From Time Series to Multi-Modality: Classifying Multivariate Time Series via Both 1D and 2D Representations

Chao Yang[1], Xianzhi Wang[1], Lina Yao[2,5], Guodong Long[3], and Guandong Xu[4]

[1] School of Computer Science, University of Technology Sydney, Australia
[2] School of Computer Science and Engineering, UNSW, Australia
[3] Australian AI Institute, University of Technology Sydney, Australia
[4] Data Science Institute, University of Technology Sydney, Australia
[5] CSIRO Data61, Australia

Abstract. Multivariate time series classification is crucial for various applications such as activity recognition, disease diagnosis, and brain-computer interfaces. Deep learning methods have recently achieved promising performance thanks to their powerful representation learning capacity. However, existing deep learning-based classifiers rely solely on temporal information while disregarding clues from the frequency perspective. In this regard, we propose a novel method for classifying multivariate time series leveraging both temporal and frequency information. We first apply Short-Time Fourier Transform (STFT) to transform time series into spectrograms, which contain a 2D representation of frequency components and their temporal positions. In particular, for each variable, we generate spectrograms with varying frequencies and temporal resolutions under different window sizes. The transformation essentially adds a new modality to 1D time series and converts the multivariate time series classification into a multi-modality data classification task, making it possible to bring powerful backbones from computer vision fields to solve the time series classification problem. We then construct a dual-stream network based on the ResNet architecture that takes in both 1D and 2D representations for accurate multivariate time series classification. Our extensive experiments on 30 public datasets show our method outperforms multiple competitive state-of-the-art baselines.

Keywords: Multivariate time series classification · multimodal learning · deep learning

1 Introduction

Multivariate time series is a type of data that exists across multiple domains and have broad applications in human activity recognition [40], heart disease diagnosis [21], and brain-computer interfaces [5]. A typical multivariate time series contains a sequence of data points at regular time intervals, where values of multiple variables or measurements from multiple sensors exist at the same time

points. Compared with traditional univariate time series classification, multivariate time series classification is inherently more challenging due to the temporal variations and correlations among multiple variables. It has thus attracted the increasing attention of researchers as a sheer amount of time series data are collected by sensors in the era of Industry 4.0 [28].

While traditional methods for multivariate time series classification have been based on statistical or machine learning methods, deep learning-based methods, represented by Long Short-Term Memory (LSTM) [14], Inception-time [9], and Time Series Transformer (TST) [42] have gained prevalence recently thanks to their outstanding capability to extract effective features and learn representation in complex scenarios. Until now, all the existing approaches have been focusing on the temporal information of multivariate time series data while disregarding the underlying frequency information, which is proven invaluable in many domains like signal processing [23]. Intuitively, real-world multivariate time series data often exhibit periodicity that is challenging to detect and model from a purely temporal perspective. This highlights the necessity of incorporating frequency information into the classifier to model and classify multivariate time series data accurately. All the above inspires us to develop a novel approach that can leverage temporal and frequency information comprehensively for more accurate multivariate time series classification.

Existing methods that extract frequency information from time series data generally aim for time series forecasting, represented by ETSformer [35] and COST [34]. These methods are commonly based on Fourier Transform [2], which decomposes time series into a set of sine functions representing different frequencies, with the amplitude of each sine function indicating the intensities of the frequency components. Fourier Transform, however, can only observe time series' global frequency components without their temporal positions, resulting in insufficient frequency information that limits the accuracy of multivariate time series classification. Therefore, it calls for new approaches that can incorporate more comprehensive frequency information to improve classification performance.

In light of the above, we aim to classify multivariate time series sequences by leveraging both temporal and frequency information. Specifically, we adopt the Short-Time Fourier Transform (STFT) [11] to address the limitations of the Fourier Transform. STFT divides time series into overlapping segments, applies a Fourier transform to each segment, and finally concatenates the resulting 2D frequency domain representations to provide more comprehensive information that covers both the frequency components and their temporal positions. In particular, we use three different window sizes to generate spectrograms for each variable; these spectrograms carry multi-resolution frequency information that reflects multi-scale temporal patterns of time series which is crucial for modeling time series [3]. Through the above transformation, we create a new data modality and transform the time series classification task into a multi-modality classification task. This further allows us to bring computer vision backbones into time series classification, which have shown effectiveness in exploiting 2D representations [36]. We further construct a dual-stream architecture based on ResNet [13],

a widely used computer vision method, to leverage the power of both 2D representations (with frequency information) and 1D representations (with temporal information) of time series. The combination of 2D and 1D representations enables us to classify time series effectively, demonstrated by our proposed method consistently outperforming state-of-the-art baselines on 30 public multivariate time series datasets.

