Distillation of Random Projection Filter Bank for Time Series Classification

Yufei Lin[1], Sen Li[1], and Qianli Ma[1,2](✉)

[1] School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China
yufeilincs@foxmail.com, awslee@foxmail.com

[2] Guangdong Key Laboratory of Big Data Analysis and Processing,
Guangzhou, China
qianlima@scut.edu.cn

Abstract. Time series is widely found in various fields such as geoscience, medicine, finance, and social sciences. How to effectively extract the features of time series remains a challenge due to its potentially complex non-linear dynamics. Recently, Random Projection Filter Bank (RPFB) [5] is proposed as a generic and simple approach to extract features from time series data. It generates the features by randomly generating numerous autoregressive filters that are convolved with input time series. Such numerous random filters inevitably have redundancy and lead to the increased computational cost of the classifier. In this paper, we propose a distillation method of RPFB, named D-RPFB, to not only maintain the high level of quantity of the filters, but also reduce the redundancy of the filters while improving precision. We demonstrate the efficacy of the features extracted by D-RPFB via extensive experimental evaluation in three different areas of time series data with three traditional classifiers (i.e., Logistic Regression (LR) [2], Support Vector Machine (SVM) [14] and Random Forest (RF) [8]).

Keywords: Random projection · Filter bank · Time series
Feature extraction

1 Introduction

Time series data are ubiquitous in many practical applications ranging from health care [3], action recognition [10], financial markets [15] to urban traffic control [16]. How to extract the features of time series effectively is a popular research topic [4,5,7,9,13]. However, time series extraction remains a challenging task due to the potentially complex non-linear dynamic system behind the time series.

Recently, Random Projection Filter Bank (RPFB) [5] is proposed as a generic and simple approach to extract features from time series data. RPFB is a set of randomly generated stable autoregressive filters that are convolved with the input time series to generate the features. These features can be used by any

© Springer Nature Switzerland AG 2018
J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 586–596, 2018.
https://doi.org/10.1007/978-3-030-03338-5_49

conventional machine learning algorithm for solving tasks such as time series prediction, classification with time series data, etc. Different filters in RPFB extract different aspects of the time series, and together they provide a reasonably good summary of the time series.

However, numerous random filters inevitably have redundancy and lead to the increased computational cost of classifier. Moreover, in some cases, redundant features will make the performance of classifier worse. How to reduce redundant features (i.e., estimate the quality of the filter) is an important issue. In this paper, with an aim of reducing the number of redundant filters, we propose a way to distil the filters of RPFB, named D-RPFB, which uses a set of specific rules to filter the filters that are most capable of guiding the classifier to get better performance. D-RPFB can reduce the number of redundant and even potentially mislead filters, thus improving the quality of the features provided to the classifier which directly improves the learning ability of the classifier and obtains a better performance.

2 Preliminaries

There is a crucial process for the distillation of RPFB, which is designed to measure the quality of a specific filter. To do that, we introduce entropy [6]. Considering that entropy is not very common in time series analysis, we first introduce the concept of entropy briefly before proposing our D-RPFB formally.

Entropy [6] is often used in information theory and probability statistics to measure the uncertainty of a variable. Entropy is always a real number larger than 0 but smaller than 1. Its value indicates the degree of uncertainty of random variables. When the entropy is equal to 0, the random variable is completely certain without any randomness. When entropy is equal to 1, the uncertainty of the random variable peaks. This property of entropy makes it possible to use the entropy to measure the classification quality of the classification subset when a classifier uses a single feature extracted by certain filter to classify an instance. The smaller the entropy of a subset, the more the feature extracted by the filter can make the classifier better complete the clustering, and vice versa, the greater the entropy value indicates that the feature extracted by the filter may lead to the confusion of the classification results.

3 Proposed Methods D-RPFB

3.1 Brief Review of Random Projection Filter Bank

The idea behind RPFB is to randomly generate many simple dynamical systems (i.e., $\frac{1}{1-Z_n^{'}z^{-1}}$ denotes a certain simple dynamical system with a given pole $Z_n^{'}$ and z^{-1} denotes the inverse of z-transform [11]) that can approximate optimal dynamical systems with a high accuracy.

