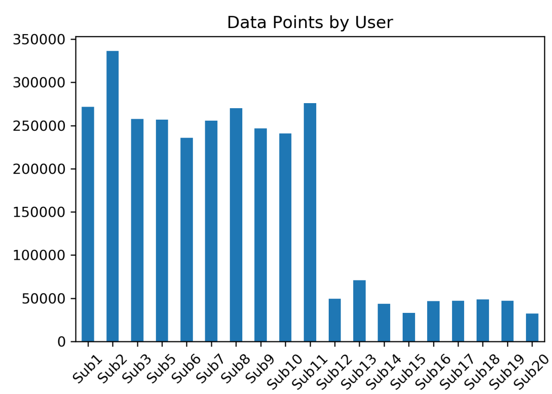


Fig. 2: Data points distribution (user).

After understanding the datasets, the proposed flow is presented as Fig.7.

3.2. Data preprocessing

3.2.1. Sampling

It is common to split signal data into windows of uniform sizes for processing by subsequent models. We choose 3 methods to split the dataset: Split without overlap ($2s$/$4s$ window), Split with 50% overlap ($2s$/$4s$ window) and Split by peaks detected ($2s$/$4s$ window). Each method is tested together with the other parameters and evaluated by best test accuracy.

Split without overlap ($2s$/$4s$ window): As the sampling frequency is 128Hz, this translates to a window size of 256/512 values respectively. For this method the windows produced do not overlap with each other, this is the most naive method of splitting the input samples.

Split with 50% overlap ($2s$/$4s$ window): For this method the samples produced overlap by 50%, meaning that 50% of the current window is also found in the previous window. This overlapping is useful to ensure the correct pattern is not missed in the window and also because there is likely a dependency with the previous time period.

Split by peaks detected ($2s$/$4s$ window): For this method, peaks in the signal data of the first input column are detected (e.g. accX_LF) and the subsequent 256/512 values from each peak are split into a window. In this method, we are not able to control the overlap percentage of the windows collected. The intuition in this method is to reduce the arbitrary nature (splitting by every $2s$/$4s$ from the start) to split the windows. Another method evaluated is to split by external data such as the Heel Strike/Toe Off dataset provided, however, it was explored and deemed inadequate as such data would be unlikely to be provided in an actual real-world situation (unlikely to have actual Heel Strike/Toe Off sensors).

Choice of $2s$/$4s$ windows: We did an exploration of the data and evaluated that $2s$/$4s$ would be able to capture 1-2 cycles of the various activities (i.e. rest, running, walking).

3.2.2. Train & test data splitting

In normal circumstances, data is split by index for training and testing dataset. However, as signal data may have different dimensions according to the business objects. For example, in HAR dataset, there are three dimensions, subject, time index, and activity. Based on different business objects, we may have three ways to split the data.