In a nutshell, we make the following contributions in this paper:

- We employ Short-Time Fourier Transform (STFT) with varying window sizes to generate 2D representations containing frequency components and corresponding temporal positions at multiple resolutions.
- We propose a dual-stream architecture based on ResNet to leverage both temporal and frequency information. A fully-connected layer with softmax function takes the fusion of the output feature maps from two streams to map them to a probability distribution of classes.
- We conducted extensive experiments on 30 public datasets and demonstrated the superior performance of our method to state-of-the-art baselines. We offer further insights by investigating convolutional backbone selection sensitivity, impact as a plugin, and ablation studies.

2 Related work

2.1 Traditional Machine Learning Methods

Statistical and traditional machine learning methods have been extensively employed for multivariate time series classification. Distance-based approaches, such as k-Nearest Neighbors (KNN) [32] combined with Dynamic Time Warping (DTW) [32], as well as feature-based methods, including Support Vector Machine (SVM) [43], TS-CHIEF [26], HIVE-COTE [19], and ROCKET [7], have been used. However, these methods typically rely on manually-crafted features and face difficulties in capturing complex relationships efficiently from high-dimensional data [1].

2.2 Deep Learning Methods

Deep learning methods are prevalent for multivariate time series classification due to the ability to capture high-dimensional non-linear relationships [17]. Convolutional Neural Networks (CNNs) [46, 20] are used to capture local temporal variations, while variants include Inception-time [9], Attentional Gated Res2Net [39], and OS-CNN [30]. As CNN lacks the ability to capture long-range dependencies, Recurrent Neural Networks (RNNs) [27] that can memorize the temporal patterns are used to classify multivariate time series, while variants include Long Short-Term Memory (LSTM) [14] and Gated Recurrent Unit (GRU) [4]. Ensemble models of RNN and CNN such as LSTM-FCN [15] and CNN-RNN Cascade model [38] incorporate both of them to exploit the CNN's

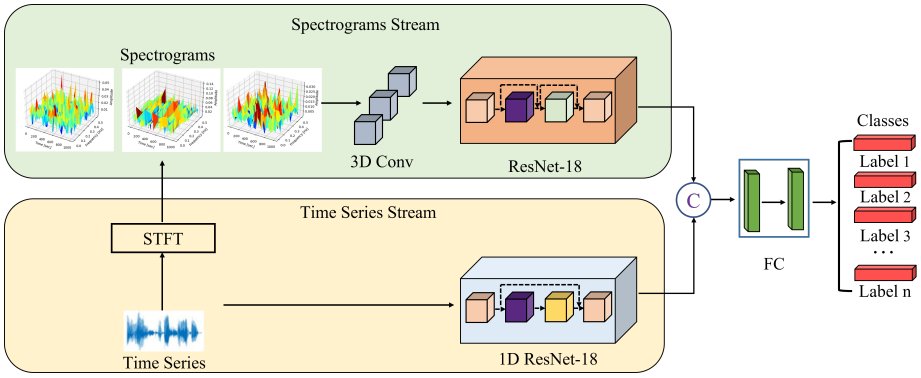


Fig. 1: The architecture of the proposed method. We use the time series with one variable to illustrate our method for simplicity. We employ Short-Time Fourier Transform with three different window sizes to generate a set of 2D spectrograms with multiple resolutions. We construct a dual-stream architecture based on ResNet to leverage both 1D representations and 2D representations. In the spectrogram stream, we use a 3D convolutional layer to fuse the spectrogram information from the resolution perspective and feed the output to ResNet-18. In the time series stream, we follow the architecture of ResNet-18 while replacing the 2D convolutional kernels using 1D convolutional kernels to adapt the shape of time series data. Finally, the output feature maps from two streams are concatenated (C in this Figure means concatenation) and fed into a fully-connected layer to map the output to the probability distribution of the classes.

ability to harness local temporal information and RNN's ability to leverage long-range dependencies for multivariate time series classification. Transformer [33] is a recently proposed method in natural language processing [8] that realizes parallel computation and multi-scale temporal information utilization, making it competitive on various tasks. Variants that are designed for multivariate time series classification include Time Series Transformer (TST) [42], Gated Transformer [22], and AutoTransformers [24]. However, all the existing methods for multivariate time series classification only focus on temporal information, ignoring the time series's inherent frequency information, which limits the capacity to classify various time series sequences.

3 Methodology

The proposed method is based on a dual-stream architecture consisting of a spectrogram stream and a time series stream, as illustrated in Fig. 1. We first implement the STFT using three different window sizes to generate a set of 2D spectrograms with varying temporal and frequency resolutions. Following this, a 3D convolutional layer is utilized to fuse the resolution-wise information of the spectrograms, while the output is fed into a ResNet-18 network to leverage 2D

representations. Concurrently, the time series data is fed into a 1D ResNet-18 network that leverages 1D representations using 1D convolutional kernels. The output feature maps of both streams are concatenated, and a fully-connected layer with softmax function is applied to map the output to the probability distribution of the classes. We elaborate on each component of the proposed method in the following sections.

