2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
October 11-14, 2020. Toronto, Canada

(n, m)-Layer MC-MHLF: Deep Neural Network for Classifying Time Series

Keiichi Tamura
Graduate School of Information Sciences,
Hiroshima City University,
Hiroshima, Japan
ktamura@hiroshima-cu.ac.jp

Shuichi Hashida
Graduate School of Information Sciences,
Hiroshima City University,
Hiroshima, Japan
shu.hashida@gmail.com

***Abstract*—Time series is now ubiquitous and time series classification has applications in many different areas; therefore, improving the accuracy of time series classification is one of the most interesting research topics. Fully convolutional neural network (FCN) and long short term memory fully convolutional network (LSTM-FCN) are leading techniques for deep-learning-based classification models. In our previous work, we proposed a new LSTM-FCN-based model, which is called multi-channel MACD histogram LSTM-FCN (MC-MHLF). The experimental results showed the MC-MHLF model had a good classification performance. To enhance the ability of the model, we propose a new deep neural model, which is called (n, m)-Layer MC-MHLF. The (n, m)-Layer MC-MHLF model is based on the MC-MHLF model and it is composed of n LSTM layers and m convolution layers. To evaluate the (n, m)-Layer MC-MHLF model, we compared the classification performance of it with that of conventional models. The experimental results showed that the (n, m)-Layer MC-MHLF model has good performance for classifying time series.**

***Index Terms*—Deep learning; LSTM-FCN model, Time series classification; MC-MHLF; MACD histogram;**

I. Introduction

With the increasing use of the Internet of things (IoT), time series are getting ubiquitous and large amounts of time series data are stored online through cloud services. Data analysis for sensing data, including temperature, radio waves, motion, electroencephalogram, and biological signals have attracted significant attention in various application domains. Time series classification, which is an important and challenging problem for time series analysis, is a data classification task that a class label of a time series is predicted using trained models. Time series classification is a simple task, yet, many academic and industrial researchers have been developing more accurate classification models because more accurate classification techniques are required for improving the performance of many kinds of applications.

Lately, deep learning has opened up a frontier for improving the classification performance of time series. Fully convolutional neural network (FCN) [1], residual network (ResNet) [1], and long short term memory fully convolutional network (LSTM-FCN) [2], [3] models are pioneers for time series classification. Conventional models that use machine learning techniques manually design the extraction of features in a time series. This causes limitations for improving the accuracy of time series classification as same as that of image classification. In contrast to the conventional models, deep neural models can extract features automatically in the process of learning the training datasets. Even though it is difficult to extract specific features in time series, this automatic feature extraction significantly improves the accuracy of time series classification.

We have been developing classification models based on the FCN and LSTM-FCN models [4]. In our previous work, the multi-channel MACD (moving average convergence divergence) histogram LSTM-FCN (MC-MHLF) [5] model was proposed. The MC-MHLF model showed good classification performance compared with other conventional models. The one of the main characteristics in this model is that the input time series is multi-channeled in the input layer. In the input layer, the MACD-histograms of the time series are extracted. The MACD histogram of a time series is its accelerator that indicates the features of local movement in the time series [6]. Especially, both long-term and short-term MACD histograms of an input time series are extracted. The time series and both of MACD histogram are transformed as a multi-channel time series data. Since the MACD histogram of the time series indicates the features of local movement in the time series; therefore, the MC-MHLF model enhances the ability of the LSTM-FCN model by the MACD histogram acting as attentions.

To enhance the ability of the MC-MHLF model, we propose a new LSTM-FCN-based model, which is called (n, m)-Layer MC-MHLF. The MC-MHLF model enables us to emphasize the local features of time series; therefore, it is essential to learn high-level feature expression to improve the classification performance of the model. The proposed model involves a specific structure with (n, m)-parallel stacked layers in it and the (n, m)-parallel stacked layers can extract the features of time series. We evaluated the (n, m)-Layer MC-MHLF model as comparing to our previous model and other deep-learning-based models. In the experiments, we used the UCR time series classification archive [7] that many researchers use it as a standard benchmark for time series classification. The experimental results using UCR time series classification and performance comparison showed that the classification accuracy of the (n, m)-Layer MC-MHLF model is better than the conventional models.

The reminder of this paper is organized as follows. Related work is explained briefly in Section II. The basic structure of the FCN and LSTM-FCN models are described in Section III. In Section IV, we propose the (n, m)-Layer MC-MHLF model. The experimental results are reported in Section V and the paper is concluded in Section VI.

