

Article

Random Subspace Ensembles of Fully Convolutional Network for Time Series Classification

Yangqianhui Zhang ¹ , Chunyang Mo ^{2,3}, Jiajun Ma ^{2,3} and Liang Zhao ^{2,3,*}

¹ School of Biomedical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada; arielzhang2018@alumni.ubc.ca

² School of Software Technology, Dalian University of Technology, Dalian 116600, China; chunyangmo@dlut.edu.cn (C.M.); jiajunma@dlut.edu.cn (J.M.)

³ Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116600, China

* Correspondence: liangzhao@dlut.edu.cn

Abstract: Time series classification (TSC) task is one of the most significant topics in data mining. Among all methods for this issue, the deep-learning-based shows superior performance for its good adaption to raw series data and automatic extraction of features. However, rare eyes are kept on composing ensembles of these superior individual classifiers to achieve further breakthroughs. The existing deep learning ensembles NNE did a heavy work of combining 60 individuals but did not maximize the deserving improvement, since it merely pays attention to the diversity of individuals but ignores their accuracy. In this paper, we propose to construct an ensemble of Full Convolutional Neural Networks (FCN) by Random Subspace Method (RSM), named RSM-FCN. FCN is a simple but outstanding individual classifier and RSM is suitable for high dimensional data such as time series, but there are few instances. Thus, the combination of these strengths, RSM-FCN provides a highly cost-effective approach to yield promising results. Experiments on the UCR dataset demonstrate the effectiveness and reasonability of the proposed method.

Keywords: time series classification; random subspace method; FCN; ensemble learning



Citation: Zhang, Y.; Mo, C.; Ma, J.; Zhao, L. Random Subspace Ensembles of Fully Convolutional Network for Time Series Classification. *Appl. Sci.* **2021**, *11*, 10957. <https://doi.org/10.3390/app112210957>

Academic Editor: Ángel González-Prieto

Received: 24 October 2021
Accepted: 17 November 2021
Published: 19 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Time series classification (TSC) has attracted great attention, for it exists in a wide variety of fields [1], such as recognition and diagnosis problems in many industries, such as finance and medicine. Meanwhile, it is deemed as a challenging problem of data mining [2], due to its high dimensionality and nonalignment. These two drawbacks make it difficult to find distinctive features from time series.

To tackle this issue, hundreds of novel methods are proposed, including distance-based [3], feature-based, deep-learning-based [4] and ensemble-based. Wherein, methods based on deep learning shows impressive performance, as they have certain tolerance for the unaligned time series and be able to perceive the features benefiting for classification task automatically. An intuitional example is: Z. Wang [5], who evaluates some simple end-to-end deep learning benchmarks, which are Multilayer Perceptions (MLP), Residual Network (ResNet) and Fully Convolutional Network (FCN). The comparisons of experiment results show that FCN and ResNet outperform several complicated non-deep-learning TSC methods. The two winners, FCN and ResNet, are two specific forms of a Convolutional Neural Network (CNN). CNN is indeed the most highly regarded deep learning model in TSC field [4], and FCN is one of its common application forms. Ismail F. et al. [6] are attracted by the robustness and good performance of FCN, so that use it as the classifier to execute transfer learning experiments among datasets in UCR archive. Karim F et al. [7] employ FCN to perceive spatial features of time series curves, and augment it by adding LSTM module to extract time features simultaneously. Their proposed LSTM-FCN actually achieved the state-of-the-art performance at that time, while

Elsayed N. et al. [8] use GRU module to realize the extraction of time-related features, and the GRU-FCN shows accurate results, as well. In fact, RNN modules like LSTM and GRU is not suitable for solving TSC problems alone [9], such superior results are mainly based on the contribution of FCN. Moreover, MFCN [10] is the proposed on the basis of FCN to learn multi-scale features of time series, and it also achieve improved performance. The successes of above models have repeatedly demonstrated the excellence of FCN for TSC problem.

