Using Motif-Based Features to Improve Signal Classification with Temporal Neural Networks

Karthikeyan Suresh[1] and Brian Kenji Iwana[2](✉)

[1] University of Lorraine, GENIAL, Lorraine, France
[2] Department of Advanced Information Technology, Kyushu University, Fukuoka, Japan
iwana@ait.kyushu-u.ac.jp

Abstract. Time series motifs are repeated subsequences within a time series. Finding motifs is a large field in time series recognition. For example, Matrix Profile is a robust and scalable data structure that helps with motif discovery. In this paper, we demonstrate that motif discovery, namely Matrix Profile, can be used as a feature to increase the accuracy of temporal neural networks. While current temporal neural networks are effective, they lack specific considerations of time series properties, such as periodicity, motifs, and discords, etc. Therefore, we propose using multi-modal fusion networks to classify signals that use both the original features and the motif-based features. We demonstrate the proposed method's effectiveness on all of the signal and device datasets of the 2018 UCR Time Series Classification Archive. The results show that the proposed Matrix Profile-based features are useful for fusion networks and can increase the accuracy of temporal neural networks.

Keywords: Matrix Profile · Fusion Neural Network · Signal Classification · Time Series Motifs

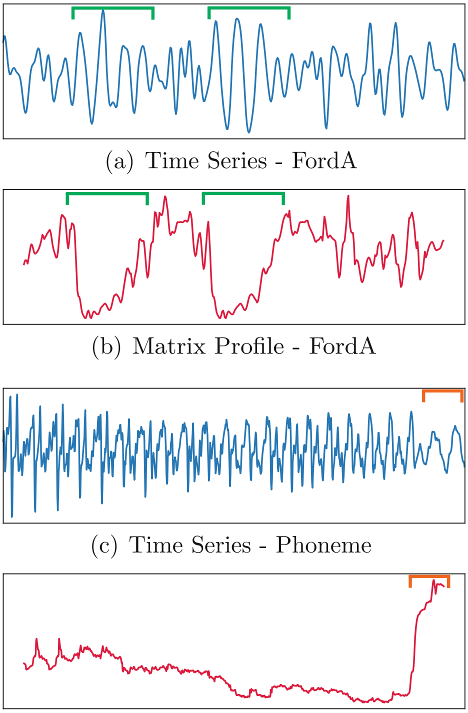
1 Introduction

Recently, many state-of-the-art time series classification methods use Artificial Neural Networks (ANN), such as Recurrent Neural Networks (RNN) [29] and Temporal Convolutional Neural Networks (TCNN) [38]. They have shown to be widely effective for time series and signal classification [3,38]. However, while neural networks might be effective for time series, they do not inherently have considerations for some aspects and elements of signals, such as motifs and discords.

Time series motifs are repeated subsequences within a time series and discords are anomalies in times series. Finding motifs and discords, or the field of *motif discovery*, is essential for finding patterns in time series. Motif discovery has been used for time series analysis in many domains, such as protein sequences [22,41], actions [36], sounds [9], and signals [23,25].

This research was partially supported by MEXT-Japan (Grant No. JP23K16949).

H. Lu et al. (Eds.): ACPR 2023, LNCS 14406, pp. 123–136, 2023.
https://doi.org/10.1007/978-3-031-47634-1_10



(a) Time Series - FordA

(b) Matrix Profile - FordA

(c) Time Series - Phoneme

(d) Matrix Profile - Phoneme

Fig. 1. Example of time series and the resulting Matrix Profile. The green brackets of (a) and (b) are an example of a motif, and the orange brackets of (c) and (d) are an example of a discord. (Color figure online)

One powerful tool used for motif discovery is Matrix Profile [40]. Matrix Profile is a robust and scalable data structure that helps with motif discovery. Specifically, Matrix Profile is a sequence calculated based on a sliding window of subsequences and the distance to its nearest neighbor subsequence. An example time series and Matrix Profile calculation result is shown in Fig. 1. In the figure, the dips in the Matrix Profile correspond to the locations of the motifs and the peaks are discords. The use of Matrix Profile has shown to be effective at large-scale motif discovery [39, 42].

We propose the inclusion of motif and discord information as supplemental information for signal classification. Namely, Matrix Profile is used to improve the classification ability of temporal neural networks by providing additional motif-based features alongside the original signal features. This is done by considering the Matrix Profile vector as a sequence and combining it with the original time series features in fusion neural networks. These motif-based features can be considered a self-augmented extra modality to represent the signal. Therefore, we are able to use both features in a single multi-modal model. Through this, we demonstrate that the motif-based features can supplement the original time series features and improve classification.