II. Related Work

In this study, we only focus on univariate time series. A univariate time series is a sequence of primitive elements (e.g., integers, real numbers, symbols, or sets of items) including stock prices, motion tracking, biomedical signals, radio waves, and electrocardiogram values. There are several types of models for classifying univariate time series using machine learning techniques, such as distance-based [8]–[10], feature-based [11], [12], ensemble-based [13], [14], and deep-learning-based models. In this section, the related work is reviewed briefly.

Distance-based models involving the one-nearest-neighbor (1-NN) algorithm has been studied well in time series classification [8]. The performance of classification depends on the distance functions that measure the distance between two time series. Euclidean distance and dynamic time warping (DTW) are prominent methods for measuring distance [9], [10]. The main advantage of distance-based models is that they are generally stable in comparison to other models; the main disadvantage is the computation time because the 1-NN algorithm requires the distance calculation for all the time series in the training dataset. To address this issue, a data compression method, called Symbolic Aggregate approximation (SAX) [15], was proposed. SAX is a simple method for transforming time series into a symbolic or string representation; therefore, we can use string matching algorithms for calculating the distance between time series.

Feature-based models are based on converting a time series to a feature vector. Once feature vectors are extracted, we can use any kind of classification algorithms, such as support vector machine (SVM). The key factor is defining the features of time series in the feature-based model. Shapelets [12] are specific sequence patterns that can discriminate the time series in a specific class from time series belonging to other classes. A bag-of-features (TSBF) algorithm was also proposed where the subsequences in time series are sampled and partitioned into intervals for feature extraction [11].

Ensemble models combine predictions from multiple models to improve the overall performance; they demonstrated better performance than conventional models. COTE [13] utilizes a collective of classifiers ensembles on different data transformations. The collective contains classifiers constructed in the time, frequency, change, and shapelet transformation domains. HIVE-COTE [14] improved the collective by proposing a new hierarchical structure with the probabilistic voting system. Ensemble-based models use multiple models based on machine learning to obtain better predictive performance than that of any constituent learning algorithms.

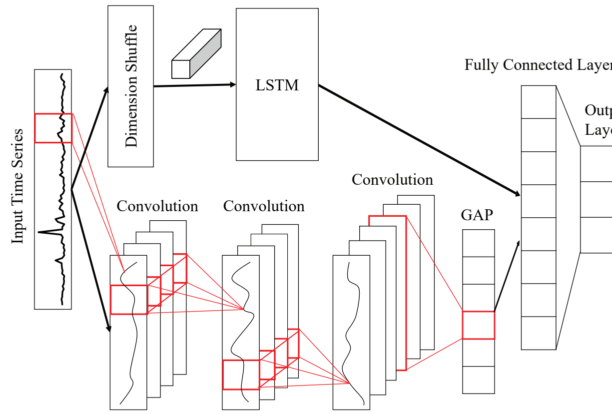

Fig. 1. LSTM-FCN Model

Recently, many researchers and practitioners have extensively investigated deep learning-based models. In the early stage of studies, multi-scale convolutional neural networks (MCNNs) was proposed in [16]. In MCNNs, layers are based on the CNN model, where there are several kinds of channel and several lengths of filters to extract various varieties of features in time series. Umeda [17] proposed a topological-data-analysis-based time series representation that can represent the hidden cycle of time series. The proposed model is based on the CNN model which was used for training the representation of time series. It was robust for noisy time series; however, it was not evaluated using the benchmark dataset. Wang et al. [1] discussed the baseline of classifying models using basic deep learning techniques such as multilayer perceptron (MLP), FCN, and residual network (ResNet). In this study, the FCN model showed the best accurate performance. Zhou et al. [18] proposed a multi-scale feature ensemble full convolutional network (MFCN). The MFCN extract multi-scale features using parallel FCN units.

As a subsequent model of the FCN model, the LSTM-FCN model was proposed by Karim et al. [2], [3]. The LSTM-FCN model is a hybrid model of the FCN and LSTM models, where the model has both FCN and LSTM layers in parallel. The ALSTM-FCN model was also proposed and it is an improved version of the LSTM-FCN model with the attention mechanism. Both the LSTM-FCN and ALSTM-FCN models showed the best accurate rates compared with other conventional time series classification models. In [19], Tamura et al. proposed a time series classification method based on the 1-NN technique where the distance function is based on MACD histogram. In addition, a deep learning model using the MACD histogram was proposed in [20]. In [20], the stacked autoencoder was used as a deep learning and the recurrence plot of the MACD histogram of time series is input to the stacked autoencoder.

