ITC 1/54
Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 16-31
DOI 10.5755/j01.itc.54.1.38771

Multi-Dimensional Temporal Feature Fusion and Density Perception for Time Series Clustering

Received 2024/09/10 | Accepted after revision 2024/12/16

HOW TO CITE: Gao, J., Guo, Y., Li, C., Wang, H., Zhao, X., Li, T., Li, X. (2025). Multi-Dimensional Temporal Feature Fusion and Density Perception for Time Series Clustering. *Information Technology and Control*, 54(1), 16-31. https://doi.org/10.5755/j01.itc.54.1.38771

Multi-Dimensional Temporal Feature Fusion and Density Perception for Time Series Clustering

Jie Gao
Software College, Shandong University, Jinan, Shandong, China PR;
Shandong Institute of Metrology, Jinan, Shandong, China PR

Yunzhen Guo
Software College, Shandong University, Jinan, Shandong, China PR

Congwei Li
Shandong Institute of Metrology, Jinan, Shandong, China PR

Haocong Wang
Software College, Shandong University, Jinan, Shandong, China PR

Xinxiao Zhao
Weifang University, Weifang, Shandong, China PR

Teng Li
Informatization Office, Shandong University, Jinan, Shandong, China PR

Xueqing Li
Software College, Shandong University, Jinan, Shandong, China PR

Corresponding authors: Xinxiao Zhao (e-mail: xinxiao.zhao1985@gmail.com) and
Teng Li (e-mail: liteng@sdu.edu.cn)

In the field of data mining and knowledge discovery, clustering algorithms have emerged as a powerful tool for unsupervised learning. However, when they are applied to time series data, some distinctive challenges emerge. The representation of time series data, which is often vast and high-dimensional. They need to be effectively reduced in dimension before the clustering process while ensuring that important information is retained. Furthermore, existing clustering methods encounter difficulties when dealing with variable density distributions. In response to these challenges, we present the Density-based Clustering Model for Time Series (DCMD). This module seamlessly integrates a temporal representation module named Multi-dimensional Representation Fusion (MDR) module, and a clustering module named K-Nearest Neighbor Weighted (NNW) clustering module. The MDR module is used to simplify those time series retains critical features by reducing data dimensions, ensuring and accuracy. This module adopts the PLR and PPA method to represent the original time series respectively and concatenates two representation results to one fusion representation for this time series. After that, the NNW module regards time series as data points and evaluates the local data point (K-Nearest data points) densities of them. Then, for each time series, we consider its local data point density and the distance to its K-nearest neighboring data points and use this to calculate its local density value. Finally, the cluster centers can be chosen according to this value. Rigorous benchmark evaluations validate the superiority of our approach.

KEYWORDS: Data mining, Time series clustering, Feature representation, Local density estimation.

1. Introduction

With the relentless growth of data, extracting hidden insights from massive datasets is paramount [38, 47, 28]. In comparison to alternative data mining algorithms [7, 21, 22], clustering methods are distinguished by their unsupervised nature and capacity to directly group analogous features into clusters, independent of the availability of category labels, and thus have attracted continuous attention and in-depth research from the scientific community [45, 30, 31]. As for Time series clustering, which is a complex and pivotal research area in the field of time series data mining [34]. To efficiently group time series into distinct clusters by identifying their similarities, we leverage the foundational principles of time series correlation representation and employ similarity measurement as the core mechanism for unsupervised learning. Clustering algorithms broadly fall into categories such as partition-based, grid-based, hierarchical, model-based, and density-based methods [43, 46, 51].

Despite significant progress in time series clustering research, several enduring challenges persist. Firstly, there's the need to enhance clustering efficiency [11, 20]. Time series data is often vast and high-dimensional, making traditional clustering techniques slow and cumbersome. Overcoming this challenge demands proficient data representation that can effectively capture essential aspects of the original time series, reducing data dimensions without compromising efficiency. Achieving a balance between data representation and efficient clustering poses a complex challenge [29]. Secondly, improving accuracy is crucial. Existing methods primarily rely on partitioning and density-based approaches. Partitioning methods are sensitive to noisy data, potentially leading to unsatisfactory results [50]. Density-based techniques excel in finding clusters with various shapes but struggle when dealing with variable or uneven density distributions [61]. This leads to difficulties in accurately identifying cluster centers, resulting in suboptimal clustering precision. Addressing these challenges is crucial to fully realize the potential of time series clustering in diverse applications.

