Multivariate Time Series Clustering With Transformer

Jiazhou Liang
Department of Mechanical & Industrial Engineering
University of Toronto
Toronto, ON
joe.liang@mail.utoronto.ca

Abstract

Multivariate Time Series (MTS) Clustering, entailing the grouping of samples exhibiting similarity across multiple temporal variables, holds significant promise for various applications. However, prevailing clustering algorithms often encounter challenges such as noise within raw time series data, diminished feature correlation, and necessitated human preprocessing. In response, this paper introduces a novel framework for MTS clustering utilizing Transformer architecture, leveraging its multi-head attention mechanism to potentially capture intricate multivariate relationships. The proposed approach entails learning a cluster-oriented univariate representation of MTS using a Transformer before applying clustering algorithms. The efficacy of this novel framework is substantiated through a series of experiments conducted on real-world datasets.

1 Introduction

Time series clustering, an unsupervised machine learning technique aimed at grouping time series samples exhibiting similar patterns, is garnering increasing attention for its wide-ranging applications. Multivariate Time Series (MTS), characterized by multiple non-independently distributed variables within a single sample, pose challenges for effective clustering in high-dimensional Euclidean spaces due to the 'Curse of Dimensionality' as elucidated by Bellman (2). Moreover, the task necessitates consideration of both temporal transitivity and variable correlation for a stratified result (8), further complicating the clustering process.

Shape-based clustering involves directly clustering time series data based on various similarity metrics, such as employing K-means clustering with Euclidean Distance or Dynamic Time Warping (3). While these methods offer computational efficiency in univariate time series data, they encounter several limitations in MTS clustering. These include addressing time series of varying lengths and handling missing values, sensitivity to data noise (11), overlooking either temporal transitivity or variable correlation, and escalating computational costs with increasing data dimensionality.

Representation-based clustering (17) attempts to resolve these limitations by clustering on a representation of samples, such as extracted features. However, prevalent representation-based time series clustering tends to extract features through their structural characteristics, such as skewness and seasonality (17). Or the segment typologies (6). While effective for univariate time series data, these methods could still struggle to capture multivariate dependencies within a sample. Some multivariate frameworks, such as 'Time2Feat' by Bontifati et al.(5) using intra-signal and inter-signal time series with supervised ranking, which demonstrates a stratified clustering result in multivariate time series clustering but requires human preprocessing works in feature extraction and selection. Therefore, this approach might suffer from generalization depending on the availability of supervised features.

More recently, deep representation-based clustering has emerged, employing deep learning models to encode multivariate samples into their latent space. For instance, utilizing an autoEncoder model's

Preprint. Under review.

'bottleneck' feature efficiently reduces the dimensionality of multivariate time series samples (15). Alternatively, employing seq2seq models leverages the sequential nature of time series data, enabling the Encoder part of the model to capture temporal transitivity within samples (9). However, a potential drawback of these approaches is their longer model training time as sample sizes increase, which contrasts with the efficiency of classic unsupervised clustering methods.

Since the attention mechanism links every value within the time series using the query/key system, the Transformer model (16) can capture the associations between each point and other points in the existing sequential training samples. The 'multi-head' feature allows it to simultaneously capture different parts of associations in samples. Moreover, since the Transformer model can parallelly compute the attention for the entire sequence, it reduces the model training time for time series data with long lengths. Using the transformer to represent time series forecasting and classification has already shown its advantage in experiments (18). This result can be extended to being a candidate for an effective feature-learning tool for representation-based clustering.

This framework aims to address the limitations of current approaches in MTS clustering by employing the Transformer model. Given the pervasive nature of time series data across various domains, this novel approach, centered on uncovering hidden patterns, holds promise for diverse industries. With potential applications spanning numerous sectors, the framework is poised to make significant contributions. The paper will commence by elucidating the framework's architecture, incorporating various techniques to enhance representation quality, clustering outcomes, and computational efficiency. Subsequently, experiments will be conducted using six datasets sourced from the 'UAE Multivariate Time Series Classification Archive' (1) to assess the framework's performance. Furthermore, an evaluation of the effectiveness of different techniques will be undertaken.