III. FCN and LSTM-FCN model

The FCN model [1] is based on convolutional neural networks and it comprises of three types of units: three

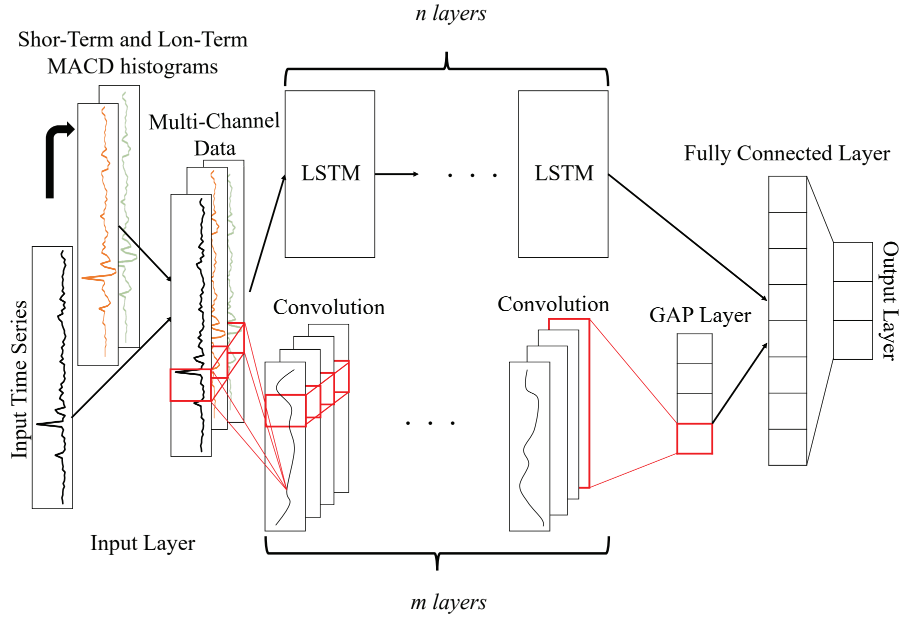

Fig. 2. $(n,\ m)$-Layer MC-MHLF

convolutional layers, a global average pooling layer, and a fully connected layer. The three convolutional layers extract the feature maps, where each convolution is one dimensional convolution and the size of the filters is $1 \times s$. To extract a feature, the filter is applied to the time series data and it slides along the time series. Additionally, the batch normalization technique is applied to each convolutional layer and the activation function is the ReLU function. After the three convolution layers, the average values in each feature channel are calculated in the global average pooling layer for compressing the feature maps. The output of global average pooling layer is fully connected to the output layer.

An FCN-based model, including a LSTM layer as a subunit, which is called the LSTM-FCN model, was proposed in [2], [3]. Fig. 1 shows the structure of the LSTM-FCN model. The input data for the LSTM-FCN model is input to two separated units that consist of an FCN unit and an LSTM unit. The FCN unit has three convolutional layers and a global averaging pool layer as same as the FCN model. In this model, the dimension shuffle technique is utilized in the input for the LSTM layer to reduce the calculation time. The input is transformed into a vector data and the vector data is input to an LSTM layer. The FCN unit can extract global features in time series and the LSTM layer can capture the temporal changes in time series data.

IV. $(n,\ m)$-Layer MC-MHLF

In this section, the $(n,\ m)$-Layer MC-MHLF model is explained.

A. Preliminaries

In this section, MACD and MACD histogram are explained briefly. Let a time series be $T = \{t_1, t_2, \cdots, t_n | t_i \in \mathbb{R}\}$. The MACD and MACD histogram of T are the velocity and the acceleration of T, respectively. The MACD histogram of T represents the features of local movement in T. In particular, the MACD and the MACD histogram of T are calculated using the past several values of T. The exponential moving average (EMA), which measures trend directions over a period of time, is used for extracting the MACD. The EMA of T is calculated as follows.

$$EMA(\omega)[T]_i = \beta \times t_i + (1-\beta) \times EMA(\omega)[T]_{i-1} \quad (1)$$

Let the i-th value in the EMA of T be $EMA(\omega)[T]_i$, where $\omega \in \mathbb{N}$ is the window size, and $\beta = 2/(\omega+1)$ represents the smoothing factor. The difference between the two EMAs is defined as MACD.

$$\begin{aligned} MACD(\omega_1, \omega_2)[T]_i = \\ EMA(\omega_1)[T]_i - EMA(\omega_2)[T]_i, \end{aligned} \quad (2)$$

where, there are two different window sizes ω_1 and ω_2, where $\omega_1 < \omega_2$. The i-th value of the MACD of T is denoted by $MACD(\omega_1, \omega_2)[T]_i$. The signal is an EMA of MACD, where the window size ω_3, which is different from ω_1 and ω_2.

