Feature selection of non-intrusive load monitoring system using RFE and RF

Zhicheng Zhu, Zhiqiang Wei*, Bo Yin, Tao Liu and Xianqing Huang
Ocean University of China, Qingdao, China

*Corresponding author e-mail: zhuzcwork@126.com

Abstract. Feature selection is a process of selecting relevant features and removing the irrelevant and redundant ones based on certain specific criteria in the original data set. This paper proposes the random forest algorithm (RF) based on recursive feature elimination (RFE) to select features in non-intrusive load monitoring system. Adopting the random forest algorithm as the basic approach, we repeatedly build models and filter features to identify the best subset of features. Last, the results of experiments using public data sets verify the effectiveness of the proposed algorithm.

1. Introduction
Energy plays such a crucial part in advancing our economy and social development. As one of the best secondary sources of energy, electricity is the essential foundation for development. To get the reliable detailed information about energy consumption, we use the real time monitoring approach as the basic tool to carry out the energy efficiency work and learn more accurate power usage data about all types of appliances. After fully understanding how their electricity is consumed, users can make informed decisions when buying different kinds of household appliances and choosing the right time and right way to use them. In this way, the citizens’ everyday life and working life won’t be affected and on the other hand, we reach the goal of reducing energy consumption and cost [1]. It is against this backdrop that non-intrusive load monitoring is invented.

Non-intrusive load monitoring serves as the core technology of power user management. Through load identification technology, users can get the load usage information in time. Feature extraction, a key step in non-intrusive load monitoring, uses signal processing techniques to extract features from voltage (V) and current (I) measurements. The ultimate goal of feature extraction is to obtain a signature (operating features or feature combination) that can uniquely identify single device. The performance of any non-intrusive load monitoring system mainly depends on the uniqueness of the device’s signature. Therefore, selecting the suitable features or feature combination is of crucial importance to improve the load identification capability of non-intrusive load monitoring system.

To precisely filter features, our research should focus on the feature selection algorithm. Feature selection algorithm refers to a process of removing irrelevant and redundant features and selecting relevant features from the original data set according to specific evaluation criteria. The subset of selected related features is known as the optimal subset of features, which has lowest dimension in feature space but contains most important information [2]. However, as the number of features continuously increases, the traditional feature selection algorithm cannot achieve accurate filter results. Therefore, this paper proposes a random forest algorithm based on recursive feature elimination. To find the optimal subset of features, we repeatedly build models and filter features over and over again. The results of experiments with data sets verify the effectiveness of the proposed algorithm.
2. Related work
There are three main approaches of feature selection: Filter, Wrapper and Embedded [3]. Filter algorithms, also known as variable ranking, perform variable ranking at the preprocessing stage to quickly eliminate some of the non-critical noise features and narrow the search range of optimized feature subsets, but it cannot guarantee a small optimal subset of features would be selected. The Pearson correlation coefficients and mutual information are two classic ranking criteria recently proposed by Lazar and other researchers [4]. Its main disadvantage is that algorithm performance is not taken into consideration while choosing the variables.

The embedded method integrates the feature selection into the training process of learning machine and realizes the feature selection by optimizing an objective function in training the classifier. The main example is lasso regression and decision tree such as the automatic selection of CART algorithms. The method of packing trains the classifier directly using the selected subset of features in the process of filtering features, and the size of the feature subset is relatively small. There’re also limitations in this approach. Since it is not possible to evaluate all subsets of variables, the wrapping methods consist of greedy strategies, such as forward or backward algorithms, when it comes to large amount of data. This algorithm is widely applied in many documents [5].

A typical problem with variable selection methods is their instability: a small perturbed training sample can completely change the set of selected variables. This problem results from the data complexity in such high-dimensional settings, especially when it comes to highly correlated features, the instability of variable selection increases [6]. For example, Buhlmann and other researchers point out that lasso tends to abandon most of the correlated variables, even though they discriminate and randomly select one to represent a set of related predictions.

Random forest algorithm, proposed by Breiman, is a modification of bagging algorithm that collects a large number of tree-based estimators. This approach has better estimation performance compared to a single stochastic tree. Although the estimates for each tree are lower, the high variance realizes the bias variance trade-off [7]. This proposed random forest algorithm based on recursive feature elimination in this paper still can find the optimal subset of features even though the number of features grows continuously.