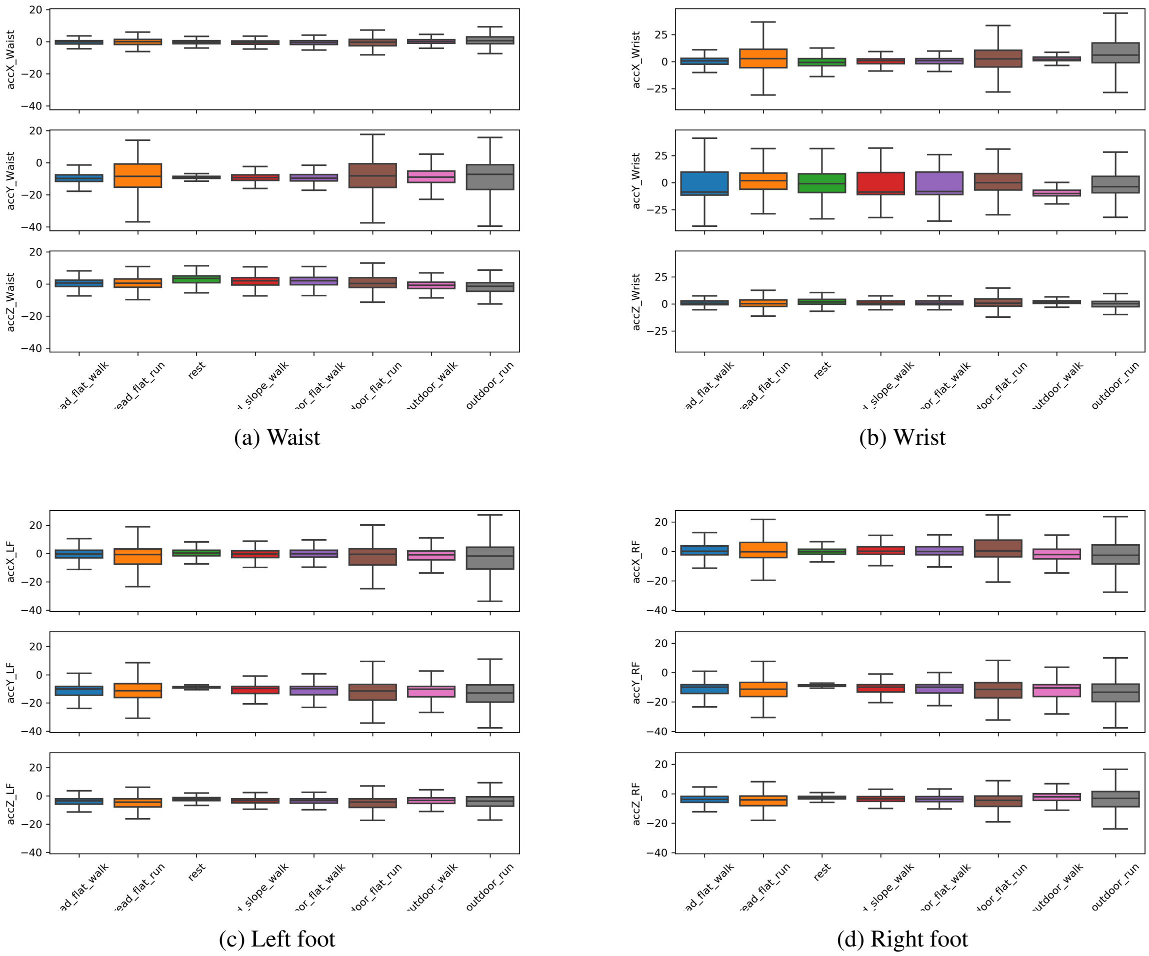
Method 1 – Random Split: Random Split will combine the data points from all subjects for all activities, shuffle and randomly split into training and testing dataset with a ratio of $80 : 20$.

Method 2 – Within-Subject Split: For each activity, data will be split within each subject. So both training and testing dataset will include all subjects and all activities.

Method 3 – Different-Subject Split: Data will be split by subjects. So some of the subjects will appear in training while the rest of the subject will appear in testing.

3.2.3. Wavelet

In this project, in order to catch all possible frequency within different time period window, our team decides to apply wavelet transformation as the signal de-noise preparation. We have set the wavelet method as 'db4' and wavelet level as 2. The comparison results for before and after wavelet transformation signals is shown in Fig.6 as an example.



(a) Waist (b) Wrist (c) Left foot (d) Right foot

Fig. 3: Acceleration magnitude distribution.

3.3. Process setup

The final potential hyper-parameters and signal processing methods were chosen after discarding a few deemed to be more inadequate (split by peaks, scaling by min-max, baseline correction). However, there were still too many hyper-parameters and sensor/axis possibilities to choose from resulting in $2 \times 2 \times 2 \times 3 \times (1 + 4 + 6 + 4) = 360$ options if we selected by window size (256/512), window split (no overlap/50% overlap), wavelet (Yes/No), split method (TrainTestSplitWithinSubject/Random/ DifferentSubjectsInTrainTest) and different combinations of the 4 sensors. Hence it was deemed that the evaluation should be done in stages:

Stage 1: Select for best hyper-parameters, keeping all 4 sensors with 3 axes each. This would result in $2 \times 2 \times 2 \times 3 = 24$ models.

Stage 2A: Select for the best 1 sensor, with the selected best hyper-parameters from Stage 1. This would result in 4 models.