In terms of enhancing clustering efficiency, researchers have dedicated their efforts to refining time series representation techniques [19, 57, 14]. Various approximate representation algorithms have emerged from these efforts, encompassing the Discrete Wavelet Transform (DWT) [48], Singular Value Decomposition (SVD) [32], Piecewise Linear Representation (PLR) [35], and Piecewise Aggregate Approximation (PAA) [41]. Among these, PLR and PAA have gained recognition as widely adopted methods in the field of time series data mining. PLR involves segmenting a given time series into contiguous subsegments and then using a sequence of linear functions to approximate each of these discrete subsegments. Notably, PLR distinguishes itself with several advantageous traits, including a lower index dimension, resulting in

a concise representation [13]. On the other hand, PAA works by dividing the time series into segmented sequences of equal length and utilizing the mean value of each segment to represent the original time series [12]. This process effectively reduces the dimensionality of the data while retaining essential information.

In terms of improving clustering accuracy, researchers have introduced a wide array of clustering methods from various perspectives. Among these methodologies, the K-means algorithm holds a distinguished position as one of the most renowned partition-based clustering techniques [40]. It operates by necessitating a priori specification of the cluster count and then iteratively forming clusters based on distinct objective functions. In stark contrast, the field of density clustering introduces a versatile paradigm capable of uncovering clusters within datasets of arbitrary shapes. Notably, this paradigm dispenses with the need for predefined cluster counts, showcasing remarkable resilience in the presence of noisy data. In the domain of density-based clustering, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm stands out prominently [52].

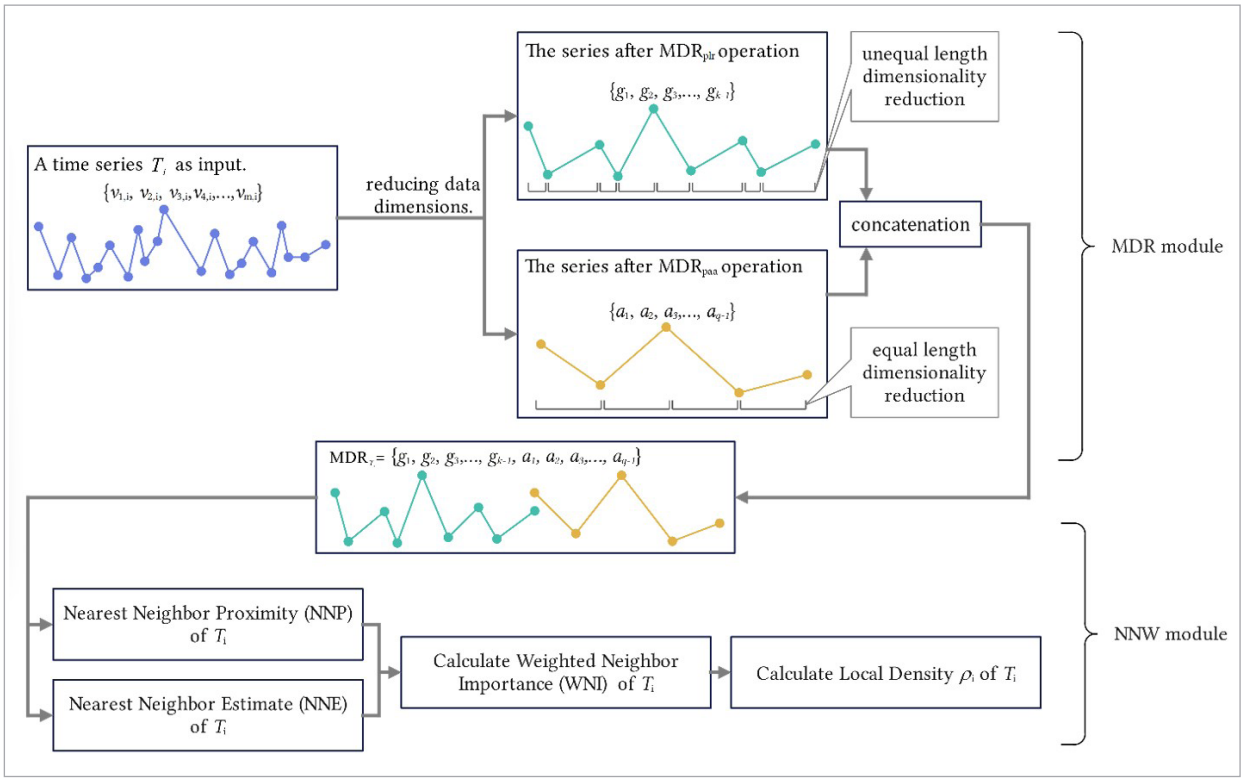
In 2014, a significant breakthrough in clustering emerged through the pioneering work of Rodriguez and Laio [42], resulting in the ingenious Fast Search and Density Peaks Clustering algorithm (DPC). What sets this algorithm apart is its operation with a single parameter, simplifying its application. DPC is built upon two fundamental assumptions: firstly, it assumes that the density of a cluster's core exceeds that of its surrounding points. Secondly, it relies on the observation that the distance between the cluster center and the point with the highest density is significantly large. The methodology of DPC involves calculating both local density (ρ) and distance (ζ) using a predefined cutoff distance. These calculations lead to the creation of a two-dimensional decision graph, which helps identify clustering centers and guides the allocation of remaining data points into their respective clusters. While DPC offers the advantages of simplicity and speed, it does come with certain limitations. Its clustering effectiveness primarily depends on the parameters of local density (ρ) and distance (ζ). Additionally, DPC faces challenges when handling datasets with complex structures, particularly those characterized by uneven density distributions and intricate manifold configurations.

