IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 September 2023, accepted 14 October 2023, date of publication 23 October 2023, date of current version 31 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327015

RESEARCH ARTICLE

LBP4MTS: Local Binary Pattern-Based Unsupervised Representation Learning of Multivariate Time Series

CHENGYANG YE[1] AND QIANG MA[2], (Senior Member, IEEE)
[1]Graduate School of Informatics, Kyoto University, Yoshida Tachibana-cho, Sakyo-ku, Kyoto 606-8303, Japan
[2]Department of Information Science, Kyoto Institute of Technology, Matsugasaki Hashigami-cho, Sakyo-ku, Kyoto 606-8585, Japan

Corresponding author: Qiang Ma (qiang@ieee.org)

This work was supported in part by the JSPS KAKENHI under Grant 19H04116 and Grant 23H03404.

ABSTRACT Representation learning of multivariate time series is a crucial and complex task that offers valuable insights for numerous applications, including time series classification, trend analysis, and regression. Unsupervised learning approaches are often favored in practical scenarios due to the limited availability of labeled data. However, most existing studies focus more on the global information of time series and ignore the local information, especially the representation learning based on the self-attention mechanism. This affects representation performance and may lead to the failure of downstream tasks. This study proposed an unsupervised representation learning model for multivariate time series by comprehensively considering multivariate time series data's global and local information. Specifically, a specially designed local binary pattern (LBP) method for multivariate time series (multivariate LBP) is introduced to the self-attention mechanism to improve the representation performance of modeling in terms of local information. Additionally, we propose a novel unsupervised approach for learning multivariate time series representations. The experimental results demonstrate significant advantages of our model over other representation learning methods and can be well applied in various downstream tasks.

INDEX TERMS Unsupervised representation learning, local binary pattern, global and local features, multivariate time series.

I. INTRODUCTION

Multivariate time series analysis is widely used in science, finance, social media, and various other fields [1], [2]. In the era of information explosion, a large amount of multivariate time series data is generated daily. Compared to other sequence data, multivariate time series data are more ubiquitous and thus have huge application prospects. This brings new challenges to discovering knowledge from big time series data. For example, in the stock market, multivariate time series analysis of stocks requires experienced and competent analysts to analyze the market changes and behavioral logic implied behind the complicated market data [3].

Recent interdisciplinary research on deep learning has positively impacted the analysis of multivariate time series [5]. A few pre-training approaches from computer vision (CV) and natural language processing (NLP) research have been applied to time series data to enhance the connection between data [6], [7]. Transformer is a typical example. The first Transformer model was proposed for natural language translation [8]. Due to the potent capabilities of self-attention in global feature extraction, this disruptive research has since inspired developments in other fields. The Vision Transformer (ViT) [9] model, proposed for image classification, broke the domain barrier and encouraged us to apply the self-attention mechanism to multivariate time series. In particular, with the widespread adoption of transformer architecture across various domains, attention mechanisms-based time series representation has become a hot research topic.

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Augusto Ribeiro Chaves.

Like text data, multivariate time series data is another important sequence data type. On the one hand, both multivariate time series and text data are sequential, meaning the order of the data points matters. But on the other, there are some differences between text data and multivariate time series data. Compared to the input in most NLP sequential models, which comprise embedded text data vectors in a semantic space via pre-training, multivariate time series data are naturally formed. This prevents individual timestamped data points within the time series from carrying more implicit information, such as trends, patterns, or dependencies. Moreover, for text data, the correlation between words is mainly determined by semantic information and is independent of their distance within the text. However, the temporal distance is directly proportional to the multivariate time series data correlation. The further the time points apart, the less significant their mutual influence becomes. These differences place more complex demands on the original self-attention mechanism.

Compared to text data, multivariate time series data exhibits similarities with image data regarding global and local characteristics [12]. Local features can emphasize trend information, a significant attribute for downstream tasks of multivariate time series data. Although attention-based characterization methods have unique advantages in learning global features, more and more studies have demonstrated that local representation learning still needs to be improved [10]. Much research on applying self-attention mechanisms in CV has focused on enhancing local features [11], which also encourages research in the multivariate time series field. A representation learning approach incorporating local and global features, without adding extra computational burden, is beneficial for multivariate time series analysis.

In addition, due to the lack of labeled data, there is widespread interest in providing efficient analysis using large amounts of unlabeled multivariate time series data [4]. Data augmentation is required for multivariate time series to constitute the training sample pairs. However, standard data augmentation techniques for time series are often inspired by CV and NLP field practices and are usually unsuited for multivariate time series. These practices carry strong inductive biases, such as transformation-invariance and cropping-invariance. Some research has already proved this issue may lead to learned representations that do not accurately encapsulate the complete information inherent to the multivariate time series [13]. This presents a significant challenge in designing sample pairs necessary for unsupervised learning in multivariate time series data.

To address these issues, this study proposes a novel unsupervised learning model named LBP4MTS (**L**ocal **B**inary **P**attern for **M**ultivariate **T**ime **S**eries). Our model enables the representation learning of multivariate time series and considers both global and local features of multivariate time series. First, the proposed model introduces a specially designed local binary pattern (LBP) method, multivariate LBP, for multivariate time series in a self-attention mechanism to improve the representation performance of the model in terms of local information. Subsequently, a variant of Dropout for multivariate time series representation, named DropLine, is designed to generate comparison sample pairs for unsupervised representation learning. Compared to conventional data augmentation methods in unsupervised representation learning, our method constructs sample pairs by network architecture instead of modifying the multivariate time series input. In this way, it's not necessary to introduce inappropriate inductive biases and assumptions.

In summary, the main contributions of our work are summarized as follows:

- We propose LBP4MTS, a novel model that can learn the representation of multivariate time series with global and local features. This model introduces an LBP-based self-attention mechanism in the transformer encoder layer (Section III-C) to learn a more comprehensive representation of multivariate time series.
- We develop an unsupervised training method (Section III-D). A variant of Dropout is also designed to construct the unsupervised sample pairs of multivariate time series.
- We conduct extensive experiments on several datasets from different fields (Section IV). The proposed model achieves better results than other baseline methods and demonstrates its applicability to various tasks.

