A Comparison of Different Models for Classifying Body Workout Activity

Yiming Wu[2702828], Chih-Chieh Lin[2700266], and Chaoran Li[2688458]

VU University, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
{wuyiming, c2.lin, c11.li}@student.vu.nl

Abstract. Mobile phones and other smart devices are usually equipped with many sensors, which enables user to record the data and gain insights from their daily activities by using machine learning methods. In this work, we collect data from some sensors to track their values when perform different body workout exercises. Outlier detection and missing values imputation are made after data aggregation. Then new features derived from original data are generated, with the hope to build better classification model. Finally, we train five different models to predict activities and evaluate their performances using some accuracy metric.

Keywords: Classification · Sensor Data · Feature Engineering

1 Introduction

Many sensors have been integrated on phones and smartwatches, which provides a method for users to collect data from the devices and track their daily activities [8]. The collected sensory data contains many valuable information. For example, avid runners might find some guides for their daily training to prepare them well for a Marathon race; while some patients may use heart rate and other measures to predict their health status.

In this work, we collected some exercises data from a number of sensors. To enable further analysis, we first pre-processed the raw data and removed outliers by exploring some single-variable plots. The missing values were imputed by k nearest neighbours (kNN) model. Then we generated new features in both time domain and frequency domain to boost the performance of our models. Furthermore, five models (four non-temporal models and one temporal model) were built and compared. Afterwards, we expected to find better approaches or models for our classification task. That is, we tried to find a relatively better model with higher accuracy on classifying user activities (body workout) by the given feature (sub)set.

2 Preparing Data

2.1 Data Collection

The data is obtained from a number of sensors during some body workout exercises of a user. Specifically, we collect sensory data in each exercise activity of the user by the “phyphox” application on the smartphone for about three minutes. We have tried to collect the data for longer period of time. However, it is hard to maintain the strength and the frequency on some workout actions(e.g., doing burpees for more than 5 minutes is exhausted).

Raw data from accelerator, linear accelerator, gyroscope and magnetometer in x, y and z axises, together with values of light sensor are recorded. The sensor rate, i.e., the frequency of data collection, is set to as fast as possible. Moreover, the smartphone was tied on the user’s right arm at the same position. Finally, consider the strength and the convenience of implementing, the following 10 body workout exercises are performed:

1. Burpees
2. Squat jump
3. High plank
4. Up to down plank
5. Mountain climber
6. Sit up
7. High knees
8. Jumping jacks
9. Bicycle crunch
10. Squat

2.2 Data Aggregation

After the raw data is collected, we need to aggregate it with a predefined granularity. Aggregation may reduce size of the dataset and eliminate noises to some degree. Also, aggregated data usually requires less computational resources to preform machine learning processes. In our case, the granularity is set to 0.25s, which leads to a dataset with sufficient information to train a fairly good model for activity classification. The new dataset contains 7253 samples approximately equally-divided into ten classes.

2.3 Outlier Detection

A sample point that has a significantly different feature value is called an outlier [6]. Outliers may cause serious problems in machine learning. To detect outliers in our dataset, we first make some exploratory analysis of samples across 13 features. The boxplot and scatter plot of all sample values in feature Magnetic field x are presented in Fig. 1.

Most features have similar results, except for feature Illuminance (lx), which is shown in Fig. 2

By inspecting these plots, we surmise that there might be outliers in feature Illuminance (lx). Thus, local outlier detection (LOF) [2] is employed (number of neighbours is set to 5) to detect such abnormal points. The reason to use LOF model is we don’t have enough prior information to assume a distribution to the samples. Also, simple distance based method may unexpectedly label points with low density as outliers. After LOF labels outliers of feature Illuminance (lx), the corresponding values are removed, which results in missing values in this variable.