Ensemble learning is an approach to achieve further improve the accuracy when basic model reaches its bottleneck. COTE [11] and PROP [12] are the representative in TSC field. One uses a total of 35 weak classifiers which trained by information from four domain to construct a strong one, the other integrates 11 kinds of classifiers which evaluates similarity by different distance measures. Except methods like those above, which combine different existing TSC techniques, classical ensemble learning ideas which provide heuristic rules for constructing diversity individual classifiers with given dataset are also employed. Ji Cv et al. [13] propose an XG-SF based on XGBoost algorithm and Shapelet features. Raza A. et al. [14] construct ensemble classifiers called EnRS-Bagging and EnRS-Boosting in a classical Boosting and Bagging manner, respectively. As we all know, the basic individual classifiers within ensemble model should be fine and diverse, whereas most of the mentioned methods employed the inferior decision tree or distance-based models, so that they cannot compete with state-of-the-art results, though achieve some improvements in accuracy. Furthermore, there is no method focuses on constructing deep neural network ensembles, until NNE [15] is proposed in 2019. NNE compose an ensemble of total 60 different neural networks that are generated by 6 kinds of models and 10 initial weights of each kind. It does pay attention to the diversity of individual classifiers but is not wise to involve the ones with more complex structure but less accurate judgment than FCN, such as Time-CNN [16] and MCD-CNN [17]. Which leads the achieved improvements not deserve the devoted efforts. Though NNE exceeds COTE [11] in accuracy under the contribution of huge amount basic classifiers, it does not maximally play the role of ensemble. In addition, approaches like XG-SF and EnRS are imperfect in another aspect that they did not choose the appropriate data perturbation method for time series. Methods such as Bagging will further lose the training instances and Boosting will be complex in fusion phase for multi-class problems, especially facing the datasets with dozens of categories.

Based on the above analysis, we propose to employ the simple-structure but excellent FCN and the classical data perturbation manner Random Subspace Method (RSM) [18] to build a deep learning ensemble. RSM, which samples in attributes subspaces, is appropriate for time series with high dimensionality except for a few instances. A classic example of RSM is the Machine Learning algorithm called Random Forest [18]. Other applications such as creating different subsets of the same features [19] or dataset [20] for sentiment classification and image classification support our task as well. However, differently than those discrete attributes or sets, the value-continuous and order-dependent time series cannot be randomly selected. Fortunately, the feature-based TSC solution Shapelet offers an inspiration. Shapelet views a high-dimensional time series as a combination of multiple shape primitives [21] and it learns several most discriminative primitives rather than whole series. Many Shapelet-based studies are proposed and achieve successes in TSC field, such as Shapelet Transformation [22], Logical Shapelet [23], as well as the COTE, XG-SF and EnRS, as mentioned before. Thus, it makes sense to only focus on the discriminative local information of time series. Despite its advantages, the brute force approach of Shapelet has a huge time complexity of $O(k^2m^4)$ [21], so we simulated but discarded the Shapelet method. Thus, in our method, the equally-divided time series intervals are regard as primitives.

Therefore, in this paper, a lightweight RSE-FCN (Random Subspace Ensembles of FCN) is designed to tackle the challenging TSC problem, where the equally-divided time series intervals are regarded as candidate subsequences, and Top-K ones with significantly discriminative feature are screened out by evaluation model, then with Random Subspace Method deploying on the Top-K and the superior FCN serving as individual classifier, the

neural network ensembles RSE-FCN can be constructed. Through processes of screening and random selection, dimension of input can be reduced without losing discriminative features, so that FCN focusing on the key task-related information will be well trained. The basic classifier FCN has superior performance and the ensemble built by RSM will make a further breakthrough. Thus, promising results can be expected.

The rest of this paper is organized as follows. Section 2 describes the details of proposed RSE-FCN. Then, experiment results are presented and analyzed in Section 3. Finally, conclusion of this paper is drawn.

2. FCN Ensemble Built by Random Subspace Method

FCN has proven to be an effective vehicle for TSC problems. Ensemble learning is able to make further improvement from the accuracy level of single classifier. Furthermore, RSM prepares diverse individuals for ensemble by offering various combination of inputs. Meanwhile, classification model trained on the low-dimensional key features via screening, usually has good performance. This work proposes a Random Subspace Ensemble of FCN (RSE-FCN), combining the strength of above all, in order to yield promising TSC results. It converts raw continuous time series into discriminative subsequences, then deploys Random Subspace Method on these subsequences with FCN classifier. The overview of RSE-FCN is shown in Figure 1. In this section, we will first introduce the computationally less expensive converting method established by simulating Shapelet discovery, and then give a description of the precise working of proposed RSE-FCN.

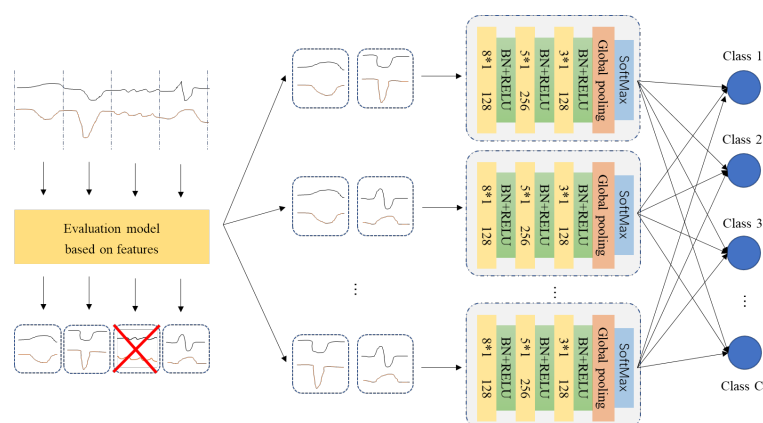


Figure 1. The overall structure of RSE-FCN.

