


Article

Adaptive Multi-Scale Wavelet Neural Network for Time Series Classification

Kewei Ouyang , Yi Hou *, Shilin Zhou and Ye Zhang

College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; ouyangkewei14@nudt.edu.cn (K.O.); slzhou@nudt.edu.cn (S.Z.); zhangye18@nudt.edu.cn (Y.Z.)

* Correspondence: yihou@nudt.edu.cn (Y.H.)

Abstract: Wavelet transform is a well-known multi-resolution tool to analyze the time series in the time-frequency domain. Wavelet basis is diverse but predefined by manual without taking the data into the consideration. Hence, it is a great challenge to select an appropriate wavelet basis to separate the low and high frequency components for the task on the hand. Inspired by the lifting scheme in the second-generation wavelet, the updater and predictor are learned directly from the time series to separate the low and high frequency components of the time series. An adaptive multi-scale wavelet neural network (AMSW-NN) is proposed for time series classification in this paper. First, candidate frequency decompositions are obtained by a multi-scale convolutional neural network in conjunction with a depthwise convolutional neural network. Then, a selector is used to choose the optimal frequency decomposition from the candidates. At last, the optimal frequency decomposition is fed to a classification network to predict the label. A comprehensive experiment is performed on the UCR archive. The results demonstrate that, compared with the classical wavelet transform, AMSW-NN could improve the performance based on different classification networks.



Citation: Ouyang, K.; Hou, Y.; Zhou, S.; Zhang, Y. Adaptive Multi-Scale Wavelet Neural Network for Time Series Classification. *Information* **2021**, *12*, 252. <https://doi.org/10.3390/info12060252>

Academic Editor: Luis Martínez López

Received: 16 May 2021
Accepted: 14 June 2021
Published: 17 June 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/>).

Keywords: wavelet transform; lifting scheme; time series classification

1. Introduction

In recent years, the research on time series classification has achieved unprecedented prosperity [1]. Time series data from the accelerometers, gyroscopes, or magnetic field sensors is used to recognize the human activity recognition [2]. Data recorded by the electroencephalogram (EEG) is important to help the doctor to study brain function and neurological disorders [3]. Mid-infrared spectroscopy analysis is also useful to discriminate the freshness of food [4]. To better compare different researches for time series classification, UCR archive [5] is built and there are at least one thousand published papers making use of at least one dataset from this archive.

The methods for time series classification can be divided into two categories: time-domain methods and frequency-domain methods [6]. Time-domain methods such as shapelets [7] and elastic distance measures [8] consider the shape of time series is important to the classification. Compared with the time-domain methods, frequency-domain methods such as Bag-of-SFA-Symbols [9] and Word Extraction for Time Series Classification [10] predict the label of the time series by analyzing the spectrum.

In the last few years, with the development of deep learning, the process of time series classification has been further advanced. Convolutional Neural Network (CNN) such as Fully Convolutional Network (FCN) and Residual Network [11] achieve the competitive performance with traditional methods. Recently, an Inception network suitable for time series called Inceptiontime [12] is proposed and achieves the state-of-the-art performance on the UCR archive. Most of the published methods learn discriminative features directly from the time domain. There are some attempts to combine the frequency representation of the time series with deep learning [5,13]. Wavelet transform is a widely used time-frequency

analysis tool that has superior time-frequency localization as compared with the Discrete Fourier Transform and Short Time Fourier Transform [14]. Wavelet transform decomposes the time series into low and high frequency components by the wavelet basis. A variety of the wavelet bases such as Harr, Morlet, and Daubechies have been proposed. Despite the remarkable achievement of the wavelet transform, there is still room for improvement. In the classical wavelet transform, the wavelet basis is artificially predefined which could be inappropriate for the task on the hand. To overcome this limitation, the second-generation wavelet emerged [15]. A lifting scheme is proposed to extract the low and high frequency components from the time series adaptively.

Inspired by the lifting scheme, an adaptive multi-scale wavelet neural network (AMSW-NN) is proposed in this paper. Instead of separating the low and high frequency components by the predefined polynomials, a multi-scale combined with a depthwise CNN is used in the AMSW-NN to obtain the candidate frequency decompositions, an optimal frequency decomposition is selected from the candidates. The primary contributions of this paper are concluded as follows:

- A multi-scale combined with a depthwise CNN is proposed to learn the candidate frequency decompositions of the time series.
- The optimal frequency decomposition is selected from the candidates by a selector.
- The experiments performed on the UCR archive [5] demonstrate that the AMSW-NN could achieve a better performance based on different classification networks compared with the classical wavelet transform.

The remainder of this paper is organized as follows. Background is reviewed in Section 2. In Section 3, AMSW-NN is proposed to extract the low and high frequency components from the time series. Next, the extensive experiments are performed on the UCR archive, and the results and discussions are presented in Section 4. Finally, a conclusion is provided in Section 5.

2. Background

This section briefly introduces the lifting scheme in the second-generation wavelet which is the building block of the proposed method.

2.1. Lifting Scheme

The second-generation wavelet is known as the lifting wavelet [16]. Compared with the classical wavelet (also called the first-generation wavelet), the lifting wavelet does not rely on the Fourier transform. Hence, a lifting scheme could be applied in the situation where the Fourier transform is unavailable [17]. The lifting scheme is usually divided into three steps including split, prediction, and update. The order of prediction and update can be reversed. The update-first structure is used in the proposed method due to the stability [18] and described in this section.

The overall flowchart of the lifting scheme is shown in Figure 1. A time series $X = (x_1, x_2, \dots, x_N)$ is split into the even component X_e and odd component X_o as presented in Equation (1):

$$\begin{aligned} X_e[n] &= X[2k - 1], \\ X_o[n] &= X[2k], \end{aligned} \quad (1)$$

where $k = 1, 2, \dots, \lfloor n/2 \rfloor$.