Stage 2B: Select for the best 2 sensors, with the best hyper-parameters from Stage 1 and the best 1 sensor with all 3 axes from stage 2A. This would result in 3 models.

Stage 3: Select the best 2 axes/1 axis from the best sensor, with the best hyper-parameters from Stage 1. This would result in $3 + 3 = 6$ models.

As it is unlikely that a commercial user would wear all 4 sensors – Left Foot (LF), Right Foot (RF), Waist (WA) & Wrist (WR), we evaluate the accuracy for each of the sensors separately with the obtained best hyper-parameters. We also evaluate the possibility of having 2 sensors instead of all 4 sensors.

With a lack of storage/ processing capacity in the wearable device, it would be even more desirable to only collect 1 or 2 axes instead of all 3 but still have comparable accuracy readings.

Fig. 4: Original x-axis signals from left foot sensor.

3.4. Evaluation

Accuracy is one of the more obvious metrics, as it is the measure of all the correctly identified cases. It is mostly used when all the classes are equally important and have similar levels of support. In our case, True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are crucial. Furthermore, the dataset does not have imbalanced classes. Therefore, accuracy will be the main metric to evaluate our model. In multi-class or binary single-label classification problem, absolute accuracy is given by the ratio (no of data instances correctly classified / total no of data instances), namely:

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1\left(\hat{y}_i = y_i\right) \qquad (7)$$

where $\hat{y}$ is the predicted value of the i-th sample and y_i is the corresponding true value, and $accuracy$ is the fraction of correct predictions over n_{samples} .

3.5. Model

At data preparation stage, we joined the raw datasets (timing and sensor), and built a dataframe for each subjects (19 of them) which has 12 columns of sensor data (x, y, z dimension for 4 sensors) and 1 column of labeling data. From Fig.8, the absolute variance of the X, Y, Z axis can give an approximate separation between classes (walking vs running vs resting). This could be theorized that a quicker swing is required for faster motions such as running, resulting in a larger absolute variance, whilst minimum swing is required when the subject is at rest, as evidenced by the figure.

As a simple exploration, we appended the 19 dataframes into 1 master dataframe, and simply fit it into the XGBoost model (12 columns as the input, label column as the target), to see whether the XYZ dimension alone can classify the activity. We used XGBoost model with the major parameters set as Table 1 below:

Table 1: XGBoost hyper-parameters.

Parameter	**Value**
base_score	0.5
booster	gbtree
Learning rate	0.1
max_depth	3

The CNN architecture in our project consists 2 convolutional layers and 2 maxpooling layers with stride of 1 and 4 respectively. Fig.9 illustrates the proposed architecture and the details involved in the structure.

To implement LSTM methodology, we create a four-layer LSTM model by using LSTM with variable of 100, dropout rate of 0.5, and two dense layers with variable of 100 and 8. There are 56k parameters used in the LSTM model. The proposed model is shown in Fig.10.

4. EXPERIMENTAL RESULTS

4.1. The best hyper-parameters

From Table 2, the best hyperparameters were window size 256, 50% overlap, using wavelet transformation and with a Random train test split. This obtained an accuracy of 97.61%.

Referencing the "split method" column of Table 2, Method 1 almost always outperformed Method 2 and Method 3. Evaluating Method 1 & 2 vs Method 3, its seems most likely that if the subjects had somewhat dissimilar walking or running styles from each other, then having a group of subjects in the train set and a separate group in the test set would produce lower accuracy values. Contemplating the fact that the experiment would likely require differing individuals for interesting results, this hypothesis seems plausible. Method 2 only separated the train test split by subject but did so in a 80-20 manner by index whilst Method 1 randomly shuffled each window evenly. As subjects did not run at a constant speed, this would mean that the initial part of the running section might be quite dissimilar from the ending part of the running section. It would be interesting to repeat the experiment with Method 2 fully randomly shuffled.

A window size of 256 (or $2s$) with 50% overlap was the best window related parameter. It seems that the window size of 256 mostly outperformed the window size of 512 for Methods 1 & 2. this could be because the window size of 256 was sufficient in collecting the required information to separate indoor/outdoor walk/run, and the additional $2s$ data in fact added more dissonance. It is strange to note that the Wavelet component alternates between Yes and No, indicating that it did not have such a strong influence on the accuracy.