Portions of this work have been presented at the IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD) in 2022 under the title "TS2V: A Transformer-Based Siamese Network for Representation Learning of Univariate Time Series Data" [14]. In this work, we improved the extant model for univariate time series data to multivariate time series data by introducing an LBP-based self-attention mechanism in the transformer encoder. Meanwhile, TS2V is a supervised representation learning model. In this work, we proposed its unsupervised learning version for various application scenarios. The results of experiments conducted on multivariate time series data demonstrate the effectiveness of the proposed model.

The remainder of this paper is organized as follows. Section II outlines previous studies on representation learning for multivariate time series, various variants of the LBP algorithm, and modifications of the Dropout method from existing literature. Section III describes the architecture of the proposed model in detail. Finally, Section IV presents the experimental results, and the study conclusions are summarized in Section V.

II. RELATED WORK

A. REPRESENTATION LEARNING OF TIME SERIES DATA

Representation learning of time series data has become a popular research topic. Most models aim to discover the spatio-temporal dependencies in time series data. Time2Graph [15] begins from Shapelet [16], which can automatically mine time series subsequences with representative features and constructs graphs for representation learning by analyzing the direct relationship between different shapelets. Time2Graph provides an inferred and interpretable temporal model with desirable performance.

Additionally, contrastive learning has been introduced into this aspect of time series analysis [17]. Constructing positive and negative data pairs achieves unsupervised representation learning of unlabeled time series data. On this basis, triplet loss is further combined with a CNN with dilation [18] to tackle long time series data. This approach is fairly easy to implement and only requires distinguishing the main features.

Studies have also been devoted to applying data augmentation to raw data inputs [19]. These models tend to construct the input view of time series data with some designed embedding methods and learn the representation of these input views by contrastive learning. These models employ the novel idea that constructing suitable time series embedding vectors as input could increase the learning performance of the model in representation learning.

The time series transformer (TST) model [20] is a recently proposed representation learning model for multivariate time series. This model essentially fills the gap in applying the transformer model to the representation learning of time series. This model achieves better learning performance than supervised training methods by introducing a transformer-based pre-training mechanism. However, the TST model is based on the original self-attention mechanism. It has limitations in capturing local information, which can emphasize trend information. Moreover, TST applied generative pre-training tasks for unsupervised representation learning. It used the masking task in the same manner as the original transformer architecture. Consequently, in the unconstrained scenario, the model could potentially learn trivial solutions, such as constant mapping, which would offer minimal utility for downstream tasks [21].

B. LBP AND ITS VARIANTS

LBP is a simple yet efficient texture operator that labels an image's pixels by thresholding each pixel's neighborhood and considers the result a binary number. LBP is wildly used in the CV field, including medical image analysis and face recognition. Many extensions have been made to the original LBP method to enhance its performance.

To reduce computational complexity and improve texture classification performance, Uniform LBP [22] was proposed to calculate uniform patterns to account for a vast majority of all patterns in texture images. In addition, Rotation Invariant LBP [23] was designed to be invariant to the rotation of the image. Furthermore, Volume Local Binary Patterns (VLBP) [24] extended LBP into three dimensions, making it suitable for the analysis of dynamic textures in videos.

For the analysis of temporal signals such as voice, audio, and electroencephalography (EEG) signals, Chatlani and Soraghan introduced the 1D-LBP [25]. 1D-LBP is an extension of the LBP operator to one-dimensional data. It demonstrates the potentiality of applying LBP methods in time series. Like the original LBP, there can be various extensions of 1D-LBP to capture more complex patterns or provide robustness against certain signal variations. Based on 1D-LBP, TTLBP [26] extend 1D-LBP from univariate series to multivariate series data.

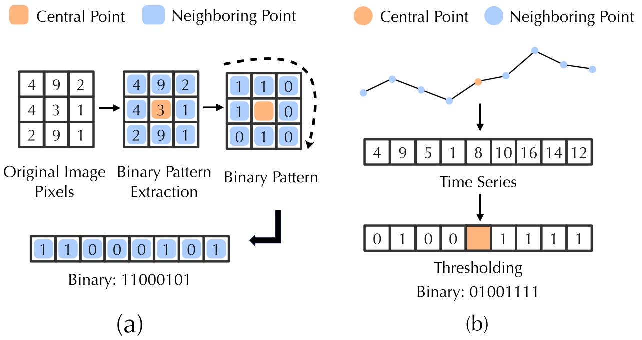


FIGURE 1. Illustration of structures of (a) LBP, (b) 1D-LBP, respectively.

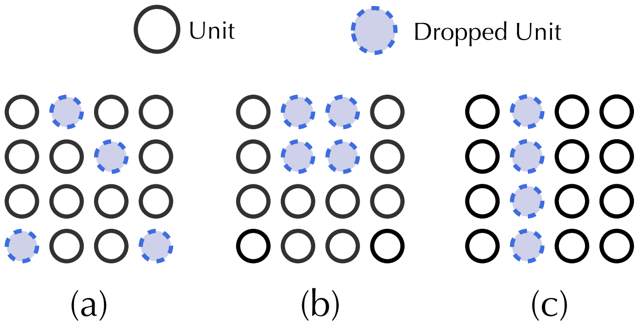


FIGURE 2. Schematic of structures of (a) Standard Dropout, (b) DropBlock, and (c) Spatial Dropout, respectively.

While these methods significantly streamline the feature extraction process for time series, they essentially remain manual feature extraction techniques, transforming the time series into histograms or distributions. Unfortunately, this transformation does not lend itself well to integration with deep learning models. Fig. 1 illustrates the original LBP method alongside the 1D-LBP variant.

C. DROPOUT AND ITS VARIANTS

Dropout is a regularization technique for reducing overfitting in neural networks. The technique temporarily drops out, or "deactivates," neurons in a layer with a certain probability during training. This forces the network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

DropBlock [27] is a form of structured dropout for the convolution layer. In standard dropout, neurons are dropped individually and randomly. In the convolution layer, other neurons in the same region may carry similar information due to spatial correlation. In DropBlock, a contiguous region of a feature map is dropped during training.

