Stacked Generalization with Wrapper-Based Feature Selection for Human Activity Recognition

Anjali Bhavan
Department of Mathematics
Delhi Technological University
New Delhi, India
anjaliblogger@gmail.com

Swati Aggarwal*
Department of Computer Engineering
Netaji Subhash Institute of Technology
New Delhi, India
swati1178@gmail.com
**Corresponding author*

***Abstract*—Human Activity Recognition has widespread usage in the fields of healthcare and human-centric computing, which is why it is important to build efficient and robust systems for accurate predictions for the same. Ensemble-based methods are also fast gaining acceptance for their ability to significantly enhance prediction quality and accuracy while also maintaining efficiency. In this context a stacked ensemble for predicting human activity as measured by a smartphone is described. Boruta, a wrapper-based all-relevant feature selection method is used before model training, and its effect on model metrics with filter-based methods and a hybrid of both methods compared. Stacking with Boruta gave an overall accuracy of 97.01%, which is an improvement over previous work (including improved accuracy in individual activities as well) and also better than simple variance-based filtering and the hybrid of both methods, which gave an accuracy of 94.07% and 93.43% respectively.**

Keywords—Ensembles, Boruta, Wrappers, Stacking, Human Activity Recognition

I. INTRODUCTION

Human Activity Recognition is a field that has been generating considerable interest in recent years, and has been studied using models like Convolution Neural Networks, Plurality Voting and Hidden Markov Models. Many approaches to this exist, such as vision-based and sensor-based recognition, where smartphone sensors are currently attracting most attention (due to convenience and ubiquity). Activity logging has several applications in daily life – for instance elderly care, location services, industry manufacture and biometric signature as every person's motion pattern is unique [1].

Several classification approaches have been proposed and studied for recognizing human activity from increasingly complex sequences of activities. He and Jin [2] proposed the usage of SVM with a discrete cosine transform for recognizing human activity from four categories namely running, jumping, standing and walking, which gave an accuracy of 97.51%. Mannini and Sabatini [3] proposed a sequential Hidden Markov Model for a different dataset for human activity recognition, which gave an accuracy of 95.6% without the Baum Welch algorithm applied after the first phase, and 98.6% with the algorithm applied as a second phase. Kwapisz, Weiss and Moore [4] developed models for predicting human activity using tri-axial cell phone accelerometers, out of which the Multi-Layer Perceptron had the best accuracy of 91.7%. Zhang and Sawchu [5] proposed a framework based on feature selection for human activity recognition employing multi-modal sensors, which achieved an accuracy of 90% using a single-layer framework, which was further improved by 3% using a multi-layer framework. Anguita, Ghio, Oneto, Parra and Reyes-Ortiz [6, 7], who also presented the dataset used in this work, proposed the use of SVM, and accuracy was reported to be 96%. Wu, Dasgupta, Ramirez, Peterson and Norman [8] employed and compared various algorithms present in the Weka toolkit and achieved 100% accuracy in the sitting activity. Ronao and Cho [9] proposed a Convolutional Neural Network framework that achieved an overall accuracy of 94.79%, which was improved to 95.75% with extra information obtained from the dataset – and a subsequent work on the same dataset using two-stage continuous Hidden Markov Models achieved an accuracy of 91.76% [10].

Ensemble learning has been demonstrated as a far superior and efficient machine learning method compared to single classifiers in several experiments, and has been used successfully in several Kaggle data science competitions as well [11]. Wolpert [12] introduced the concept of stacked generalization as a more sophisticated and efficient scheme compared with single classifiers or even other ensemble methods like bagging or boosting. Dzeroski and Ženko [13] constructed ensembles of diverse classifiers using stacking and demonstrated their performance as being comparable to selecting the best classifier from the ensemble by cross-validation. Ravi, Dandekar, Mysore and Littman [14]

978-1-5386-9276-9/18/$31.00

demonstrated that a Plurality Voting ensemble for human activity recognition displays the best performance across a wide range of settings on the Weka Toolkit.
Catal, Tufekci, Pirmit and Kocabag [15] used a voting ensemble comprising of Decision Trees, Logistic Regression and Multi-Layer Perceptron (MLP) models, and improved upon the work done by Kwapisz, Weiss and Moore [4] on the same dataset. This paper similarly aims to improve on the work done by Ronah and Cho [9, 10] and Anguita, Ghio, Oneto, Parra and Reyes-Ortiz [6, 7] in terms of accuracy and overall algorithm efficiency. This is significant and an improvement on previous work done in this field.