2 Related work

This section will further discuss the state-of-the-art models and related works in each type of multivariate time series clustering.

2.1 Shape based clustering

Shape-based clustering draws inspiration from traditional clustering algorithms such as *k-means*, *k-medoids* (12), and *fuzzy c-mean* (4), adapting them to accommodate the unique characteristics of time series data. Unlike conventional methods that rely on Euclidean distance or Pearson's correlation coefficient, shape-based clustering algorithms manipulate distance metrics tailored to the temporal nature of the data. One such metric is *Dynamic Time Warping (DTW)* (3), which employs dynamic programming to determine the optimal alignment of two sequences. DTW facilitates similarity measurement for samples with varying numbers of timestamps, addressing a limitation of the original Euclidean distance metric. Additional distance metrics like *Short Time Series (STS)* (10) with *fuzzy c-mean* measure dissimilarity based on the sum of squared differences between the slopes of two time series. However, while effective for univariate time series clustering, these approaches often falter in multivariate settings. For tasks specific to MTS, there are metrics such as the disparity between the $V \times T$ spectral matrices (13) of samples, where V represents the dimension of variables and T denotes the total timestamps.

In contrast to manipulating distance metrics, alternative shape-based clustering methods, such as the *cross-sectional approach* proposed by Košmelj et al. (7) involve applying k-means clustering to generate cross-sectional typologies of samples. However, this method overlooks temporal variable correlations by treating each timestamp of each sample independently. Another approach by Liao et al. employs a two-step procedure for MTS, incorporating an additional clustering algorithm to convert raw multivariate samples into discrete, univariate time-stripped data.

2.2 Representation based clustering

Representation-based clustering encompasses two primary types: feature-based clustering and deep representation-based clustering. Feature-based clustering involves extracting potential features within samples that can distinguish each sample from the rest of the group. For instance, *Characteristic-Based Clustering* (17) extracts features based on structural characteristics such as mean, skewness, and seasonality, subsequently applying *k-means* using these extracted features instead of raw samples.

Additionally, *Principal Component Analysis (PCA)* serves as an effective tool for dimension reduction, applicable in MTS clustering to filter the dimensionality of raw spectra (14). Bontifati et al. employ techniques from audio processing to utilize intra- and inter-signal in time series data. This approach further reduces dimensionality by adding a semi-supervised feature selection process, ranking feature relevance through available labels, and utilizing *Analysis of Variance (ANOVA)* with *p-value* to quantify feature significance. Although this yields interpretable features with improved performance in experiments, it necessitates existing labels for feature selection, which are often absent in clustering scenarios.

State-of-the-art approaches in MTS Clustering involve employing deep learning models to learn sample representations from the latent space, often referred to as the embedding layer, of models. One approach combines attention and gate mechanisms in *Gated Recurrent Units (GRU)* with the 'bottleneck' features of an *Autoencoder* model to create a GRU-based Autoencoder for representation learning. Another approach that utilizes the attention is *Seq2Seq* model, with the raw sample in the Encoder part and the original sample reconstruction in the decoder part. The hidden state passed from the Encoder to the decoder network serves as the latent space representation for the sample. This approach also incorporates a novel k-means loss to guide the model towards representations beneficial for clustering. However, both models encounter computational efficiency issues in long-length time series data due to the inherent limitation of RNN and its variants, which can only process one timestamp per time step.