Spatial Dropout [28] performs dropout along specific dimensions only. During training, Spatial Dropout randomly selects a certain percentage of the channels in a convolutional

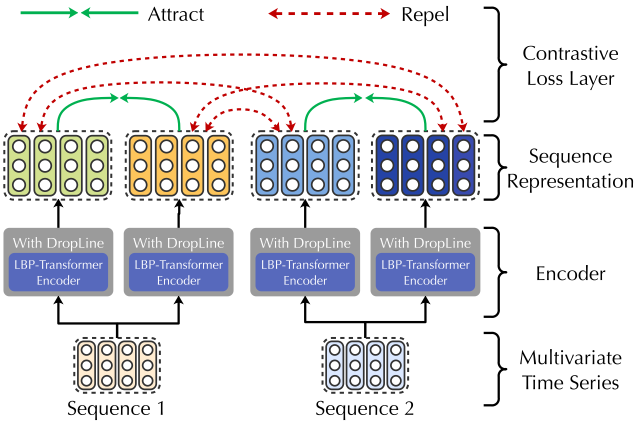


FIGURE 3. Structure of proposed unsupervised representation learning for multivariate time series.

layer and sets all values in these channels to zero for a given forward pass. This can often result in improved generalization and better performance on unseen data. Fig. 2 shows these four Dropout methods.

Some research [29] applied Dropout in contrastive unsupervised learning. These methods use random characteristics of Dropout to generate sample pairs by passing one input through the model with the Dropout layer twice. This inspired us to apply unsupervised contrastive learning for multivariate time series.

III. METHODOLOGY

A. OVERVIEW

This section describes the proposed model LBP4MTS and the relevant algorithms. The structure of LBP4MTS is shown in Fig. 3. First, each sequence of multivariate time series goes through the encoder part twice to generate positive pairs in contrastive learning. In traditional unsupervised contrastive learning, data augmentation is usually applied to generate sample pairs. However, most existing data augmentation methods may change the original data's distribution or multivariate time series pattern information. Model-based methods are then widely used for a variety of data and tasks. These methods construct sample pairs by stochasticity in specially designed models. This can avoid issues of changing certain information of original data.

Subsequently, an LBP-based self-attention mechanism is introduced to the encoder of transformer architecture as a representation learning model. It uses a specially designed LBP module, multivariate LBP, to extract local features of multivariate time series. Inspired by 1D-LBP and other LBP methods, such as TTLBP, The multivariate LBP module is designed for calculating the local feature relationship matrix of tensors.

Furthermore, a novel Dropout method, DropLine, is proposed. It can be regarded as a one-dimension version of DropBlock [27]. Like DropBlock, DropLine also obstructs the transfer of pertinent information from units adjacent to the dropped unit to the subsequent layer. Then, a contrastive loss is employed to train the representation of multivariate time series.

B. PROBLEM DEFINITION

Given a training sample $X \in \mathbb{R}^{n \times m}$, which is a multivariate time series of length n and dimensions m, the input sequence with n vector is $x_t \in \mathbb{R}^m : X \in \mathbb{R}^{n \times m} = [x_1, x_2, \ldots, x_n]$. The proposed unsupervised representation learning model aims to train a mapping function that transforms each input data point x_t into its corresponding representation r_t. Such a representation is designed to capture the input data's most informative and distinguishing features, allowing it to describe itself effectively.

Therefore, the representation of the training sample is denoted as $R = [r_1, r_2, \ldots, r_n]$, where each vector $r_t \in \mathbb{R}^k$ represents the learned representation of the input at a particular timestamp t. Here, k denotes the dimension of representation vectors. Essentially, the model transforms each input data point x_t into a representation vector r_t of size k, capturing the essential features and characteristics of the input. The resulting representation sequence R consists of these vectors corresponding to the individual timestamps.

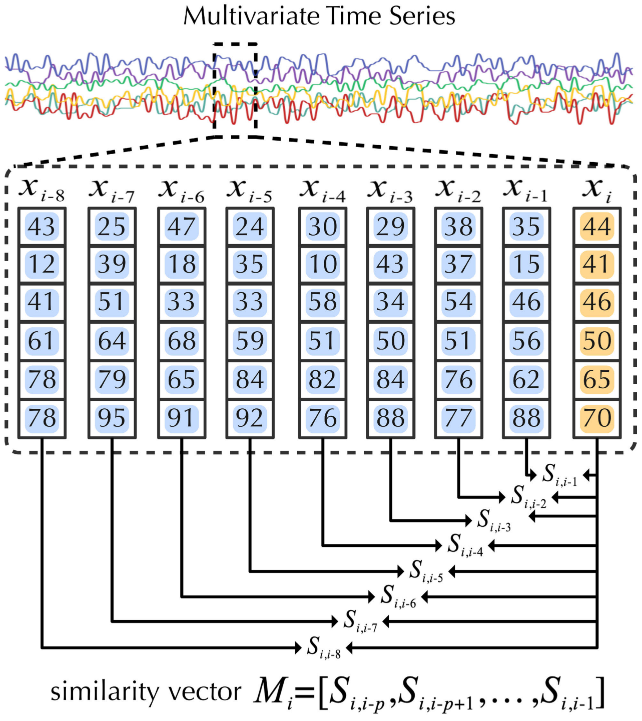
C. LBP-BASED TRANSFORMER ENCODER

As previously discussed, the original self-attention mechanism falls short of adequately representing the local characteristics inherent in multivariate time series data. Hence, numerous modifications have been suggested for the original self-attention model to improve its ability to portray local features found in sequential and multivariate time series data. The feature dependence of multivariate time series in local space is similar to that of image data, i.e., for any given encoded data point, its neighboring data points exert a more significant influence than data points located further away. Thus, convolutional Layers, a widely-used module to extract local information in CV, could be used to improve the performance of self-attention in extracting local information.

A straightforward method to encode local information is to use a convolutional layer before the self-attention mechanism. This allows the model to extract local features in the input sequence. However, both the convolutional layer and self-attention mechanism use learnable structures that are continuously updated during training. This continuous updating can lead to high computational costs, especially for deep networks with many layers and extensive training data. Several studies have opted for using non-learnable modules, like LBP, as substitutes for convolutional layers within a network [30]. These techniques can enhance computational efficiency and reduce susceptibility to overfitting. This motivates us to use LBP in the self-attention mechanism to improve the performance of extracting local features.

1) LBP FOR MULTIVARIATE TIME SERIES

Our multivariate LBP method is an operator for multivariate time series. Given a training sample of multivariate time



x_{i-8}	x_{i-7}	x_{i-6}	x_{i-5}	x_{i-4}	x_{i-3}	x_{i-2}	x_{i-1}	x_i
43	25	47	24	30	29	38	35	44
12	39	18	35	10	43	37	15	41
41	51	33	33	58	34	54	46	46
61	64	68	59	51	50	51	56	50
78	79	65	84	82	84	76	62	65
78	95	91	92	76	88	77	88	70

FIGURE 4. Illustration of calculation process of multivariate LBP.