II. METHODOLOGY

A. Dataset information:

The dataset was released publicly by Anguita, Ghio, Oneto, Parra and Reyes-Ortiz [6] and is now available in the UCI Machine Learning Repository. It consisted of 561 attributes and a total of 10299 samples, which were split 70-30% to yield 7352 training examples and 2947 testing examples. Table 1 [6] describes the list of measures for computing feature vectors.
There were a total of 6 classes for prediction (Walking, Laying, Sitting, Walking Upstairs, Walking Downstairs and Standing), and the string class labels were converted to numeric values for ease of computation using the mapping in Table 2.

A. Data pre-processing

All subsequent work is done using the Scikit-learn [16] and Mlxtend [17] libraries.

TABLE I. LIST OF MEASURES FOR COMPUTING FEATURE VECTORS

Function	Description
Mean	Mean value
std	Standard deviation
mad	Mean absolute deviation
max	Largest value in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of squares
iqr	Interquartile range
entropy	Signal Entropy
arCoeff	Autoregression coefficients
correlation	Correlation coefficient
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
skewness	Frequency signal skewness
kurtosis	Frequency signal kurtosis
energyBand	Energy of frequency interval
angle	Angle between two vectors

- Outlier detection

We first examined the data to detect outliers using Isolation Forest feature in Scikit-learn, which randomly selects features and then a random split value is chosen between the maximum and minimum value of the selected feature. The number of splits is equivalent to the path length, which is quite noticeably short in case of an outlier. So a sample is an outlier if the forest repeatedly produces shorter path lengths for it. Running Isolation Forest returned zero outliers (indicating a clean dataset), so we moved ahead to the next step of data pre-processing.

- Feature selection

Since the feature vector was very high-dimensional, it was necessary to apply feature selection methods to remove unnecessary features and keep only the relevant ones. There are two types of feature selection methods i.e. filter-based [18, 19] and wrapper-based [20], out of which filter-based methods are computationally faster but ignore feature dependencies and interaction with the classifier, while wrapper-based methods, though more computationally expensive than filter methods, take into account interactions with the classifier and feature dependencies and thus provide a far better feature subset. The risk of overfitting due to these methods is taken care of by K-Fold cross validation performed further on.
Filter-based methods are independent of the classifier. Many types of filter methods, like those based on correlation (Fisher's discriminant criterion) or simple statistical or variance-based tests like t-test are used for feature selection, while wrapper methods use the classifier to score the subsets of features based on their prediction performance [19].
In this work Boruta [21] is employed, an all-relevant wrapper-based feature selection method. Instead of the *minimal optimal* problem, which focuses on coming up with the minimum possible subset which can provide the best accuracy, an *all-relevant* method like Boruta focuses on finding all attributes which are relevant in terms of classification and prediction – a more difficult problem [21].
So for solving such problems, filter-based feature selection methods cannot be used because the fact that a given feature is not important cannot be concluded simply from an absence of direct correlation between features and decisions [21], hence Boruta, a wrapper-based method comes into the picture.

TABLE II. MAPPING FROM STRING CLASS LABELS TO INTEGER

Activity	Assigned Numeric Value
Standing	0
Sitting	1
Walking	2
Laying	3
Walking upstairs	4
Walking downstairs	5

We compare the model performance on the feature subset produced by Boruta with that produced by a variance-based filter feature selection method (that eliminates all those features with variance beyond a certain threshold) and also a hybrid of these two: the feature subset produced from Boruta is next fed into the filter method, and the final feature subset is used for prediction.
Number of selected features of each method:
Boruta reduced the size of the feature vector from 561 to 482 (in 100 iterations), while variance-based filter method reduced it to 212, and the hybrid method reduced it to 182.

B. Model building and evaluation

Our ensemble consisted of the following classifiers:

- Random Forest
- Logistic Regression
- Support Vector Machine with linear kernel
- Multi-Layer Perceptron

with Logistic Regression as meta-classifier. We used the StackingCV classifier provided in Mlxtend, which modifies the standard stacking procedure for better performance.
The standard procedure consists of training the base classifiers on the training set, and feeding their predictions as input to the meta-classifier, which in turn gives the final predictions. In stacking with cross-validation, the dataset is split into K folds, and K-fold cross validation is performed on each base classifier before giving input to the meta-classifier. Cross-validation, particularly 10-fold (which has been used in this work) is one of the best methods for assessing the model building process, and enhances the efficiency of the algorithm.
The ensemble was thus trained, and 10-fold cross validation was performed. For tuning the hyper-parameters, Grid Search was used. Tables III and IV demonstrate the performance of the base classifiers and the stacked ensemble during 10-fold cross validation.