3 Methology

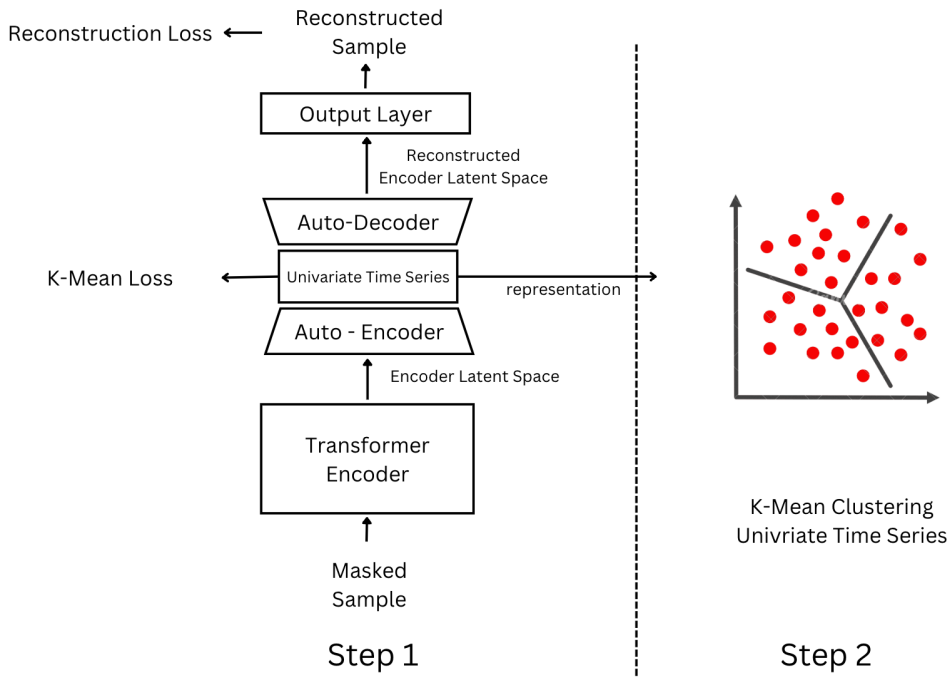

Figure 1: The overall architecture of the purposed clustering framework for given MTS samples.It involves to two steps: creating univariate time series representations of MTS samples through self-supervised learning and applying *k-mean* clustering using the learned representations of samples.

Figure 1 illustrates the overall architecture of the proposed framework for Multivariate Time Series (MTS) clustering. This approach comprises two steps: learning representation and clustering based on representation. Initially, a group of MTS samples, characterized by V variables and T timestamps, undergoes masking and serves as input to the Transformer Encoder. The primary objective of the

Transformer Encoder is to reconstruct the original samples before masking. Subsequently, the latent space of the trained model is utilized as the representation of input samples. Moreover, this latent space needs to be compressed from $D_{model} \times T$, where D_{model} represents the dimension of the Transformer model, into a $1 \times T$ representation (univariate time series). This representation is employed in the second step, where existing clustering algorithms, such as *k-means* in this study, are utilized to obtain clustering results for the given MTS samples. This section will go through several techniques used in this framework, adapted from previous studies, to achieve a satisfactory clustering result.

3.1 Self-supervised learning using masking

In most clustering tasks, samples lack true labels necessary for supervised model training. Previous studies employing AutoEncoder and RNN models utilize reconstruction loss, where the raw MTS samples serve as both input and expected output. This approach enables the model to learn a representation of the input in its latent space when successfully reconstructing the original samples. This approach is effective for AutoEncoder models as their 'bottleneck' challenges the model to reconstruct samples accurately. Similarly, RNN models process timestamps separately. However, traditional Transformer models lack a 'bottleneck', and all timestamps are fed into the model simultaneously, which lacks proper challenges to help the model learn a good representation during reconstructions since the model can directly use the input as output.

Zerveas et al. propose masking a proportion of timestamps in each dimension's time series to train a robust transformer model (18), and demonstrate favorable results through experiments. This paper adopts the same masking technique as Zerveas et al., employing the following steps. Initially, for each sample, the framework randomly selects a subset of timestamps of a specified size for each dimension's time series from a Gaussian distribution. This subset of timestamps is then masked before being used as input for the Transformer Encoder. The rationale behind generating a random subset for each dimension separately is to ensure that the masked timestamps are evenly distributed across all dimensions, thus preventing scenarios where one dimension contains an excessive or insufficient number of masked timestamps, which could hinder the Encoder's training. The masked sample is subsequently fed into the transformer Encoder, while the original sample serves as the expected output. This setup aids in training the Encoder to learn a representation in its latent space that effectively captures the features within the samples.