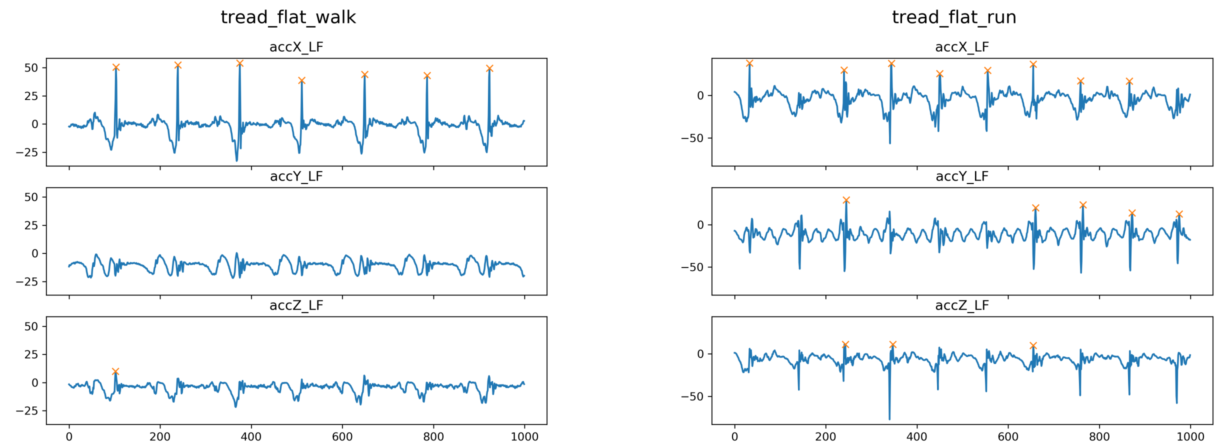


Fig. 5: Find peak by distance and height.

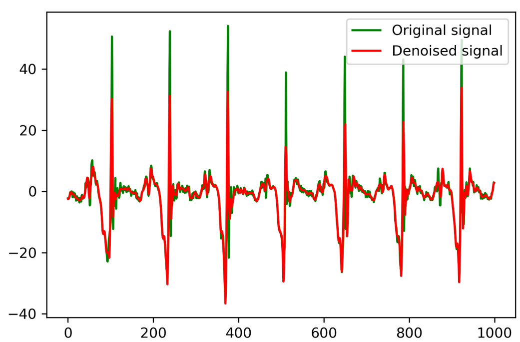


Fig. 6: Wavelet transformation (db4) for x-axis signals from left foot sensor.

4.2. The best 1 & 2 sensor(s)

From Table 4, the best 1 sensor, with all 3 axes was the LF, which obtained an accuracy of 90.25%. The best 2 Sensors, with all 3 axes was the LF & WA, which obtained an accuracy of 96.96%.

The Left Foot (LF) sensor had the highest accuracy of the single sensor readings, this could be attributed to the fact that when walking/running different individuals had different movement styles with regards to their Wrist (swinging it strongly forward and backward/ keeping it close to the body) and Waist (rotating the hips or keeping it steady). It should be noted that the Right Foot (RF) sensor had the second highest accuracy of the single sensor readings at 84.89%. It is intriguing to note that the Wrist had the worst single sensor accuracy at 62.87The Left Foot (LF) and Waist (WA) had the highest accuracy of the 2 sensors' readings at 96.96%, followed closely by the LF & RF at 96.73%. This could be that most of the inference came from the LF readings and the secondary sensor provided a contrast to separate the close cases.

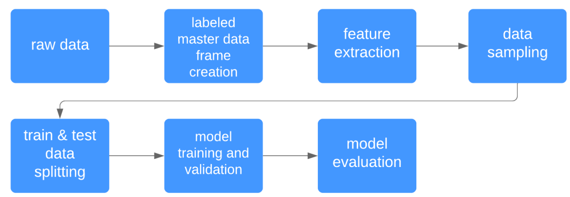


Fig. 7: Proposed data flow.