$$\begin{aligned} Signal(\omega_1, \omega_2, \omega_3)[T]_i = \\ EMA(\omega_3)[MACD(\omega_1, \omega_2)[T]]_i \end{aligned} \quad (3)$$

The difference between $MACD(\omega_1, \omega_2)[T]_i$ and $Signal(\omega_1, \omega_2, \omega_3)[T]_i$ is the MACD histogram of T.

$$\begin{aligned} Histogram(\omega_1, \omega_2, \omega_3)[T]_i = \\ MACD(\omega_1, \omega_2)[T]_i - Signal(\omega_1, \omega_2, \omega_3)[T]_i, \end{aligned} \quad (4)$$

where, $Histogram(\omega_1, \omega_2, \omega_3)[T]_i$ is the i-th value of the MACD histogram of T.

TABLE I
HYPER PARAMETERS OF (n, m)-LAYER MC-MHLF MODEL

Layers of LSTM n	Layers of FCN m	Number of Units in LSTM	Number of Filters/Filter Size
1	3	128	{128/8, 256/5, 128/3}
1	6	128	{128/8, 128/8, 256/5, 256/5, 128/3, 128/3}
1	9	128	{128/8, 128/8, 128/8, 256/5, 256/5, 256/5, 128/3, 128/3 128/3}
3	3	128, 64, 64	{128/8, 256/5, 128/3}
3	6	128, 64, 64	{128/8, 128/8, 256/5, 256/5, 128/3, 128/3}
3	9	128, 64, 64	{128/8, 128/8, 128/8, 256/5, 256/5, 256/5, 128/3, 128/3 128/3}

TABLE II
COMPARISON WITH ALL MODELS

	HIVE-COTE	MLP	FCN	ResNet	LSTM-FCN	ALSTM-FCN	MC-MHLF	(n, m)-Layer MC-MHLF
Average of Ranking	4.729	7.165	5.247	5.553	3.529	3.082	3.012	**1.788**
Number of Best Accuracy	14	2	7	9	15	17	18	**55**

B. Deep Structure of (n, m)-Layer MC-MHLF

Fig. 2 shows the structure of the (n, m)-Layer MC-MHLF model. It has two main characteristics, (1) multi-channel data and (2) deep layers. The input time series is transformed into the multi-channel data, including the input time series and the two MACD histograms extracted from the input time series. There are two paths from the input to the output layer: one path consists of n LSTM layers and the other path consist of m FCN layers. The n LSTM layers learns the time-variant features of time series and the m FCN layer learn the specific sub-patterns representing the characteristics of classes. The combination of these deep layers can extract different features of time series; moreover, we can adjust the length of layers for each dataset depending on its characteristics.

C. Multi-Channel Input using MACD histogram

To enhance the performance of the deep neural models, we input not only time series, but also its MACD histogram as well. We use two types of MACD histograms; one is a short-term MACD histogram and a long-term MACD histograms a short-term MACD histogram is extracted by the short-term parameters and it represents the short-term variation of the time series. The long-term MACD histogram is extracted by the long-term parameters and it represents the long-term variations of the time series. To make a multi-channel input, the values of the time series, short-term MACD histogram, and long-term MACD histogram are set to the first, second, and third dimensions, respectively. The short-term MACD histogram parameters are set as $\omega_1 = 3$, $\omega_2 = 5$, and $\omega_3 = 4$. In contrast, the long-term MACD histogram parameters are set as $\omega_1 = 13$, $\omega_2 = 26$, and $\omega_3 = 9$.

V. EVALUATION

In this section, evaluation for the (n, m)-Layer MC-MHLF model is reported.

A. Setup

To evaluate the $n, m)$-Layer MC-MHLF model, we used the UCR time series classification archive [7] for measuring classification accuracy and comparison. The UCR time series classification archive has been updated three times, the latest version of the dataset includes 120 time-series datasets including various types of time series obtained in various environments. There used to be 85 datasets in the UCR time series classification archive in the last version; therefore, we used 85 datasets to compare the conventional methods. There are different characteristics for each dataset according to the target application of the dataset, the length of time series, total number of training and test dataset, the number of classes. In the experiments, we measured the accuracy of the test data after a model learned using the training dataset. For each dataset, the (n, m)-Layer MC-MHLF model was compared with the accuracy of the following seven models; HIVE-COTE, MLP, FCN, ResNet, LSTM-FCN, ALSTM-FCN, and MC-MHLF. Table I shows the hyper parameters of the (n, m)-Layer MC-MHLF model. The number of epochs is 2,000. For each dataset, we conducted a grid search on the hyper parameters.