The procedure of Shapelet discovery can be summarized in four stages: enumerate candidate subsequences, evaluate candidates and form the Shapelet binary tuple, screen and re-rank. The converting operation adopted in our method generally follows the logic of Shapelet discovery. However, we rather to roughly eliminate part of irrelevant and redundant subsequences than to accurately search out K most representative primitives as Shapelet. As shown in Algorithm 1, first, N equally-divided intervals of time series are viewed as N candidate subsequences. Then, with classification accuracy as score, each subsequence is evaluated by a simple feature-based classifier. Here, roughly trained FCN are leveraged as the evaluator. Finally, K subsequences who have the highest score will be retained for Random Subspace Method.

Algorithm 1 GetTopKBestSubsequences (D, T, Y_d, Y_t, N, K).

```

1: Input: Training set  $D$  and its label set  $Y_d$ , Test set  $T$  and its label set  $Y_t$ , Subsequences amount  $N$ ; Number of optimal subsequences  $K$ 
2: Output:  $kBestID, D_s, T_s$ 
3:  $L = X.demision$  //The time step (attribute dimension) of the instances
4:  $SL = \lfloor L/N \rfloor$  //Rounding down, the end of the series will be discarded.
5: SubsequencesID =  $\emptyset$ 
6: for  $i$  in range(1,K) do
7:   start =  $SL*(i - 1)$ , end =  $SL*i$ 
8:    $D_i = D$  [start:end] ,  $T_i = T$  [start:end]
9:   model.train( $D_i, Y_d$ )
10:  Score = model.test( $T_i, Y_t$ )
11:  SubsequencesID.append( $i, Score$ ) //Record every part and its score
12: end for
13: SubsequencesID.sortByScore()
14:  $kBestID =$  SubsequencesID.topK() //The set of ordinal numbers of top K test accuracy
15:  $D_s = \{D_1, D_2, D_3, \dots, D_N\}$  // N intervals of train set after it is divided.
16:  $T_s = \{T_1, T_2, T_3, \dots, T_N\}$  // N intervals of test set after it is divided.
17: return  $kBestID, D_s, T_s$ 

```

This ensemble framework requires four structural hyperparameters, they are: number of intervals to be divided N , number of optimal subsequences K , subsequences number M received by classifier and number of basic classifiers E . They directly correspond to the performance of proposed method. Figure 2 gives an example with hyperparameter value. A big N will lead each subsequence too short to be learned. The diversity of subsequence selections is positively related to N and the amount of individual classifiers E is also positively related to that diversity, so we think N and E within 10 is enough. Furthermore, obviously, M should be less than or equal to K . Moreover, $M/N > 1/2$ should be guaranteed, since above discovery process of key discriminative subsequence is rough, on the basis, retaining too little proportion as inputs may miss some beneficial features which hides in the discard ones.

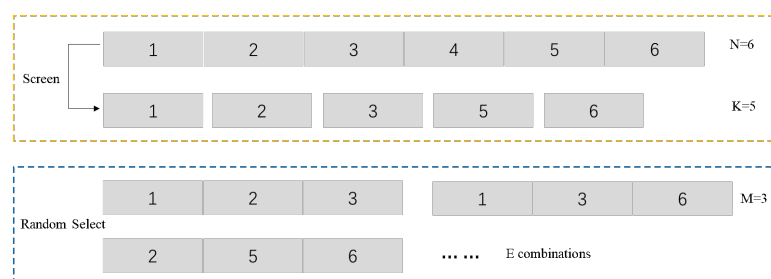


Figure 2. Explanatory diagrams of the four hyperparameters.

In the ensemble phase, Random Subspace Method is deployed on FCNs (Figure 3), which keeps the same structure with that found in another paper [5]. FCNs are trained by different combinations selected from top-K discriminative subsequences. Finally, the soft classification scores given by each individual classifier will be combined by soft voting.

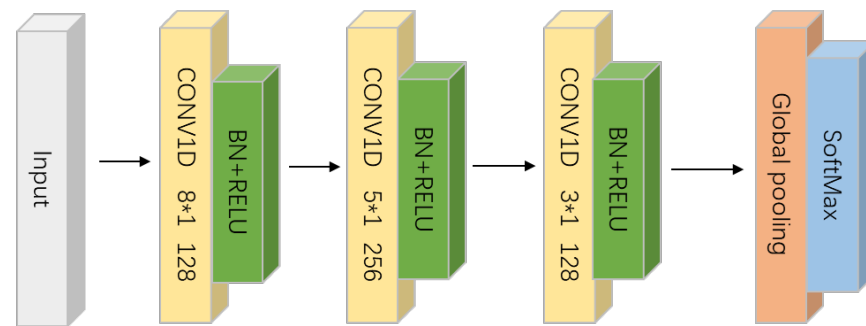


Figure 3. The commonly used structure of FCN in TSC field.