The contribution of this paper is as follows:

- We propose the use of Matrix Profile-based features to supplement time series in classification. This is done using fusion networks to combine the features.
- We demonstrate that the proposed method can improve the accuracy of neural networks in signal classification. To do this, we evaluate the proposed method on 24 datasets taken from the 2018 University of California Riverside (UCR) Time Series Archive [7]. The 24 datasets are all of the sensor and device datasets with at least 100 training patterns from the archive.
- We examine the effect that the window size of Matrix Profile has on the accuracy of the proposed method.
- The code for the proposed method can be found at https://github.com/uchidalab/motif-based-features

2 Related Work

The use of fusion neural networks is a common solution for multi-modal data recognition [11]. They have been used for a wide range of applications. Of them, there are a few works that use fusion neural networks with different features extracted from the same time series. For example, the Long Short-Term Memory Network Fully Convolutional Network (LSTM-FCN) [17] uses a fusion network to combine an LSTM branch and an FCN branch for time series classification. Similarly, Song et al. [34] combine the features from an LSTM and a CNN for time series retrieval. Features can also be derived or learned from the original time series representation. For example, Iwana et al. [16] propose using local distance-based features with the original features in fusion 1D CNNs and Oba et al. [24] combines data augmentation methods in a gated fusion network. Matsuo et al. [20] uses a learned self-augmentation by converting the time series into images and then uses a multi-modal network. Wang et al. [37] uses a fusion network with multi-scale temporal features and distance features.

3 Using Matrix Profile as a Feature Extraction Method

3.1 Motif Discovery

A motif is a repeated pattern in a time series. Specifically, given time series $\mathbf{t} = t_1, \ldots, t_n, \ldots, t_N$ of length N and $t_n \in \mathbb{R}$, a continuous subsequence $\mathbf{t}_{s,M} = t_s, \ldots, t_{s+M-1}$ of length M starting from position s, where $1 \leq s \leq N - M + 1$, is a motif if it shares similar values with any other subsequence $\mathbf{t}_{s',M}$ within $\mathbf{t}$ with a different start position s'. Note, time series element t_n can one dimension (univariate) or multiple dimensions (multivariate).

Motif discovery refers to finding sets of similar short sequences in a large time series dataset. Motifs are essential as these primitive patterns can be used as inputs for algorithms to perform segmentation, classification, anomaly detection, etc. Further, studying motifs can provide insight into the functional properties of the time series [43].

3.2 Matrix Profile

Matrix Profile [40] is a powerful motif discovery algorithm that represents times series based on the distances of subsequences to their nearest neighbors. Specifically, using a sliding window, we can extract all of the subsequences from the time series. We can then compute the pairwise distances between these subsequences and store them in the form of a matrix. This matrix is then stored in a vector that only holds the information on the distances of each subsequence to the nearest neighbor of that subsequence. This vector is called the *Matrix Profile*.

Namely, given time series $\mathbf{t}$, first the *all-subsequences set* is created. The all-subsequences set $\mathcal{A}$ is an ordered set of all possible subsequences of time series $\mathbf{t}$, obtained by a sliding window of length M across $\mathbf{t} = t_1, \ldots, t_{N-M+1}$, where M is a user-defined subsequence length. We use $\mathcal{A}[s]$ to denote the subsequence $\mathbf{t}_{s,M} = t_s, \ldots, t_{s+M-1}$.

Next, a *Distance Profile* $\mathbf{d}_i$ is created for each subsequence in $\mathcal{A}$. The Distance Profile is the ordered vector of distances between each subsequence in all-subsequences set $\mathcal{A}$ and all other subsequences in $\mathcal{A}$. For this distance, traditionally, the Euclidean distance is used. Using each Distance Profile, a *similarity join set* $\mathcal{S}$ is constructed with each subsequence $\mathcal{A}$ with its nearest neighbor,

$$\mathcal{S}_s = \arg\min_{s'} \mathbf{d}_s[s'], \tag{1}$$

for each s-th subsequence of $\mathcal{A}$. Matrix Profile $\mathbf{p}$ is the vector of distances between $\mathcal{A}$ and $\mathcal{S}$, or:

$$\mathbf{p} = ||\mathcal{A}_1 - \mathcal{S}_1||, \ldots, ||\mathcal{A}_s - \mathcal{S}_s||, \ldots, ||\mathcal{A}_{N-M+1} - \mathcal{S}_{N-M+1}||. \tag{2}$$

An example of the result of a Matrix Profile calculation is shown in Fig. 1. Matrix Profile has many advantages over conventional methods of motif discovery and anomaly detection representations. To state a few, it is space efficient, parallelizable, scalable, and can be used efficiently on streams of data [40].

3.3 Matrix Profile as a Motif-Based Feature

As described previously, Matrix Profile is a vector that is used to identify motifs and discords by containing the distance of each subsequence to its nearest neighbor. In other words, the values of the Matrix Profile will be small for repeated subsequences and large for anomalies. The values of the Matrix Profile vector have a nonlinear relationship to the original time series features. Thus, it is possible to exploit Matrix Profile to create a feature vector that contains information that is not inherent to the original time series features.