B. Results

Table II summarizes the experimental results of the (n, m)-Layer MC-MHLF and other conventional models. This table shows the average of ranking and the number of best accuracy. The average of ranking is the average value of the ranking, where the ranking is determined according to the sorted accuracy values. Regarding the average of ranking, lower values of it indicate better performance. The number of best accuracy shows the number of datasets with the highest accuracy rate. Table II demonstrates that the average rank of the proposed model is the lowest, while its number of best accuracy is the largest. The (n, m)-Layer MC-MHLF has the best performance in comparison to conventional models.

The datasets of the UCR time series classification archive are divided into several categories. There are seven categories for a total of 85 datasets: "Device," "ECG," "Image," "Motion," "Sensor," "Simulated," and "Spectro." The "Device" dataset mainly contains data observed during the load measurement of electronic devices. The "ECG" dataset uses

TABLE III
COMPARISON FOR DATA SET IN "DEVICE" CATEGORY.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
Computers	0.7600	0.5400	0.8480	0.8240	0.8600	**0.864**	0.8040	0.8160
ElectricDevices	0.7703	0.5800	0.7230	0.7280	0.7681	0.7672	**0.7797**	**0.7797**
LargeKitchenAppliances	0.8640	0.4800	0.8960	0.8930	**0.92**	0.9067	0.8987	0.9173
Refrigeration Devices	0.5573	0.3710	0.5330	0.5280	0.5813	0.5840	0.6000	**0.6187**
ScreenType	0.5893	0.4080	0.6670	**0.707**	0.6693	0.6907	0.6320	0.6320
SmallKitchenAppliances	**0.8533**	0.3890	0.8030	0.7970	0.8080	0.7947	0.8133	0.8293
Average of Ranking	5.0000	8.0000	5.0000	5.0000	3.0000	3.5000	3.5000	**2.6667**
Number of Best Accuracy	1	0	0	1	1	1	1	**2**

TABLE IV
COMPARISON FOR DATA SET IN "ECG" CATEGORY.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
ECG200	0.8500	0.9200	0.9000	0.8700	0.9000	0.9100	0.9300	**0.95**
ECG5000	0.9462	0.9350	0.9410	0.9310	0.9473	0.9484	0.9460	**0.9491**
ECGFiveDays	**1**	0.9700	0.9850	0.9550	0.9919	0.9954	**1**	**1**
NonInvasiveFatalECGThorax1	0.9303	0.9420	0.9610	0.9480	0.9654	**0.9751**	0.9644	0.9644
NonInvasiveFatalECGThorax2	0.9445	0.9430	0.9550	0.9510	0.9623	0.9664	0.9644	**0.9674**
TwoLeadECG	0.9965	0.8530	**1**	**1**	0.9991	0.9991	**1**	**1**
Average of Ranking	5.8333	6.6667	4.6667	6.0000	4.0000	3.0000	2.5000	**1.3333**
Number of Best Accuracy	1	0	1	1	0	1	2	**5**