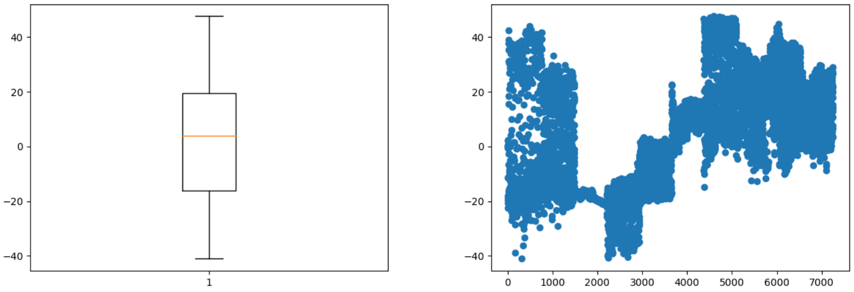


Fig. 1. Boxplot and scatter plot of all sample values in feature Magnetic field x

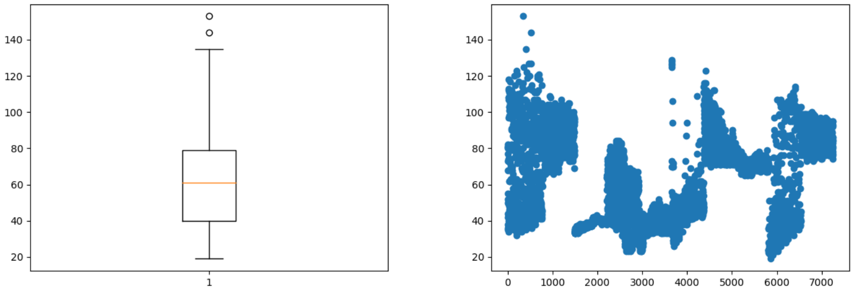


Fig. 2. Boxplot and scatter plot of all sample values in feature Illuminance (lx)

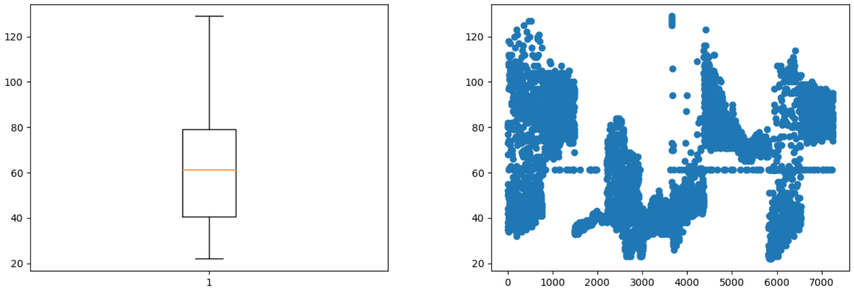


Fig. 3. Boxplot and scatter plot of all sample values after imputation in feature Illuminance (lx)

2.4 Missing Values Imputation

We utilize kNN model (k is set to 4) to impute these missing values. The reason is that the samples tends to have very similar values with its local neighbours. The result is shown in Fig. 3.

3 Feature Engineering

In this section, we focus on generating new features using our original dataset. These new features will not only be based on the time domain but also be generated from the frequency domain. Moreover, PCA features are also utilized in our work.

3.1 Time Domain

In the time domain, we decide to aggregate our features with a window size of 20. Median values in this dataset can reveal the middle value according to the given window size. Another feature we use is the standard deviation, which could tell us the degree of dispersion of this dataset's distribution; and further help us to find patterns when building our model. Fig.4 shows the aggregation results and original feature values of acc_x. The patterns showed in the new features are very similar to the original feature in this figure. We also add the median and standard deviation values of all the other features as new features in our dataset.

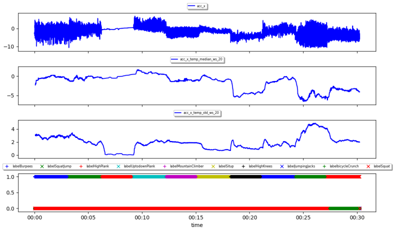


Fig. 4. Time Domain

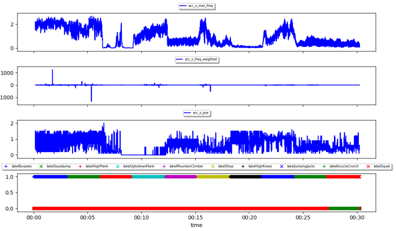


Fig. 5. Frequency Domain

3.2 Frequency Domain

For generating features in the frequency domain, we need to use Fourier transformation to transform our time domain features into the frequency domain. We decided to use real amplitudes with a specific frequency, the frequency with maximum amplitude, frequency weighted signal average, and power spectral entropy(PSE) to act as new features in the frequency domain. These features will give us an indication of the most important frequency in the windows and tell us if there are one or more different frequencies standing out of all others[8]. Fig.5 shows the aggregated features in the frequency domain of acc_x. We can see that almost every activity has its own patterns, for example, HighKnees always has low frequency and high PSE.