After the split, the information contained in the time series X is decomposed into the even component X_e and odd component X_o . The low frequency component X_c of the time series X is approximated by the running average as shown in Equation (2):

$$X_c[n] = X_e[n] + U(X_o[n]), \quad (2)$$

where $U()$ is an update filter.

When the low frequency component X_c is obtained, the high frequency component X_d could be predicted by the X_c and X_o as presented in Equation (3):

$$X_d[n] = X_o[n] - P(X_c[n]), \quad (3)$$

where $P(\cdot)$ is a prediction filter.

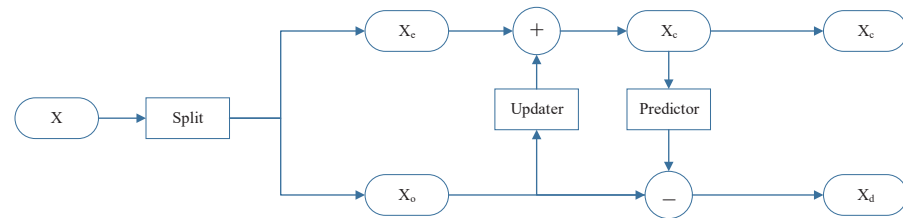


Figure 1. The flowchart of the lifting scheme.

2.2. Adaptive Lifting Scheme

The predictor and updater in the original lifting scheme are constructed by the predefined polynomials which is a suboptimal solution. Consider the excellent mapping and self-learning ability of the Back Propagation (BP) network. The predictor and updater in the adaptive lifting scheme are constructed by the BP networks [19]. The loss function $loss$ of the adaptive lifting scheme consists of two parts as shown in Equation (4):

$$loss = loss_l + loss_h, \quad (4)$$

The first part is low frequency loss $loss_l$ which maintains the coarse coefficients as Equation (5):

$$loss_l = \sum_{n=1} (X_o[n] - P(X_c[n]))^2 \quad (5)$$

The second part is high frequency loss $loss_h$ which minimizes the detail coefficients as Equation (6) [16]:

$$loss_h = \sum_{n=1} (X_o[n] - X_e[n] - U(X_o[n]))^2. \quad (6)$$

3. Adaptive Multi-Scale Wavelet Neural Network (AMSW-NN)

In this section, the proposed AMSW-NN is introduced. Compared with the BP network in the adaptive lifting scheme for one-dimensional signal, the updater and predictor in the AMSW-NN are based on a multi-scale CNN and a depthwise CNN [20]. The flowchart of the AMSW-NN is presented in Figure 2. From Figure 2, AMSW-NN consists of a frequency decomposition network (FD-Network) and a classification network (C-Network). FD-Network contains an updater, a predictor, and a selector which would be detailed introduced in the following. C-Network could be a CNN such as FCN and ResNet.

3.1. Updater

For the adaptive lifting scheme, $X_e[n]$ is updated by a fixed order polynomial. A predefined neighborhood is not always an optimal solution due to the noise and data distribution. To better obtain the low frequency component $X_c[n]$, a multi-scale neighborhood is considered in the AMSW-NN. The structure of the updater is presented in Figure 3. Similar to [16], reflection padding is first applied to the $X_o[n]$ instead of the zero padding. Then, an Inception-like module is proposed to update the $X_e[n]$ in the multiple scales. It consists of the 1×1 , 3×1 and 5×1 convolution kernels followed by the Rectified Linear Unit (ReLU) activation and the 1×1 depthwise convolution (DWConv) kernels followed by the hyperbolic tangent (Tanh) activation. $X_c[n]$ could be obtained from the output of updater and $X_e[n]$ as Equation (2).

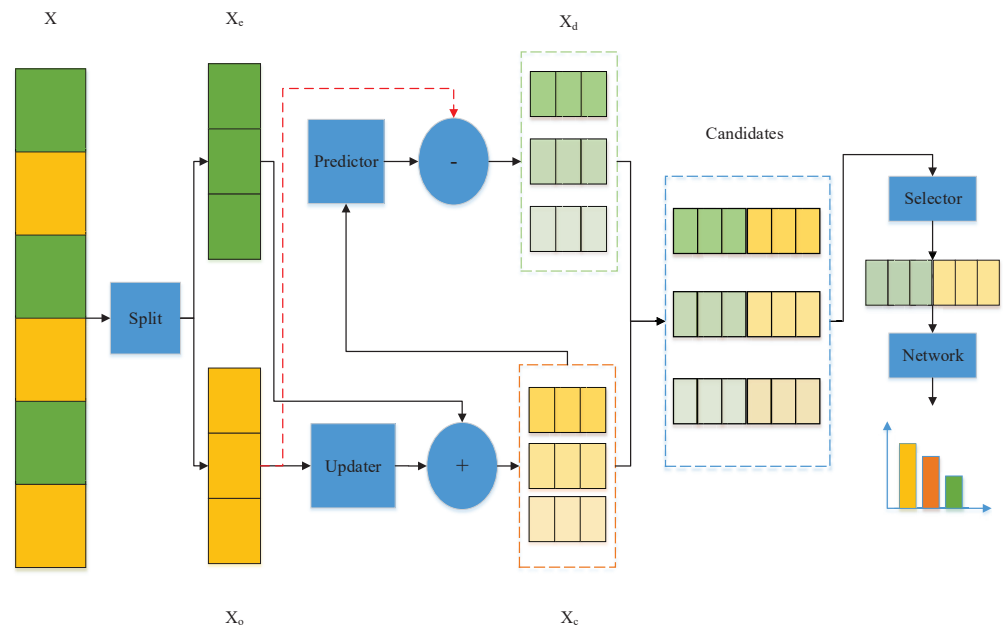


Figure 2. The flowchart of the AMSW-NN.

The rationale behind this design is that each branch of the updater models the relationship between $X_e[n]$ and $X_o[n]$ with polynomials of different orders. The different convolution kernels in each branch model this relationship with polynomials of different coefficients. DWConv guarantees the channel-dependent update without coupling. Meanwhile, DWConv could effectively reduce the number of parameters.