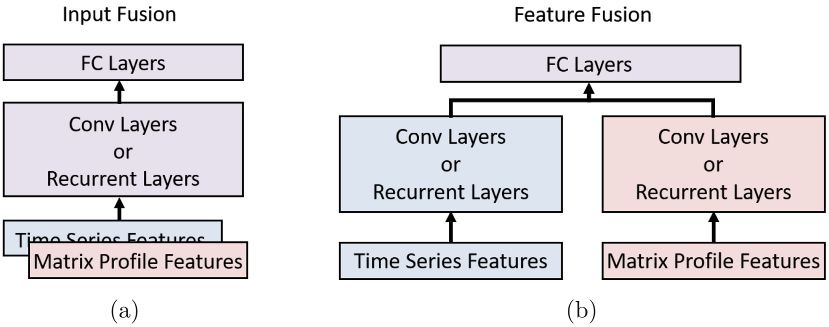
In order to use the Matrix Profile features, we consider vector $\mathbf{p}$ as a sequence, or:

$$\mathbf{f} = \mathbf{p}^\top = p_1, \ldots, p_s, \ldots, p_{N-M+1}. \tag{3}$$

This gives a sequence $\mathbf{f}$ of length $N - M + 1$, which is similar in size to the original time series features $\mathbf{t}$. The motif feature sequence $\mathbf{f}$ can now be used alongside the original $\mathbf{t}$ in multi-modal classification.

4 Multi-modal Classification with Fusion Neural Networks

The original features **t** and the motif features **f** are different modalities that contain different information about the same signal. Therefore, we combine the two features in one multi-modal model to improve classification. Using the additional motif features can supplement the original time series.



(a)

(b)

Fig. 2. Different arrangements of combining features in multi-modal fusion networks.

There are various methods of creating a multi-modal classification model. We propose to use a fusion neural network. Specifically, we implement a multi-modal neural network and combine the modalities through model fusion. The modalities can be combined at different points in the neural network. Some common places modality fusion can take place is at the input level or at the feature level, as shown in Fig. 2. Input-level fusion combines the inputs by concatenating them. The combined input is then used with a typical temporal neural network. Feature-level fusion concatenates separate modality branches within a network.

5 Experimental Results

5.1 Data

In order to evaluate the proposed method, we use 24 time series datasets from the UCR Time Series Archive [7]. The datasets are all of the device and sensor type datasets with at least 100 training patterns. There are 8 device datasets, ACSF1, Computers, ElectricDevices, LargeKitchenAppliances, PLAID, RefrigerationDevices, ScreenType, and SmallKitchenAppliances and 16 sensor datasets, AllGestureWiimoteX, AllGestureWiimoteY, AllGestureWIimoteZ, ChlorineConcentration, Earthquakes, FordA, FordB, FreezerRegularTrain, GesturePebbleZ1, GesturePebbleZ2, InsectWingbeatSound, Phoneme, Plane, StarLightCurves, Trace, and Wafer. The sensor and device datasets are used since they are examples of signal data. However, there is no theoretical limitation to the type of time series used. Furthermore, datasets with less than 100 training patterns are not used because very small datasets are not suitable for neural networks.

The datasets used in the experiments have a wide range of lengths. The dataset with the shortest time series is the ElectricDevices dataset with 96 time steps and the longest one is the ACSF1 dataset with 1,460 time steps. Six of the datasets, AllGestureWiimoteX, AllGestureWiimoteY, AllGestureWiimoteZ, GesturePebbleZ1, GesturePebbleZ2, and PLAID have varying number of time steps. For these datasets, we pre-process them by post pattern zero padding. In addition, all the datasets except for the six previously mentioned datasets were already z-normalized by the creators of the datasets.

For the motif-based features, the Matrix Profile algorithm is applied to the signal. We use a Matrix Profile window size of 7% of the longest time series in the training dataset. This window size is determined through a parameter search, shown in Sect. 5.6. Also, for the input-fusion networks, because the feature sequence lengths are different by $M - 1$, the motif features $\mathbf{f}$ are post zero-padded. Finally, the motif features are z-normalized based on the training set.

5.2 Architecture and Settings

Two time series recognition architectures were used as the foundation of the experiments, a 1D Very Deep Convolutional Network (VGG) [33] and a Bidirectional LSTM (BLSTM) [31]. The 1D VGG is a VGG adapted for time series in that it uses 1D convolutions instead of the standard 2D convolutions. It has multiple blocks of convolutional layers followed by max pooling layers. There are two fully-connected layers with 1,024 nodes and dropout with a probability of 0.5. The number of blocks and filters per convolution are determined by the suggestions of Iwana and Uchida [15]. For the BLSTM, there are two layers of 100 cells each. The hyperparameters of the BLSTM are the optimal suggestions by Reimers et al. [28]. In the case of the feature-fusion networks, two streams with the same hyperparameters are used and the concatenation is performed before the first fully-connected layer of each.

For training the 1D VGG, Stochastic Gradient Decent (SGD) with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4} is used. These settings are suggested by [33]. For the BLSTM, following [28], we use an Nadam [8] optimizer with an initial learning rate of 0.001. For both networks, we use batch size 50 and train for 10,000 iterations. The datasets used have fixed training and test sets that are provided by the dataset authors.