It is pertinent to note that since time series data already constitutes a numerical sequence, there is no need for an embedding layer to convert textual data into numerical form, as required in the original transformer model proposed by Vaswani et al. Consequently, this framework bypasses the embedding layer, directly feeding the raw MTS samples into the positional encoding layer. To ensure that the dimensionality of the raw MTS is sufficient for effectively utilizing the "multi-head attention" mechanism, the raw MTS is linearly projected into a higher-dimensional space of size $D_{model} \times T$, which serves as the input embedding space for the Transformer Encoder. Additionally, the model incorporates an extra learnable linear layer after the Encoder to transform the Encoder's output back to the original dimension of the samples. Sizes of masked samples and D_{model} are both hyperparameters to tune based on specific datasets.

3.2 Autoencoder model for latent space compression

The output space of the Transformer Encoder shares the same dimension as the input embedding space, denoted as $D_{model} \times T$, which represents a high-dimensional spectrum. However, directly utilizing this high-dimensional representation for *k-mean* clustering in the second step of the framework presents challenges analogous to clustering raw multivariate time series samples, thus conflicting with the overarching objective of this innovative framework. Consequently, it is imperative to compress the encoder output into a single-dimensional array with a length of T to serve as a more practical representation of the sample. Although directly aggregating each dimension of the output into a single dimension using mean or median can be computationally efficient, the resulting representation derived from mean aggregation fails to yield satisfactory clustering results in experiments. One plausible explanation is that the mean of the encoder output may not serve as a robust representation of the sample, as the reconstruction process only incorporates the original uncompressed latent space, thereby compromising the evaluation of representation quality.

To address this limitation, this paper introduces an Autoencoder model between the Encoder's output and the final output layer. The Encoder component of the autoencoder network reduces the dimensionality of the output latent space into a single-dimensional array, which subsequently serves as the final representation of the sample. Subsequently, the decoder reconstructs the compressed representation back to the original latent space spectrum with the same dimension as $D_{model} \times T$, passing it to the output layer for sample reconstruction. In this approach, the compressed representation becomes an integral part of the reconstruction process and undergoes evaluation and updating through the loss function. Furthermore, the inclusion of a 'bottleneck' in the entire model facilitates the training of a more effective encoder model for representation learning.

3.3 K-mean loss

While previous techniques may aid in learning a robust representation of MTS, the representation obtained solely from the reconstruction loss may not be optimally suited for clustering tasks. Therefore, this paper introduces an additional loss function to guide the model in learning a representation specifically tailored for clustering purposes. However, the original objective of the k-means clustering algorithm is non-differentiable, precluding its direct use as the model's loss function for updating via Stochastic Gradient Descent (SGD). Ma et al. (9) propose a novel loss function for time series k-means clustering utilizing the spectral relaxation of k-means clustering introduced by Zha et al. (19). This process can be outlined as follows.

Let $H \in \mathbb{R}^{T \times N_{batch}}$ denote a matrix, where N_{batch} represents the batch size of the model training process in SGD, and $F \in \mathbb{R}^{N_{batch} \times k}$ denote another matrix, where k denotes the total number of clusters in k-means clustering. The sum of squares between each sample in H and its cluster centroid can be expressed as follows:

$$Tr(H^T H) - Tr(F^T H^T H F) \tag{1}$$

Here, H comprises all univariate time series samples (i.e., the compressed latent space representation) in each batch, and F can be interpreted as an indicator matrix indicating the cluster to which each time series sample in H belongs. Notably, spectral relaxation is effective only when all samples have equal length, a condition ensured in this scenario as raw MTS samples with different lengths are padded to the same length before undergoing transformation by the Transformer Encoder. The objective of k-means clustering is to minimize Equation 1, which can be reformulated as:

$$\max(Tr(F^T H^T H F)) \tag{2}$$

Subject to the constraint:

$$s.t.\ \ F^T F = I \tag{3}$$

Zha et al. also provide a closed-form solution for F using the Ky Fan Theorem, which consists of the top-k singular vectors of H. Hence, this approach furnishes both a closed-form solution for updating the cluster assignment in k-means clustering and a differentiable sum-of-squares cost, suitable for use as the loss function.