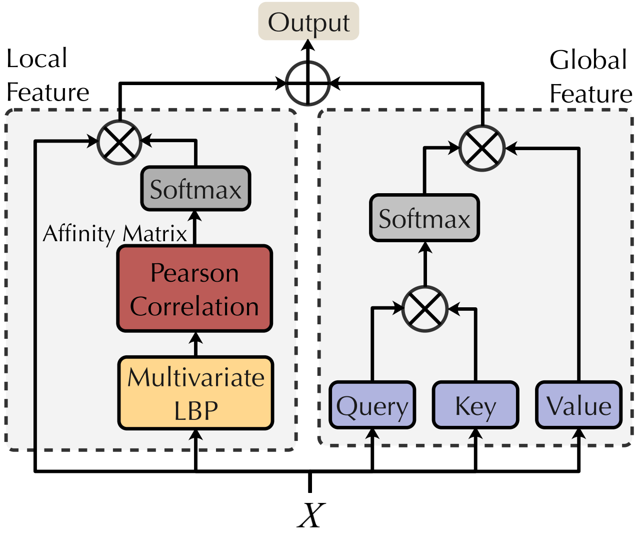


FIGURE 5. Schematic of LBP-based self-attention mechanism.

series $X = [x_1, x_2, \ldots, x_n]$, for each timestamp of multivariate time series, x_i, multivariate LBP defines the variant M_i as a combined similarity vector between x_i and p timestamp data points before x_i.

$$M_i = [S_{i,i-p}, S_{i,i-p+1}, \ldots, S_{i,i-1}] \tag{1}$$

where $S_{i,i-j}$ is the similarity value between x_i and the j^th data point before x_i. It can be expressed as follows:

$$S_{i,i-j} = s(x_i, x_{i-j}), j \in [1, p] \tag{2}$$

where $s(\cdot)$ denotes similarity calculation. The similarity determination in multivariate LBP cannot be made directly, as in LBP, by comparing the values of two scalars. There are numerous similarity measures for vectors that can be utilized. The selection of an appropriate similarity measure can be tailored according to the specific situation. A commonly employed measure is Cosine similarity. For any timestamp data point x_i and its neighboring data point x_{i-j}, the Cosine similarity is calculated as follows:

$$cos(x_i, x_{i-j}) = \frac{\langle x_i, x_{i-j}\rangle}{\sqrt{\langle x_i, x_i\rangle} \cdot \sqrt{\langle x_{i-j}, x_{i-j}\rangle}} \tag{3}$$

where $\langle\cdot\rangle$ represents the inner product.

Unlike the LBP and most variants, multivariate LBP is not symmetric and has no central point. This asymmetrical design ensures that the multivariate LBP value for each timestamp data point in multivariate time series is solely influenced by its neighboring historical data but not by any future data. Thus, the coefficient of each timestamp data point, x_i, is influenced by its immediate p neighboring data points, $[x_{i-p}, x_{i-p+1}, \ldots, x_{i-1}]$. Fig. 4 illustrates the calculation process of multivariate LBP. As for parameter p, i.e., the number of neighboring historical data, the experiment in TTLBP proved that eight neighboring historical data get the best performance for multivariate time series. Therefore, we also choose eight neighboring historical data in this paper to calculate multivariate LBP value. For the initial timestamp data point input of multivariate time series, i.e., x_i where $i < 9$, we populate their historical data using the constant composition to compute its multivariate LBP operation.

Furthermore, unlike other LBP-based methods for extracting local features from multivariate time series, our approach does not rely on histograms to represent the local information. Instead, we directly employ the computed similarity results to create a similarity vector. This vector is then utilized to calculate an affinity matrix (AM), like the weight matrix in the self-attention mechanism. The resulting affinity matrix captures the local features of the multivariate time series in the encoder layer and is combined with the attention mechanism to enhance the overall representation.

2) LBP-BASED SELF-ATTENTION MECHANISM

Based on the similarity vector calculated by the multivariate LBP method, we propose an LBP-based self-attention mechanism in the encoder of the transformer architecture to add local features to the representation learning of multivariate time series. The diagram of the proposed LBP-based self-attention mechanism is shown in Fig. 5, where X represents the entire sequence of multivariate time series. In this mechanism, the local feature is represented by similarity vector M_i calculated by the multivariate LBP method. An affinity matrix is then generated according to the similarity vector to reveal the degree of similarity between inputs. To enhance the robustness of the affinity matrix, we can utilize the Pearson correlation coefficient among similarity vectors of timestamp data points in the multivariate time series. The correlation coefficient serves as a centered version of cosine similarity since it involves subtracting the mean from the data points before computation.

The formula to calculate the Pearson correlation coefficient is as follows:

$$p_{i,j} = \frac{\langle M_i - \overline{M_i}, M_j - \overline{M_j} \rangle}{\sqrt{\langle M_i - \overline{M_i}, M_i - \overline{M_i} \rangle} \cdot \sqrt{\langle M_j - \overline{M_j}, M_j - \overline{M_j} \rangle}} \tag{4}$$

where $\overline{M_i}$ is the mean of similarity vector m_i. By calculating the Pearson correlation coefficient of among each timestamp data point in multivariate time series, the affinity matrix is formed as follows:

$$AM(X) = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix} \tag{5}$$

where AM is an abbreviation for affinity matrix.

In original self-attention, for any input X, the function of self-attention is expressed as follows:

$$Q = XW^Q; K = XW^K; V = XW^V \tag{6}$$

$$\text{Attention} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \tag{7}$$

where Q,K,V represent the matrices of queries, keys (with dimension d_k),and values (with dimension d_v), respectively. As shown in equation (3), queries, keys, and values are transformed through linear projections by $W^Q \in \mathbb{R}^{d_m \times d_k}$, $W^K \in \mathbb{R}^{d_m \times d_k}$ and $W^V \in \mathbb{R}^{d_m \times d_v}$, respectively, where d_m is the dimension of the input.

After adding a multivariate LBP module, the function of LBP-based self-attention can be described as follows:

$$\text{LBPAttention} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V + softmax\,(AM(X)) \cdot X \tag{8}$$

In Equation (8), the first component $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$ represents the global feature and the second component $softmax\,(AM(X)) \cdot X$ represents local feature.