This study introduces a novel time series clustering approach based on the fusion of multiple temporal representations. This methodology involves the extraction of key features from diverse perspectives within the time series data. As shown in Figure 1, by amalgamating these multiple representations, we create a comprehensive time series representation, striving to preserve vital information while reducing data dimensions for enhanced clustering efficiency. Furthermore, we have improved the regional density calculation method as a significant contribution. This improvement is particularly beneficial when dealing with time series that exhibit variable density distributions or uneven density patterns. It enables more precise identification of cluster centers, resulting in higher precision clustering outcomes.

This work makes several significant contributions, which are summarized as follows:

1. We introduce the Temporal Density Clustering Model Based on Multi-Dimensional Feature Fusion (DCMD). This model integrates a temporal representation module and a temporal clustering module, aligning well with the requirements of clustering tasks in terms of both efficiency and accuracy.
2. We propose a novel method for time series representation called the Time Series Multi-Dimensional Representation Fusion Method (MDR). By leveraging elements from PLR and PAA techniques and incorporating key change points within time series data, MDR effectively retains essential information from the original time series while reducing data dimensions, thereby enhancing clustering efficiency.
3. We design the K-Nearest Neighbor Weighted Clustering Method (NNW), a unique density clustering algorithm. NNW assigns distinct weights to each K-nearest neighbor and incorporates these weights into regional density calculations. This innovation allows for a more efficient reflection of regional distribution characteristics within time series data, considering both local and global data distribution, ultimately leading to improved clustering performance.
4. To validate the effectiveness of our proposed method, we conduct a comprehensive evaluation across a wide range of benchmark time series datasets. The experimental results clearly illustrate the superior performance and effectiveness of our approach in the field of time series clustering applications.

Figure 1
Schematic representation of the processing of time series T_i through the model



2. Related Work

Clustering algorithms are fundamental components of unsupervised data mining, offering valuable insights across diverse applications [26, 37, 76]. These algorithms, grouping similar data points together, have proven to be indispensable in numerous domains. They can be broadly categorized into several types, with their strengths and limitations. One category of clustering algorithms is partition-based methods, such as K-Means and K-Medoids. These methods are relatively intuitive and easy to implement. However, they are sensitive to noisy data and are particularly effective at detecting clusters with round or spherical shapes [46]. One significant advantage of hierarchical clustering is that it does not require the pre-definition of the number of clusters. Instead, it reveals the hierarchical relationships between clusters, providing a more comprehensive view of the data structure. However, these methods often come with higher computational complexity [51]. Density-based methods form a third category. Like hierarchical methods, they don't necessitate specifying the number of clusters in advance. Moreover, density-based algorithms can identify clusters with arbitrary shapes, making them versatile in capturing complex data patterns. However, their performance can be sensitive to the threshold parameters used [9]. Beyond these primary categories, various specialized clustering approaches exist, addressing specific data characteristics and analytical needs.