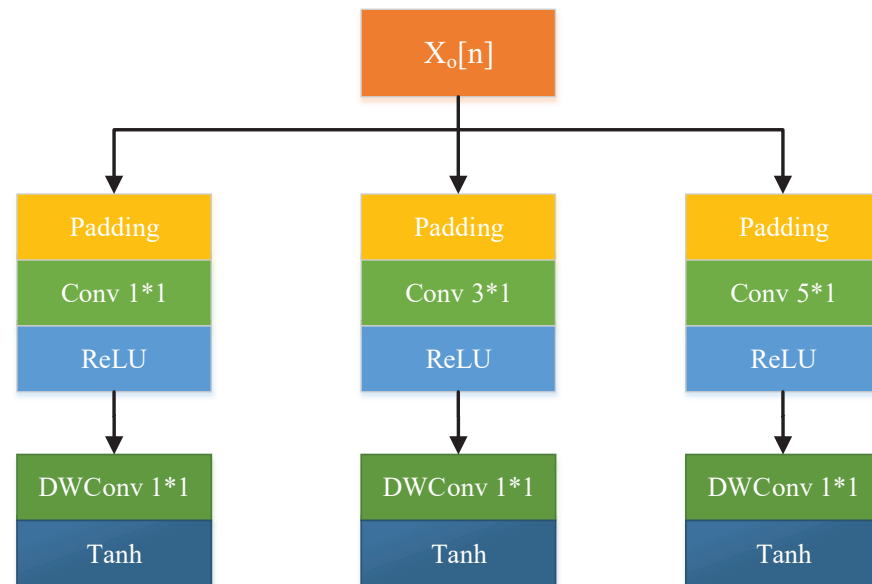


Figure 3. The structure of the updater. Padding in the updater denotes the reflection padding.

3.2. Predictor

When the $X_c[n]$ is updated, the predictor is applied to obtain the $X_d[n]$. The structure of the predictor is presented in Figure 4. It contains the reflection padding with 1×1 , 3×1 and 5×1 DWConv kernels followed by the ReLU activation and the 1×1 DWConv kernels followed by the Tanh activation. $X_d[n]$ could be predicted by the output of predictor and $X_c[n]$ as Equation (3). DWConv is also used to guarantee the channel-dependent prediction.

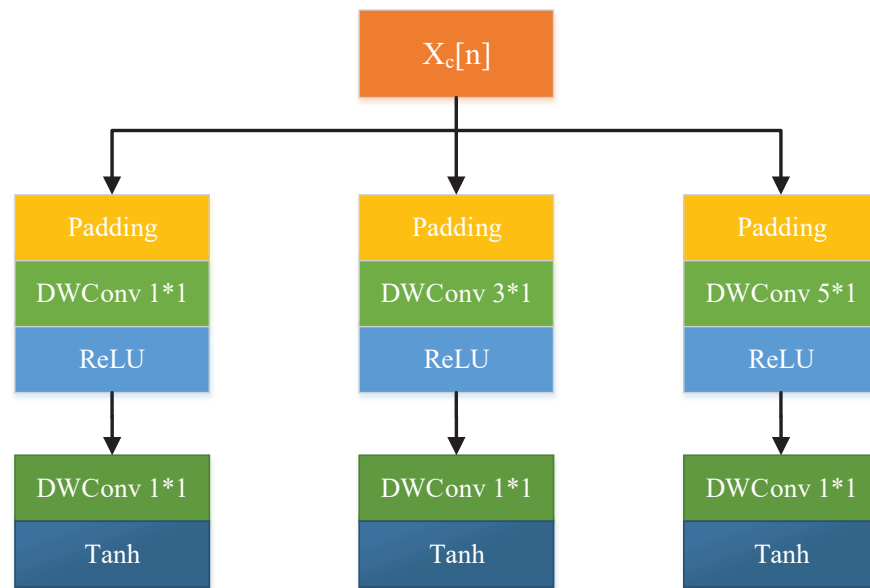


Figure 4. The structure of the predictor. Padding in the predictor denotes the reflection padding.

3.3. Selector

The frequency decomposition of the time series is determined after the update and prediction in the original lifting scheme. However, the Inception-like module used in the updater and predictor results in a multi-channel feature map as Figure 2. Each channel of the feature map could be considered as a candidate frequency decomposition of the time series. The function of the selector is to choose the optimal frequency decomposition from the candidates. The structure of the selector is presented in Figure 5. A squeeze-and-excitation module [21] is applied to put the channel attention on each channel and select the optimal channel from the candidates. Given the candidate frequency decompositions $\{D_1, D_2, \dots, D_M\}$, a global average pooling (GAP) layer combined with a two-layer Multi-layer Perceptron (MLP) as Equation (7) is used to learn the importance of each candidate frequency decomposition.

$$s_i = \sigma(W_2 \delta(W_1 D_i)), \quad (7)$$

where $W_1 \in \mathbb{R}^{\frac{M}{r} \times M}$ and $W_2 \in \mathbb{R}^{M \times \frac{M}{r}}$ are the weights of the two-layer MLP. $\sigma()$ and $\delta()$ are the ReLU and sigmoid function, respectively.

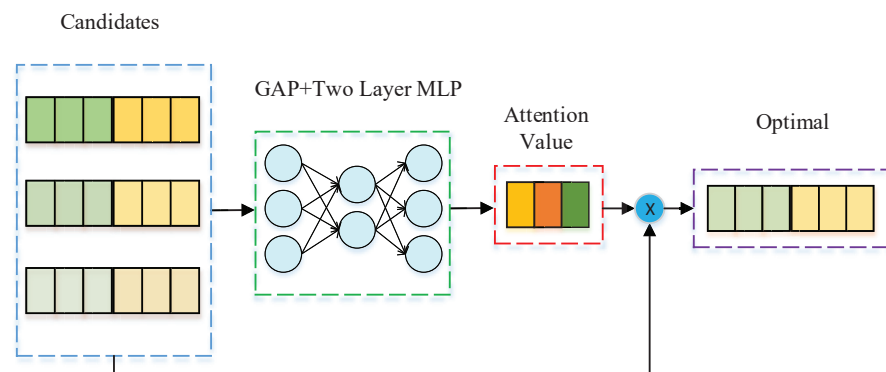


Figure 5. The structure of the selector.