D. UNSUPERVISED TRAINING

Unsupervised learning is particularly pertinent to multivariate time series data analysis, given the considerable effort and expense often associated with obtaining labeled data.

Most existing research on unsupervised learning for multivariate time series relies on data augmentation from fields of CV or NLP to generate sample pairs. These techniques might not always be suitable due to the unique characteristics of multivariate time series data, such as temporal dependency. The inductive biases transformation-invariance, like rotating an image in CV, or cropping-invariance, like cropping part of a sentence in NLP, might not hold true in the case of multivariate time series data.

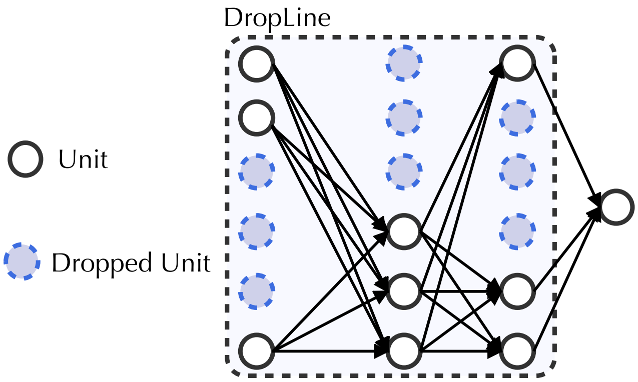


FIGURE 6. Schematic of dropLine method in standard neural network.

Besides utilizing data augmentation to create sample pairs during the preprocessing phase, a rising number of studies are now turning to apply model stochasticity, like the Dropout layer, in the training phase to generate positive training pairs [29]. This strategy aims to avoid the potential negative impacts that data augmentation could impose on the input data. Inspired by these, a variate of Dropout is proposed for multivariate time series to generate training sample pairs without traditional data augmentation methods. This method is named DropLine and can be added to most training models for multivariate time series. The diagram of DropLine is shown in Fig. 6.

Compared with standard Dropout, DropLine randomly discards continuous neuron units within the layer, i.e., a line of neuron units is dropped. This design is based on the understanding that for any given timestamp data point, neighboring neural nodes could potentially hold similar information because of temporal continuity. Essentially, it suggests that random deactivation of individual nodes does not necessarily lead to a total loss of relevant information.

After the DropLine operator, the sample pairs of unsupervised training are obtained. Then, the training object of contrastive learning is to learn an encoder such that:

$$score\left(f(x), f(x^+)\right) \gg score\left(f(x), f(x^-)\right) \tag{9}$$

where x^+ is the positive sample, and x^- is the negative sample. $score$ is often expressed as a distance function, which means that the training object can be formulated by computing the distance among the anchor, positive, and negative samples. It is articulated as follows:

$$max(d(x, x^+) - d(x, x^-) + margin, 0) \tag{10}$$

where $d(\cdot)$ is the distance between the sample pairs, and the margin is a hyperparameter to control the distances. The loss function can be defined as follows:

$$-\log \frac{e^{\cos(x_i, x_i^+)/\tau}}{\Sigma_{j=1}^{n}\left(e^{\cos(x_i, x_j^+)/\tau} + e^{\cos(x_i, x_j^-)/\tau}\right)} \tag{11}$$

where τ is a temperature hyperparameter and n is training batch size.

TABLE 1. Summary of UEA multivariate classification datasets.

Dataset	Train Size	Test Size	Length	Classes	Dimensions	Type
EthanolConcentration	261	263	1751	4	3	Other
FaceDetection	5890	3524	62	2	144	EEG
Handwriting	150	850	152	26	3	HAR
Heartbeat	204	205	405	2	61	AUDIO
JapaneseVowels	270	370	29	9	12	AUDIO
PEMS-SF	267	173	144	7	983	MISC
SelfRegulationSCP1	268	293	896	2	6	EEG
SelfRegulationSCP2	200	180	1152	2	7	EEG
SpokenArabicDigits	6599	2199	93	10	13	SPEECH
UWaveGestureLibrary	2238	2241	315	8	3	HAR

TABLE 2. Accuracy results of classification of the proposed and baseline methods.

Dataset	LBP4MTS	DTW_D	XGBoost	TST	TS2Vec
EthanolConcentration	**0.429**	0.305	0.417	0.258	0.288
FaceDetection	**0.661**	0.526	0.635	0.535	0.500
Handwriting	0.361	0.278	0.175	0.215	**0.479**
Heartbeat	0.725	0.727	0.732	**0.739**	0.694
JapaneseVowels	0.951	0.909	0.917	**0.980**	0.943
PEMS-SF	0.692	0.703	**0.967**	0.737	0.677
SelfRegulationSCP1	**0.845**	0.753	0.823	0.714	0.818
SelfRegulationSCP2	**0.597**	0.528	0.489	0.550	0.570
SpokenArabicDigits	**0.997**	0.959	0.712	0.931	0.973
UWaveGestureLibrary	0.910	0.907	0.772	0.900	**0.912**
Average Accuracy	**0.717**	0.660	0.664	0.656	0.686
Average Rank	**1.9**	3.5	3.3	3.2	3.1

IV. EXPERIMENTS

In this section, we assess the performance of our model by analyzing its performance across various tasks. We employ classification and regression tasks as downstream tasks to evaluate the value of local features in the representation learning of multivariate time series.

In the subsequent experiments outlined below, we employ the predefined training-test splits of the benchmark datasets and ensure all models are sufficiently trained to achieve convergence. An initial adjustment of the hyper-parameters (such as the number of training batch sizes, the number of encoder blocks, or the representation dimension) for each distinct dataset can contribute to enhanced performance. After the hyper-parameters were determined, the complete training set was leveraged for model training, which was ultimately assessed using the test set.

To more accurately assess the performance of our algorithm, we employed K-fold cross-validation (ten-fold cross-validation) on each dataset and repeated the experiment 5 times for each fold.

A. CLASSIFICATION

In this subsection, we report the experiments conducted to evaluate the effectiveness of our proposed model on the UEA dataset [31], using the classification task as a downstream task. The UEA dataset is significant for researching and analyzing multivariate time series time data. Benefited from its expansive collection of real-world multivariate time series data, the UEA dataset provides a consistent benchmark for researchers. Its ongoing updates and expansions not only ensure its enduring relevance in the ever-evolving research landscape but have also led to its increasing adoption in a multitude of time series studies worldwide. It currently has 128 univariate and 30 multivariate time-series classification datasets. We conducted repeat experiments on ten multivariate time series datasets to verify the performance, providing multiple datasets from different domains, with varying dimensions, unequal length dimensions, and missing values. The summary of these datasets is shown in Table 1.