In order to do this, what we should do first is to determine the number of filters in the filter bank. After that, given the certain number of filters N, we draw

N random real numbers or the imaginary numbers $Z_1^{'}, \cdots, Z_n^{'}$ from the unit circle to construct a filter bank defined by filter $\phi(z^{-1}) = (\frac{1}{1-Z_1^{'}z^{-1}}, \cdots, \frac{1}{1-Z_n^{'}z^{-1}})$ which contains N random projection filters. Then, we pass each input time series through every filter in RPFB to do convolution and generate N features corresponding to each time series at each time step. For example, assuming the length of the each input time series is T, we will get $N * T$ features after passing it through RPFB. Finally, we can input the obtained features into different classifiers for conducting time series classification.

3.2 The Distillation of Random Projection Filter Bank

Introduce the Entropy into Time Series. The entropy is used in the traditional decision tree ID3 algorithm [12] for feature selection. That motivates us to use entropy to evaluate the quality of a certain filter. However, in the traditional decision tree ID3 algorithm [12], the entropy is only applicable to a discrete variable. To solve this issue, we use an extra classifier to introduce the entropy into time series and achieve the purpose of evaluating the quality of a certain filter. In general, assuming the length of the each input time series is T, we will get T features through time after passing it through a certain filter. Then, we input the T features into a certain classifier to get the classification result. In this way, for each time series example, we get a classification result which makes a certain filter become a discrete variable. And, we propose evaluation method combined with entropy and classification result to evaluate the quality of a certain filter.

Computation of Subset Uncertainty and Evaluation of Filters. After using RPFB to generate filter, each filter will be executed with the proposed evaluation algorithms to get their evaluation value. The overall algorithm flow is shown in Algorithm 1. First, in the training data set, randomly select the same number of instances in each category to form data set D_m for avoiding unbalanced sample. For each filter in RPFB, randomly select the half number of instances in D_m as training data D_t, the other half as validation data D_v and then pass the train and valid data into the filter, extracting the corresponding features (denoted by F_t and F_v). Then, fitting the classifier with the F_t. When the remaining features F_v are classified by the classifier, each category (totally M category) will produce a corresponding subset $D_m^{'}$. Each subset $D_m^{'}$ may contain the instances that belong to the subset or contains instances that do not belong to the subset. Thirdly, we can calculate the uncertainty of each subsets $D_m^{'}$ by entropy. If the uncertainty of the subset $D_m^{'}$ is small it means that $D_m^{'}$ contains many instances of the same category, which means that the feature extracted by the filter can guide the classifier to complete the clustering of the time series. However, only clustering results cannot evaluate whether a filter is really efficient because if a subset $D_m^{'}$ contains many instances of the same category that do not belong to $D_m^{'}$, the feature extracted by the filter is quite bad which misleads the classifier. Therefore, we have to consider the classification accuracy as the second characteristics of each subset $D_m^{'}$. In this way, the two important measurements, the clustering effect and the classification accuracy are both considered. Both of them are equally important for evaluating the quality of the feature extracted by a filter. Therefore, D-RPFB proposes a method for calculating the evaluation value of a certain filter as follow:

Algorithm 1. The distillation of random projection filter bank

```
   Input: Dataset = $(X_{i,1}, Y_{i,1}), \cdots, (X_{i,T_i}, Y_{i,T_i})_{i=1}^{m}$
   Output: Classifier $\hat{f}$ and new filter bank $\phi_{new}$
1  $l : Y^{'} \times Y \rightarrow \mathbb{R}$ : Loss function;
2  $\mathcal{F}$ : Function space;
3  $n$ : The number of filters in random projection filter banks;
4  $\rho$ : The percentage of remaining filters after the screening filter;
5  Draw $Z_1^{'}, \cdots, Z_n^{'}$ uniformly random within the unit circle.
6  Define filter $\phi(z^{-1}) = (\frac{1}{1-Z_1^{'}z^{-1}}, \cdots, \frac{1}{1-Z_n^{'}z^{-1}})$.
7  In the training data set, randomly select the same number of instances in each
   category to form data set $D_m$.
8  foreach $\phi_i(z^{-1})$ in $\phi(z^{-1})$ do
9      Pass each time series in $D_m$ through filter $\phi_i(z^{-1})$.
10     Randomly select the half number of instances in $D_m$ as training data $D_t$,
       the other half as valid data $D_v$.
11     Input the corresponding features $F_t$ generated by training data $D_t$ in step 9
       into the classifier to fit the model. (The type of classifier used here is the
       same as the $f$ in line 19.)
12     Input the corresponding features $F_v$ generated by valid data $D_v$ into the
       fitted model got by step 11 to get the classification subsets $D_m^{'}$.
13     Use Equation (3) to calculate the evaluation $E_{\phi(z^{-1})}$ of the filter $\phi_i(z^{-1})$.
14 end
15 Sort all filters according to their evaluation value $E_{\phi(z^{-1})}$.
16 Select the corresponding number of filters based on $\rho$ with higher evaluation
   values to form new filter banks $\phi_{new}$.
17 Pass each time series in training set through every filter in the new filter bank
   $\phi_{new}$.
18 Use the new extracted features $(X_{i,1:T_i}^{'})$ generated by new filter bank $\phi_{new}$ to
   construct the estimator, we use regularized empirical risk minimization to solve
   it and $J(f)$ controls the complexity of the function space:
19         $\hat{f} \leftarrow \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{m} \sum_{t=1}^{T_i} l(f(X_{i,t}^{'}, Y_{i,t})) + \lambda J(f)$
   where $l$ denotes the cross entropy cost function, $J$ can be lasso or ridge
   regression regularization.
20 Return $\hat{f}$ and $\phi_{new}$
```

$$H(D_m^{'}) = -\sum_{m=1}^{M} p_m log p_m \tag{1}$$

$$Recall_{D_m^{'}} = \frac{TP}{TP + FN} \tag{2}$$

$$E_{\phi(z^{-1})} = \sum_{m=1}^{M} (1 - H(D_m^{'})) \times (Recall_{D_m^{'}}) \tag{3}$$

where $H(D_m^{'})$ is the entropy of a classification subset of the filter i, M is the total number of category, p_m is proportion of an instance of M category in the classification subset $D_m^{'}$, TP is the number of the samples classified correctly in this category, $TP + FN$ is the number of the total samples in this category, $Recall_{D_m^{'}}$ is the recall of classification subsets $D_m^{'}$ and $E_{\phi(z^{-1})}$ is the total evaluation value E of the i filter.

4 Experiment

In order to verify that the proposed D-RPFB can reduce the redundancy of the numerous filters while also keeping or even improving the performance of classification, we evaluate it in three different areas of time series data with three traditional classifiers (i.e., LR, SVM and RF) compared with RPFB. First, we investigate the effect of the proposed evaluation method for measuring the quality of a specific filter. Then, we show the experimental results on other two time series. Finally, we give an analysis of the screening percentage of the filters to empirically decide how many filters should be retained.

4.1 Analyzing the Effect of the Proposed Evaluation on Star Curve Data Set

The proposed evaluation method in the Eq. (3) for measuring the quality of a certain filter plays an important role in our D-RPFB. We first investigate the effect of the proposed evaluation method on the Star curve data set [1]. We assess the effect of the Eq. (3) by answering the question: Can we use the Eq. (3) to get three group filter banks that correspond to an excellent, inferior, and average property and get the corresponding performance on the test set? If this happens, then the proposed evaluation method is considered to be effective.