Based on above settings and descriptions, the details of RSE-FCN can be presented in Algorithm 2. First, by **GetTopKBestSubsequences** method (Algorithm 1), **KBestId**, the set of ordinal number corresponding to K optimal intervals, can be obtained. The sets of N intervals D_s and T_s , which the train set D and test set T are divided into, should be reserved. Second, M ordinal numbers **RsID** are randomly selected out from **KBestId** every time and this operation should be repeated for E times. Next, according to D_s , T_s and current **RsID**, new training set D' and test set T' can be generated. To preserve the features which stretch over division nodes, the subsequences combining order should be in accordance with the number-ascending order rather than the selection order. Or reversely, we can get rid of the subsequences whose ordinal number not in the selected M ones from the interval sets of whole time series. Then, FCN is trained by current dataset D' , to yield the judgment for the test set T' . Finally, the classification results of E individual FCN classifiers are integrated by soft voting, thus the ensemble decision for test set T can be obtained.

Algorithm 2 Random Subspace Ensembles of FCN.

```

1: Input: Training set  $D$  and its label set  $Y_d$ , Test set  $T$  and its label set  $Y_t$ , and  $N, K, M, E$ 
2: Output: The classification result  $Y$  of  $T$ 
3:  $kBestID, D_s, T_s = \text{GetTopKBestSubsequences}(D, T, Y_d, Y_t, N, K)$ 
4:  $U = \{1, 2, 3, \dots, N\}$  // The full set of ordinal numbers
5: IndividualResults =  $\emptyset$ 
6: for  $i$  in range(1, E) do
7:    $RsID = \text{RandomSelectFrom}(kBestID, M)$ 
8:   Drop =  $U - RsID$  // Subtraction of set, getting the ordinal numbers to be eliminated
9:    $D' = \text{combineExcept}(D_s, \text{Drop}), T' = \text{combineExcept}(T_s, \text{Drop})$ 
10:   $FCN.train(D', Y_d)$ 
11:   $r_i = FCN.test(T')$ 
12:  IndividualResults.append( $r_i$ )
13: end for
14:  $Y \leftarrow \text{IndividualResults}$  // Vote
15: return  $Y$ 

```

3. Experiments

The 44 datasets in the UCR archives [24], adopted by Wang's work [5], are employed by the most existing methods. In this scope, we consider the ones that fit our background question, which are picked from the following aspects: the number of training instances, number of their categories and their time step. Thus, validation experiments are carried out on 26 datasets.

The training iteration of evaluator FCN in Algorithm 1 is set as 300 and that of the classifier FCN is 1600. The four hyper-parameters N, K, M and E in the method are up to specific dataset. Towards the special dataset ItPwDmd with only 24 dimensions, $K = N$ is

taken, that means no elimination in time step. Settings of other hyper-parameters for the experiment are given in Table 1.

Table 1. Hyperparameters of RSE-FCN validation experiment.

Hyper-Parameters	Setting
Batch size	128
Optimizer	Adam
Learning rate	10^{-3} to 10^{-4}
Loss function	Cross-entropy

Comparisons will be conducted between RSE-FCN and related benchmarks, including: (1) classical single modal deep learning TSC methods [5] FCN, ResNet and MLP, (2) the multimodal neural network MCNN [25] and MFCN [10], (3) non-deep-learning and deep-learning ensemble model: SE (Shaplet Ensemble), COTE [11] and NNE. In terms of the authors of the UCR archive also noted that “beating the performance of DTW on some data sets should be considered a necessary”, we select the classical DTW [26] and an elastic-distance based ensemble PROP [12]. In addition, we conduct another sets of comparison among RSE-FCN, ENSR [14] and XG-BF [13] on several common datasets as well, since they are all the ensemble classifier built by learning different local discriminative information.

Four evaluation metrics are adopted in above comparisons, they are the average accuracy, the times of wins, the win times excluding ties and the average ranking of each algorithm. Further, the average ranking with a critical distance (CD) given by hypothesis testing [27] could measure the performance difference.

The experiments results are shown in Tables 2 and 3. Furthermore, Figure 4 is the CD diagram.

Table 2. Comparison between RSE-FCN and all benchmarks.