3. Feature selection algorithm

3.1. Random forest algorithm
In 2001, Breiman put forward an effective integrated learning classification algorithm - random forest algorithm. Not only is it a classification algorithm with universal application, but also a method for data dimensionality reduction. It has wide range of application and can avoid overfitting, achieving better results than decision tree.

Random forest algorithm is a combination of Bagging algorithm and Random Subspace algorithm, with decision tree as basic building block. Through the combination of multiple decision trees \( h_1(x), h_2(x), ..., h_{n\text{Tree}}(x) \) (could be binary or multi-way tree) to improve classification accuracy. At this point, we get a random forest classifier (Fig. 1), and classify unknown samples by final vote.
The generalization is affected to some extent by the randomness of random forest algorithm. To address the above-mentioned shortcomings, we recur to recursive feature elimination algorithm to improve random forest algorithm.

3.2. Random forest algorithm based on recursive feature elimination

RFE-RF initializes the needed feature set to the entire data set. Each time the data with lowest ranking criteria score is removed until the last feature set is obtained. This process therefore shows RFE-RF is a sequential backward selection algorithm based on maximum interval principle of RF. The ranking criteria score for the feature i in RFE-RF is defined as (1).

$$c_i = w_i^2$$ (1)

In each iteration, the feature with lowest ranking criteria score is removed, and then the remaining features are used to train the RF for the next iteration. The specific algorithm flow is as follows:

1. Preprocess the original data, and change the set X into (n (n-1))/2 subsets $X_j$. $X_j$ denotes the total number of j training samples.

$$X_j = \begin{cases} 
\{(x_i, y_i)\}_{i=1}^{N_1+N_j}, & j = 1, 2, \ldots, n-1 \\
\{(x_i, y_i)\}_{i=1}^{N_2+N_j+n+3}, & j = n, 2, \ldots, 2n-3 \\
\{(x_i, y_i)\}_{i=1}^{N_{n-1}+N_n}, & j = (n(n-1))/2 \\
\end{cases}$$

$$v_i = j - n + 3, y_i = -1;$$

$$v_i = n - 1, y_i = 1;$$

2. Use RFE-RF to select features from data set $X_j$, the results and get the corresponding feature subset $F_j \subseteq \{1, 2, \ldots, D\}, j = 1, 2, \ldots, (n(n-1))/2$.

3. Combine feature subset $F_j$ to obtain the final feature subset by using (2).

$$F = \bigcup_{j=1}^{(n(n-1))/2} F_j$$ (2)
4. Experiment

4.1. Data preparation

In the experiments, we take the selected PLAID data for algorithm verification, mainly because of its long monitoring period and large amount of data. Short-term monitoring data cannot completely and truly reflect household energy consumption due to human factors, weather and other causes.

PLAID data collection equipment collects thousands of current and voltage data of 11 different types of load from as many as 56 households, Since the load characteristics have to be calculated from both current and voltage data, we remove some of the failure data from the data set. There’re a total of 737 records left that are shown in TABLE 1.

| Load type            | Number of records |
|----------------------|-------------------|
| Hair dryer           | 121               |
| Microwave oven       | 115               |
| Fluorescent lamp     | 110               |
| Electric fan         | 102               |
| Energy-saving lamp   | 100               |
| Air conditioner      | 55                |
| Vacuum cleaner       | 36                |
| Electric heater      | 31                |
| Refrigerator         | 29                |
| Dish washer          | 20                |
| Laptop               | 18                |

Using these data, we can calculate 38 steady-state features and 14 transient features. Next, an RFE-RF based feature selection algorithm is evaluated using a leave-one-out-cross-validation method for each family in the dataset.

4.2. Performance demonstration of feature selection algorithm

It can be seen from Fig. 2 that as the number of load characteristics increases, the random forest algorithm rapidly rises in the cross-validation score. Its accuracy tends to stay steady after about 20 features, and then remains almost unchanged as the features continue to increase.

![Figure 2. Results of Load character cross-validation schematic diagram of random forest.](image-url)
Therefore, Gini index is not an effective way to select the optimal subset, because most of the features are highly correlated with each other, and the random nature of the algorithm makes it more difficult to choose threshold. Therefore, we adopt a relatively conservative approach - only removing the 12 most lower-ranking features.