Our experimental scheme is as follows. Firstly, a sufficient number of filters are generated to form an initial filter group. Then, we input a part of the training data and the initial filter bank into the filter method to get the evaluation of all the filters by Eq. 3). Third, sorting the filter by the respective evaluation value of E, we divide the filter into four intervals according to the evaluation value of E (i.e., $0 < E < 0.25$ for worst, $0.25 < E < 0.5$ for worse, $0.5 < E < 0.75$ for better, $0.75 < E < 1$ for best). Finally, we construct three group filter banks with 200 filters in each that corresponds to excellent, inferior, and average distribution by randomly selecting a specific number of filters in a specific interval to meet the scheme we need. The corresponding distribution is shown in Fig. 1.

Figure 2 shows clearly the ability of the evaluation method to distinguish high quality filters from inferior filters. Generally speaking, the classification error of the inferior distribution is far higher than the classification error rate of the average distribution and the classification errors of the excellent distribution

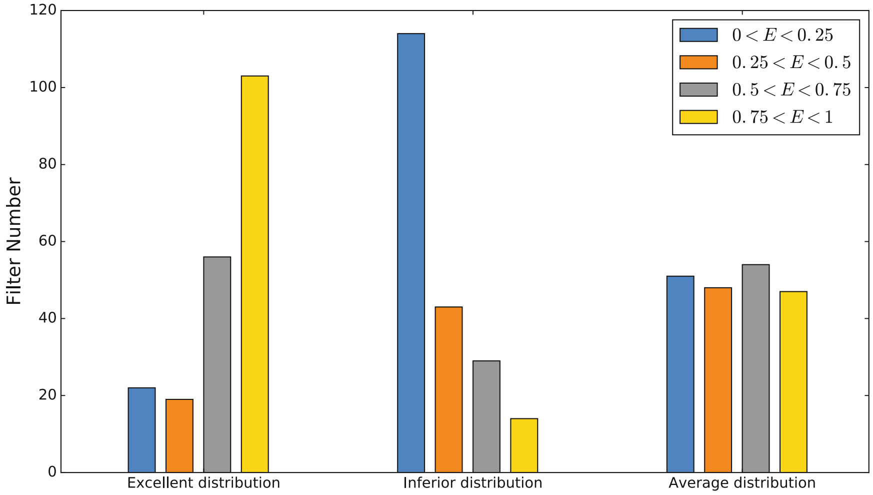

Fig. 1. The number of filters with different evaluation values in the three group of filter banks.

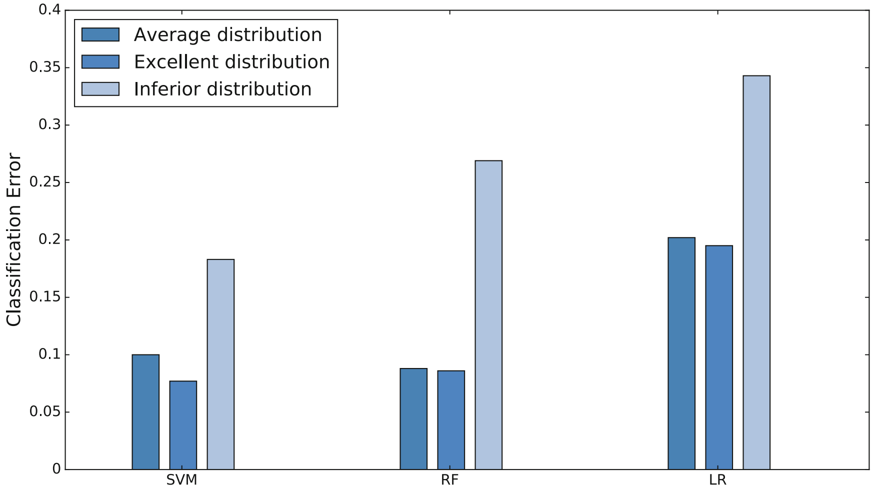

Fig. 2. The performance of classification comparison among three filter banks distribution with three classifiers.

are lower than the average distribution on the three classifiers, which shows that the proposed evaluation method can effectively distinguish high quality and low quality filters.