Dataset	FCN	ResNet	MCNN	MLP	COTE	DTW	PROP	SE	NNE	MFCN	RSE-FCN
Adiac	0.857	0.826	0.769	0.752	0.767	0.604	0.647	0.627	0.852	0.829	0.867
Beef	0.750	0.767	0.633	0.833	0.867	0.633	0.633	0.867	0.767	0.767	0.800
CBF	0.992	0.994	0.998	0.860	0.999	0.997	0.998	0.990	0.994	1.000	0.995
CinC_ECG	0.813	0.828	0.942	0.842	0.936	0.651	0.938	0.979	0.897	0.810	0.867
Coffee	1.000	1.000	0.964	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
DiaSizeRed	0.931	0.931	0.977	0.964	0.918	0.967	0.941	0.931	0.301	1.000	0.980
FaceAll	0.929	0.834	0.765	0.885	0.895	0.808	0.885	0.753	0.864	0.876	0.940
FaceFour	0.932	0.932	1.000	0.833	0.909	0.833	0.909	0.966	0.955	0.955	0.932
Fish	0.971	0.989	0.949	0.874	0.971	0.823	0.966	0.943	0.977	0.994	0.971
Gun_Point	1.000	0.993	1.000	0.933	0.993	0.907	0.993	0.940	1.000	1.000	1.000
InlineSkate	0.411	0.365	0.382	0.351	0.449	0.384	0.433	0.347	0.384	0.338	0.484
ItPwDmd	0.970	0.960	0.970	0.966	0.964	0.950	0.961	0.947	0.965	0.965	0.959
Lighting7	0.863	0.836	0.781	0.644	0.753	0.726	0.767	0.726	0.904	0.808	0.863
Mallat	0.980	0.979	0.943	0.936	0.964	0.934	0.950	0.908	0.969	0.973	0.970
MoteStrain	0.950	0.895	0.921	0.869	0.915	0.835	0.886	0.887	0.939	0.938	0.918
SwedishLeaf	0.966	0.958	0.934	0.893	0.954	0.792	0.915	0.880	0.973	0.973	0.973
Symbols	0.962	0.872	0.951	0.853	0.954	0.950	0.951	0.917	0.959	0.945	0.980
Synth_Cntr	0.990	1.000	0.997	0.950	1.000	0.993	0.990	0.967	1.000	1.000	0.983
Trace	1.000	1.000	1.000	0.820	0.990	1.000	0.990	0.950	1.000	1.000	0.990
TwoLeadECG	0.897	1.000	0.999	0.853	0.985	1.000	1.000	0.971	1.000	0.999	0.980
uWavGestX	0.754	0.787	0.820	0.768	0.804	0.728	0.801	0.752	0.821	0.790	0.780
uWavGestY	0.725	0.668	0.732	0.703	0.733	0.634	0.717	0.678	0.732	0.713	0.727
uWavGestZ	0.729	0.755	0.768	0.705	0.735	0.658	0.710	0.654	0.780	0.735	0.764
Wafer	0.997	0.997	0.998	0.996	0.999	0.980	0.997	0.998	0.998	0.998	0.999
WordSynon	0.580	0.632	0.724	0.594	0.734	0.490	0.774	0.643	0.669	0.679	0.668
Yoga	0.902	0.858	0.888	0.855	0.887	0.836	0.879	0.841	0.893	0.871	0.891
AVERAGE ACC	0.879	0.871	0.877	0.828	0.888	0.812	0.870	0.849	0.869	0.883	0.895
Wins	7	4	4	1	4	3	3	3	9	8	8
Wins (Not tie)	3	0	2	0	2	0	1	1	3	3	4
AVERAGE Rank	4.654	5.577	4.615	8.346	4.538	8.385	5.731	8.038	3.385	4.115	3.962

Table 3. Comparison on common datasets with the models based on shapelets information.

Dataset	EnRS-Bagging	EnRS-Boosting	XG-SF	RSE-FCN
Adiac	0.629	0.661	0.762	0.867
Beef	0.671	0.653	0.833	0.800
Coffee	0.956	0.968	0.929	1.000
DiaSizeRed	0.933	0.886	1.000	0.980
ItPwDmd	0.957	0.957	0.985	0.959
Lighting7	0.710	0.682	0.757	0.863
MoteStrain	0.878	0.871	0.950	0.908
Symbols	0.865	0.841	0.920	0.980
Trace	1.000	1.000	1.000	0.990
TwoLeadECG	0.946	0.953	1.000	0.980
AVERAGE	0.854	0.847	0.914	0.933
BEST	1	1	6	4

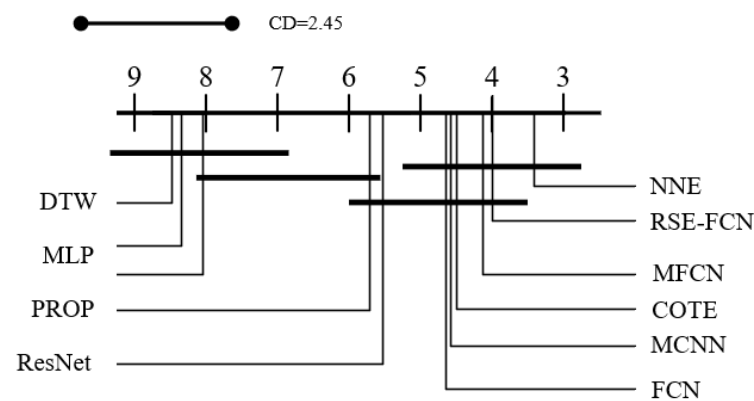


Figure 4. Critical difference diagram of models involved in comparison.