TABLE V
COMPARISON FOR DATA SET IN "IMAGE" CATEGORY.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
50words	0.8088	0.7120	0.6790	0.7270	0.8044	0.8242	0.8088	**0.8286**
Adiac	0.8107	0.7520	0.8570	0.8260	0.8593	0.8670	0.8696	**0.8772**
ArrowHead	0.8629	0.8230	0.8800	0.8170	0.9086	0.9257	0.9314	**0.9371**
BeetleFly	0.9500	0.8500	0.9500	0.8000	0.9500	**1**	**1**	**1**
BirdChicken	0.8500	0.8000	0.9500	0.9000	**1**	**1**	**1**	**1**
DiatomSizeReduction	0.9412	0.9640	0.9300	0.9310	0.9673	0.9739	0.9837	**0.9967**
DistalPhalanx OutlineAgeGroup	0.7626	0.8270	0.8350	0.7980	0.8600	0.8625	0.8675	**0.8675**
DistalPhalanx OutlineCorrect	0.7717	0.8100	0.8120	0.8200	0.8250	0.8417	**0.845**	**0.845**
DistalPhalanxTW	0.6835	0.7470	0.7900	0.7400	**0.8175**	**0.8175**	0.8100	0.8125
FaceAll	0.8030	0.8850	0.9290	0.8340	0.9402	0.9657	0.9710	**0.9822**
FaceFour	0.9545	0.8300	0.9320	0.9320	0.9432	0.9432	0.9659	**1**
FacesUCR	0.9629	0.8150	0.9480	0.9580	0.9293	0.9434	0.9629	**0.9751**
Fish	0.9886	0.8740	0.9710	0.9890	0.9829	0.9771	0.9943	**1**
HandOutlines	**0.9324**	0.8070	0.7760	0.8610	0.8930	0.9030	0.8880	0.8950
Herring	0.6875	0.6870	0.7030	0.5940	**0.7656**	0.7500	0.6875	0.7344
MedicalImages	0.7776	0.7290	0.7920	0.7720	**0.8013**	0.7961	0.7895	0.7895
MiddlePhalanx OutlineAgeGroup	0.5974	0.7350	0.7680	0.7600	0.8125	**0.8175**	0.8075	0.8075
MiddlePhalanx OutlineCorrect	0.8316	0.7600	0.7950	0.7930	0.8217	0.8400	**0.8583**	**0.8583**
MiddlePhalanxTW	0.5714	0.6090	0.6120	0.6070	0.6165	0.6466	0.6466	**0.6491**
OSULeaf	0.9793	0.5700	0.9880	0.9790	0.9959	0.9959	**1**	**1**
Phalanges OutlinesCorrect	0.8065	0.8300	0.8260	0.8250	0.8368	0.8380	0.8485	**0.8601**
ProximalPhalanx OutlineAgeGroup	0.8585	0.8240	0.8490	0.8490	**0.8927**	0.8878	0.8829	0.8927
ProximalPhalanx OutlineCorrect	0.8797	0.8870	0.9000	0.9180	**0.945**	0.9313	0.9278	0.9313
ProximalPhalanxTW	0.8146	0.7970	0.8100	0.8070	0.8350	**0.8375**	0.8250	0.8325
ShapesAll	0.9050	0.7750	0.8980	0.9120	0.9017	0.9183	**0.935**	**0.935**
SwedishLeaf	0.9536	0.8930	0.9660	0.9580	0.9792	**0.9856**	0.9824	0.9824
Symbols	0.9739	0.8530	0.9620	0.8720	0.9839	0.9869	0.9829	**0.9879**
Words Synonyms	0.7382	0.5940	0.5800	0.6320	0.6708	0.6677	0.7398	**0.7524**
Yoga	0.9177	0.8550	0.8450	0.8580	0.9177	0.9190	0.9363	**0.9407**
Average of Ranking	5.6552	7.1724	5.7931	6.2759	3.4483	2.6552	2.5517	**1.5862**
Number of Best Accuracy	1	0	0	0	6	6	6	**20**

data measured from ECG as time series data. The "Image" dataset deals with the outlines of substances in image data as series data. The "Motion" dataset comprises data that observe human motion. The "Sensor" dataset includes data obtained by observing the phenomena from sensor devices. The "Simulated" dataset comprises artificial sequence data calculated through simulation. The "Spectro" dataset contains data that treat the values of each frequency of the spectrogram for various entities (such as food). obtained by the total reflection measurement method as a series.

Table III shows the results for the "Device" category. The number of best accuracy is the first place, and it is shown

TABLE VI

Comparison for data set in "Motion" category.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
CricketX	0.8231	0.5690	0.8150	0.8210	0.8077	0.8051	0.8333	**0.8538**
CricketY	0.8487	0.5950	0.7920	0.8050	0.8179	0.8205	0.8462	**0.859**
CricketZ	0.8308	0.5920	0.8130	0.8130	0.8103	0.8308	**0.8795**	**0.8795**
GunPoint	**1**	0.9330	**1**	0.9930	**1**	**1**	**1**	**1**
Haptics	0.5195	0.4610	0.5510	0.5060	0.5747	0.5649	0.6039	**0.6071**
InlineSkate	0.5000	0.3510	0.4110	0.3650	0.4655	0.4927	**0.5327**	**0.5327**
ToeSegmentation1	0.9825	0.6010	0.9690	0.9650	0.9825	0.9868	0.9825	**0.9868**
ToeSegmentation2	0.9538	0.7460	0.9150	0.8620	0.9308	0.9308	0.9769	**0.9923**
UWaveGestureLibraryX	0.8398	0.7680	0.7540	0.7870	**0.849**	0.8481	0.8289	0.8423
UWaveGestureLibraryY	0.7655	0.7030	0.7250	0.6680	**0.7672**	0.7658	0.7468	0.7594
UWaveGestureLibraryZ	0.7831	0.7050	0.7290	0.7550	0.7973	**0.7982**	0.7795	0.7870
UWaveGestureLibraryAll	**0.9685**	0.9540	0.8260	0.8680	0.9618	0.9626	0.8961	0.9151
Worms	0.5584	0.3430	0.6690	0.6190	0.6685	0.6575	**0.7238**	**0.7238**
WormsTwo Class	0.7792	0.5970	0.7290	0.7350	0.7956	0.8011	0.8011	**0.8232**
Average of Ranking	3.6429	7.5714	5.7143	6.3571	3.5000	3.0714	2.7857	**1.7857**
Number of Best Accuracy	2	0	1	0	3	2	4	**10**