4.2 Detection of Bearing Defects

To compare D-RPFB and RPFB, we employ the bearing defect detection data set [5] used by the RPFB. We extract 40 time series of length 3333 in each class time series for filtering screening and testing. First, we select 15 time series (3 categories in total 45) in each category to screen the filter. Next, we generate a set of filter banks, each of which will be used in the D-RPFB and RPFB respectively. In RPFB, the filter group will maintain the number of the filters and participate in the classification of time series, and finally produce the classification error rate. In D-RPFB, the filter group will be firstly screened and then participate in the classification of time series. In this case, if the classification error rate of the D-RPFB is the same with that of the RPFB, it can verify that D-RPFB can reduce the number of redundancy and even potentially mislead filters, thus obtaining a better performance.

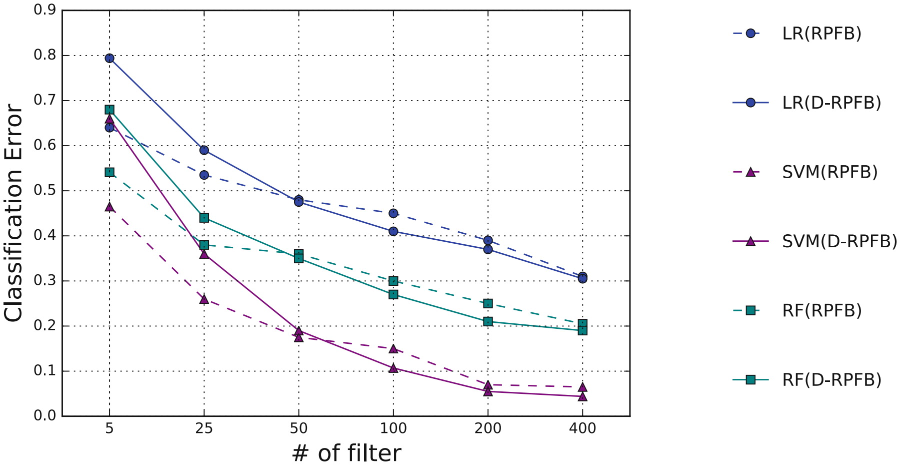

Fig. 3. The performance of classification comparison between the RPFB and D-RPFB with different classifiers on data set detection of bearing defects [5].

In our experiment, we empirically retain 75% filters (i.e., reduced number of filters in RPFB by 25%) in D-RPFB. As shown in Fig. 3, both of D-RPFB and RPFB are decreasing with the increasing of the number of filters. On this data set, the SVM can provide a lower error rate than the LR or RF. This conclusion is consistent in both the D-RPFB and RPFB. On the one hand, the error rate of the RPFB and D-RPFB is relatively high when the number of filters is relatively small. Besides, D-RPFB is worse than PRFB. This implies that the RPFB has a limited ability to summarize the time series when there are only a few filters. Meanwhile, D-RPFB further reduces the number of filters with relatively poor quality by distillation mechanism results in fewer filters, which reduces the accuracy of the D-RPFB. On the other hand, with the increasing of the number of filters, the error rate of the D-RPFB and RPFB has decreased, but the D-RPFB declines more. This is because the D-RPFB has gradually obtained the filter which can accurately summarize the time series through the screening mechanism and remove some filters that can produce a misleading effect. The RPFB, because there is no screening mechanism to distinguish the redundant and misleading filters, the effect of some inefficient filters hinders classifier from getting a better performance.

4.3 Heart Rate Classification

To show more that the D-RPFB can improve the performance of classification, we apply the heart rate data set [5] used in the RPFB. There are two time series with a length of 1800, which belong to category A and B respectively. We firstly divide the time series of category A into 30 short time series with 60 length, 15 of which are training data sets and 15 others are test data sets. Next, we conduct the same operations on the time series of category B. After dividing two long time series, we get 30 training time series (15 of them are category A and the remaining 15 are category B) and 30 test time series (also 15 of them are category A and the remaining 15 are category B). Then, we generate a set of filter banks, each of which will be used in the D-RPFB and RPFB respectively. Finally, again, RPFB uses all the generated filters for classifier. And D-RPFB uses the screened filters for classifier.