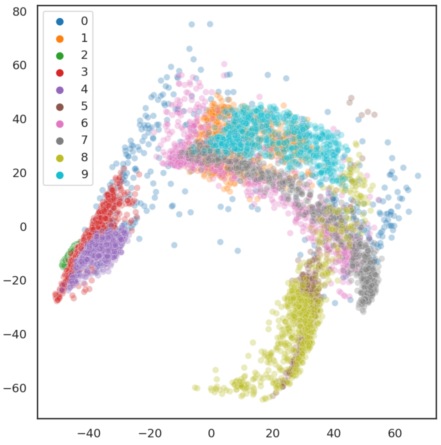


Fig. 6. 2D Plot of PCA

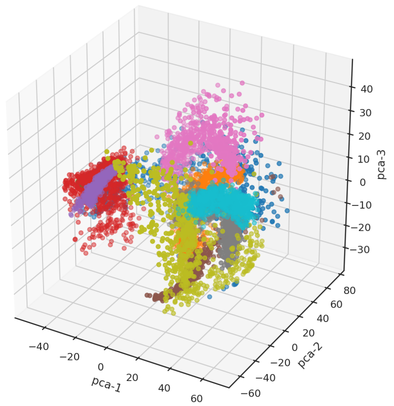


Fig. 7. 3D Plot of PCA

3.3 PCA Features

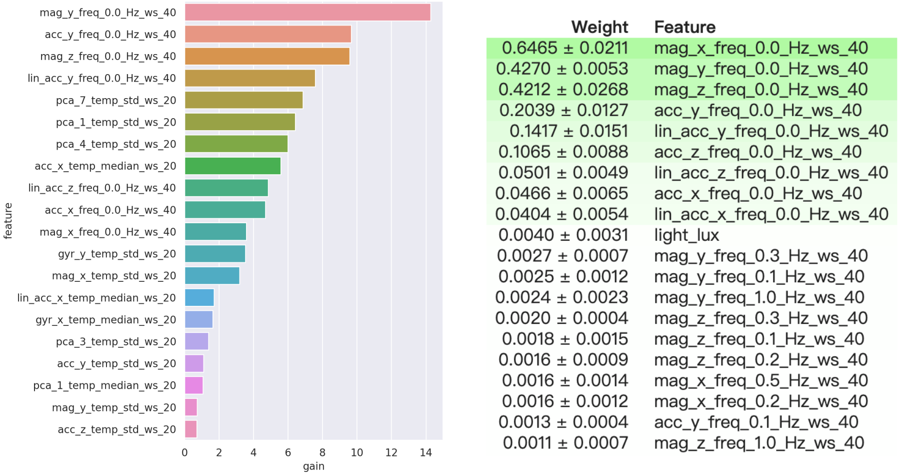
The PCA (Principal components analysis) values are also considered to be good features to build our model. The Fig.6 and Fig.7 show that the values of PCA have decent performance on representing the original features. Therefore, we incorporate these PCA features into our dataset for training the classification models.

3.4 Feature Importance

Two different approaches are utilized to figure out the feature importance in our training process. The first metric is provided by LightGBM package, whcih considers the “gain”(the average gain of the feature when it is used in trees) or “split”(the number of times a feature is used to split the data across all trees).

Another approach: permutation importance, which is calculated by the decrease of the score of a model when values of a feature are randomly assigned. If the score of the model decreases to a very low number when we replace the values of one feature with random values, it represents that this feature is important. This approach does not depend on a specific model[1].