4.3. The best 1 & 2 axis

The best 2 axes of the LF Sensor is the Y and Z Axis, which obtained an accuracy of 86.38%.

The best 1 axis of the LF Sensor is the Y Axis, which obtained an accuracy of 73.49%.

Of the single axis, the Y axis made the most contribution, from the MAREA explanation the Y axis is the downward axis [1]. This could be because the ankle needs to be raised more abruptly during Toe-Off when running as compared to when walking, resulting in higher acceleration in the Y axis.

Our best result is with the accuracy of 97.61% where all 4 sensors and all 3-dimensions data were used. This is also the highest score among all the combinations of chosen sensors and dimensions, the result proved that the more comprehensive data we use for training, the better results we would expect for the model. We have observed that with all 4 sensors, the highest accuracy was achieved. While, with less sensors, we could also achieve decent results for the classification:

- With 2 sensors: LF & Waist of all 3 dimensions data, we obtained 96.96% accuracy.

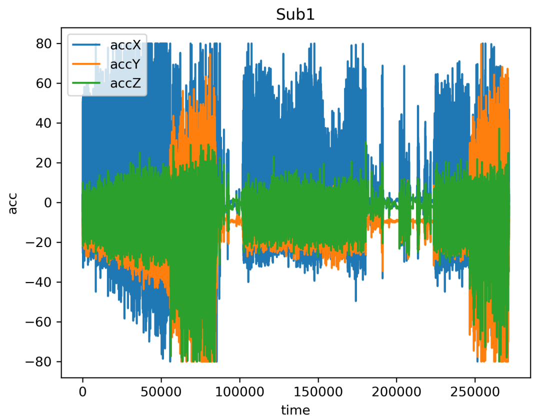


Fig. 8: Original signal of subject 1.

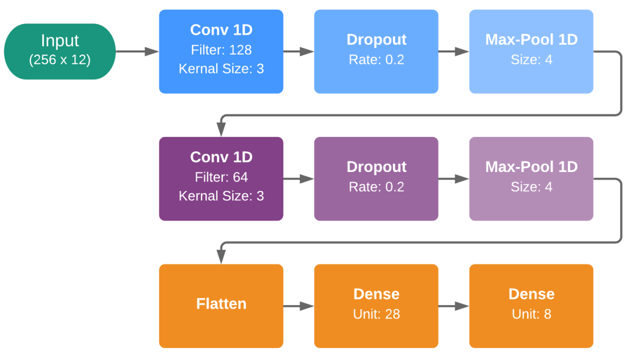


Fig. 9: Proposed CNN model.

- With 1 sensor alone: LF with all 3 dimensions data, we obtained 90.28% accuracy.
- With 1 axis alone: LF with Y,Z dimensions data, we obtained 86.38% accuracy.

With 2 sensors: LF & Waist of all 3 dimensions data, we obtained 96.96% accuracy. With 1 sensor alone: LF with all 3 dimensions data, we obtained 90.28% accuracy. With 1 axis alone: LF with Y,Z dimensions data, we obtained 86.38% accuracy

From Table 4, we observed with LF sensor alone, we are able to reach more than 90% accuracy, this is understandable, as persons' foot movement changes significantly for different activities such as running, walking, slop walk or flat walk. In other words, without looking at a person's full body movement, but only one of his feet, we will be confident enough to tell what he is doing.

While for LF sensor alone, dimension Y, Z gives the accuracy of 86.38%, as X dimension data shows the stride of the person, which does not change much for activities such as run and walk and not much different, while Y (vertical), Z(sideways) data change significantly for activities of walking and running.

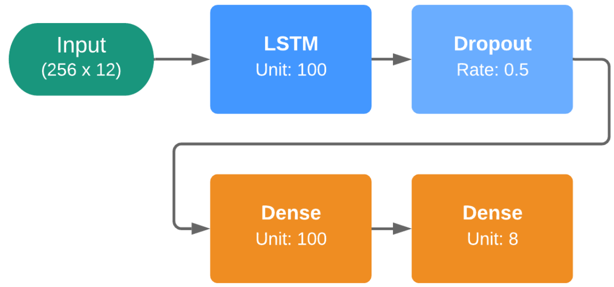


Fig. 10: Proposed LSTM model.