3.1 Short-Time Fourier Transform

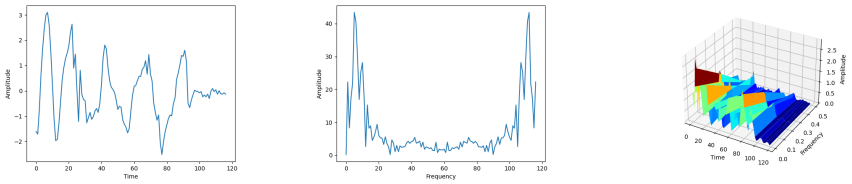
Real-world time series data are typically sampled from continuous data streams at specific sampling rates. In signal processing, the Discrete Fourier Transform (DFT) is commonly used to extract frequency components from time series data, which can be described as follows:

$$X(k) = \sum_{t=0}^{T-1} x(t) e^{-i2\pi kt/T} \tag{1}$$

where x_t is the time series sequence, and $t \in (0, T-1)$, T is the length of the time series. $X(k)$ is the frequency component obtained after DFT, while k is the index. However, the DFT lacks temporal position information of the frequency components, resulting in insufficient frequency information. To address this limitation, the Short-Time Fourier Transform (STFT) is performed, which involves using a sliding window to divide a time series sequence into short time intervals and performing the Fourier Transform on each interval to obtain the frequency components and their temporal positions. The STFT can be described as:

$$X(j, \omega) = \sum_{t=0}^{L-1} x(t) w(j-t) e^{-i\omega t} \tag{2}$$

where $x(t)$ represents the input time series in the time domain, $w(j-t)$ represents truncating the time series $x(t)$ with a window function in time to obtain the short time interval $x(t)w(j-t)$, L is the length of the window, j represents the center position of the current window, and ω represents the frequency of interest. In this case, STFT provides more sufficient frequency information including the frequency components and their temporal positions compared with the Fourier Transform. We applied the Fourier Transform and STFT to a sequence sampled from the **Handwriting** dataset to illustrate the differences between the spectrograms obtained through the Fourier Transform and STFT, and the results are shown in Figure 2. The STFT requires a balance between frequency and temporal resolutions, which presents a challenge in selecting an optimal window size. A larger window size provides more precise frequency information but results in poorer temporal resolution, while a smaller window size provides better temporal resolution but less precise frequency information. We follow a traditional signal processing approach [12] to address this issue, where the window size is chosen based on the time series's bandwidth. To select an appropriate window



(a) Time Domain (b) Spectrogram generated by Fourier Transform (c) Spectrogram generated by STFT

Fig. 2: The time domain of a time series sampled from the Handwriting dataset and the spectrograms generated by the Fourier Transform and Short-Time Fourier Transform (STFT). The spectrogram generated by the STFT provides more comprehensive frequency information, including both the frequency components and their temporal positions, in contrast to the spectrogram generated by the Fourier Transform

size, we calculate the maximum bandwidth among all variables, which can be described as:

$$\begin{aligned} \text{Bandwidth}_{\,n} &= \lceil f^n_{\max} - f^n_{\min} \rceil \\ \text{Bandwidth} &= \max(\,\text{Bandwidth}_{\,0},\ \text{Bandwidth}_{\,1}, \ldots,\ \text{Bandwidth}_{\,N}) \end{aligned} \tag{3}$$

where f^n_{max} and f^n_{min} are the maximum frequency and the minimum frequency present in the time series's nth variable, respectively, and N is the variable number. We then use three window sizes: the two, three, and four times the time series's frequency bandwidth, respectively, with an overlap of 50%, to generate three spectrograms with multi-level resolutions. For a time series sequence $x \in \mathbb{R}^{N \times T}$, where N is the variable number and T is the sequence length, the corresponding spectrogram generated by STFT is $s \in \mathbb{R}^{N \times 3 \times H \times W}$ where 3 means three different window sizes that we use, and H and W are the spectrogram's height and width. In this way, we extract the frequency components and their temporal positions from the time series and create a new data modality by converting the 1D time series sequence into a set of 2D representations, enabling us to borrow the powerful backbones from the computer vision field for leveraging 2D representation.

3.2 ResNet-18

We propose a dual-stream architecture based on ResNet [13] to leverage both the 2D representations in the spectrogram stream and the 1D representations in the time series stream for representation learning. ResNet is a popular deep neural network architecture that addresses the issue of vanishing gradients, which arises when the gradients become too small to effectively update the weights during backpropagation, particularly in very deep networks. This property has made it a competitive backbone for various computer vision tasks, motivating us to adopt

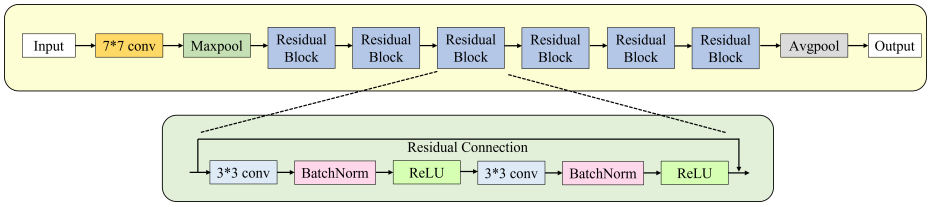


Fig. 3: The architecture of ResNet-18.