The results generated by two approaches show in Fig.8. We could observe that the magnetic features have top places in both two figures. Also, PCA features have decent places in the first figure. Indeed, this result agrees our efforts on previous feature engineering work.



Weight	Feature
0.6465 ± 0.0211	mag_x_freq_0.0_Hz_ws_40
0.4270 ± 0.0053	mag_y_freq_0.0_Hz_ws_40
0.4212 ± 0.0268	mag_z_freq_0.0_Hz_ws_40
0.2039 ± 0.0127	acc_y_freq_0.0_Hz_ws_40
0.1417 ± 0.0151	lin_acc_y_freq_0.0_Hz_ws_40
0.1065 ± 0.0088	acc_z_freq_0.0_Hz_ws_40
0.0501 ± 0.0049	lin_acc_z_freq_0.0_Hz_ws_40
0.0466 ± 0.0065	acc_x_freq_0.0_Hz_ws_40
0.0404 ± 0.0054	lin_acc_x_freq_0.0_Hz_ws_40
0.0040 ± 0.0031	light_lux
0.0027 ± 0.0007	mag_y_freq_0.3_Hz_ws_40
0.0025 ± 0.0012	mag_y_freq_0.1_Hz_ws_40
0.0024 ± 0.0023	mag_y_freq_1.0_Hz_ws_40
0.0020 ± 0.0004	mag_z_freq_0.3_Hz_ws_40
0.0018 ± 0.0015	mag_z_freq_0.1_Hz_ws_40
0.0016 ± 0.0009	mag_z_freq_0.2_Hz_ws_40
0.0016 ± 0.0014	mag_x_freq_0.5_Hz_ws_40
0.0016 ± 0.0012	mag_x_freq_0.2_Hz_ws_40
0.0013 ± 0.0004	acc_y_freq_0.1_Hz_ws_40
0.0011 ± 0.0007	mag_z_freq_1.0_Hz_ws_40

Fig. 8. Feature Importance by LightGBM(left) and by Permutation(right)

4 Models

4.1 Non-temporal Model

Four non-temporal models are selected to finish our task; and would be described in this section.

Logistics Regression Although logistic regression(LR) is called regression, it is actually a classification model. The advantages of LR are simplicity, parallelization, and strong interpretability. The theory of LR is to assume that the data conforms to a specific distribution and is linearly separable, and then we can use maximum likelihood estimation to estimate the parameters[5].

Support Vector Machine Support vector machine(SVM) is a robust model using in supervise learning field. However, the basic SVM model is designed to deal with binary classification problems, and cannot directly handle multi-classification tasks. But we can use the calculation methods provided by standard SVM to build multiple decision boundaries in order to successfully classify datasets with more than two labels. The usual implementations are “one-against-all” and “one-against-one”[10]. In our case, the “one-against-all” is utilized in our SVM model.

Random Forest Random forest is an ensemble method that can deal with multiple labels dataset. To solve classification tasks, the random forest will give an output that is selected by most base models[7]. Every base model in the random forest is a decision tree. It always has a better performance than a single decision tree, but it is not as good as a gradient boost decision tree.

Gradient Boost Decision Tree The fourth model is GBDT (Gradient Boost Decision Tree), GBDT is an ensemble machine learning model that uses multiple decision trees as its base model. Decision trees in this model are not independent, because the newly added decision tree will learn some information about the incorrectly assigned samples so that the loss in each iteration will decrease and get a better result[4]. Finally, the prediction result is determined based on the sum of all decision tree's results[11]. In our case, the LightGBM package is selected to implement this approach.

4.2 Temporal Model

LSTM For the temporal model, a LSTM (Long Short-Term Memory) is selected to implement for classification task. Several previous work[3][12] were presented that the LSTM model was good at the task of human activity recognition.

However, unlike the non-temporal models which could predict the results by each data point, the LSTM would predict the results by each time serie we separate. Consider the frequency that we did each workout action, each time serie is set to be 4 seconds with 50 % overlap. That is, for instance, the series used to train and predict would be: (1s ∼ 4s), (2s ∼ 6s), (4s ∼ 8s), (6s ∼ 10s)...