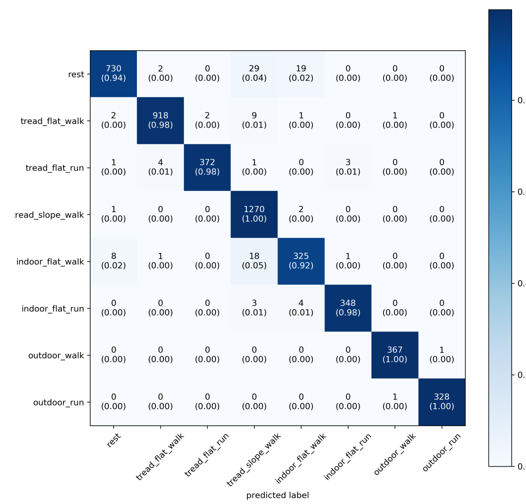


Fig. 11: LSTM best result confusion matrix.

The above observations also give us an idea regarding cutting down insignificant sensors, thus, instead of using 4 sensors, we would only need 1 sensor, this is a 75% saving of cost and effort for the researchers.

However, this suggestion only applicable for specific circumstances such as activity classification. If there are other research focuses, we still need to have more sensors to provide comprehensive readings from all parts of the body.

Inspecting Figure 10, we can see that the outdoor_walk and outdoor_run classes were classified with almost perfect recall by the LSTM model using 4 sensor, 3 axes with the best hyper-parameters. This could be because the outdoor surfaces are not as even as indoor surfaces, allowing for a better separation with the various axis. It is interesting to note that the 'rest' class had the lowest recall amongst the classes where 29 instances were predicted as tread_slope_walk and 19

Table 2: Accuracy score for each pre-processing scenario using LSTM Model Approach.

Sliding window		Wavelet	Split method	Accuracy
Size	Overlap			
256	50%	Yes	Method 3	42.23%
256	50%	No	Method 3	48.49%
512	No	No	Method 3	49.57%
512	No	Yes	Method 3	51.04%
256	No	No	Method 3	51.62%
512	50%	No	Method 3	59.39%
256	No	Yes	Method 3	59.41%
512	50%	Yes	Method 3	60.39%
512	No	Yes	Method 2	79.92%
512	No	No	Method 2	80.64%
512	50%	No	Method 2	84.30%
256	No	No	Method 2	84.79%
512	50%	Yes	Method 2	85.51%
256	No	Yes	Method 2	86.07%
512	No	No	Method 1	86.15%
256	50%	Yes	Method 2	87.23%
256	50%	No	Method 2	87.27%
512	No	Yes	Method 1	90.43%
512	50%	No	Method 1	93.60%
512	50%	Yes	Method 1	94.27%
256	No	No	Method 1	95.02%
256	No	Yes	Method 1	95.86%
256	50%	No	Method 1	97.46%
256	50%	Yes	Method 1	97.61%

instances were predicted as indoor_flat_walk, these instances could be very early on in the experiment where the subjects were warming up and not walking too quickly.

4.3.1. Limitation

As the data is obtained from MAREA dataset, we have no control over the experiment conditions provided. It could likely be that to obtain meaningful results, 20 subjects of various heights/weights and walking styles were specifically obtained for the experiment. However, if this were the case then each user's activity (e.g. walking) would not be similar to another user's activity, this could be why the Different Subjects Split method has the worst results. The code was structured to allow for different options for window size, overlap percentage and hyper-parameters for wavelet transform/ peaks detection. However due to a lack of time an arbitrary decision was made to choose certain parameters, such as 50% overlap instead of 20%, 25%, 50%, 75%, etc. Many of the hyper-parameters were also pre-chosen in this same manner such as window ($2s$/$4s$) and the parameters of the wavelet transform level of 2.

Table 3: Accuracy score across models.

Model	**Accuracy**
XGBoost	76.43%
CNN	97.97%
LSTM	97.61%

Table 4: Accuracy score with different sensors and axes.