5.3 Comparison Methods

To demonstrate the effectiveness of the proposed method, the following evaluations were performed:

- *Single-Modality Network with Time Series Features (Single, TS)*: The original time series features are used as a baseline.

- *Single-Modality Network with Matrix Profile Features (Single, MP)*: This uses the same networks as TS, but using only the Matrix Profile-based features. Single MP (7%) refers to using a 7% window for matrix profile. Single MP (Best) uses the best window for each dataset.
- *Input-Level Fusion with Time Series and Matrix Profile Features (Input Fusion, TS+MP)*: For the input-level fusion network, the time series features and Matrix Profile features are concatenated in the dimension direction and then fed to the neural network.

The evaluation of all of the comparisons use the same hyperparameters and training scheme, with exception of the feature-level fusion which has two modality streams with their own set of feature extraction layers.

5.4 Results

Training and testing were performed five times, and the average of the five results was recorded as the final value in order to obtain an accurate representation of the accuracy of each method. The results are shown in Table 1. For the applicable comparisons, the window size was set to a fixed percentage of the longest length time series in the training set. For the MP (7% Win.) features, all datasets use a window of 7% of the maximum time series length. For MP (Best Win.), the best accuracy is used for each dataset.

From the table, it can be seen that input fusion with the time series features and the proposed Matrix Profile features had the highest accuracy for most datasets. The results with the best window for each dataset predictably had the highest accuracy. However, a window of 7% of the time series length still performed better than without using the fusion network. This is true for both the 1D VGG and the BLSTM.

We also compare the proposed method to other comparison methods. A Nemenyi test is performed using results reported from literature. Figure 3 compares the proposed BLSTM and VGG using TS+MP (7%) and (Best) to Bag of Patterns (BoP) [18], Bag of Symbolic Fourier Approximation Symbols (BOSS) [30], Collective of Transformation Ensembles (COTE) [2], Complexity Invariant Distance (CID) [4], Derivative DTW (DD_{DTW}) [12], Derivative Transform Distance (DTD_C) [13], Elastic Ensemble (EE) [19], Fast Shapelets (FS) [26], Feature Fusion CNN using Local Distance Features and series features (CNN LDF+TS) [16], Learned Pattern Similarity (LPS) [6], Learned Shapelets (LS) [21], Multilayer Perceptron (MLP) [1], Random Forest (RandF) [1], Residual Network (ResNet) [10], Rotation Forest (RotF) [1], Shapelet Transform (ST) [14], Symbolic Aggregate Approximation - Vector Space Model (SAXVSM) [32], SVM with a linear kernel (SVML) [1], Time Series Bag of Features (TSBF) [5], SVM with a quadratic kernel (SVMQ) [1], 1-NN with Euclidean Distance (1-NN ED) [7], 1-NN with DTW (1-NN DTW) [7], 1-NN with DTW with the best warping window (1-NN DTW (Best)) [27], and 1-NN with Move-Split-Merge (1-NN MSM) [35]. In the figure, BLSTM and VGG refer to the previous models with only time series features. It can be seen that the