In this experiment, we empirically retain 75% filters (i.e., reduced number of filters in RPFB by 25%) in D-RPFB. As shown in Fig. 4, with the small amounts of filters, the performance of D-RPFB is inferior to RPFB again. This implies that there is no need for distillation when the number of filter is very small. However, with the increase of the filters, most of the points on the classification

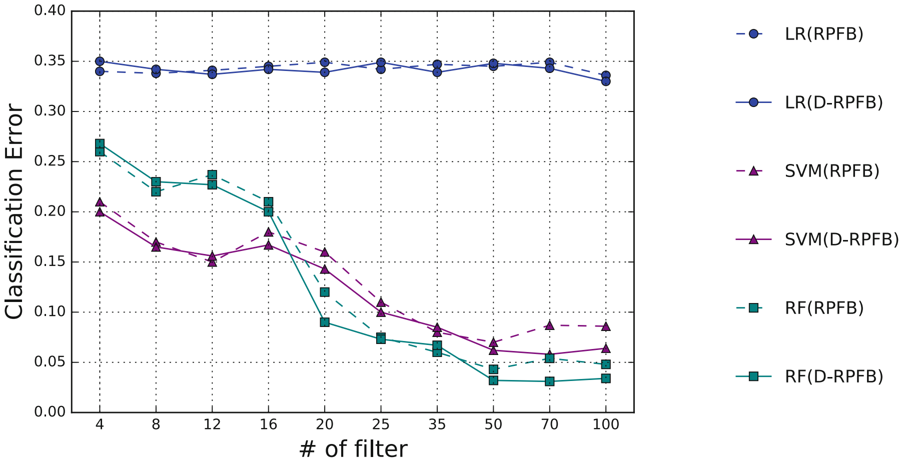

Fig. 4. The performance of classification comparison between the RPFB and D-RPFB with different classifiers on data set heart rate [5].

error curve using the features provided by D-RPFB are under the classification error curve of using the features provided by RPFB, even if some points are not under the classification error curve of RPFB, they are not much higher than in the original method. That is to say, such numerous filters randomly generated by RPFB are indeed redundant and have some misleading filters. D-RPFB distil the filters obtained by RPFB to reduce redundancy or some misleading filters to achieve the high quality of the filters and then input to the classifier, resulting a better performance.

4.4 Analyzing the Choosing of the Screening Percentage of the Filters on Hand Profile Data Set

How many filters can be kept to obtain a good summary of the input time series remains to be a question. The above reported result is under the 75% retainment (i.e., the corresponding percentage of screening is 25%) of the filters case. In this section, we analyze the choosing of the screening percentage of the filters on Hand profile data set [1]. We first generate 200 filters and then adjust the remaining filter ratio by selecting the high ranking filters, obtaining the corresponding results.

As shown in Fig. 5, if the number of filters retained is too small, the features extracted by these filters may not provide a good summary of the input time series, thus resulting a worse performance. With the percentage of retainment is increasing, the performance is better. Combined with the conclusions of experiments 4.2 and 4.3, there is no redundant or misleading information which could