Observation shows that RSE-FCN is in the leading position among all models within comparison. Its performance is slightly better than NNE, the 60 deep learning classifier ensembles, and much better than FCN, COTE and all the other methods. Figure 4 shows most ensemble learning methods and deep learning methods are in the first level and they have no significant difference. However, RES-FCN ranks high in this group of methods, and exceeds the competitors in other specific metrics, and it is on a par with NNE. Through further comparative analysis, it can be concluded that:

- According to the three metrics in Table 1, the proposed lightweight RSE-FCN consisted within 10 FCN is at the same level as NNE which constructed by 60 neural networks, even if excluding the judgment accident of NNE on dataset DiaSizeRed (Then, the average accuracy of NNE on 25 datasets will be approximately 0.892). Furthermore, if take that dataset into consideration, RES-FCN has an obvious advantage over NNE in accuracy and robustness. This is because the superior individual FCN employed in our method establishes more accurate results than that in NNE, and it has stable performance. The ensemble built by them makes further breakthrough from this basis.
- RSE-FCN outperforms three single modal neural networks and multimodal neural networks MCNN and MFCN. This illustrates building neural network ensembles by RSM manner is an effective way to solve TSC problem. RSM offers various combinations as input and thus trains many FCN classifiers which focus on different features. Compared with the multimodal neural networks like MCNN and MFCN, RSE-FCN can integrate more comprehensive information, so it is no wonder that overwhelming performance can be achieved. RSE-FCN is also more competitive than the notable non-deep-learning ensemble model COTE, showing the advantage of employing powerful deep learning method as individuals.

- RSE-FCN achieves higher accuracy and more winning times than COTE and EnRS. XG-SF gives more top result than RSE-FCN but has less average accuracy, which also means it is not as robust as RSE-FCN. The three ones are methods which learn Shapelets information. Consequently, it suggests our method makes use of discriminative local information successfully, as well. Although simple and rough the converting process is, it retains the key subsequences and eliminates the information weakly related to TSC task.

After witnessing the experimental results, we briefly analyze the running time of the model. It should be noted that currently in the TSC field, in order to make fair comparisons, FCN will be trained for 2000 times conventionally, since Wangs did so in the initial experiment. For other deep learning models, a fixed number of training periods are kept as well. We assume that NNE also follow this default rule. In our RSE-FCN, the feature picking and basic model training process of RSE-FCN is approximately equivalent to totally construct eight FCNs. In FCN, only a single model needs to be trained, while NNE integrated 60 models. According to the volume of these models, within the same hardware condition, the time consumption of RSE-FCN is about 6–10 times of that of single FCN, and will be 1/6–1/10 of that of NNE.

Above comparison establish the significance of our proposed RSE-FCN. Furthermore, its mechanism is analyzed by the representative dataset Beef with balanced class and middle-level time steps. Instances in Beef have dimensionality of 470 and categories of 5. Taking $N = 8$ as an example, as given in Figure 5, the first image in the upper left corner is the complete curve of the five categories instances. Images from 2 to 9 show the 8 equally divided intervals and their evaluating accuracy.