Table 1. Average Test Accuracy (%) Trained Five Times

Type:	Single	Single	Single	Input Fusion	Input Fusion
Feature:	TS	MP (7%)	MP (Best)	TS+MP (7%)	TS+MP (Best)
1D VGG Backbone					
ACSF1	34.6	27.6	31.6 (15%)	36.8	**43.0** (13%)
AllGestureWiimoteX	**70.3**	27.1	28.2 (5%)	68.6	69.6 (9%)
AllGestureWiimoteY	**72.2**	25.5	26.3 (5%)	68.9	71.3 (3%)
AllGestureWiimoteZ	**63.9**	29.7	31.0 (13%)	58.9	**63.9** (15%)
ChlorineConcentration	**83.4**	69.0	69.0 (7%)	73.2	74.3 (15%)
Computers	60.7	50.0	55.0 (5%)	62.8	**63.4** (1%)
Earthquakes	68.8	72.9	**74.8** (Multi)	70.2	71.4 (11%)
ElectricDevices	70.4	63.8	63.8 (7%)	**72.3**	**72.3** (7%)
FordA	**94.0**	55.0	86.0 (3%)	93.5	**94.0** (13%)
FordB	82.6	55.9	75.7 (1%)	82.3	**83.1** (15%)
FreezerRegularTrain	**99.2**	55.5	88.8 (5%)	94.9	97.8 (15%)
GesturePebbleZ1	16.0	44.8	53.8 (5%)	59.5	**74.4** (3%)
GesturePebbleZ2	15.3	34.7	34.7 (7%)	71.3	**84.4** (1%)
InsectWingbeatSound	59.1	37.2	42.6 (13%)	**59.5**	**59.5** (7%)
LargeKitchenAppliances	79.8	53.7	14.4 (3%)	78.6	**81.8** (1%)
Phoneme	15.9	10.7	14.4 (3%)	15.5	**17.6** (5%)
PLAID	12.2	36.0	**72.9** (5%)	12.2	12.2 (Multi)
Plane	98.7	96.0	**99.0** (13%)	97.1	98.7 (11%)
RefrigerationDevices	47.9	42.0	47.8 (3%)	51.0	**52.2** (9%)
ScreenType	39.9	38.3	43.4 (1%)	40.7	**51.4** (1%)
SmallKitchenAppliances	70.5	54.8	72.4 (3%)	75.5	**78.4** (13%)
StarLightCurves	**96.7**	92.0	93.7 (11%)	96.1	96.6 (15%)
Trace	83.4	78.2	87.0 (5%)	99.6	**100** (5%)
Wafer	99.6	99.6	**100** (3%)	99.6	99.9 (3%)
Total	63.97	52.08	60.44	68.27	**71.38**
BLSTM Backbone					
ACSF1	**73.0**	34.0	34.0 (7%)	61.0	68.5 (5%)
AllGestureWiimoteX	**68.4**	24.6	30.4 (13%)	67.5	67.7 (9%)
AllGestureWiimoteY	65.2	24.5	26.1 (9%)	65.9	**68.7** (9%)
AllGestureWiimoteZ	63.9	31.6	34.1 (15%)	59.8	**66.0** (15%)
ChlorineConcentration	**61.8**	58.4	59.4 (9%)	61.1	61.1 (7%)
Computers	60.0	60.5	61.4 (7%)	59.9	**62.9** (11%)
Earthquakes	73.4	69.4	**74.6** (15%)	73.6	74.1 (15%)
ElectricDevices	69.6	63.0	63.0 (7%)	**73.3**	**73.3** (7%)
FordA	60.2	76.5	80.5 (3%)	**91.1**	**91.1** (7%)
FordB	69.5	66.1	71.1 (3%)	78.0	**82.6** (1%)
FreezerRegularTrain	92.5	83.3	89.5 (15%)	93.4	**95.9** (11%)
GesturePebbleZ1	43.5	51.7	55.2 (11%)	44.6	**67.6** (9%)
GesturePebbleZ2	40.3	43.2	57.0 (13%)	44.5	**70.6** (15%)
InsectWingbeatSound	**55.8**	29.5	35.8 (15%)	51.9	54.5 (11%)
LargeKitchenAppliances	85.5	53.9	58.8 (1%)	84.6	**86.2** (5%)
Phoneme	11.7	**14.8**	**14.8** (7%)	13.0	14.1 (3%)
PLAID	50.5	73.8	75.9 (5%)	**76.0**	**76.0** (7%)
Plane	97.1	92.2	97.0 (15%)	99.0	**99.2** (11%)
RefrigerationDevices	45.3	46.6	49.4 (3%)	48.7	**53.4** (15%)
ScreenType	38.6	39.7	46.5 (3%)	41.3	**49.3** (1%)
SmallKitchenAppliances	68.7	73.0	75.7 (3%)	**80.3**	**80.3** (7%)
StarLightCurves	91.8	87.2	93.4 (15%)	95.8	**95.9** (9%)
Trace	84.0	70.5	**98.0** (15%)	80.0	90.7 (13%)
Wafer	99.1	99.0	99.5 (3%)	99.0	**99.6** (3%)
Total	65.39	56.96	61.70	68.47	**72.99**

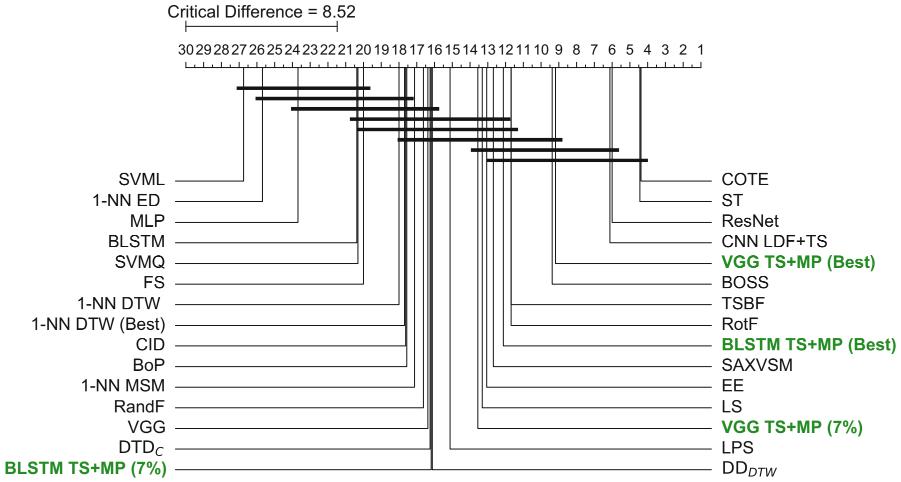


Fig. 3. Critical difference diagram using a Nemenyi test comparing the proposed method to reported methods. Green highlighted methods are the proposed input fusion networks. (Color figure online)