Sensor	**Axis**	**Accuracy**	**Remarks**
LF	Z	57.13%	
LF	X	58.36%	
LF	Y	73.49%	BEST 1 sensor, 1 axis
LF	X, Z	71.56%	
LF	X, Y	84.45%	
LF	Y, Z	86.38%	BEST 1 sensor, 2 axes
LF, Wrist	X, Y, Z	94.07%	
LF, RF	X, Y, Z	96.73%	
LF, Waist	X, Y, Z	96.96%	BEST 2 sensors, 3 axes
Wrist	X, Y, Z	62.87%	
RF	X, Y, Z	84.89%	
Waist	X, Y, Z	87.28%	
LF	X, Y, Z	90.28%	BEST 1 sensor, 3 axes

4.4. Future works

A future work would be further exploration of the various hyper-parameters and a more extensive search of their combinations. Also, other activities could be added into the dataset for more interesting inference.

4.5. Source code for reproducing results

The source code is freely available at `https://github.com/davidygp/a-comprehensive-comparison-of-data-processing-methods-for-human-activity-recognition`.

5. CONCLUSIONS

From our experiments we have determined that 1 sensor, the Left Foot (LF), is able to obtain 90.28% accuracy whereas including all 4 sensors only had an increase of 7.33% to 97.61%. This means that a single sensor is sufficient in meaningful separation and classification between the different human activities tested. It should also be noted that the Wrist sensor had the worst accuracy of 62.87% amongst the 4 single sensors, this could indicate that the purported activity readings from common wrist watches like Samsung Gear/Garmin series/Suunto are not very accurate in practice. We have also experimented with various signal processing methods and obtained optimal settings from the levels trialed. Furthermore, we have proven that a LSTM neural network can accurately differentiate between various human activities based solely on accelerometer-based data.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge Siddhartha Khandelwal and Nicholas Wickström from Halmstad University for sharing the MAREA (Movement Analysis in Real-world Environments using Accelerometers) gait datasets that is used in our experiment.

7. REFERENCES

[1] Siddhartha Khandelwal and Nicholas Wickström, "Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the marea gait database," *Gait & posture*, vol. 51, pp. 84–90, 2017.

[2] F Foerster, M Smeja, and J Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[3] Radoslaw M Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva, "Deep neural networks predict hierarchical spatio-temporal cortical dynamics of human visual object recognition," *arXiv preprint arXiv:1601.02970*, 2016.

[4] Raja Bose and Abdelsalam Helal, "Observing walking behavior of humans using distributed phenomenon detection and tracking mechanisms," in *2008 International Symposium on Applications and the Internet*. IEEE, 2008, pp. 405–408.

[5] Andreas Bulling, Ulf Blanke, and Bernt Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 33, 2014.

[6] TL Kasteren, Gwenn Englebienne, and BJ Kröse, "An activity monitoring system for elderly care using generative and discriminative models," *Personal and ubiquitous computing*, vol. 14, no. 6, pp. 489–498, 2010.

[7] Guto Leoni Santos, Patricia Takako Endo, Kayo Henrique de Carvalho Monteiro, Elisson da Silva Rocha, Ivanovitch Silva, and Theo Lynn, "Accelerometer-based human fall detection using convolutional neural networks," *Sensors*, vol. 19, no. 7, pp. 1644, 2019.

[8] Roman Chereshnev and Attila Kertész-Farkas, "Rapidhare: A computationally inexpensive method for real-time human activity recognition from wearable sensors," *Journal of Ambient Intelligence and Smart Environments*, vol. 10, no. 5, pp. 377–391, 2018.

[9] Hui Xing Tan, Nway Nway Aung, Jing Tian, Matthew Chin Heng Chua, and Youheng Ou Yang, "Time series classification using a modified lstm approach from accelerometer-based data: A comparative study for gait cycle detection," *Gait & posture*, vol. 74, pp. 128–134, 2019.

[10] Neelam Mehala and Ratna Dahiya, "A comparative study of fft, stft and wavelet techniques for induction machine fault diagnostic analysis," in *Proceedings of the 7th WSEAS international conference on computational intelligence, man-machine systems and cybernetics, Cairo, Egypt*, 2008, vol. 2931.

[11] Tianqi Chen and Carlos Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] Sepp Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[14] Nils Y Hammerla, Shane Halloran, and Thomas Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.