In this framework, the loss function $\mathcal{L}_{k-means}$ for the univariate time series representations is defined as the sum-of-squares cost in Equation 1:

$$\mathcal{L}_{k-means} = Tr(H^T H) - Tr(F^T H^T H F) \tag{4}$$

The objective of weight updating using this loss function is to minimize the sum of the squared distances between samples and their cluster centroids. The assigned clusters for each sample F will be periodically updated through Singular Value Decomposition (SVD) of H to further minimize the sum of square distances. Therefore, the total loss function for the model will be the summation of reconstruction loss, the Mean Square Error between all masked input and predicted output, and the k-mean loss. And an additional regularization term, such as l_2 regularization, to prevent overfitting the model.

$$\mathcal{L}_{model} = \mathcal{L}_{reconstruction} + \mathcal{L}_{k-mean} + \textit{regularization} \tag{5}$$

Some other techniques, such as using a learnable position encoding layer instead of the fixed encoding in original Transformer models, are also applied to improve the clustering result. 1 shows the complete 2-step procedure of the proposed novel framework in MTS clustering with Transformer.

Algorithm 1 Complete 2-step procedure for MTS clustering with Transformer Encoder

Input: $S \in \mathbb{R}^{N \times V \times T}$: samples, K : number of clusters, $M(s.t. M \leq T)$: masked samples, D_{model} : dimension of input embedding space
Output: C: Cluster label of S
Mask M timestamps for each variable in S
Linearly project input Samples into $\mathbb{R}^{N \times D_{model} \times T}$
Initialize F randomly
for each epoch **do**
    for each batch with samples $S_{batch} \in \mathbb{R}^{N_{batch} \times V \times T}$ **do**
        $R \leftarrow$ TransformerEncoder(S_{batch}) ▷ Original Representation
        $R_{compressed} \leftarrow$ AutoEncoder(R)
        $\mathcal{L}_{k-mean} \leftarrow$ Equation 1 $(R_{compressed}, F)$
        $R_{reconstructed} \leftarrow$ AutoDecoder($R_{compressed}$)
        $S_{reconstructed} \leftarrow$ Linear(R$_{reconstructed}$) ▷ $[N_{batch}, D_{model}, T]$ to $[N_{batch}, V, T]$
        $\mathcal{L}_{reconstruction} \leftarrow$ MSE(S_{batch}, $S_{reconstructed}$)
        $\mathcal{L}_{total} \leftarrow \mathcal{L}_{k-mean}$ + $\mathcal{L}_{reconstruction}$
        Update model weights using $\mathcal{L}_{total}$ with SGD
    end for
    for every J number of epochs: $F \leftarrow$ SVD(compressed_representation)
end for
$C \leftarrow$ KMeans(compressed_representation)
Return C

4 Experiment

This paper evaluates the proposed framework using six real-world datasets sourced from the UAE Multivariate Time Series Classification Archive (1). The selection of datasets originally designed for classification tasks is deliberate; such datasets come with pre-defined labels and a specified number of classes. These attributes facilitate a fair comparison between models via external metrics. While clustering tasks do not inherently provide a "true" label, the labels associated with classification tasks serve as the most appropriate proxy for the intrinsic nature of the samples.

The six datasets were carefully chosen to represent a diverse array of challenges one may encounter with MTS samples, including variations in dimensionality, series length, and sample size. This diversity ensures a comprehensive evaluation of the framework's performance across different scenarios. A summary of the statistics of the six datasets is presented in Table 1. Among them, the BasicMotions and Epilepsy datasets exemplify typical scenarios in MTS clustering, characterized by balanced dimensions, sample sizes, and series lengths. In contrast, the PenDigits dataset, with its substantial sample size yet minimal variable count and short timestamps, represents a less complex clustering challenge, potentially simplifying model training and evaluation.