To further reduce feature range, we employ the random forest algorithm to continue training the rest of features. For features with low impact, many features have similar behavior, so no obvious threshold can be found. We therefore have to use that conservative method again to remove the relatively unimportant features among the remaining load features. The total number of features drops from 40 to 30. The experiment results of this step are shown in in Fig. 3.

![Figure 3. Average accuracy of 30 selected features.](image)

Then we continue iterating to eliminate feature, using the random forest to continuously train the rest of features and rank them based on average accuracy. The total number of features falls to 20 after the third around of feature selection and to 10 after the fourth around. The experiment results are shown in Fig. 4 and Fig. 5.

![Figure 4. Average accuracy of 20 selected features.](image)

It should be noted that average accuracy is improved compared to previous iteration as the number of retained feature decreases after each iteration and feature changes in subsequent iterations. This result confirms that the load features with more influence will be reserved after feature selection.
Figure 5. Average accuracy of 10 selected features.

Finally, when there’re only ten 10 features left after iteration, the average accuracy becomes lower than the previous feature selection process. This indicates that the load features with low impact have been completely excluded and the final results would be affected if we continue to remove the rest of load features. So far, this feature selection process comes to end.

4.3. Specific load analysis

In this section, we will further discuss algorithm performance from the perspective of specific load. First, the F-score is employed to analyze the changes of each load during the process of load feature selection, which is shown in Fig. 6.

Figure 6. Change of each load manifested in feature selection process.

Fig. 6 illustrates that the recognition rate for all 11 types of loads in the PLAID dataset basically experiences an upward trend in every round of feature selection. Also, the lower the initial F-score of a load is, the more significantly the recognition rate increases. In addition, considering the amount of load data, the loads with fewer entries in the data set have relatively poor F-score. It can be concluded that the more data that can be trained, the better results of the load feature selection algorithm.
To further analyze cause of load misclassification and to understand which loads are more likely to be misclassified, we study the changes of confusion matrices for each round of feature selection, as shown in Fig. 7.

Figure 7. Confusion matrix varies as the number of features changes: (a) 52 features, (b) 40 features, (c) 30 features, (d) 20 features, (e) 10 features

With the exception of diagonal cells, each cell shows the probability of the corresponding load that is misclassified. In general, the false detection rate of load is improved by iterative feature elimination for most loads, and new misclassifications are not created in the iteration. For example, air...
conditioners are misclassified into four types of loads when using 52 features. As feature selection continues until there’re only 20 features left, air conditions are misclassified into three types and the false detection is reduced.

5. Conclusion
In this paper we propose an improved features selection method based on random forest algorithm and recursive feature elimination in NILM system, which can be used to accurately removing the irrelevant and redundant features. A series of experimental using public data sets verify the effectiveness of the proposed algorithm. Further research work may include optimization of algorithm operation efficiency and create a comprehensive data set (with a wide range of household appliances in different households) to perform our feature selection process.

Acknowledgments
This work was supported in part by China International Scientific and Technological Cooperation Special (2015DFR10490), Qingdao national laboratory for marine science and technology Aoshan science and technology innovation project (2016ASKJ07-4) and Qingdao innovation and entrepreneurship leading talent project (13-cx-2).

References
[1] T. Liang, C. Gao, and B. Wang. "Applications of demand side management in smart grid," Electric Power Automation Equipment, vol. 32, no. 5, pp. 81-85, May. 2012.
[2] H. Najmeddine et al., "State of art on load monitoring methods," Power and Energy Conference, pp.1256-1258, 2009.
[3] Archer K J, Kimes R V. Empirical Characterization of Random Forest Variable Importance Measures [J]. Computational Statistics and Data Analysis, 2008, (52).
[4] Lewis P. The characteristic selection problem in recognition systems [J]. Information Theory, IRE Transactions on, 1962, 8(2): 171-178.
[5] Auret L, Aldrich C. Empirical Comparison of Tree Ensemble Variable Importance Measures [J].Chemometrics and Intelligent Laboratory Systems, 2011, (105).
[6] Neville P G. Controversy of Variable Importance in Random Forests [J]. Journal of Unified Statistical Techniques, 2013, (1).
[7] Nicodemus K K. Letter to the Editor: On the Stability and Ranking of Predictors From Random Forest Variable Importance Measures [J]. Briefings in Bioinformatics, 2011, (12).