it in our approach. In the spectrogram stream, the sets of 2D representations generated by the STFT are first fed into a 3D convolutional layer for resolution-wise information fusion. This layer down-samples the input spectrograms from the resolution perspective and generates a single 2D representation for each variable. The calculation process can be described as:

$$y = W * x + b \tag{4}$$

where $x \in \mathbb{R}^{N\times 3\times H\times W}$ and $y \in \mathbb{R}^{N\times H\times W}$, and W and b are the convolutional kernel and bias term, respectively.

The resulting 2D representation and the original 1D time series are then fed into two separate neural networks, namely ResNet-18 and 1D ResNet-18, respectively. The architecture of ResNet-18, illustrated in Figure 3, comprises six residual blocks, each consisting of two convolutional layers with a kernel size 3×3. The output feature maps are fed into an average pooling layer for down-sampling from the spatial perspective, generating the latent vector of the input feature maps. In the 1D ResNet-18, we follow the same architecture as ResNet-18 but replace the 2D convolutional kernels with 1D convolutional kernels to accommodate the shape of the 1D representations. This enables us to process the time series data while retaining the advantages of ResNet-18's architecture. We then concatenate the output of the two streams and feed them into the fully-connected layer with a softmax function to map them to the probability distribution of the classes.

4 Experiments

4.1 Datasets

We evaluated our method using the UEA Time Series Classification Repository [6], which contains 30 public multivariate time series datasets. These datasets concern different domains and reflect diverse data characteristics in terms of sequence lengths and variable numbers, etc. All datasets had been preprocessed and split into training and test sets. The detailed statistics of each dataset are summarized in Table 1.

We further normalized them to zero mean and unit standard deviation and applied zero paddings to ensure that each dataset contains sequences of the same lengths.

Table 1: Statistics of the 30 UEA datasets used in experimentation.

Dataset	Train Cases	Test Cases	Dimensions	Length	Classes
ArticularyWordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
CharacterTrajectories	1,422	1,436	3	182	20
Cricket	108	72	6	1,197	12
DuckDuckGeese	60	40	1345	270	5
EigenWorms	128	131	6	17,984	5
Epilepsy	137	138	3	206	4
EthanolConcentration	261	263	3	1,751	4
ERing	30	30	4	65	6
FaceDetection	5,890	3,524	144	62	2
FingerMovements	316	100	28	50	2
HandMovementDirection	320	147	10	400	4
Handwriting	150	850	3	152	26
Heartbeat	204	205	61	405	2
JapaneseVowels	270	370	12	29	9
Libras	180	180	2	45	15
LSST	2,459	2,466	6	36	14
InsectWingbeat	30,000	20,000	200	78	10
MotorImagery	278	100	64	3,000	2
NATOPS	180	180	24	51	6
PenDigits	7,494	3,498	2	8	10
PEMS-SF	267	173	963	144	7
Phoneme	3,315	3,353	11	217	39
RacketSports	151	152	6	30	4
SelfRegulationSCP1	268	293	6	896	2
SelfRegulationSCP2	200	180	7	1,152	2
SpokenArabicDigits	6,599	2199	13	93	10
StandWalkJump	12	15	4	2,500	3
UWaveGestureLibrary	120	320	3	315	8

4.2 Baselines

We consider several popular machine learning methods and recently proposed deep learning models as baselines. The selected competitive methods include ROCKET [7], Time Series Transformer (TST) [42], ShapeNet [18], Dynamic Time Warping (DTW), TS2Vec [41], MLSTM-FCN [15], OS-CNN [30], TapNet [45], Temporal Neighborhood Coding (TNC) [31], and WEASEL+ MUSE [25].

4.3 Model Configuration and Evaluation Metric

We trained our model for 500 training epochs using Adam [16] optimizer. The learning rate is initialized to 0.001; it scales down with a coefficient of 0.1 every 50 epochs after the first 100 epochs. We repeated the training and test processes five times and took the average of multiple runs as the final results to mitigate

Table 2: Accuracy of different models on 30 benchmark datasets. The best performance values are bolded, and the second-best performance values are underlined.