TABLE III. CROSS VALIDATION SCORE OF BASE CLASSIFIERS

Model	10-fold cross validation score
Logistic Regression	95%+/-5%
Random Forest	92%+/-4%
Multi-Layer Perceptron	93%+/-3%
SVM with Linear Kernel	95%+/-5%

TABLE IV. CROSS VALIDATION SCORE OF STACKED ENSEMBLE

Model	10-fold cross validation score
Stacked ensemble	94%+/-5%

III. EXPERIMENTAL RESULTS

We choose accuracy and F1 scores as our metrics for model performance. Accuracy is the ratio of the number of accurately predicted samples to the total number of samples, while F1 score is the weighted mean of precision and recall, given by

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \ldots(i)$$

Precision is the number of observations that have been correctly predicted positive over all the observations predicted positive, while recall is the number of observations that have been correctly predicted positive over the total quantity of observations in that class. Precision is a measure of how exact the model is: a system with high precision returns fewer results, but mostly correctly classified labels. Recall is a measure of how complete the model is: it returns more number of results, but most of its predicted values are not correct when compared to the training values. The trade-off between these two, which is a measure of the model's performance, is the F1 score.
Tables V, VI, VII and VIII summarize the accuracy and F1 scores obtained for the model and the reports for the best model:

TABLE V. MODEL ACCURACY

Classifier	Boruta	Variance based filtering	Boruta + variance based filtering
Random Forest + Multi-Layer Perceptron + Logistic Regression + SVM with linear kernel	97.01%	94.07%	93.43%

TABLE VI. MODEL F1 SCORE

Classifier	Boruta	Variance based filtering	Boruta + variance based filtering
Random Forest + Multi-Layer Perceptron + Logistic Regression + SVM with linear kernel	97%	94%	93%

TABLE VII. CLASSIFICATION REPORT OF BEST MODEL

Activity	Precision	Recall	F1-Score	Support
Standing	92%	98%	94%	532
Sitting	97%	90%	93%	491
Walking	97%	100%	98%	496
Laying	98%	100%	99%	537
Walking Upstairs	99%	95%	97%	471
Walking Downstairs	100%	98%	99%	420
Average/total	97%	97%	97%	2947

TABLE VIII. CONFUSION MATRIX OF BEST MODEL

Activity	Standing	Sitting	Walking	Laying	Walking Upstairs	Walking Downstairs	Accuracy
Standing	519	13	0	0	0	0	97.55%
Sitting	48	443	0	0	0	0	90.22%
Walking	0	0	494	2	0	0	99.59%
Laying	0	0	0	537	0	0	100%
Walking Upstairs	0	2	14	8	447	0	94.9%
Walking Downstairs	0	0	3	0	5	412	98.09%

IV. CONCLUSIONS

The algorithm demonstrates 100% accuracy for the laying activity, which is better or comparable with the other work done on this dataset. It also improves upon the works previously mentioned in the walking (99.59%, with just two samples misclassified), sitting (90.22%) and standing (97.55%) activities.

Wrapper-based methods surpass filter-based methods for feature selection in multiple kinds of ensembles and situations, and also perform better than a hybrid feature selection method consisting of both filter and wrapper methods. Their limitation, however, is the time they take for execution – again, which is comparable to the time and effort required for deep learning methods or the number of parameters to be set for Hidden Markov Models.

Nevertheless, it is important to establish valid comparisons with the performance of various deep learning methods as described in [22], which focuses on variations on deep networks and reinforcement learning applied in mining data in the biological domain. For instance, Zhao, Yang, Chevalier and Gong [23] demonstrated the usage of bi-directional long short term memory networks (Bi-LSTM) and various other

deep networks on the same dataset as used in this paper – and reported an overall best accuracy of 93.57%.

The results demonstrate that stacked ensembles can be used successfully for human activity recognition, and provide computationally efficient, accurate and robust classification compared to the prevalent methods.

Our algorithm, however, does not do as well on the walking upstairs activity (94.9%) as compared to others. This is a possible scope for improvement in further work on this ensemble. There is also apparent confusion on the model's part in differentiating between standing and sitting activities – due to their orientations being relatively similar, which warrants the need of another sensor to appropriately differentiate between the two. The model could be trained to differentiate between standing and sitting better as well, which will constitute future research. Future work could also include ways to even further improve computational efficiency and work on the walking-upstairs activity. Our future plans also include creating our own dataset and adding improvements to our current work.