3.4. Loss Function

The loss function used to train the AMSW-NN is shown in Equation (8) which is similar to [16]. It includes a cross-entropy loss, a detail loss and a mean loss. Detail loss

prefers low-magnitude detailed coefficients and mean loss promotes the $X_c[n]$ to maintain coarse coefficients,

$$\text{loss} = - \sum_{i=1}^K y_i \log(p_i) + \lambda_1 H(D) + \lambda_2 (m_{X_c} - m_X)^2, \quad (8)$$

where K is the number of categories, $H()$ is the Huber norm. λ_1 and λ_2 are the hyperparameters.

4. Experiment

In this section, extensive experiments are performed to validate the effectiveness of the AMSW-NN. This section is divided into four parts including experimental settings, experimental results, ablation studies and complexity analysis.

4.1. Experimental Settings

In this section, the dataset used to evaluate the performance is first introduced. Then, the compared method and evaluation metric are presented. Finally, the parameter settings are provided.

4.1.1. Dataset

One of the most famous datasets for time series classification is the UCR archive. UCR archive is first introduced in 2002 [5] and updated many times. It contains time series data from different applications such as ECG and HAR. In this paper, the UCR archive including 85 datasets is used which is consistent with many published papers.

4.1.2. Compared Methods

As the discussion in Section 2, consists of a FD-Network and a C-Network. The structure of the C-Network could be designed according to the application. In this experiment, FCN, ResNet, and Inception are chosen because FCN, ResNet [11] and Inception [12] are the strong baselines and the superior methods on the UCR archive, respectively. The advantage of AMSW-NN is data-adaptive frequency decomposition. To demonstrate the performance of the FD-Network, FD-Network is replaced by a Daubechies-4 (db4) decomposition as [6] to build the compared methods.

4.1.3. Evaluation Metrics

The evaluation metrics used in this experiments include Number of Win, Average Arithmetic Ranking (AVG-AR), Average Geometric Ranking (AVG-GR) and Mean Per-Class Error (MPCE). The definitions of AVG-AR, AVG-GR, and MPCE are presented in Equations (9)–(11):

$$\text{AVG-AR}_i = \frac{1}{K} \sum r_k, \quad (9)$$

$$\text{AVG-GR}_i = \sqrt[K]{\prod r_k}, \quad (10)$$

$$\begin{aligned} \text{PCE}_k &= \frac{e_k}{c_k}, \\ \text{MPCE}_i &= \frac{1}{K} \sum \text{PCE}_k, \end{aligned} \quad (11)$$

where k is the index of different datasets and i is the index of different methods, K is the number of datasets, r_k , c_k , and e_k are the rank, the number of categories, and error rates for the k th dataset, respectively.

The critical difference defined by Equation (12) is also tested to statistically compare different methods over multiple datasets [22].

$$\text{Critical Difference} = q_{\alpha} \sqrt{\frac{N_c(N_c + 1)}{6K}} \quad (12)$$

where critical value q_{α} is the studentized range statistic divided by $\sqrt{2}$, N_c is the number of methods. α is set to 0.05 in the experiments.

4.1.4. Parameter Settings

AMSW-NN consists of FD-Network and C-Network. The parameter settings for FD-Network and training are listed in Table 1 and the parameter settings of C-Network is the same as [11,12]. The number of the channel used for each branch in the updater and predictor is 32. Hence, the number of the candidate frequency decomposition is 96. The ratio r in the selector is 8. AMSW-FCN is trained for 2000 epochs, and AMSW-ResNet and AMSW-Inception are trained for 1500 epochs. The Adam optimizer is employed to train the AMSW-NN with an initial learning rate $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. λ_1 and λ_2 in the loss function is set to 0.01 and 0, respectively. The model with minimum training loss is used to evaluate the performance on each dataset.

Table 1. Parameter settings for FD-Network and training.

Parameter	Value
Kernel size	5, 3, 1
FD channel	32
Ratio	8
Training epoch	1500/2000
Learning rate	0.001
λ_1	0.01
λ_2	0

4.2. Experimental Results

In this section, the performance of the AMSW-NN on the UCR archive is reported. The accuracy rates and evaluation metrics of the AMSW-NN and compared method are shown in Table 2. DW-FCN, DW-ResNet, and DW-Inception are the abbreviations of db4 decomposition with FCN, ResNet, and Inception, respectively. To mitigate the influence of the random initialization, the evaluation is performed five times on each dataset and the average is reported to compare different methods. From Table 2, AMSW-Inception achieves the highest performance on 25 datasets and the lowest AVG-GR. AMSW-ResNet achieves the lowest AVG-AR and the second best MPCE which is just a little difference between the ResNet. Figure 6 shows the critical difference comparison of DW-FCN, DW-ResNet, DW-Inception, FCN used for the C-Network in AMSW-NN(AMSW-FCN), ResNet used for the C-Network in AMSW-NN(AMSW-ResNet), and Inception used for the C-Network in AMSW-NN(AMSW-Inception) on the UCR archive. AMSW-ResNet obtains the smallest rank compared to the other methods. Moreover, a pairwise comparison is presented in Figure 7. Compared with the DW-FCN, AMSW-FCN is better on 47 datasets and worse on 35 datasets. AMSW-ResNet is better on 47 datasets and worse on 33 datasets than DW-ResNet. AMSW-Inception is much better than DW-Inception which wins on 51 datasets and loses on 29 datasets. It proves that no matter what C-Network is selected, FD-Network obtains a better frequency decomposition than db4 decomposition.