5 Result and Evaluation

A common metric: Accuracy[9],is utilized to evaluate our models; and the corresponded formula was presented as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FT + FN}$$

Table 1. Results of Four Models

	Model	Accuracy on Test set
Non-temporal	LR	0.9178
	SVM	0.9834
	RF	0.9921
	GBDT	0.9999
Temporal	LSTM	1.0

Table 1. shows our result for five different model's accuracy on predicting body workout activity. The most successful model in non-temporal models is the one using GBDT. It only predicts three cases of false prediction in more than two thousands instances in the test set.

For the temporal model, the LSTM obtains the perfect results on predicting body workout activity as the accuracy derived was 1. Also, it could be observed in the confusion matrix at Fig. 10. That is, all the values which bigger than one are placed on the diagonal in the confusion matrix.

According to the result we obtained, the Random Forest, the GBDT and the LSTM are the comparatively better choice for classifying body workout activity in our case. Moreover, further discussion and details would be described in the next section.

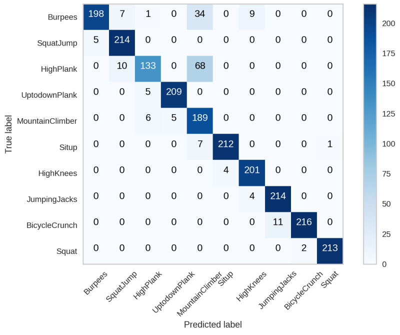


Fig. 9. Confusion Matrix of LR

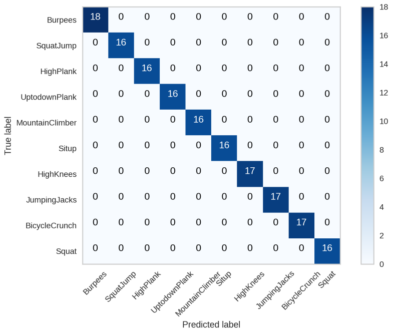


Fig. 10. Confusion Matrix of LSTM

6 Discussion and Conclusion

Undoubtedly, we derived satisfied results and obtained comparatively better models for the classifying body workout task. Several reasons why our model could perform well on this task are provided as follows.

First, the workout activities are selected after careful consideration. That is, the strength and variability of the activities are taken into consideration. We ensure that the selected actions are as various as possible in order to make models easier recognize the actions. However, it still has certain level of similarity between several actions. For instance, all of the burpees, high plank, and mountain climber, have the action of putting hands on floor and holding the body like a plank. We could observe that the LR model can not properly predict the results among these three actions in Fig. 9.

Second, the reason why LSTM have such great success on our task is that in addition to the one in the previous paragraph, the dataset we generated is quite small, and the time serie made for training and predicting enable LSTM to obtain more information than the non-temporal models do.

Finally, to conclude our article, we collected the sensor data after careful consideration in order to ensure the variability of activities. It ultimately brings us with great success on this classification task. Afterwards, after comparing five different models' performance on our classification task, we find relatively better models for this task.

References

1. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics **26**(10), 1340–1347 (2010)
2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
3. Chevalier, G.: Lstms for human activity recognition (2016)
4. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
5. Gourieroux, C., Monfort, A.: Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. Journal of Econometrics **17**(1), 83–97 (1981)
6. Grubbs, F.E.: Procedures for detecting outlying observations in samples. Technometrics **11**(1), 1–21 (1969)
7. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
8. Hoogendoorn, M., Funk, B.: Machine learning for the quantified self. On the art of learning from sensory data (2018)
9. Hossin, M., Sulaiman, M.: A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process **5**(2), 1 (2015)
10. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE transactions on Neural Networks **13**(2), 415–425 (2002)
11. Yuan, Y., Li, S., Zhang, X., Sun, J.: A comparative analysis of svm, naive bayes and gbdt for data faults detection in wsns. In: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). pp. 394–399. IEEE (2018)
12. Zhao, Y., Yang, R., Chevalier, G., Gong, M.: Deep residual bidir-lstm for human activity recognition using wearable sensors. CoRR **abs/1708.08989** (2017), http://arxiv.org/abs/1708.08989