REFERENCES

[1] Sunny, J. T., George, S. M., Kizhakkethottam, J. J. "Applications and challenges of human activity recognition using sensors in a smart environment." *IJIRST Int. J. Innov. Res. Sci. Technol* 2 (2015): 50-57.

[2] He, Z., & Jin, L. (2009, October). Activity recognition from acceleration data based on discrete consine transform and SVM. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 5041-5044). IEEE.

[3] Mannini, A., & Sabatini, A. M. "Machine learning methods for classifying human physical activity from on-body accelerometers." *Sensors* 10.2 (2010): 1154-1175.

[4] Kwapisz, Jennifer R., Gary M. Weiss, and Samuel A. Moore. "Activity recognition using cell phone accelerometers." *ACM SigKDD Explorations Newsletter* 12.2 (2011): 74-82.

[5] Zhang, M., & Sawchuk, A. A. "A feature selection-based framework for human activity recognition using wearable multimodal sensors." *Proceedings of the 6th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[6] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. "A Public Domain Dataset for Human Activity Recognition using Smartphones." *ESANN*. 2013.

[7] Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine." *International workshop on ambient assisted living*. Springer, Berlin, Heidelberg, 2012.

[8] Wu, W., Dasgupta, S., Ramirez, E. E., Peterson, C., & Norman, G. J. "Classification accuracies of physical activities using smartphone motion sensors." *Journal of medical Internet research* 14.5 (2012).

[9] Ronao, C. A., & Cho, S. B. "Human activity recognition with smartphone sensors using deep learning neural networks." *Expert Systems with Applications* 59 (2016): 235-244.

[10] Ronao, C. A., & Cho, S. B. "Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models." *Natural computation (ICNC), 2014 10th international conference on*. IEEE, 2014.

[11] Hendrik Jacob van Veen, Le Nguyen The Dat, Armando Segnini. 2015. Kaggle Ensembling Guide. [accessed 2018 Feb 6]. https://mlwave.com/kaggle-ensembling-guide/

[12] Wolpert, David H. "Stacked generalization." *Neural networks*5.2 (1992): 241-259.

[13] Džeroski, Saso, and Bernard, Ženko. "Is combining classifiers with stacking better than selecting the best one?." *Machine learning* 54.3 (2004): 255-273.

[14] Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. "Activity recognition from accelerometer data." *Aaai*. Vol. 5. No. 2005.

[15] Catal, C., Tufekci, S., Pirmit, E., & Kocabag, G. "On the use of ensemble of classifiers for accelerometer-based activity recognition." *Applied Soft Computing* 37 (2015): 1018-1022.

[16] "Scikit-learn: Machine Learning in Python", Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[17] Sebastian Raschka, Reiichiro Nakano, James Bourbeau, Will McGinnis, Guillaume Poirier-Morency, Colin, … Adam Erickson. (2018, March 15). rasbt/mlxtend: Version 0.11.0 (Version v0.11.0). Zenodo. http://doi.org/10.5281/zenodo.1198892

[18] Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." *Journal of machine learning research* 3.Mar (2003): 1157-1182.

[19] Ladha, L., and T. Deepa. "Feature selection methods and algorithms." *International journal on computer science and engineering* 3.5 (2011): 1787-1797.

[20] Chrysostomou, Kyriacos. "Wrapper feature selection." *Encyclopedia of Data Warehousing and Mining, Second Edition*. IGI Global, 2009. 2103-2108.

[21] Kursa, Miron B., and Witold R. Rudnicki. "Feature selection with the Boruta package." *J Stat Softw* 36.11 (2010): 1-13.

[22] Mahmud, Mufti, Mohammed Shamim Kaiser, Amir Hussain, and Stefano Vassanelli. "Applications of deep learning and reinforcement learning to biological data." IEEE transactions on neural networks and learning systems 29, no. 6 (2018): 2063-2079.

[23] Yu, Zhao, Yang Rennong, Chevalier Guillaume, and Gong Maoguo. "Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors." arXiv preprint arXiv:1708.08989 (2017).

[24] Lara, Oscar D., and Miguel A. Labrador. "A survey on human activity recognition using wearable sensors." *IEEE Communications Surveys and Tutorials* 15.3 (2013): 1192-1209.

[25] Sewell, Martin. "Ensemble learning." *RN* 11.02 (2008)