Furthermore, it could be observed that no model could achieve the best performance on all datasets from the results listed in Table 2. However, an empirical guidance could be summarized. AMSW-Inception adopts the Inception architecture to discover the patterns in the different scales. Hence, AMSW-Inception obtains the highest accuracy on the datasets such as “CricketX” and “UWaveGestureLibraryX” which have the large intra-class

difference because a single-scale convolution is insufficient to extract the discriminative pattern on these datasets. In contrast, AMSW-FCN and AMSW-ResNet are more suitable for the datasets such as “Beef” and “Meat” which have the small intra-class difference.

Table 2. Accuracy rates and evaluation metrics of the DW-FCN (DWF), DW-ResNet (DWR), DW-Inception (DWI), AMSW-FCN (AMSWF), AMSW-ResNet (AMSWR), and AMSW-Inception (AMSWI) on the UCR archive. The accuracy rate listed in this Table for each dataset is the average of five evaluations on the testing set. For each evaluation, the model corresponding to the minimum training loss is used to predict the label and calculate the accuracy on the testing set. The accuracy rates keep three decimal places for clarity. The highest value (bold) in each dataset is actually based on the original results.

Dataset	DWF	AMSWF	DWR	AMSWR	DWI	AMSWI
Adiac	0.849	0.850	0.838	0.837	0.765	0.770
ArrowHead	0.867	0.864	0.848	0.853	0.834	0.838
Beef	0.760	0.800	0.747	0.780	0.713	0.727
BeetleFly	0.890	0.900	0.910	0.910	0.780	0.810
BirdChicken	0.900	0.910	0.920	0.890	0.880	0.860
Car	0.903	0.930	0.907	0.920	0.910	0.917
CBF	0.982	0.974	0.989	0.968	0.996	0.997
ChlorineConcentration	0.796	0.785	0.835	0.801	0.856	0.824
CinCECGTorso	0.852	0.866	0.837	0.841	0.844	0.855
Coffee	1.000	1.000	1.000	1.000	1.000	1.000
Computers	0.774	0.785	0.764	0.768	0.748	0.738
CricketX	0.774	0.769	0.811	0.818	0.838	0.838
CricketY	0.773	0.779	0.810	0.827	0.841	0.843
CricketZ	0.798	0.791	0.843	0.843	0.845	0.855
DiatomSizeReduction	0.907	0.917	0.939	0.941	0.931	0.944
DistalPhalanxOutlineAgeGroup	0.706	0.714	0.725	0.725	0.747	0.695
DistalPhalanxOutlineCorrect	0.773	0.761	0.785	0.766	0.778	0.778
DistalPhalanxTW	0.660	0.694	0.676	0.691	0.653	0.642
Earthquakes	0.757	0.731	0.744	0.748	0.737	0.741
ECG200	0.904	0.894	0.882	0.896	0.898	0.902
ECG5000	0.940	0.941	0.934	0.937	0.944	0.944
ECGFiveDays	0.996	0.978	1.000	1.000	0.999	0.999
ElectricDevices	0.662	0.657	0.666	0.660	0.661	0.662
FaceAll	0.878	0.867	0.825	0.818	0.824	0.808
FaceFour	0.932	0.930	0.955	0.955	0.927	0.932
FacesUCR	0.954	0.948	0.962	0.964	0.956	0.956
FiftyWords	0.705	0.711	0.765	0.766	0.831	0.818
Fish	0.981	0.976	0.987	0.985	0.986	0.983
FordA	0.940	0.931	0.961	0.948	0.957	0.958
FordB	0.822	0.825	0.826	0.826	0.848	0.857
GunPoint	0.996	1.000	1.000	0.999	0.992	0.992
Ham	0.722	0.709	0.754	0.752	0.670	0.678
HandOutlines	0.869	0.887	0.929	0.931	0.959	0.964
Haptics	0.523	0.527	0.571	0.550	0.535	0.545
Herring	0.644	0.697	0.588	0.603	0.688	0.700
InlineSkate	0.400	0.441	0.411	0.377	0.518	0.461
InsectWingbeatSound	0.453	0.498	0.597	0.602	0.638	0.638
ItalyPowerDemand	0.959	0.949	0.960	0.944	0.960	0.948
LargeKitchenAppliances	0.910	0.901	0.909	0.889	0.890	0.891
Lightning2	0.738	0.754	0.721	0.797	0.770	0.800
Lightning7	0.838	0.803	0.833	0.814	0.833	0.819
Mallat	0.964	0.965	0.965	0.966	0.959	0.959
Meat	0.860	0.933	0.977	0.977	0.957	0.947
MedicalImages	0.761	0.766	0.765	0.773	0.783	0.769
MiddlePhalanxOutlineAgeGroup	0.490	0.516	0.460	0.535	0.490	0.516
MiddlePhalanxOutlineCorrect	0.751	0.800	0.764	0.814	0.792	0.790
MiddlePhalanxTW	0.512	0.534	0.487	0.531	0.512	0.547
MoteStrain	0.906	0.921	0.910	0.922	0.877	0.885
NonInvasiveFetalECGThorax1	0.961	0.951	0.952	0.941	0.962	0.958
NonInvasiveFetalECGThorax2	0.958	0.943	0.957	0.950	0.958	0.958

Table 2. Cont.