In the classification task, the output vector of our model was passed through a SoftMax function to obtain a distribution over classes, and its cross-entropy with the categorical ground truth labels was considered as the sample loss. This experiment can directly verify the performance of the proposed representation learning model.

The UEA archives also provide an initial benchmark for the existing models, with accurate baseline information including classification accuracy. The benchmarks facilitate consistency in evaluations, ensuring that methodologies are compared under standardized conditions. Based on these information, we chose these four models as our baseline for multivariate time series classification: dimension-dependent dynamic time warping (DTW_D) [32], TST [20], XGBoost [33] and TS2Vec [13]. Adhering to the approach outlined by the TST model, we utilize the best-performing method, DTW_D that the authors of the UEA archive examined, as our benchmark for comparison. Meanwhile, as the first and the most famous model that introduces transformer architecture to representation learning of multivariate time series, TST is also considered as the baseline. Additionally,

TABLE 3. Details of multivariate regression datasets.

Dataset	Train Size	Test Size	Length	Dimensions	Missing Values
AppliancesEnergy	96	42	144	24	No
BenzeneConcentration	3433	5445	240	8	Yes
BeijingPM10Quality	12432	5100	24	9	Yes
BeijingPM25Quality	12432	5100	24	9	Yes
LiveFuelMoistureContent	3493	1510	365	7	No
IEEEPPG	1768	1328	1000	5	No

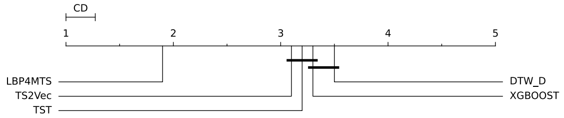


FIGURE 7. Critical Difference (CD) diagram of representation learning methods on time series classification tasks with a confidence level of 95%.

XGBoost is among the most frequently utilized models for both univariate and multivariate time series analysis, which can also be used as a baseline to evaluate the performance of our model. Finally, TS2Vec is currently the most advanced representation learning model for multivariate time series, which also be included for comparison. These methods are the best-performing methods studied by the creators of the archive. Among these four methods, TST and TS2Vec are neural network-based models, while DTW_D and XGBoost are traditional methods. Table 2 presents our model's and baseline models' classification results for the multivariate time series, where bold indicates the best values. The Critical Difference diagram for the Nemenyi test applied to these datasets is depicted in Fig. 7. Classifiers not linked by a bold line exhibit significant differences in their average ranks. This provides strong evidence that our algorithm notably surpasses other methods.

Table 2 reveals that our proposed model exhibited superior performance on five out of the ten datasets, achieving an average ranking of 1.9^{th}. This was followed by TS2Vec and TST, which outperformed the remaining two datasets and achieved average ranks of 3.1^{th} and 3.2^{th}, respectively. XGBoost performed best on the remaining 1 dataset, ranked 3.3^{th} on average. The table clearly indicates that methods based on neural networks generally yield superior results, aligning with the current understanding of the significant role neural networks play in the advancement of multivariate time series analysis. We note that all datasets where TS2Vec surpassed our model's performance were extremely low-dimensional, specifically 3-dimensional. Compared with other methods, TST achieves the best result of performance in multivariate time series with the type of AUDIO. In terms of XGBoost, it demonstrates robust performance on highly dimensional data, highlighting potential limitations of methods grounded in neural networks.

Interestingly, the data presented in the table also suggests a clear positive relationship between the efficacy of our model and the volume of available data, especially for large-scale training data. This indicates that as the quantity of data increases, the performance of our model also significantly improves. It further underscores the importance of large datasets in enhancing the model's predictive power and generalization capabilities, which is crucial in machine learning and data-driven decision-making. This correlation between data volume and model effectiveness could pave the way for future research and developments in optimizing data collection and utilization methods.

B. REGRESSION

In this subsection, the regression task is introduced as the downstream task to evaluate the effectiveness of our proposed model. multivariate Time series regression is a statistical method that is used to analyze multivariate time series data. multivariate Time series regression aims to create a mathematical model that can predict future responses based on the behavior observed in past data. This method can be used to forecast trends, cycles, or other patterns in the data that tend to repeat over time.

We chose various datasets from UEA&UCR Time Series Regression Archive [34]. Table 3 presents detailed characteristics of these datasets. As mentioned in experiments of TST, this selection was made to ensure a diverse representation concerning the dimensionality and length of multivariate time series samples and the number of samples.

In the regression task, we choose root mean square error (RMSE) to evaluate the performance of different models. RMSE is a commonly used metric in regression analysis and forecasting to measure the model's prediction error. The RMSE represents the sample standard deviation of the differences between predicted and observed values. Essentially, it tells you how concentrated the data is around the line of best fit. RMSE is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2} \tag{12}$$

where n is the number of observations, P_i is the predicted value for observation i and O_i is the observed value for observation i.

Meanwhile, inspired by the TST paper, we also incorporate the "average relative difference from the mean" evaluation criterion. This addition can help RMSE in mitigating the impact of different magnitudes across various datasets, thereby providing a more accurate measure of different models' performance across diverse datasets. The metric average relative difference from the mean (represented as r_j

TABLE 4. Performance of regression task for our and baseline models on multivariate regression datasets (RMSE).

Dataset	LBP4MTS	ROCKET	XGBoost	1-NN-ED	Random Forest	TST
AppliancesEnergy	2.355	**2.240**	3.494	5.273	3.415	2.359
BenzeneConcentration	**0.461**	3.160	0.662	6.296	0.815	0.506
BeijingPM10Quality	84.783	113.943	94.589	130.583	96.946	**82.996**
BeijingPM25Quality	55.789	60.874	60.352	84.806	65.905	**53.153**
LiveFuelMoistureContent	**43.795**	44.651	48.897	57.901	47.685	44.785
IEEEPPG	**23.909**	35.115	30.877	31.685	30.879	26.469
Ave Rel. diff. from mean	**-0.280**	0.095	-0.110	0.650	-0.084	-0.270
Average Rank	**1.5**	3.8	3.6	5.8	4.1	2

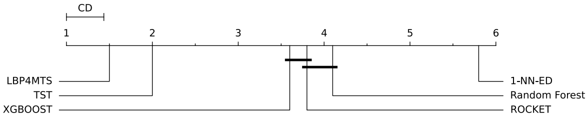


FIGURE 8. Critical Difference (CD) diagram of representation learning methods on time series regression tasks with a confidence level of 95%.

for each model j) can be defined as follows:

$$r_j = \frac{1}{N} \sum_{i=1}^{N} \frac{R(i,j) - \bar{R}_i}{\bar{R}_i} \tag{13}$$

$$\bar{R}_i = \frac{1}{M} \sum_{k=1}^{M} R(i,k) \tag{14}$$

where $R(i,j)$ is the RMSE of model j on dataset i, N is the number of datasets, and M is the number of models. Upon analyzing this particular metric, it is obvious that a smaller value of the average relative difference from the mean corresponds to superior model performance.