Table 1: Summary statistics of datasets

Name	Sample Size	Dimension	Length	Classes
BasicMotions	60	6	100	4
Epilepsy	137	3	206	4
PenDigits	7494	2	8	10
EigenWorms	128	6	17984	5
NATOPS	180	24	52	6
DuckDuckGeese	60	1345	270	5

The EigenWorms dataset stands out due to its significantly longer time series for each variable, necessitating a larger input embedding space for the Transformer Encoder. Such datasets have historically posed challenges for RNNs and their variants, often resulting in protracted processing times. The final two datasets, NATOPS and DuckDuckGeese, are illustrative of MTS samples with high dimensionality. DuckDuckGeese, in particular, with over 1300 variables, highlights the complexities involved in capturing correlations between a large number of variables. Traditional

shape-based clustering algorithms, which rely on direct comparisons of raw data, may struggle with this level of complexity.

Given the availability of true labels within the selected datasets, this study employs the Rand Index to assess the similarity between the results of the clustering algorithm and the actual labels. The Rand Index quantitatively measures the concordance between two sets of clusterings by evaluating all possible sample pairs within the dataset. It calculates the proportion of sample pairs that are either assigned to the same cluster or to different clusters in both the predicted clustering and the true labeling. Mathematically, the Rand Index is defined as follows:

$$Rand\ Index = \frac{number\ of\ agreeing\ pairs}{total\ number\ of\ pairs} \tag{6}$$

This metric is particularly suitable for the validation of clustering results, as it provides a normalized score that directly reflects the accuracy of the clustering relative to the ground truth.

This paper selects a baseline model that applies the k-means clustering algorithm directly to the raw MTS samples. The k-means algorithm is chosen for its efficiency and effectiveness in clustering tasks. Utilizing the raw MTS samples allows for a straightforward comparison, highlighting the enhancements brought by the proposed framework over conventional clustering approaches. The comparison aims to demonstrate the superiority of the proposed method in handling the complexities of MTS data, particularly in capturing the correlation between variables that may not be readily discernible through traditional k-means clustering.

This paper utilizes a pre-defined train and test dataset split within the original dataset. The proposed framework is trained separately on each dataset using the designated training split. The model demonstrates the lowest validation loss in each dataset, derived from a 20% validation set partitioned from the training data, which will be selected. The selected model is then applied to learn the representation of samples in the test dataset, subsequently enabling clustering based on the learned representations, and evaluating cluster results using the Rand Index.

Regarding hyperparameters, the dimensionality of all datasets is set at $D_{model} = 64$. An exception is made for the 'DuckDuckGeese' dataset, which consists of 1345 variables and therefore does not require projection into a higher dimensional space. The size of the mask is set to 15% of the time series length for each variable, adapting to the various lengths present across datasets by using a percentage rather than a fixed size. The training process uses 400 epochs for all datasets. The epoch with the lowest validation loss is selected as the final model for each dataset. This adaptive approach ensures that each dataset is optimally trained to capture the essential temporal features without unnecessary computational expense.

Table 2: Rand Index and clustering time (seconds) of purposed framework and baseline k-mean

Name	Transformer	k-mean	Transformer Time	k-mean Time
BasicMotions	**0.5270**	0.4192	0.0129	0.0152
Epilepsy	**0.6351**	0.6123	0.0536	0.0806
PenDigits	0.8184	**0.9140**	0.7338	3.7740
EigenWorms	**0.6860**	0.4912	0.0483	0.9896
NATOPS	0.7577	**0.8002**	0.0313	0.0660
DuckDuckGeese	**0.5494**	0.2832	0.0161	1.3116

Table 2 presents the Rand Index comparisons across six different datasets between the Transformer-based method and the traditional *k-means* MTS clustering. The Transformer approach demonstrates superior performance in achieving higher Rand Index values in two general cases—'BasicMotions' and 'Epilepsy'—as well as in datasets with longer sequences such as 'EigenWorms'. A notably significant improvement is observed in the 'DuckDuckGeese' dataset, characterized by its high dimensionality, thereby underscoring the efficacy of this novel framework in clustering high-dimensional MTS samples relative to conventional shape-based clustering techniques using raw data.