Dataset	DWF	AMSWF	DWR	AMSWR	DWI	AMSWI
OliveOil	0.693	0.720	0.867	0.853	0.727	0.740
OSULeaf	0.979	0.983	0.964	0.976	0.926	0.929
PhalangesOutlinesCorrect	0.804	0.815	0.807	0.825	0.810	0.824
Phoneme	0.299	0.309	0.302	0.304	0.290	0.285
Plane	1.000	1.000	1.000	1.000	1.000	1.000
ProximalPhalanxOutlineAgeGroup	0.841	0.825	0.860	0.827	0.844	0.842
ProximalPhalanxOutlineCorrect	0.892	0.888	0.918	0.899	0.903	0.902
ProximalPhalanxTW	0.787	0.771	0.771	0.777	0.755	0.759
RefrigerationDevices	0.522	0.479	0.528	0.523	0.508	0.474
ScreenType	0.598	0.550	0.572	0.534	0.535	0.536
ShapeletSim	0.833	0.736	0.966	0.711	0.853	0.669
ShapesAll	0.912	0.910	0.920	0.931	0.916	0.923
SmallKitchenAppliances	0.777	0.759	0.732	0.759	0.757	0.782
SonyAIBORobotSurface1	0.953	0.892	0.963	0.942	0.859	0.780
SonyAIBORobotSurface2	0.950	0.938	0.919	0.947	0.905	0.895
StarLightCurves	0.975	0.975	0.973	0.977	0.978	0.978
Strawberry	0.982	0.982	0.984	0.984	0.982	0.979
SwedishLeaf	0.965	0.967	0.958	0.952	0.962	0.952
Symbols	0.983	0.985	0.979	0.979	0.971	0.969
SyntheticControl	0.991	0.969	0.993	0.982	0.994	0.973
ToeSegmentation1	0.963	0.978	0.939	0.944	0.956	0.959
ToeSegmentation2	0.925	0.911	0.922	0.928	0.945	0.948
Trace	1.000	1.000	1.000	1.000	1.000	1.000
TwoLeadECG	0.992	0.995	0.999	0.998	0.963	0.983
TwoPatterns	0.915	0.956	1.000	1.000	1.000	1.000
UWaveGestureLibraryAll	0.867	0.857	0.885	0.891	0.963	0.964
UWaveGestureLibraryX	0.769	0.778	0.793	0.791	0.822	0.824
UWaveGestureLibraryY	0.669	0.674	0.707	0.706	0.764	0.767
UWaveGestureLibraryZ	0.731	0.734	0.739	0.745	0.766	0.771
Wafer	0.998	0.998	0.999	0.998	0.997	0.997
Wine	0.596	0.730	0.674	0.789	0.785	0.796
WordSynonyms	0.618	0.621	0.664	0.671	0.740	0.753
Worms	0.779	0.805	0.753	0.764	0.795	0.771
WormsTwoClass	0.722	0.730	0.719	0.730	0.751	0.745
Yoga	0.885	0.872	0.889	0.883	0.917	0.912
Number of win	13	16	23	16	16	25
AVG-AR	3.824	3.729	3.153	3.082	3.271	3.141
AVG-GR	3.297	3.138	2.612	2.658	2.765	2.536
MPCE	0.047	0.046	0.044	0.044	0.045	0.046

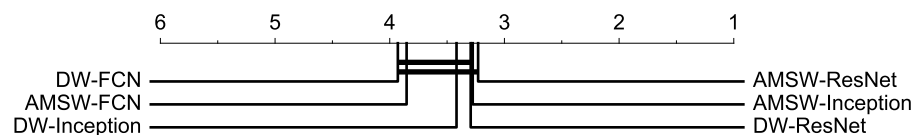


Figure 6. Critical difference diagram showing statistical difference comparison of DW-FCN, DW-ResNet, DW-Inception, AMSW-FCN, AMSW-ResNet, and AMSW-Inception on the UCR archive.

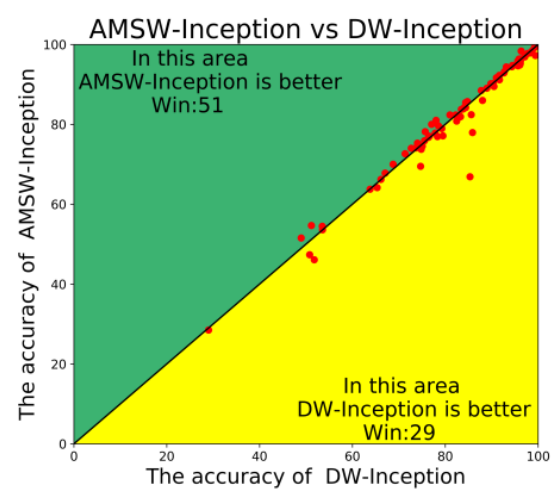
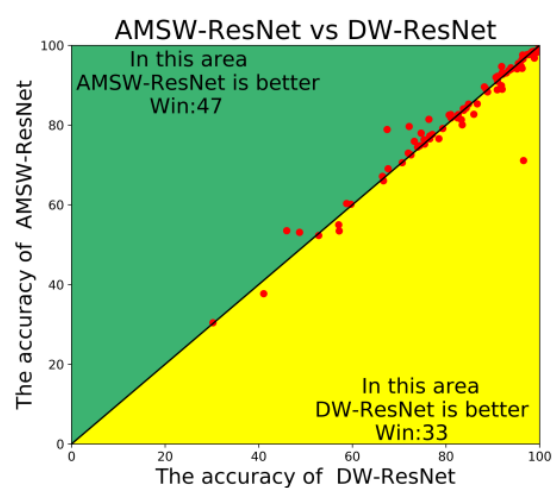
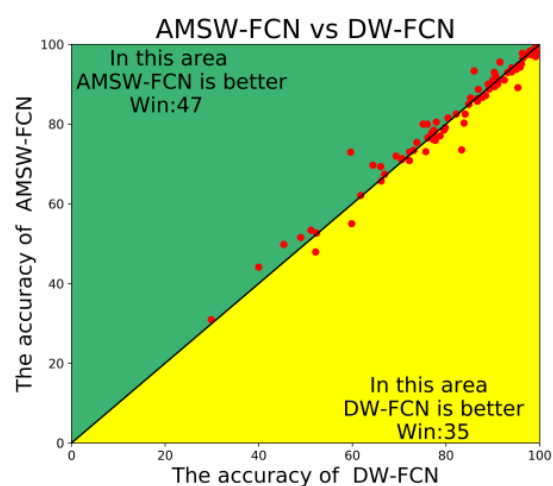


Figure 7. The results of the pairwise comparison. (a) shows the accuracy of AMSW-FCN against DW-FCN, (b) shows the accuracy of AMSW-ResNet against DW-ResNet, (c) shows the accuracy of AMSW-Inception against DW-Inception.