TABLE VII

Comparison for data set in "Sensor" category.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
Car	0.8667	0.8330	0.9170	0.9330	0.9500	**0.9667**	0.9167	0.9500
Chlorine Concentration	0.7120	**0.872**	0.8430	0.8280	0.8099	0.8070	0.8115	0.8656
CinCECGtorso	**0.9964**	0.8420	0.8130	0.7710	0.8862	0.9058	0.8964	0.9580
Earthquakes	0.7482	0.7920	0.8010	0.7860	0.8354	0.8292	0.8292	**0.8354**
FordA	**0.9644**	0.7690	0.9060	0.9280	0.9272	0.9267	0.9347	0.9381
FordB	0.8235	0.6290	0.8830	0.9000	0.9180	0.9158	0.9224	**0.9282**
InsectWingbeatSound	0.6551	0.6310	0.4020	0.5310	0.6616	**0.6823**	0.6005	0.6101
ItalyPower Demand	0.9631	0.9660	0.9700	0.9600	0.9631	0.9602	0.9679	**0.9718**
Lighting2	0.8197	0.7210	0.8030	0.7540	0.8033	0.7869	0.8852	**0.918**
Lighting7	0.7397	0.6440	0.8630	0.8360	0.8356	0.8219	0.8630	**0.8904**
MoteStrain	0.9329	0.8690	**0.95**	0.8950	0.9393	0.9361	0.9225	0.9249
Phoneme	**0.3824**	0.0980	0.3450	0.3240	0.3776	0.3671	0.3803	0.3803
Plane	**1**	0.9810	**1**	**1**	**1**	**1**	**1**	**1**
SonyAIBORobotSurface	0.7654	0.7270	0.9680	0.9850	0.9817	0.9700	0.9817	**0.985**
SonyAIBORobotSurfaceII	0.9276	0.8390	0.9620	0.9620	**0.978**	0.9748	0.9213	0.9549
StarlightCurves	**0.9815**	0.9570	0.9670	0.9750	0.9756	0.9767	0.9671	0.9693
Trace	**1**	0.8200	**1**	**1**	**1**	**1**	**1**	**1**
Wafer	0.9994	0.9960	0.9970	0.9970	0.9992	0.9981	0.9992	**0.9995**
Average of Ranking	4.1111	6.9444	4.5000	4.9444	3.1667	3.7778	3.6111	**2.2222**
Number of Best Accuracy	6	1	3	2	3	4	2	**9**

TABLE VIII

Comparison for data set in "Simulated" category.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
CBF	0.9989	0.8600	**1**	0.9940	0.9978	0.9967	0.9900	**1**
MALLAT	0.9620	0.9360	0.9800	0.9790	0.9808	0.9838	0.9706	**0.9915**
ShapeletSim	**1**	0.4830	0.8670	**1**	0.9722	0.9833	**1**	**1**
Synthetic Control	0.9967	0.9500	0.9900	**1**	0.9933	0.9900	0.9900	0.9933
TwoPatterns	**1**	0.8860	0.8970	**1**	0.9968	0.9968	**1**	**1**
Average of Ranking	2.8000	8.0000	4.8000	2.8000	4.4000	4.4000	4.0000	**1.4000**
Number of Best Accuracy	2	0	1	3	0	0	2	**4**

TABLE IX

Comparison for data set in "Spectro" category.

	HIVE-COTE	MLP	FCN	ResNet	LSTM -FCN	ALSTM -FCN	MC-MHLF	(n, m)-Layer MC-MHLF
Beef	0.9333	0.8330	0.7500	0.7670	0.9000	0.9333	0.9333	**0.9667**
Coffee	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Ham	0.6667	0.7140	0.7620	0.7810	0.7714	**0.8381**	0.8190	0.8286
Meat	0.9333	0.9330	0.9670	**1**	0.9167	0.9833	0.9500	**1**
OliveOil	0.9000	0.4000	0.8330	0.8670	0.8667	0.9333	0.9333	**0.9667**
Strawberry	0.9703	0.9670	0.9690	0.9580	**0.9838**	**0.9838**	0.9804	0.9837
Wine	0.7778	0.7960	0.8890	0.7960	0.8704	0.9074	0.8704	**0.963**
Average of Ranking	4.8571	6.0000	5.0000	4.5714	4.2857	1.8571	3.1429	**1.4286**
Number of Best Accuracy	1	1	1	2	2	3	1	**5**