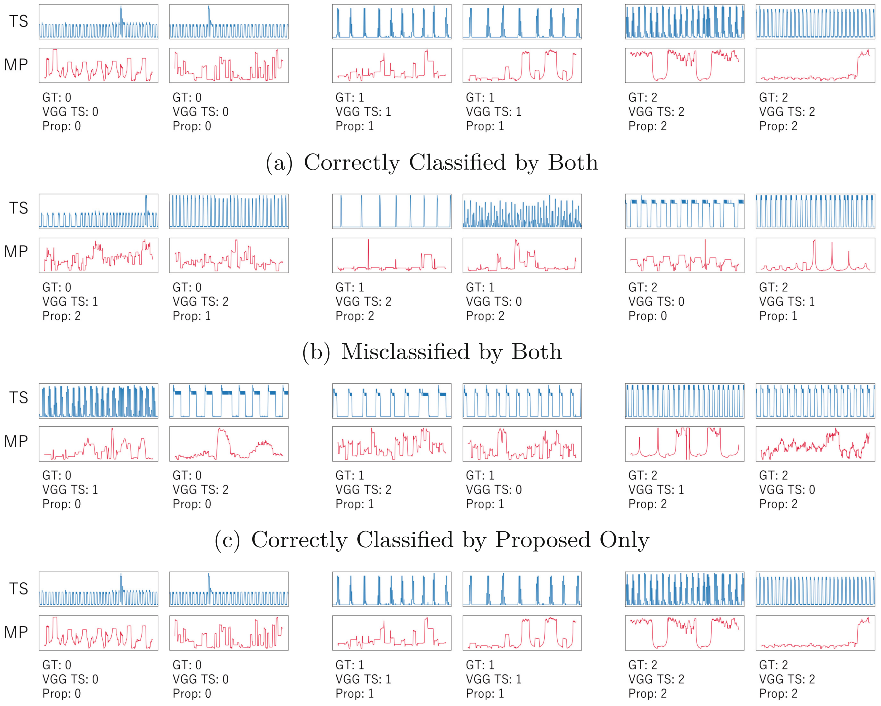
proposed method performed on the upper end of the methods. Using the best window with VGG was ranked higher than most of the models. The methods with better overall scores were a large ensemble of many classifiers (COTE), a classical method, and two other neural networks.

5.5 Qualitative Analysis

Figure 4 is a comparison of test samples classified by a standard VGG using the normal time series features and the proposed Input Fusion network with a VGG backbone using both the time series features and the matrix profile features. Noticeably, the proposed method performed better when the matrix profile features had more discords. In Figs. 4(b) and (c), the matrix profile features were often small with only sparse and narrow peaks. Conversely, when the proposed method excelled, the discord peaks were wider and more frequent. Thus, it can be inferred that the proposed method is better suited to signals with more frequent discords.

5.6 Ablation

An ablation study is performed to demonstrate the usefulness of adding matrix profile-based features. Table 2 compares using a network on a single feature (time series features or matrix profile features) as well as the difference in using an input fusion network versus a feature fusion network. Feature Fusion refers to a network that has branches for both features and concatenates them at the fully connected layer. Also, in order to demonstrate that the improved results



(a) Correctly Classified by Both

(b) Misclassified by Both

(c) Correctly Classified by Proposed Only

(d) Correctly Classified by VGG TS Only

Fig. 4. Examples of test patterns classified by Single VGG TS and the proposed Input Fusion VGG TS+MP (7%) from the RefrigerationDevices dataset. The upper row of each subfigure is the original Time Series feature and the lower row is the Matrix Profile feature. Two examples from each class are shown.

Table 2. Average Test Accuracy (%) of the 24 Datasets Trained Five Times

Network	Feature	Model	
		1D VGG	BLSTM
Single	TS	63.97	65.39
Single	MP (7% Window)	52.08	56.96
Single	MP (Best Window)	60.44	61.70
Input Fusion	TS+TS	63.32	66.47
Feature Fusion	TS+TS	64.07	64.33
Input Fusion	TS+MP (7% Window)	68.27	68.47
Input Fusion	TS+MP (Best Window)	**71.38**	**72.99**
Feature Fusion	TS+MP (7% Window)	66.08	66.95
Feature Fusion	TS+MP (Best Window)	67.91	70.45

of the proposed method are not strictly due to having more parameters, a fair comparison is made using only the time series features for modalities of the fusion networks (TS+TS).

Table 2 shows that using a fusion network with only time series does not have a significant difference with just having a single network. The results between Single TS and Input Fusion/Feature Fusion TS+TS are within a percent of each other. However, the TS+MP trials all have large increases over Single TS. This indicates that the matrix profile-based features provide supplemental information for the network to learn from.

5.7 Effect of Window Size

Matrix Profile has one hyperparameter, the window size. The value of this hyperparameter has an effect on the robustness of the proposed method. Figure 5 shows the average accuracy at different window sizes. As can be seen in the figure, after about a window size 7% of the time series, the accuracy starts to decrease. Furthermore, when using the Matrix Profile features only, the accuracy of the VGG quickly decreases after a small window. Despite this, the Input Fusion and Feature Fusion still increases.