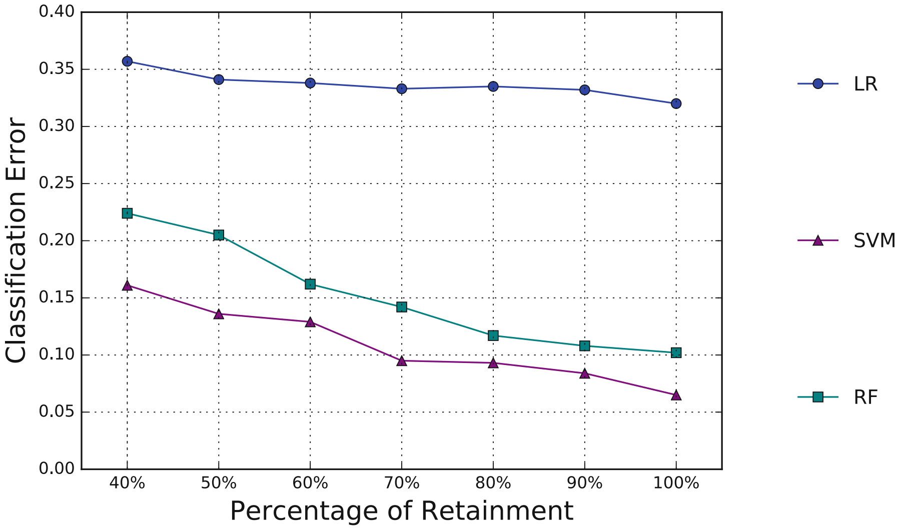

Fig. 5. The performance of classification obtained by using different classifiers under different percentages of retainment on D-RPFB.

harm the performance among such 200 filters. We can see that the original time series has been well summarized at the 80% of retainment (i.e., the corresponding percentage of screening is 20%), because the benefits from retaining more are already very small. Besides, more filters retained mean more running-time consuming when combined with specific classifier. So, in our experiment, we retain the number of filters at the original 80% while making further adjustments and finally retain 75% (i.e., the corresponding percentage of screening is 25%) to get a better performance.

5 Conclusion

In this paper, we proposed the distillation of random projection filter bank (D-PRFB) for time series classification, which is an improvement method of the random projection filter bank (PRFB). Before directly applying the features generated by the randomly generated numerous autoregressive filters that are convolved with the input time series, we add filter screening in the original method for screening the filters that are most capable of guiding the classifier to get better performance. We evaluated the D-PRFB in three different areas of time series data with three traditional classifiers. Extensive experimental results demonstrate that D-RPFB can reduce redundancy and even potentially misleading filters, thus improving the quality of the features provided to the classifier which directly improves the learning ability of the classifier to obtain a better performance.

Acknowledgment. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 61502174, 61872148), the Natural Science Foundation of Guangdong Province (Grant No. 2017A030313355, 2017A030313358), the Guangzhou Science and Technology Planning Project (Grant No. 201704030051), the Opening Project of Guangdong Province Key Laboratory of Big Data Analysis and Processing (Grant No. 2017014) and the Guangdong University of Finance & Economics Big Data and Educational Statistics Application Laboratory (Grant No. 2017WSYS001).

References

1. Chen, Y., et al.: The UCR time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/
2. Cucchiara, A.: Applied logistic regression. Technometrics **44**(1), 81–82 (1989)
3. Elmoaqet, H., Tilbury, D.M., Ramachandran, S.K.: Multi-step ahead predictions for critical levels in physiological time series. IEEE Trans. Cybern. **46**(7), 1704–1714 (2016)
4. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases, vol. 23 (1994)
5. Farahmand, A.M., Pourazarm, S., Nikovski, D.: Random projection filter bank for time series data. In: Advances in Neural Information Processing Systems, pp. 6565–6575 (2017)

6. Jaynes, E.T.: Information theory and statistical mechanics. Phys. Rev. **106**(4), 620 (1957)
7. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowl. Inform. Syst. **3**(3), 263–286 (2001)
8. Liaw, A., Wiener, M.: Classification and regression by randomForest. R news **2**(3), 18–22 (2002)
9. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11 (2003)
10. Ma, Q., Shen, L., Chen, E., Tian, S., Wang, J., Cottrell, G.W.: Walking walking walking: action recognition from action echoes. In: International Joint Conference on Artificial Intelligence, pp. 2457–2463 (2017)
11. Oppenheim, A.V., Schafer, R.W.: Discrete-time signal processing **23**(2), 157 (1989)
12. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
13. Susto, G.A., Schirru, A., Pampuri, S., Mcloone, S.: Supervised aggregative feature extraction for big data time series regression. IEEE Trans. Ind. Inform. **12**(3), 1243–1252 (2016)
14. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
15. Xu, Z., Kersting, K., von Ritter, L.: Stochastic online anomaly analysis for streaming time series. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 3189–3195 (2017)
16. Zhan, H., Gomes, G., Li, X.S., Madduri, K., Sim, A., Wu, K.: Consensus ensemble system for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 1–12 (2018)