Dataset	Ours	WEASEL+MUSE	TST	ROCKET	DTW	TS2Vec	MLSTM-FCN	OS-CNN	TapNet	TNC	ShapeNet
ArticularyWordRecognition	**0.996**	0.990	0.977	**0.996**	0.987	0.987	0.973	0.988	0.987	0.973	0.987
AtrialFibrillation	**0.524**	0.333	0.067	0.249	0.200	0.200	0.267	0.233	0.333	0.133	0.400
BasicMotions	**1.000**	**1.000**	0.975	0.990	0.975	0.975	0.950	**1.000**	**1.000**	0.975	**1.000**
CharacterTrajectories	0.997	0.990	0.975	0.967	0.989	0.995	0.985	**0.998**	0.997	0.967	0.980
Cricket	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.972	0.917	0.993	0.958	0.958	0.986
DuckDuckGeese	**0.767**	0.575	0.562	0.461	0.492	0.680	0.675	0.540	0.575	0.460	0.725
EigenWorms	**0.897**	0.890	0.748	0.863	0.618	0.847	0.504	0.414	0.489	0.840	0.878
Epilepsy	**1.000**	**1.000**	0.949	0.991	0.964	0.964	0.761	0.980	0.971	0.957	0.987
Ering	0.875	0.133	**0.964**	0.447	0.133	0.874	0.133	0.881	0.133	0.852	0.133
EthanolConcentration	**0.476**	0.430	0.326	0.452	0.323	0.308	0.373	0.240	0.323	0.297	0.312
FaceDetection	**0.683**	0.545	0.681	0.647	0.529	0.501	0.545	0.575	0.556	0.536	0.602
FingerMovements	**0.601**	0.490	0.560	0.553	0.530	0.480	0.580	0.568	0.530	0.470	0.580
HandMovementDirection	0.443	0.365	0.243	**0.446**	0.231	0.338	0.365	0.443	0.378	0.324	0.338
HandWriting	**0.672**	0.605	0.359	0.567	0.286	0.515	0.286	0.668	0.357	0.249	0.451
HeartBeat	**0.863**	0.727	0.776	0.717	0.717	0.515	0.663	0.489	0.751	0.746	0.756
JapaneseVowels	0.967	0.984	**0.994**	0.962	0.949	0.984	0.976	0.991	0.965	0.978	0.984
Libras	**0.981**	0.973	0.656	0.906	0.870	0.867	0.856	0.950	0.850	0.817	0.856
LSST	0.782	**0.878**	0.408	0.632	0.551	0.537	0.373	0.413	0.568	0.595	0.590
MotorImagery	**0.632**	0.590	0.500	0.531	0.500	0.510	0.510	0.535	0.590	0.500	0.610
NATOPS	0.941	0.500	0.850	0.885	0.883	0.928	0.889	**0.968**	0.939	0.911	0.883
PEMS-SF	**0.932**	0.870	0.919	0.751	0.711	0.682	0.699	0.760	0.751	0.699	0.751
PenDigits	0.991	0.968	0.560	**0.996**	0.977	0.989	0.978	0.985	0.980	0.979	0.977
Phoneme	0.287	0.190	0.085	0.284	0.151	0.233	0.110	**0.299**	0.175	0.207	0.298
RacketSports	**0.934**	0.190	0.809	0.928	0.803	0.855	0.803	0.877	0.868	0.776	0.882
SelfRegulationSCP1	**0.961**	0.934	0.925	0.908	0.775	0.812	0.874	0.835	0.652	0.799	0.782
SelfRegulationSCP2	**0.738**	0.710	0.589	0.533	0.539	0.578	0.472	0.532	0.550	0.550	0.578
SpokenArabicDigits	0.994	0.460	0.993	0.712	0.963	0.988	0.990	**0.997**	0.983	0.934	0.975
StandWalkJump	**0.659**	0.333	0.267	0.456	0.200	0.467	0.067	0.383	0.400	0.400	0.533
UWaveGestureLibrary	**0.951**	0.916	0.903	0.944	0.903	0.906	0.891	0.927	0.894	0.759	0.906
InsectWingBeat	**0.697**	0.163	0.105	0.168	0.105	0.466	0.167	0.667	0.208	0.469	0.250
Average Accuracy	**0.808**	0.658	0.658	0.698	0.628	0.698	0.621	0.704	0.657	0.670	0.699
Average Rank	**1.57**	4.93	6.63	4.93	7.83	6.17	7.77	4.77	6.07	8.00	4.83

the impact of randomized parameter initialization. We used dropout to avoid possible overfitting. Training and testing are done on a single Nvidia GTX 3080 Ti.

We use *accuracy*, which is currently used by all baseline methods, as the metric for comparison. We additionally use macro *precision*, *recall*, and *F1-Score* in our parameter and ablation studies to gain further insights into our model's performance.