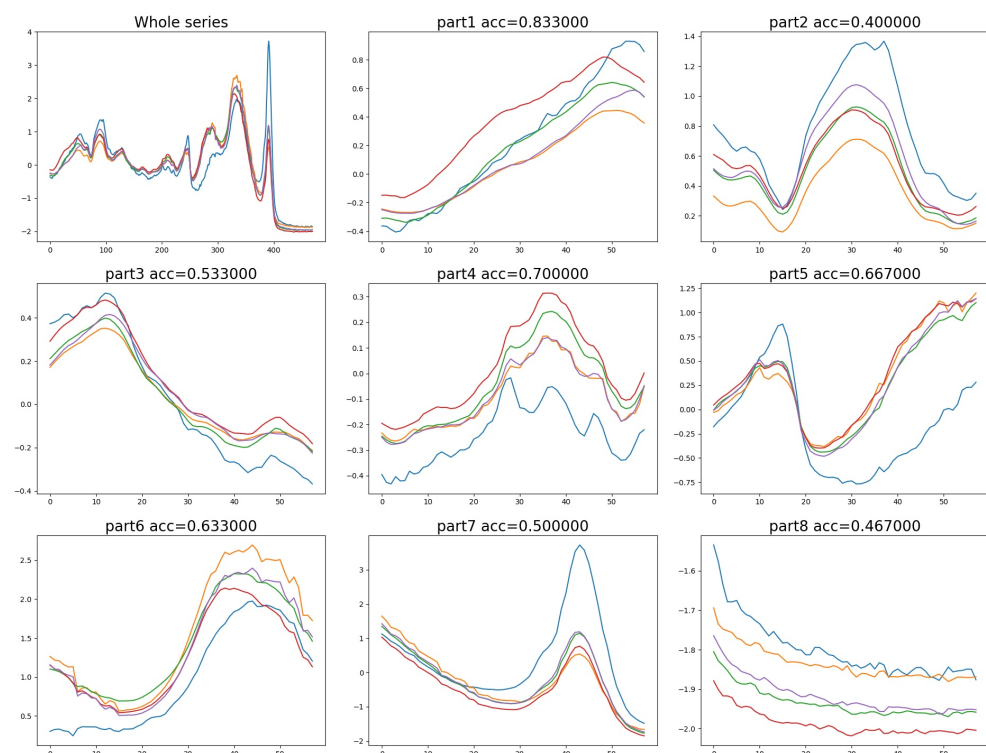


Figure 5. Curves of time series instances of Beef dataset and the corresponding accuracy given by candidates evaluation model.

It can be seen from Figure 5 that in the second, seventh and eighth intervals, all curves mainly differ in amplitude. Furthermore, though there exists some fluctuations, compared with that in the others, they are tiny and irregular, which may be caused by noise and not category-related. Thus, in these intervals, all instances produce similar activation values

when perform convolution operations with the filter matrix, making themselves to be indistinguishable, while in the rest intervals, differences appear in aspects of amplitude, change tendency, fluctuations, etc., thus they can be distinguished more easily and achieve higher accuracies. During the converting procedure of RSE-FCN, some intervals with weak discriminative feature will be discarded, so that all the random combinations will involve key information and FCNs trained by them will give fine judgment. Actually, on this dataset, some individual models which receive the discriminative local information generated by RSM did achieve higher accuracies than that based on whole series. This result proves our application of time series is reasonable and provides certain interpretability for the success of RSE-FCN.

4. Conclusions

Deep-learning-based methods show impressive performance for TSC task. However, few studies pay attention to building neural network ensembles, and the only existing NNE are not cost-effective. This paper proposes a method called RSE-FCN using Random Subspace Method to integrate the simple but outstanding FCN. It combines the strengths of learning key local information, employing superior classifier FCN and bringing promotion by ensemble. Firstly, time series are equally divided into subsequences and the top-K discriminative ones can be obtained via evaluating and screening. In that process, the main features of raw time series retain when its dimension are reduced. Then, individual models are generated with RSM deploying on the reversed subsequences and FCN serving as classifier. Finally, ensembles can be built by voting.

In experiments on UCR datasets, the lightweight RSE-FCN within 10 individuals shows same competitive with the heavy NNE, outperforming the other benchmarks. The well-trained FCNs establish a superior performance basis, the ensemble purely built by them makes further breakthrough, thus promising results can be achieved. Further analysis indicate that our usage of time series can indeed retain discriminative feature and eliminate the less relevant information, which contributes to generating strong individual classifiers. Furthermore, Random Subspace Method remains an effective manner to build good ensemble by deploying on time series subsequences.

Author Contributions: Y.Z. and C.M. came up with the creation and methodology of the models, provided the design of experiment and wrote the initial draft; J.M. participated in the programming phase and the analysis of the experimental data; L.Z. supervised the research activity planning and execution and was also responsible for ensuring that the descriptions are accurate and agreed upon by all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work is supported by National Science Foundation of China under Grants 61871013, 71671009, 61906030, Consulting Project of China Academy of Engineering 2021XY39, the Fundamental Research Funds for the Central Universities DUT20RC(4)009 and the Natural Science Foundation of Liaoning Province 2020-BS-063.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bagnall, A.J.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E.J. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [[CrossRef](#)] [[PubMed](#)]
2. Yang, Q.; Wu, X. 10 Challenging Problems in Data Mining Research. *Int. J. Inf. Technol. Decis. Mak.* **2006**, *5*, 597–604. [[CrossRef](#)]
3. Abanda, A.; Mori, U.; Lozano, J.A. A review on distance based time series classification. *Data Min. Knowl. Discov.* **2019**, *33*, 378–412. [[CrossRef](#)]
4. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [[CrossRef](#)]
5. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1578–1585. [[CrossRef](#)]

6. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P. Transfer learning for time series classification. In Proceedings of the IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, 10–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1367–1376. [[CrossRef](#)]
7. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access* **2018**, *6*, 1662–1669. [[CrossRef](#)]
8. Elsayed, N.; Maida, A.S.; Bayoumi, M.A. Gated Recurrent Neural Networks Empirical Utilization for Time Series Classification. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCoM/SmartData 2019, Atlanta, GA, USA, 14–17 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1207–1210. [[CrossRef](#)]
9. Ma, Q.; Shen, L.; Chen, W.; Wang, J.; Wei, J.; Yu, Z. Functional echo state network for time series classification. *Inf. Sci.* **2016**, *373*, 1–20. [[CrossRef](#)]
10. Zhou, W.; Hao, K.; Tang, X.; Xiao, Y.; Wang, T. Time Series Classification Based on FCN Multi-scale Feature Ensemble Learning. In Proceedings of the 2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), Dali, China 24–27 May 2019; pp. 901–906. [[CrossRef](#)]
11. Bagnall, A.J.; Lines, J.; Hills, J.; Bostrom, A. Time-series classification with COTE: The collective of transformation-based ensembles. In Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, 16–20 May 2016; pp. 1548–1549. [[CrossRef](#)]
12. Lines, J.; Bagnall, A.J. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Discov.* **2015**, *29*, 565–592. [[CrossRef](#)]
13. Ji, C.; Zou, X.; Hu, Y.; Liu, S.; Lyu, L.; Zheng, X. XG-SF: An XGBoost Classifier Based on Shapelet Features for Time Series Classification. In Proceedings of the 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018, Beijing, China, 19–21 October 2018; Elsevier: Amsterdam, The Netherlands, 2018; Volume 147, pp. 24–28. [[CrossRef](#)]
14. Raza, A.; Kramer, S. Ensembles of Randomized Time Series Shapelets Provide Improved Accuracy while Reducing Computational Costs. *arXiv* **2017**, arXiv:1702.06712.
15. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P. Deep Neural Network Ensembles for Time Series Classification. *arXiv* **2019**, arXiv:1903.06602.
16. Zhao, B.; Lu, H.; Chen, S.; Liu, J.; Wu, D. Convolutional neural networks for time series classification. *J. Syst. Eng. Electron.* **2017**, *28*, 162–169. [[CrossRef](#)]
17. Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J.L. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In *Web-Age Information Management, Proceedings of the 15th International Conference, WAIM 2014, Macau, China, 16–18 June 2014*; Lecture Notes in Computer Science; Springer: Amsterdam, The Netherlands, 2014; Volume 8485, pp. 298–310. [[CrossRef](#)]
18. Ho, T.K. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844. [[CrossRef](#)]
19. Kotsiantis, S.B. A random subspace method that uses different instead of similar models for regression and classification problems. *Int. J. Inf. Decis. Sci.* **2011**, *3*, 173–188. [[CrossRef](#)]
20. Wang, G.; Zhang, Z.; Sun, J.; Yang, S.; Larson, C.A. POS-RS: A Random Subspace method for sentiment classification based on part-of-speech analysis. *Inf. Process. Manag.* **2015**, *51*, 458–479. [[CrossRef](#)]
21. Ye, L.; Keogh, E.J. Time series shapelets: A new primitive for data mining. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J., Eds.; ACM: New York, NY, USA, 2009; pp. 947–956. [[CrossRef](#)]
22. Hills, J.; Lines, J.; Baranauskas, E.; Mapp, J.; Bagnall, A.J. Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.* **2014**, *28*, 851–881. [[CrossRef](#)]
23. Mueen, A.; Keogh, E.J.; Young, N.E. Logical-shapelets: An expressive primitive for time series classification. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; Apté, C., Ghosh, J., Smyth, P., Eds.; ACM: New York, NY, USA, 2011; pp. 1154–1162. [[CrossRef](#)]
24. Dau, H.A.; Bagnall, A.J.; Kamgar, K.; Yeh, C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E.J. The UCR Time Series Archive. *arXiv* **2018**, arXiv:1810.07758.
25. Cui, Z.; Chen, W.; Chen, Y. Multi-Scale Convolutional Neural Networks for Time Series Classification. *arXiv* **2016**, arXiv:1603.06995.
26. Rakthanmanon, T.; Campana, B.J.L.; Mueen, A.; Batista, G.E.A.P.A.; Westover, M.B.; Zhu, Q.; Zakaria, J.; Keogh, E.J. Searching and mining trillions of time series subsequences under dynamic time warping. In Proceedings of the The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, 12–16 August 2012; Yang, Q., Agarwal, D., Pei, J., Eds.; ACM: New York, NY, USA, 2012; pp. 262–270. [[CrossRef](#)]
27. Demsar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.