In contrast, for less complex datasets with lower dimensions, *k-means* MTS clustering maintains a higher Rand Index than the Transformer approach. It is important to note that these experiments were conducted using consistent hyperparameters across all datasets, selected based on achieving

the lowest validation loss. Optimizing hyperparameters specifically tailored to each dataset might further enhance the clustering performance of the Transformer approach, indicating a potential area for future research to explore.

Table 2 additionally provides insights into the clustering time of the two approaches compared. It is important to note that this timing excludes the model training time for the Transformer approach, which is significantly longer than the clustering time itself. Therefore, the reported times should be considered primarily as a reference for understanding the speed improvements achieved when utilizing a univariate representation for clustering, as opposed to using raw MTS samples. This distinction is particularly relevant for datasets such as 'PenDigits' and 'DuckDuckGeese', where the simplification to a univariate representation significantly expedites the clustering process.

The Rand Index for the Transformer approach presented in Table 2 is derived from the best results obtained using various techniques discussed earlier in the paper. The paper explores different configurations of the Transformer model to assess the effectiveness of different techniques on clustering performance shown in Table 3. In the 'Without Autoencoder' column, a simplified approach is utilized where the compressed univariate time series representation is achieved through mean aggregation. In this result, using this substitution reduces the Rand Index significantly in all dataset. The 'Without K-means Loss' column presents results from a model trained solely on reconstruction loss. There is a reduction in the Rand Index in most of the dataset, which validates the effectiveness of including $\mathcal{L}_{k-mean}$.

Another point to note is that the less complex dataset, 'PenDigits', exhibited poorer performance when applying all proposed techniques. This underperformance could likely be attributed to overfitting within the model. Given the simplicity of the dataset, the Transformer-based framework might have led to the model learning not only the underlying patterns but also the noise within the data, thus negatively impacting its generalizability and effectiveness in clustering tasks.

Table 3: Rand Index of six datasets using Transformer approach without different techniques

Name	All	Without Masking	Without Autoencoder	Without K-mean Loss
BasicMotions	0.5217	**0.5270**	0.3153	0.3474
Epilepsy	0.5517	0.4943	0.5172	**0.6351**
PenDigits	0.6555	**0.8184**	0.7955	0.7496
EigenWorms	**0.6860**	0.6289	0.5631	0.6307
NATOPS	**0.7577**	0.7470	0.6006	0.7368
DuckDuckGeese	**0.5494**	0.3992	0.4008	0.4106

5 Limitation

The novel framework proposed in this paper exhibits superior performance in clustering multivariate time series (MTS) data compared to the baseline shape-based MTS clustering using *k-means* with raw samples. However, this approach is not without its limitations. Firstly, the self-supervised training required by our framework demands significantly more time to learn a robust representation of the model compared to the more straightforward shape-based clustering method. This is compounded by the fact that the framework trains on different datasets separately, increasing the overall time required for clustering across multiple datasets.

A potential solution to mitigate these training time issues could involve the use of transfer learning. By pre-training a model on a large corpus of MTS data, it may be possible to reduce the training time required when clustering individual datasets or even eliminate the need for retraining the model to obtain satisfied representations for new datasets.

Another limitation of the current approach is its lack of interpretable representations, unlike other approaches such as Time2Feat, which provide interpretable feature spaces. Interpretable representations of MTS can be particularly beneficial for data summarization, as they serve as a lower-dimensional projection of raw samples that can capture and elucidate the relationships between variables. Enhancing the interpretability of the model's output could greatly increase the utility of the framework

in practical applications where understanding the underlying patterns and structures in the data is crucial.

6 Conlusion

This paper proposed a novel framework in Multivariate Time Series Clustering utilizing self-supervised learning with Transformer Encoder. The experiment in this paper demonstrates this novel approach shows a superficial result compared to the state-of-the-art shape-based clustering approach in some real-world datasets. Given the ubiquitous nature of time series data in various domains, our novel approach, which focuses on uncovering hidden patterns, holds promise for diverse industries. This framework has the potential to find applications across numerous sectors.