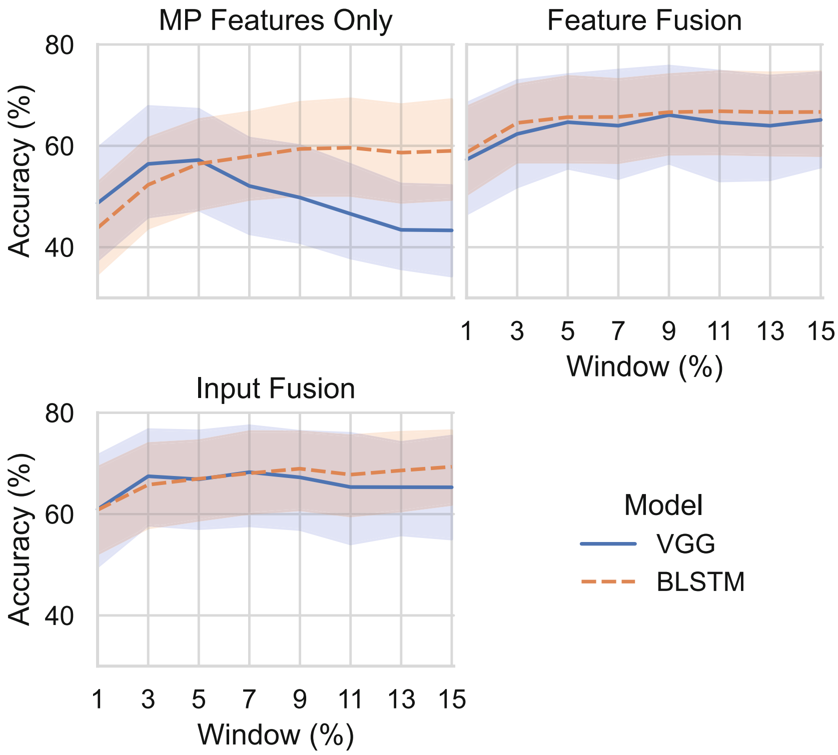


Fig. 5. Effect of the Matrix Profile window size.

6 Conclusion

In this paper, we propose the use of motif-based features to supplement time series in classification. The motif-based features are features that represent motifs and discords in time series. The features are created by using Matrix Profile to generate a second modality of data to represent the time series. Due to the features being similar in length to the original time series, we can use them as a sequence in multi-modal neural networks.

Through the experiments, we demonstrate that using the motif-based features alongside the time series features can be used to increase the accuracy of BLSTMs and Temporal CNNs in fusion networks. We performed an extensive evaluation using all of the signal and device time series patterns from the UCR Time Series Archive, which includes 24 time series datasets.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **31**(3), 606–660 (2016). https://doi.org/10.1007/s10618-016-0483-9
2. Bagnall, A., Lines, J., Hills, J., Bostrom, A.: Time-series classification with COTE: The collective of transformation-based ensembles. In: ICDE (2016). https://doi.org/10.1109/icde.2016.7498418
3. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
4. Batista, G.E.A.P.A., Keogh, E.J., Tataw, O.M., de Souza, V.M.A.: CID: an efficient complexity-invariant distance for time series. Data Min. Knowl. Disc. **28**(3), 634–669 (2013). https://doi.org/10.1007/s10618-013-0312-3
5. Baydogan, M.G., Runger, G., Tuv, E.: A bag-of-features framework to classify time series. IEEE Trans. Pattern Analy. and Mach. Intell. **35**(11), 2796–2802 (2013). https://doi.org/10.1109/tpami.2013.72
6. Baydogan, M.G., Runger, G.: Time series representation and similarity based on local autopatterns. Data Min. Knowl. Disc. **30**(2), 476–509 (2015). https://doi.org/10.1007/s10618-015-0425-y
7. Dau, H.A., et al.: Hexagon-ML: The UCR time series classification archive (2018). https://www.cs.ucr.edu/~eamonn/time_series_data_2018/
8. Dozat, T.: Incorporating nesterov momentum into adam. In: ICLR Workshops (2016)
9. Exadaktylos, V., Silva, M., Ferrari, S., Guarino, M., Taylor, C.J., Aerts, J.M., Berckmans, D.: Time-series analysis for online recognition and localization of sick pig (sus scrofa) cough sounds. J. Acous. Soc. Am. **124**(6), 3803–3809 (2008)
10. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Data augmentation using synthetic data for time series classification with deep residual networks. arXiv preprint arXiv:1808.02455 (2018)
11. Gao, J., Li, P., Chen, Z., Zhang, J.: A survey on deep learning for multimodal data fusion. Neural Comput. **32**(5), 829–864 (2020). https://doi.org/10.1162/neco_a_01273