4.4 Comparison Results

The performance comparison results (shown in Table 2) reveal that our model has demonstrated superior performance to all the baseline methods across a wide range of experimental datasets. Specifically, our model achieved the best results on 21 datasets, the second-best performance on six datasets, and the third-best on two datasets out of 30 experimental datasets. It demonstrated superior performance compared to all baselines, achieving a 14.7% increase in average classification accuracy compared to the second-best method, OS-CNN, and a 15.6% increase compared to the third-best method, ShapeNet. Furthermore, our model achieved an average rank of 1.57, outperforming the second-best method, OS-CNN, which had an average rank of 4.77. Figure 4 shows the result of the Wilcoxon signed-rank test (with a confidence level of 95%) on the base-

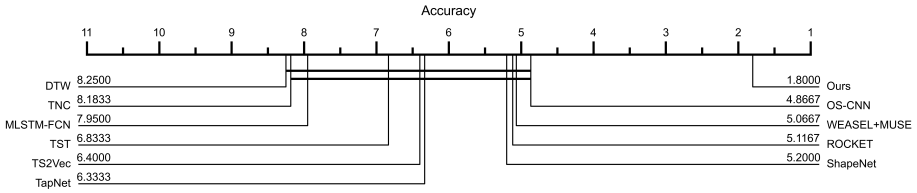


Fig. 4: Critical Difference (CD) diagram of the selected baselines and our method with a confidence level of 95%.

line methods' performance, consistently showing that our method achieved the highest classification performance among all the compared methods.

Traditional machine learning methods, including WEASEL+MUSE, DTW, and ROCKET, are limited in handling such large datasets, reflected in their inferior performance on datasets including InsectWingBeat and FaceDetection, which contain 50,000 and 9,114 samples, respectively. Furthermore, existing deep learning models often ignore the inherent frequency information in time series data, which can be crucial for accurately classifying time series with significant differences in the frequency domain rather than the time domain.

We attribute this improvement to two key factors. First, our method's dual-stream architecture effectively captures both temporal and frequency information, enhancing its ability to discriminate time series sequences between different classes. Second, by utilizing the Short-Time Fourier Transform (STFT), our method leverages the frequency components and their temporal locations of the time series to provide more comprehensive frequency information compared to the Fourier Transform. Our results from the Wilcoxon signed-rank test, conducted with a confidence level of 95%, further confirm that our method achieved the best classification performance among all compared methods.

4.5 Convolutional Backbone Selection Sensitivity

We replaced the ResNet with other popular computer vision backbones including ResNeXt [37], Res2Net [10], ResNeSt [44], and Inception [29] to explore the impact of the backbone selection on the performance. We conducted experiments on three datasets including DuckDuckGeese, HeartBeat, and HandWriting. The results can be found in Table 3. The tested backbones have more complex architectures and parameters compared to ResNet, leading to better performance during the training phase but overfitting on the test sets. We believe that with the increase of the dataset scale, implementing more complicated backbones may enhance the classifier's classification capacity. As most of the datasets we use contain limited samples in the training set (fewer than 1000), we selected ResNet as the optimal solution based on our evaluation of the performance metrics.

Table 3: The Training and test results of different backbones from the computer vision field. The best performance values are bolded.

		Training				Test			
Dataset	Models	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
DuckDuckGeese	ResNet	0.862	0.813	0.764	0.788	**0.767**	**0.741**	0.736	**0.738**
	ResNeXt	0.866	0.809	**0.866**	0.837	0.673	0.645	0.639	0.642
	Res2Net	0.894	0.815	0.827	0.821	0.639	0.613	0.632	0.622
	ResNeSt	**0.907**	**0.915**	0.855	**0.884**	0.692	0.704	**0.761**	0.731
	Inception	0.859	0.897	0.811	0.852	0.734	0.729	0.736	0.732
HeartBeat	ResNet	0.906	0.891	0.882	0.886	0.863	0.772	0.795	0.783
	ResNeXt	0.916	0.902	0.914	0.908	0.741	0.726	0.719	0.722
	Res2Net	**0.931**	**0.919**	0.922	0.920	0.714	0.678	0.669	0.673
	ResNeSt	0.928	0.917	**0.927**	**0.922**	0.665	0.640	0.608	0.624
	Inception	0.909	0.874	0.907	0.890	**0.782**	**0.738**	**0.806**	**0.771**
HandWriting	ResNet	0.735	0.702	0.744	0.722	**0.672**	**0.654**	**0.661**	**0.657**
	ResNeXt	0.849	0.865	0.872	0.868	0.533	0.542	0.591	0.565
	Res2Net	0.856	0.802	0.874	0.836	0.592	0.607	0.586	0.596
	ResNeSt	**0.857**	**0.886**	**0.883**	**0.884**	0.557	0.573	0.605	0.589
	Inception	0.764	0.753	0.773	0.763	0.597	0.612	0.596	0.604

Table 4: The experimental results when using our frequency stream with 2D representations as a plugin. W/o means that the method does not contain the spectrogram stream and vice versa. The best performance values are bolded.