As same as the classification task, The UEA&UCR Time Series Regression Archive also provides an initial benchmark for the existing models, with accurate baseline information. Based on the performance metrics provided by the archives, we chose these five models as our baseline for multivariate time series classification: ROCKET [35], XGBoost [33], 1-NN-ED [36], Random Forest [37], and TST [20]. According to the results reported in the archive, these methods emerge as the top five-performing algorithms. Table 4 presents the RMSE of regression results of our model and baseline models for the multivariate time series, where bold indicates best values. The Critical Difference diagram illustrating the results from the Nemenyi test for various datasets can be seen in Fig. 8. If algorithms are not connected by a bold line, it indicates significant disparities. Such evidence compellingly underscores the superiority of our algorithm over the other methods.

As the results in Table 4 indicate, our model yields the best performance on three datasets, outperforming all other models. On the remaining three datasets, where our model didn't achieve optimal performance, it secured the second position. The second one is the TST model, which proves optimal on two datasets, while the ROCKET model, securing the third position, is optimal on one dataset. Thus, the overall ranking for our model stands at 1.5. The outcomes from both TST and our model underscore the efficacy of deep learning models in the representation learning of multivariate time series. Even though our model managed to achieve second rank on three datasets, the analysis of these datasets uncovers a limitation in our model's capability to utilize local features when dealing with multivariate time series data of shorter lengths (such as BeijingPM10Quality and BeijingPM25Quality datasets). Moreover, deep learning-based models tend not to perform well with smaller sample datasets (such as the Appliances dataset). Several factors might contribute to these limitations. For one, smaller datasets limit the diversity and variability within the data, constraining the model's learning process. Without a broad range of data to train on, the model might miss subtle patterns or nuances. When working with compact datasets, the model may not have sufficient information to train effectively, potentially leading to overfitting or reduced generalization capabilities. Furthermore, short data lengths may not offer a comprehensive view of the temporal dependencies and patterns inherent in longer sequences, which can be vital for accurate predictions in time series analysis. Meanwhile, by comparing the results of our model with those of the TST model, it can be seen that generative unsupervised learning could potentially outperform contrastive unsupervised learning when it comes to learning representations of shorter sequences. This insight outlines our prospective direction for enhancement.

C. ABLATION STUDY

To validate the efficacy of the proposed components in our model, i.e., the LBP-based self-attention and DropLine, we compare the full LBP4MTS model against its three variants across ten UEA datasets outlined in Table 1. To swiftly and effectively demonstrate the efficacy of each module within the proposed LBP4MTS model, a classification experiment is adopted for the ablation study. The results of the ablation study were evaluated based on the accuracy of the classification results and their percentage change.

Table 5 presents the results of this ablation study, where (1) **w/o LBP** removes the LBP-based self-attention module and employs original self-attention mechanism, (2) **w/o DropLine** removes DropLine designed in this paper and applies Dropout to unsupervised train the model, (3) **w/o LBP & DropLine** remove both LBP-based self-attention module and DropLine. The results demonstrate that every

TABLE 5. Ablation results for LBP4MTS and its variants.

	Average Accuracy	Accuracy Decline
LBP4MTS	**0.719**	-
w/o LBP	0.681	-3.8%
w/o DropLine	0.693	-2.6%
w/o LBP & DropLine	0.670	-4.9%

component within the LBP4MTS structure is essential and irreplaceable.

Meanwhile, the comparison of the results from LBP4MTS and its variate without LBP-based self-attention suggests that local features play a pivotal role in the representation learning of multivariate time series. This is attributed to the fact that the trends of its neighboring data heavily influence the timestamp data points within a multivariate time series. Capturing local features enables the model to more accurately depict the underlying pattern of change within the multivariate time series. In addition, by comparing the difference in results between LBP4MTS and its variate without DropLine, we can observe that the DropLine module is more adept at constructing sample pairs for unsupervised training of multivariate time series. This outcome is credited to DropLine's capacity to avoid the leakage of information from neighboring timestamp data points, resulting in a more effective unsupervised model training process.

V. CONCLUSION

Given the inherent nature of multivariate time series data, local features play a crucial role in the representation learning process. The identification of local patterns and trends can provide a wealth of insights that global analysis might miss. However, in the original self-attention mechanism, these local aspects were not effectively captured, potentially losing important information. In this study, our LBP-based transformer encoder is proposed as a mechanism to represent multivariate time series. This model aims to overcome the shortcomings of the original model in local feature extraction. In addition, a variate of Dropout, DropLine, is designed to construct the sample pairs of multivariate time series and to achieve unsupervised contrastive learning. DropLine is based on the understanding that neighboring neural nodes could potentially hold similar information because of temporal continuity. The conducted experiments reveal that the proposed model exhibits substantial improvement in the representation learning of multivariate time series. An ablation study proves the effectiveness of components within the LBP4MTS structure. Consequently, it can be employed in various downstream tasks, such as classification and regression.

In future research, our efforts will be devoted to improving the performance of our model in datasets with small data sizes and short data lengths. These include leveraging transfer learning from pre-trained models, integrating domain-specific or external data sources for added context, and exploring hybrid models to bolster the model's adaptability to short data sequences. By harnessing these approaches, we anticipate marked improvements in model efficacy across diverse data scenarios. Meanwhile, we find the design of the loss function to be a captivating aspect of unsupervised representation learning of multivariate time series. So far, a variety of loss functions have been engineered to cater to diverse applications. Thus, we believe that optimizing the loss function could further enhance the performance of our model.