12. Górecki, T., Łuczak, M.: Using derivatives in time series classification. Data Mining Knowl. Disc. **26**(2), 310–331 (2012). https://doi.org/10.1007/s10618-012-0251-4
13. Górecki, T., Łuczak, M.: Non-isometric transforms in time series classification using DTW. Knowl.-Based Sys. **61**, 98–108 (2014). https://doi.org/10.1016/j.knosys.2014.02.011
14. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Min. Knowl. Disc. **28**(4), 851–881 (2013). https://doi.org/10.1007/s10618-013-0322-1
15. Iwana, B.K., Uchida, S.: An empirical survey of data augmentation for time series classification with neural networks. PLoS ONE (2021). https://doi.org/10.1371/journal.pone.0254841
16. Iwana, B.K., Uchida, S.: Time series classification using local distance-based features in multi-modal fusion networks. Pattern Recogn. **97**, 107024 (2020). https://doi.org/10.1016/j.patcog.2019.107024
17. Karim, F., Majumdar, S., Darabi, H., Chen, S.: LSTM fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2018). https://doi.org/10.1109/access.2017.2779939
18. Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. J. Intell. Info. Sys. **39**(2), 287–315 (2012). https://doi.org/10.1007/s10844-012-0196-5
19. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Min. Knowl. Disc. **29**(3), 565–592 (2014). https://doi.org/10.1007/s10618-014-0361-2
20. Matsuo, S., Iwana, B.K., Uchida, S.: Self-augmented multi-modal feature embedding. In: ICASSP (2021). https://doi.org/10.1109/icassp39728.2021.9413974
21. Mitzev, I.S., Younan, N.H.: Time series shapelets: training time improvement based on particle swarm optimization. Int. J. Mach. Learn. and Comput. **5**(4), 283–287 (2015). https://doi.org/10.7763/ijmlc.2015.v5.521
22. Mohamed, S.A.E.H., Elloumi, M., Thompson, J.D.: Motif discovery in protein sequences. Pattern Recogn.-Analy. and App. (2016)
23. Mueen, A.: Time series motif discovery: dimensions and applications. Data Mining Knowl. Disc. **4**(2), 152–159 (2014)
24. Oba, D., Iwana, B.K., Matsuo, S.: Dynamic data augmentation with gating networks for time series recognition. In: ICPR (2022)
25. Patel, P., Keogh, E., Lin, J., Lonardi, S.: Mining motifs in massive time series databases. In: ICDM, pp. 370–377 (2002)
26. Rakthanmanon, T., Keogh, E.: Fast shapelets: A scalable algorithm for discovering time series shapelets. In: SIAM ICDM (2013). https://doi.org/10.1137/1.9781611972832.74
27. Ratanamahatana, C.A., Keogh, E.: Three myths about dynamic time warping data mining. In: SIAM ICDM (2005). https://doi.org/10.1137/1.9781611972757.50
28. Reimers, N., Gurevych, I.: Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799 (2017)
29. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986). https://doi.org/10.1038/323533a0
30. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Min. Knowl. Disc. **29**(6), 1505–1530 (2014). https://doi.org/10.1007/s10618-014-0377-7
31. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Sig. Proc. **45**(11), 2673–2681 (1997). https://doi.org/10.1109/78.650093

32. Senin, P., Malinchik, S.: SAX-VSM: Interpretable time series classification using SAX and vector space model. In: IEEE ICDM (2013). https://doi.org/10.1109/icdm.2013.52
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
34. Song, D., Xia, N., Cheng, W., Chen, H., Tao, D.: Deep r -th root of rank supervised joint binary embedding for multivariate time series retrieval. In: KDD (2018). https://doi.org/10.1145/3219819.3220108
35. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE Trans. Knowl. Data Engin. **25**(6), 1425–1438 (2013). https://doi.org/10.1109/tkde.2012.88
36. Tanaka, Y., Iwamoto, K., Uehara, K.: Discovery of time-series motif from multi-dimensional data based on mdl principle. Mach. Learn. **58**(2), 269–300 (2005)
37. Wang, T., Liu, Z., Zhang, T., Hussain, S.F., Waqas, M., Li, Y.: Adaptive feature fusion for time series classification. Knowl.-Based Sys. **243**, 108459 (2022)
38. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: IJCNN, pp. 1578–1585 (2017). https://doi.org/10.1109/ijcnn.2017.7966039
39. Yeh, C.C.M., Kavantzas, N., Keogh, E.: Matrix profile vi: Meaningful multidimensional motif discovery. In: ICDM, pp. 565–574 (2017)
40. Yeh, C.C.M., et al.: Matrix profile i: All pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: ICDM (2016). https://doi.org/10.1109/icdm.2016.0179
41. Zambelli, F., Pesole, G., Pavesi, G.: Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. Brief. Bioinform. **14**(2), 225–237 (2013)
42. Zhu, Y., et al.: Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In: ICDM, pp. 739–748 (2016)
43. Zimmerman, Z., et al.: Matrix profile xiv: Scaling time series motif discovery with gpus to break a quintillion pairwise comparisons a day and beyond. In: SoCC (2019). https://doi.org/10.1145/3357223.3362721