		Train				Test			
Dataset	Method	Accuracy	Recall	Precision	F1-Score	Accuracy	Recall	Precision	F1-Score
EigenWorms	MLSTM-FCN (w/o)	0.587	0.574	0.624	0.598	0.504	0.519	0.479	0.498
	MLSTM-FCN (w)	**0.721**	**0.714**	**0.677**	**0.695**	**0.629**	**0.624**	**0.595**	**0.609**
	TST (w/o)	0.839	0.832	0.816	0.824	0.748	0.791	0.778	0.784
	TST (w)	**0.882**	**0.893**	**0.885**	**0.889**	**0.826**	**0.828**	**0.819**	**0.823**
RacketSports	MLSTM-FCN (w/o)	0.828	0.779	**0.833**	0.805	0.803	0.709	0.702	0.705
	MLSTM-FCN (w)	**0.843**	**0.811**	0.805	**0.808**	**0.814**	**0.727**	**0.751**	**0.739**
	TST (w/o)	0.854	0.819	0.822	0.820	0.809	0.712	0.705	0.708
	TST (w)	**0.894**	**0.833**	**0.882**	**0.857**	**0.824**	**0.762**	**0.793**	**0.777**

4.6 Impact of Our Spectrogram Stream as a Plugin

We incorporate the spectrogram stream as a plugin into the existing architectures including TST [42] and MLSTM-FCN [15] to evaluate the effectiveness of the 2D representations with frequency information in improving the performance of the existing methods. The outcomes of our investigation, as presented in Table 4, indicate a significant improvement in the average classification accuracy and the F1-Score of both methods during both the training and testing phases. Specifically, we observed an increase of 8.6% and 7.3% in the average classification accuracy and F1-Score, respectively, during the training phase, and an increase of 9.6% and 10.4% in the average classification accuracy and F1-Score, respectively, during the test phase. These findings suggest that the utilization of 2D representations with frequency information can enhance the performance of existing methods.

Table 5: Ablation test for our method. Fourier Transform means we use Fourier Transform instead of STFT to extract frequency information. Single window size means we only use one window size (three times the bandwidth) to generate the spectrogram. Time Series and Spectrogram Stream only mean using information from one stream separately instead of both to classify time series. The best performance values are bolded.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
DuckDuckGeese	Fourier Transform	0.675	0.669	0.688	0.678
	Single Window Size	0.689	0.707	0.725	0.716
	Time Series Stream Only	0.632	0.619	0.661	0.639
	Spectrogram Stream Only	0.718	0.711	0.724	0.717
	Ours	**0.767**	**0.741**	**0.736**	**0.738**
FaceDetection	Fourier Transform	0.575	0.602	0.552	0.576
	Single Window Size	0.627	0.585	0.673	0.626
	Time Series Stream Only	0.630	0.615	0.622	0.618
	Spectrogram Stream Only	0.647	0.651	0.642	0.646
	Ours	**0.681**	**0.622**	**0.716**	**0.666**
PEMS-SF	Fourier Transform	0.751	0.643	0.637	0.640
	Single Window Size	0.794	0.718	0.698	0.708
	Time Series Stream Only	0.819	0.822	0.803	0.812
	Spectrogram Stream Only	0.874	0.856	0.877	0.866
	Ours	**0.932**	**0.957**	**0.889**	**0.922**

4.7 Ablation Study

We conducted ablation studies on three datasets, including DuckDuckGeese, FaceDetection, and PEMS-SF, to investigate the effectiveness of individual components of our proposed method. We compared the performance of the method with the use of Fourier Transform instead of STFT to extract frequency information. Besides, for STFT, we use a single window size (three times the bandwidth) for spectrogram generation instead of three window sizes. Additionally, we tried to use information from one single stream (either the time series or spectrogram stream) individually to classify time series instead of both. The experimental results are summarized in Table 5.

Our analysis reveals that each component improves the classifier's performance. Notably, STFT demonstrates a more significant impact on the classification accuracy of the model on two of the datasets. This finding implies that the utilization of 2D representations with frequency information, provided by STFT, is crucial for enhancing the classification capacity of the model.

5 Conclusion and Future Work

This study proposes a novel dual-stream architecture for accurately classifying multivariate time series sequences. The method leverages the inherent frequency

information in the time series data by implementing STFT to obtain the frequency components and their temporal positions. We construct a dual-stream architecture based on ResNet, which can leverage both 1D and 2D representations effectively to classify multivariate time series sequences. We evaluate the proposed model on diverse datasets containing sequences of various lengths and variable numbers. The experimental results show that our method outperforms several baseline and state-of-the-art methods by a significant margin. We also conduct a thorough investigation of the effect of different components and settings on the model's performance. Our future work includes exploring the interpretability of our proposed method through visualization technologies for convolutional neural networks from the computer vision field. Additionally, we plan to extend our work to more time series-related tasks, such as time series imputation, forecasting, and abnormal detection.