that the proposed model has the best value throughout the category even for the average rank. It turns out that the model is an effective model for some datasets. Table IV shows the results for the "ECG" category. The MC-MHLF model, which is a model based on the MACD histogram, and the proposed model have the highest classification accuracy. The dataset is an electrocardiogram, and it is easy to capture the difference between classes using the MACD histogram. In addition, the proposed model shows a large improvement in classification accuracy compared with the conventional model. The classification accuracy could be further improved by adjusting the hyper-parameter at the layer level for the data set.

Tables V and VI show the results for the "Image" and "Motion" categories. It can be seen that both categories have the highest classification accuracy in the MC-MHLF model and the proposed model. In the "Image" category, the LSTM-FCN model, the ALSTM-FCN model, and the MC-MHLF model have the same number of best accuracy, but the average of ranking is higher for the MC-MHLF model. Therefore, the MACD-based model showed good performance in these datasets. The effect of the improvement of the classification accuracy used was confirmed, and the data set with improved classification accuracy has a relatively long sequence length, and the time-series data classification using the MACD histogram has a long sequence length. It can be considered that this is an effective model for, because MACD histograms can emphasize changes in the time-series data and extract good features.

Table VII shows the results of the "Sensor" category. The classification accuracy of the proposed model is highest compared with other models. However, focusing on the MC-MHLF model and the LSTM-FCN model, the improvement was not seen because the number of parameters of the MC-MHLF model was small, or it was necessary to adjust the parameters when calculating the MACD histogram. Table VIII shows that the classification accuracy of the ResNet model and the proposed model is high. Table IX shows that the classification accuracy of the ALSTM-FCN model and the proposed model is high.

VI. Conclusion

In this paper, we proposed a new deep neural model, which is called $(n,\ m)$-Layer MC-MHLF. The MC-MHLF model enables us to emphasize the local features of time series; therefore, it is essential to learn high-level feature expression to improve the classification performance of the model. The proposed model involves a specific structure with $(n,\ m)$-parallel stacked units in it and the $(n,\ m)$-layer can represent the structure of time series. We evaluated the proposed method as comparing to other conventional models. We used the UCR time series classification archive in the experiments. The performance comparison in the experiments showed that the performance of $(n,\ m)$-Layer MC-MHLF was better than the other conventional models.

Acknowledgment

The authors would like to thank Prof. Eamonn Keogh and all the people who have contributed to make "the UCR time series classification archive" for their contributions. This work was supported by JSPS KAKENHI Grant Number JP18K11320.

References

[1] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1578–1585.

[2] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.

[3] F. Karim, S. Majumdar, and H. Darabi, "Insights into lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 7, pp. 67 718–67 725, 2019.

[4] S. Hashida and K. Tamura, "Macd-histogram-based fully convolutional neural networks for classifying time series," in *Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 1049–1054.

[5] ——, "Multi-channel mhlf: Lstm-fcn using macd-histogram with multi-channel input for time series classification," in *Proceedings of the 2019 IEEE 11th International Workshop on Computational Intelligence and Applications (IWCIA)*, 2019, pp. 67–72.

[6] J. J. Murphy, *Technical analysis of the financial markets*. New York Institute of Finance, 1999.

[7] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML, "The ucr time series classification archive," October 2018, https://www.cs.ucr.edu/ eamonn/time_series_data_2018/.

[8] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[9] T. M. Cover, P. Hart *et al.*, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1542–1552, 2008.

[11] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2796–2802, 2013.

[12] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 289–297.

[13] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.

[14] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles," vol. 12, no. 5, 2018.

[15] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: A novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[16] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *CoRR*, vol. abs/1603.06995, 2016. [Online]. Available: http://arxiv.org/abs/1603.06995

[17] Y. Umeda, "Time series classification via topological data analysis," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 32, no. 3, pp. 1–12, 2017.

[18] W. Zhou, K. Hao, X. Tang, Y. Xiao, and T. Wang, "Time series classification based on fcn multi-scale feature eensemble learning," in *2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS)*, 2019, pp. 901–906.

[19] K. Tamura, T. Sakai, and T. Ichimura, "Time series classification using macd-histogram-based SAX and its performance evaluation," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 2419–2424.

[20] K. Tamura and T. Ichimura, "Time series classification using macd-histogram-based recurrence plot," *International Journal of Computational Intelligence Studies*, vol. 7, no. 3-4, pp. 192–213, 2018.