4.3. Ablation Studies

In this section, the effectiveness of the multi-scale structure and hyperparameters of loss function are analyzed. To validate the superiority of the multi-scale updater and predictor for AMSW-NN, a single-scale version of AMSW-FCN called ASSW-FCN is designed. Compared to the AMSW-FCN, ASSW-FCN only applies the 1×3 convolution kernel size to update and predict. The pairwise comparison between AMSW-FCN and ASSW-FCN is shown in Figure 8. Compared to the ASSW-FCN, AMSW-FCN achieves a better performance on the UCR archive which proves the effectiveness of the multi-scale structure.

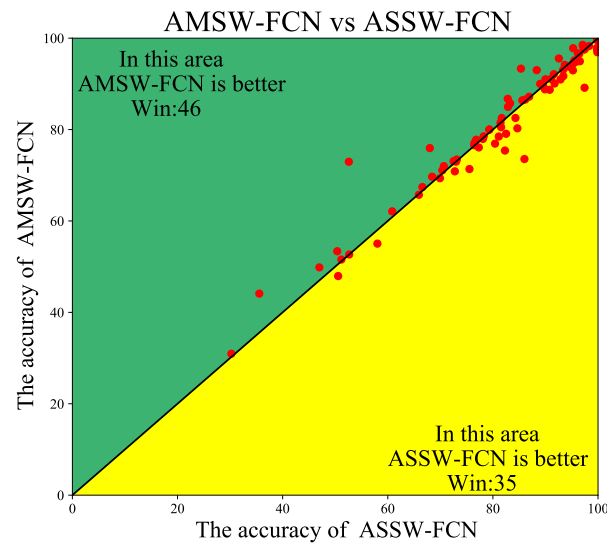


Figure 8. The pairwise comparison between AMSW-FCN and ASSW-FCN.

The loss function for training the AMSW-NN contains the detail loss and mean loss as presented in Equation (8). In Section 4.1, λ_2 is set to 0 which means the high frequency is not suppressed. In this section, λ_2 is set to 0.01 as [16] to suppress the detailed coefficients. AMSW-FCN with this loss function called AMSW-FCN(L) is trained on the UCR archive again. The pairwise comparison between AMSW-FCN and AMSW-FCN(L) is shown in Figure 9.

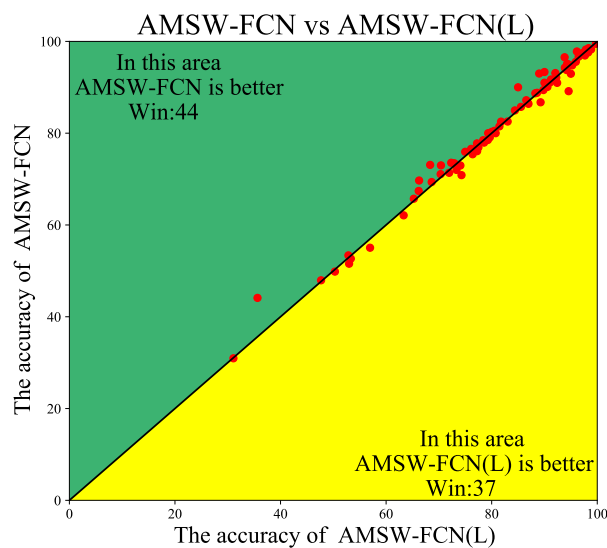


Figure 9. The pairwise comparison between AMSW-FCN and AMSW-FCN(L).

As shown in Figure 9, the performance of AMSW-FCN is slightly better than AMSW-FCN(L). The reasonable explanation is that AMSW-FCN suppresses the high frequency and AMSW-FCN(L) does not. If the high frequency is noise rather than detail, it is expected that AMSW-FCN is better than AMSW-FCN(L), and vice versa. For instance, AMSW-FCN achieves the higher accuracy on the “CricketX”, “CricketY” and “CricketZ”. Figure 10 presents some training samples from the “CricketX”, “CricketY” and “CricketZ”. It indicates that high frequency noise exists.