References

1. Bengio, Y., LeCun, Y., et al.: Scaling learning algorithms towards ai. Large-scale kernel machines **34**(5), 1–41 (2007)
2. Bracewell, R.N., Bracewell, R.N.: The Fourier transform and its applications, vol. 31999. McGraw-Hill New York (1986)
3. Chen, Z., Ma, Q., Lin, Z.: Time-aware multi-scale rnns for time series modeling. In: IJCAI. pp. 2285–2291 (2021)
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. Coyle, D., Prasad, G., McGinnity, T.M.: A time-series prediction approach for feature extraction in a brain-computer interface. IEEE transactions on neural systems and rehabilitation engineering **13**(4), 461–467 (2005)
6. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The ucr time series classification archive (October 2018)
7. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery **34**(5), 1454–1495 (2020)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. 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 Mining and Knowledge Discovery **34**(6), 1936–1962 (2020)
10. Gao, S., Cheng, M.M., Zhao, K., Zhang, X.Y., Yang, M.H., Torr, P.H.: Res2net: A new multi-scale backbone architecture. IEEE transactions on pattern analysis and machine intelligence (2019)
11. Griffin, D., Lim, J.: Signal estimation from modified short-time fourier transform. IEEE Transactions on acoustics, speech, and signal processing **32**(2), 236–243 (1984)

12. Harris, F.J.: On the use of windows for harmonic analysis with the discrete fourier transform. Proceedings of the IEEE **66**(1), 51–83 (1978)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Karim, F., Majumdar, S., Darabi, H., Harford, S.: Multivariate lstm-fcns for time series classification. Neural Networks **116**, 237–245 (2019)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
18. Li, G., Choi, B., Xu, J., Bhowmick, S.S., Chun, K.P., Wong, G.L.H.: Shapenet: A shapelet-neural network approach for multivariate time series classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 8375–8383 (2021)
19. Lines, J., Taylor, S., Bagnall, A.: Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. ACM Transactions on Knowledge Discovery from Data **12**(5) (2018)
20. Liu, C.L., Hsaio, W.H., Tu, Y.C.: Time series classification with multivariate convolutional neural network. IEEE Transactions on Industrial Electronics **66**(6), 4788–4797 (2018)
21. Liu, M., Kim, Y.: Classification of heart diseases based on ecg signals using long short-term memory. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 2707–2710. IEEE (2018)
22. Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z., Song, W.: Gated transformer networks for multivariate time series classification. arXiv preprint arXiv:2103.14438 (2021)
23. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.Y., Sainath, T.: Deep learning for audio signal processing. IEEE Journal of Selected Topics in Signal Processing **13**(2), 206–219 (2019)
24. Ren, Y., Li, L., Yang, X., Zhou, J.: Autotransformer: Automatic transformer architecture design for time series classification. In: Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I. pp. 143–155. Springer (2022)
25. Schäfer, P., Leser, U.: Multivariate time series classification with weasel+ muse. arXiv preprint arXiv:1711.11343 (2017)
26. Shifaz, A., Pelletier, C., Petitjean, F., Webb, G.I.: Ts-chief: a scalable and accurate forest algorithm for time series classification. Data Mining and Knowledge Discovery **34**(3), 742–775 (2020)
27. Smirnov, D., Nguifo, E.M.: Time series classification with recurrent neural networks. Advanced analytics and learning on temporal data **8** (2018)
28. Spiegel, S., Gaebler, J., Lommatzsch, A., De Luca, E., Albayrak, S.: Pattern recognition and classification for multivariate time series. In: Proceedings of the fifth international workshop on knowledge discovery from sensor data. pp. 34–42 (2011)
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

30. Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., Jiang, J.: Omni-scale cnns: a simple and effective kernel size configuration for time series classification. In: International Conference on Learning Representations (2021)
31. Tonekaboni, S., Eytan, D., Goldenberg, A.: Unsupervised representation learning for time series with temporal neighborhood coding. In: International Conference on Learning Representations (2020)
32. Tran, T.M., Le, X.M.T., Nguyen, H.T., Huynh, V.N.: A novel non-parametric method for time series classification based on k-nearest neighbors and dynamic time warping barycenter averaging. Engineering Applications of Artificial Intelligence **78**, 173–185 (2019)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
34. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. arXiv preprint arXiv:2202.01575 (2022)
35. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: Etsformer: Exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381 (2022)
36. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 (2022)
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
38. Yang, C., Jiang, W., Guo, Z.: Time series data classification based on dual path cnn-rnn cascade network. IEEE Access **7**, 155304–155312 (2019)
39. Yang, C., Wang, X., Yao, L., Long, G., Jiang, J., Xu, G.: Attentional gated res2net for multivariate time series classification. Neural Processing Letters pp. 1–25 (2022)
40. Yang, J., Nguyen, M.N., San, P.P., Li, X., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: Ijcai. vol. 15, pp. 3995–4001. Buenos Aires, Argentina (2015)
41. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. arXiv preprint arXiv:2106.10466 (2021)
42. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2114–2124 (2021)
43. Zhang, D., Zuo, W., Zhang, D., Zhang, H.: Time series classification using support vector machine with gaussian elastic metric kernel. In: 2010 20th International Conference on Pattern Recognition. pp. 29–32. IEEE (2010)
44. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., et al.: Resnest: Split-attention networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2736–2746 (2022)
45. Zhang, X., Gao, Y., Lin, J., Lu, C.T.: Tapnet: Multivariate time series classification with attentional prototypical network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6845–6852 (2020)
46. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics **28**(1), 162–169 (2017)