REFERENCES

[1] O. C. Yolcu, E. Egrioglu, E. Bas, and U. Yolcu, "Multivariate intuitionistic fuzzy inference system for stock market prediction: The cases of Istanbul and Taiwan," *Appl. Soft Comput.*, vol. 116, Feb. 2022, Art. no. 108363.

[2] F. Mubang and L. O. Hall, "VAM: An end-to-end simulator for time series regression and temporal link prediction in social media networks," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 4, pp. 1479–1490, Aug. 2022.

[3] M. van Deurs, M. E. Brooks, M. Lindegren, O. Henriksen, and A. Rindorf, "Biomass limit reference points are sensitive to estimation method, time-series length and stock development," *Fish Fisheries*, vol. 22, no. 1, pp. 18–30, Jan. 2021.

[4] H. Gweon and H. Yu, "A nearest neighbor-based active learning method and its application to time series classification," *Pattern Recognit. Lett.*, vol. 146, pp. 230–236, Jun. 2021.

[5] A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. D. Delgado, and L. A. Akanbi, "Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting," *Mach. Learn. Appl.*, vol. 7, Mar. 2022, Art. no. 100204.

[6] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 3988–4003.

[7] Y. Yuan, L. Lin, Q. Liu, R. Hang, and Z.-G. Zhou, "SITS-former: A pre-trained spatio-spectral–temporal representation model for Sentinel-2 time series classification," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 106, Feb. 2022, Art. no. 102651.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[10] X. Pan, C. Ge, R. Lu, S. Song, G. Chen, Z. Huang, and G. Huang, "On the integration of self-attention and convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 805–815.

[11] I. Beltagy, M. E. Peters, and A. Cohan, "LongFormer: The long-document transformer," 2020, *arXiv:2004.05150*.

[12] H. Xu, J. Li, H. Yuan, Q. Liu, S. Fan, T. Li, and X. Sun, "Human activity recognition based on Gramian angular field and deep convolutional neural network," *IEEE Access*, vol. 8, pp. 199393–199405, 2020.

[13] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "TS2Vec: Towards universal representation of time series," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 8980–8987.

[14] C. Ye and Q. Ma, "TS2V: A transformer-based Siamese network for representation learning of univariate time-series data," in *Proc. IEEE 25th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2022, pp. 1245–1250.

[15] Z. Cheng, Y. Yang, W. Wang, W. Hu, Y. Zhuang, and G. Song, "Time2Graph: Revisiting time series modeling with dynamic shapelets," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3617–3624.

[16] C. Ji, C. Zhao, S. Liu, C. Yang, L. Pan, L. Wu, and X. Meng, "A fast shapelet selection algorithm for time series classification," *Comput. Netw.*, vol. 148, pp. 231–240, Jan. 2019.

[17] A. Jansen, M. Plakal, R. Pandya, D. P. W. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 126–130.

[18] J. Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 4650–4661.

[19] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A survey on data augmentation for text classification," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–39, Jul. 2023.

[20] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 2114–2124.

[21] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5753–5763.

[22] D. Q. Zeebaree, H. Haron, A. M. Abdulazeez, and D. A. Zebari, "Trainable model based on new uniform LBP feature to identify the risk of the breast cancer," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, Apr. 2019, pp. 106–111.

[23] S. Wang, M. Jiang, J. Qin, H. Yang, and Z. Gao, "A secure rotation invariant LBP feature computation in cloud environment," *Comput., Mater. Continua*, vol. 68, no. 3, pp. 2979–2993, 2021.

[24] G. Zhao and M. Pietikinen, "Dynamic texture recognition using volume local binary patterns," in *Proc. Int. Workshop Dyn. Vis.*, 2005, pp. 165–177.

[25] N. Chatlani and J. J. Soraghan, "Local binary patterns for 1-D signal processing," in *Proc. 18th Eur. Signal Process. Conf.*, Aug. 2010, pp. 95–99.

[26] X. Hu and G. Li, "Temporal tensor local binary pattern: A novel local tensor time series descriptor," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6393–6402, Oct. 2020.

[27] G. Ghiasi, T. Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 10727–10737.

[28] S. Lee and C. Lee, "Revisiting spatial dropout for regularizing convolutional neural networks," *Multimedia Tools Appl.*, vol. 79, nos. 45–46, pp. 34195–34207, Dec. 2020.

[29] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," 2021, *arXiv:2104.08821*.

[30] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4284–4293.

[31] M. Middlehurst, P. Schäfer, and A. Bagnall, "Bake off redux: A review and experimental evaluation of recent time series classification algorithms," 2023, *arXiv:2304.13029*.

[32] A. Bagnall, H. Anh Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," 2018, *arXiv:1811.00075*.

[33] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, and T. Zhou, "XGBoost: extreme gradient boosting," *R Package Version*, vol. 1, pp. 1–4, Jan. 2015.

[34] C. Tan, C. Bergmeir, F. Petitjean, and G. Webb, "Time series extrinsic regression: Predicting numeric values from time series data," *Data Mining Knowl. Discovery*, vol. 35, pp. 1–29, Mar. 2021.

[35] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining Knowl. Discovery*, vol. 34, no. 5, pp. 1454–1495, Sep. 2020.

[36] H. A. Dau, A. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR time series archive," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.

[37] H. Wu, Y. Cai, Y. Wu, R. Zhong, Q. Li, J. Zheng, D. Lin, and Y. Li, "Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression," *BioSci. Trends*, vol. 11, no. 3, pp. 292–296, 2017.

CHENGYANG YE received the B.S. degree in control science and engineering from the Chongqing University of Posts and Telecommunications, in 2015, and the M.S. degree in control science and engineering from Chongqing University, in 2018. He is currently pursuing the Ph.D. degree with Kyoto University, Kyoto, Japan.

QIANG MA (Senior Member, IEEE) received the Ph.D. degree from the Graduate School of Informatics, Kyoto University, in 2004. He was a Research Fellow of JSPS, from 2003 to 2004. He joined the National Institute of Information and Communication Technology, as an Expert Researcher, in 2004. From 2006 to 2007, he was an Assistant Manager with NEC. In October 2007, he joined Kyoto University, as an Assistant Professor, and has been an Associate Professor, since August 2010. In April 2023, he joined the Kyoto Institute of Technology, as a Full Professor. His research interests include multimedia information systems, data mining, social informatics, sightseeing informatics, investment informatics, and informational nutrition (information complementation).