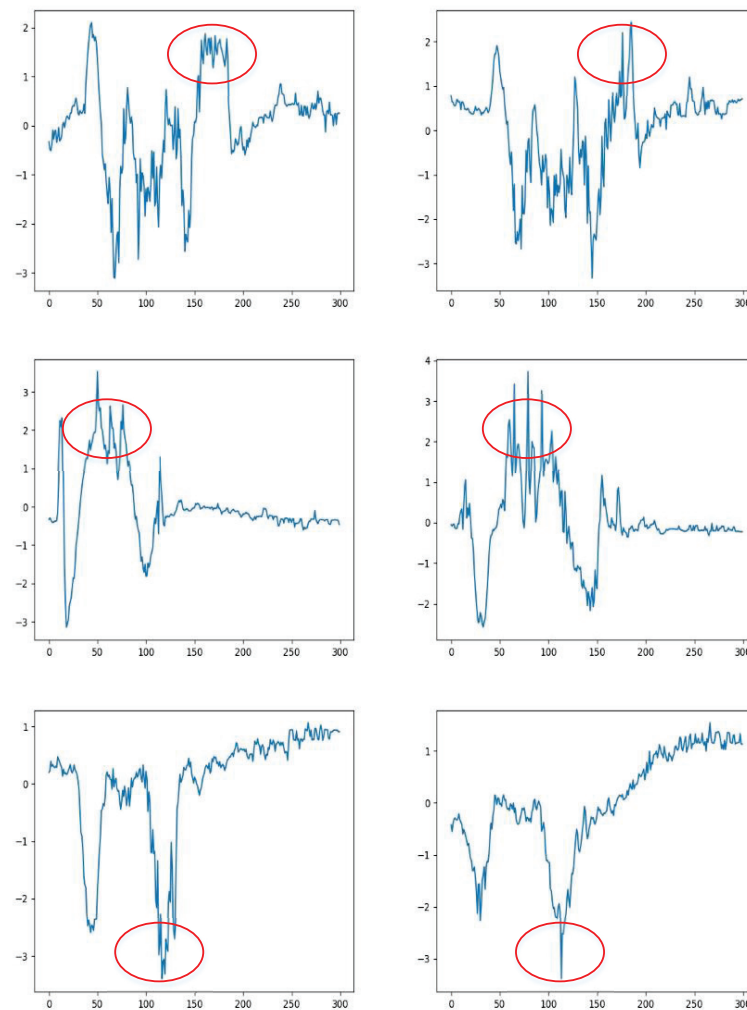


Figure 10. The training samples from the “CricketX”, “CricketY” and “CricketZ”. The samples from the same class are listed in the same row. High frequency noise could be observed in the red circle.

4.4. Complexity Analysis

AMSW-NN is composed of the FD-Network and C-Network. Compared with the DW-NN, the extra computational complexity is from the FD-Network. It is proportional to the number of the convolution kernel of the updater and predictor. Moreover, it is also proportional to the ratio r for the selector. Compared with the C-Network, the parameter learnt in the FD-Network is relatively small because the DWConv is used. The number of the learnable parameters for FD-Network and different classification networks is shown in Table 3.

Table 3. The number of the learnable parameters for AMSW-NN.

Component	Parameter Amount
FD-Network	3564
FCN	271,154
ResNet	526,964
Inception	426,642

5. Conclusions

In this paper, an adaptive multi-scale wavelet neural network called AMSW-NN for Time Series Classification is proposed. Compared with the frequency decomposition by the predefined wavelet basis, AMSW-NN adopts the multi-scale and depthwise convolution with the squeeze-and-excitation module to build the learnable updater, predictor and selector to adaptively separate the low frequency component and high frequency component from the time series which has a better generalization performance. Extensive experiments on the UCR archive show that the AMSW-NN indeed achieves a better performance than the classical wavelet decomposition combined with the neural network. In future work, we will attempt to extend the AMSW-NN to more complex applications. First, we want to modify the AMSW-NN to classify multivariate time series. Furthermore, second, we hope to find an adaptive strategy to better split the time series before the update.

Author Contributions: Methodology, K.O.; supervision, Y.H. and S.Z.; writing—original draft, K.O.; writing—review and editing, Y.H. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China under Grant No.61903373.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study is from the UCR archive which can be found here: <http://www.timeseriesclassification.com/>, accessed on 16 June 2021.

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

References

1. Liu, C.L.; Hsiao, W.H.; Tu, Y.C. Time series classification with multivariate convolutional neural network. *IEEE Trans. Ind. Electron.* **2018**, *66*, 4788–4797. [\[CrossRef\]](#)
2. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Übeyli, E.D. Wavelet/mixture of experts network structure for EEG signals classification. *Expert Syst. Appl.* **2008**, *34*, 1954–1962. [\[CrossRef\]](#)
4. Al-Jowder, O.; Kemsley, E.; Wilson, R.H. Detection of adulteration in cooked meat products by mid-infrared spectroscopy. *J. Agric. Food Chem.* **2002**, *50*, 1325–1329. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Dau, H.A.; Bagnall, A.; Kamgar, K.; Yeh, C.C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E. The UCR time series archive. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 1293–1305. [\[CrossRef\]](#)
6. Wang, J.; Wang, Z.; Li, J.; Wu, J. Multilevel wavelet decomposition network for interpretable time series analysis. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2437–2446.
7. Ye, L.; Keogh, E. 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–12 July 2009; pp. 947–956.
8. Lines, J.; Bagnall, A. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Discov.* **2015**, *29*, 565–592. [\[CrossRef\]](#)
9. Schäfer, P. The BOSS is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.* **2015**, *29*, 1505–1530. [\[CrossRef\]](#)

10. Schäfer, P.; Leser, U. Fast and accurate time series classification with weasel. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 637–646.
11. 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), Anchorage, AK, USA, 14–19 May 2017; pp. 1578–1585.
12. Fawaz, H.I.; Lucas, B.; Forestier, G.; Pelletier, C.; Schmidt, D.F.; Weber, J.; Webb, G.I.; Idoumghar, L.; Muller, P.A.; Petitjean, F. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.* **2020**, *34*, 1936–1962. [[CrossRef](#)]
13. Li, D.; Bissyandé, T.F.; Klein, J.; Traon, Y.L. Time series classification with discrete wavelet transformed data. *Int. J. Softw. Eng. Knowl. Eng.* **2016**, *26*, 1361–1377. [[CrossRef](#)]
14. Akansu, A.N.; Haddad, P.A.; Haddad, R.A.; Haddad, P.R. *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*; Academic Press: Cambridge, MA, USA, 2001.
15. Sweldens, W. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.* **1998**, *29*, 511–546. [[CrossRef](#)]
16. Rodriguez, M.X.B.; Gruson, A.; Polania, L.; Fujieda, S.; Prieto, F.; Takayama, K.; Hachisuka, T. Deep adaptive wavelet network. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikola, HI, USA, 5–9 January 2021; pp. 3111–3119.
17. Sweldens, W. Wavelets and the lifting scheme: A 5 minute tour. *Zamm-Z. Angew. Math. Mech.* **1996**, *76*, 41–44.
18. Ma, H.; Liu, D.; Xiong, R.; Wu, F. iWave: CNN-Based Wavelet-Like Transform for Image Compression. *IEEE Trans. Multimed.* **2019**, *22*, 1667–1679. [[CrossRef](#)]
19. Zheng, Y.; Wang, R.; Li, J. Nonlinear wavelets and BP neural networks adaptive lifting scheme. In Proceedings of the 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding, Chengdu, China, 17–19 December 2010; pp. 316–319.
20. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
22. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.