References

[1] A. BAGNALL, H. A. DAU, J. LINES, M. FLYNN, J. LARGE, A. BOSTROM, P. SOUTHAM, AND E. KEOGH, *The UEA multivariate time series classification archive, 2018*, arXiv, (2018).

[2] R. BELLMAN, R. CORPORATION, AND K. M. R. COLLECTION, *Dynamic Programming*, Rand Corporation research study, Princeton University Press, 1957.

[3] D. J. BERNDT AND J. CLIFFORD, *Using dynamic time warping to find patterns in time series*, in Proceedings of the 3rd international conference on knowledge discovery and data mining, 1994, pp. 359–370.

[4] J. C. BEZDEK, R. EHRLICH, AND W. FULL, *FCM: The fuzzy c-means clustering algorithm*, Comput. Geosci., 10 (1984), pp. 191–203.

[5] A. BONIFATI, F. D. BUONO, F. GUERRA, AND D. TIANO, *Time2Feat: learning interpretable representations for multivariate time series clustering*, Proc. VLDB Endow., 16 (2022), pp. 193–201.

[6] D. GUIJO-RUBIO, A. M. DURÁN-ROSAL, P. A. GUTIÉRREZ, A. TRONCOSO, AND C. HERVÁS-MARTÍNEZ, *Time series clustering based on the characterisation of segment typologies*, arXiv, (2018).

[7] K. KOŠMELJ AND V. BATAGELJ, *Cross-sectional approach for clustering time varying data*, Journal of Classification, 7 (1990), pp. 99–109.

[8] H. LI AND Z. LIU, *Multivariate time series clustering based on complex network*, Pattern Recognit., 115 (2021), p. 107919.

[9] Q. MA, J. ZHENG, S. LI, AND G. W. COTTRELL, *Learning Representations for Time Series Clustering*, Advances in Neural Information Processing Systems, 32 (2019).

[10] C. S. MÖLLER-LEVET, F. KLAWONN, K.-H. CHO, AND O. WOLKENHAUER, *Fuzzy clustering of short time-series and unevenly distributed sampling points*, in International symposium on intelligent data analysis, Springer, 2003, pp. 330–340.

[11] J. PAPARRIZOS AND L. GRAVANO, *Fast and accurate time-series clustering*, ACM Transactions on Database Systems (TODS), 42 (2017), pp. 1–49.

[12] H.-S. PARK AND C.-H. JUN, *A simple and fast algorithm for k-medoids clustering*, Expert systems with applications, 36 (2009), pp. 3336–3341.

[13] T. SERIES, *Discrimination and clustering for multivariate*, Journal of the American Statistical Association, 93 (1998), p. 328.

[14] C. SHAW AND G. KING, *Using cluster analysis to classify time series*, Physica D: Nonlinear Phenomena, 58 (1992), pp. 288–298.

[15] N. TAVAKOLI, S. SIAMI-NAMINI, M. ADL KHANGHAH, F. MIRZA SOLTANI, AND A. SIAMI NAMIN, *An autoencoder-based deep learning approach for clustering time series data*, SN Applied Sciences, 2 (2020), pp. 1–25.

[16] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is All you Need*, Advances in Neural Information Processing Systems, 30 (2017).

[17] X. WANG, K. SMITH, AND R. HYNDMAN, *Characteristic-Based Clustering for Time Series Data*, Data Min. Knowl. Disc., 13 (2006), pp. 335–364.

[18] G. ZERVEAS, S. JAYARAMAN, D. PATEL, A. BHAMIDIPATY, AND C. EICKHOFF, *A Transformer-based Framework for Multivariate Time Series Representation Learning*, arXiv, (2020).

[19] H. ZHA, X. HE, C. DING, M. GU, AND H. SIMON, *Spectral relaxation for k-means clustering*, Advances in neural information processing systems, 14 (2001).