Within the domain of density-based clustering, significant strides have been made in enhancing traditional algorithms and introducing novel approaches. These developments have greatly expanded the capabilities of density-based clustering for various applications. One noteworthy category of density-based clustering methods is based on Density-Peak-based Clustering (DPC) algorithms. These algorithms were introduced to efficiently identify cluster centers by locating density peaks in the data distribution. An exemplar of this class is the "Clustering by

fast search and find of density peaks" (CFSFDP) algorithm proposed by Rodriguez et al. [42]. CFSFDP has gained widespread recognition for its ability to effectively detect clusters of arbitrary shapes within large-scale datasets. Importantly, CFSFDP achieves this with low computational complexity and high efficiency, making it particularly well-suited for the processing of big data. Furthermore, in the context of dealing with large-scale noisy datasets, robust clustering algorithms have been explored. These algorithms leverage the concept of density peaks to detect cluster centers and assign data points to clusters. Notably, they incorporate a fuzzy weighted K-nearest neighbors (KNN) [17] strategy to enhance robustness in the presence of noise. This approach improves the clustering performance when confronted with noisy and complex datasets, making it a valuable addition to the toolkit of density-based clustering techniques. Overall, the advancements in density-based clustering algorithms, especially those centered around density peaks, have significantly expanded the applicability of these methods, allowing them to effectively address clustering challenges in large-scale, noisy, and complex datasets. In the realm of density-based clustering, Zheng et al. [75] proposed an approximate nearest neighbor search method for multiple distance functions with a single index. An adaptive method was presented in [44] for clustering, where heat-diffusion is used to estimate density and cutoff distance is simplified. In [42], an adaptive density-based clustering algorithm was introduced in spatial databases with noise, which uses a novel adaptive strategy for neighbor selection based on spatial object distribution to improve clustering accuracy. Aiming at clustering ensemble, an automatic clustering approach was introduced via outward statistical testing on density metrics in [55]. Yu et al. [63] proposed an adaptive ensemble framework for semi-supervised clustering solutions. Yu et al. [64] introduced an incremental semi-supervised clustering ensemble approach for high-dimensional data clustering.

Compared with existing clustering algorithms, the K-Nearest Neighbor Weighted Clustering Method proposed in this paper can more accurately identify clusters with variable density distribution or uneven density distribution, based on shrinking the data dimensions. This ensures high efficiency and accuracy in clustering.

2. Proposed Method

In addressing the limitations of the DPC algorithm, which is significantly influenced by variables and less effective in clustering regions with variable or uneven density distributions, we propose a time series clustering method based on time series multi-representation fusion (DCMD). This method involves extracting main time series features from multiple perspectives, leading to the realization of a fused time series representation. Our approach aims to retain key feature information in time series while reducing data dimensionality, thus ensuring efficient clustering. Additionally, we have enhanced the regional density calculation method to more accurately identify the cluster center and achieve higher precision clustering, particularly for time series with variable density distribution or uneven density distribution. Table 1 summarizes some key notations used throughout the rest of this paper.

Table 1
Notations and their definitions

Notation	Definition
P	A time series set containing n time series elements.
T_i	The ith time series in P.
$v_{j,i}$	The jth element of T_i.
KP_p	The pth key change point in a KP set.
CAV_p	The importance weight of KP_p.
T_{i-plr}	The PLR series of T_i.
g_x	The xth element of T_{i-plr}.
T_{i-paa}	The PAA series of T_i.
a_x	The xth element of T_{i-paa}.
MDR_{plr}	The PLR series for all elements in P.
MDR_{paa}	The PAA series for all elements in P.
P_d	The MDR series set for P, which contains n element $d_1, d_2, ..., d_n$.
d_i	The MDR series of T_i, which is concatenated by T_{i-plr} and T_{i-paa}.
NNP_i	The nearest neighbor proximity (NNP) of d_i.
Δ_i	The nearest neighbor estimate (NNE) of d_i.
$\Theta_{k,i}$	The importance of d_i's kth nearest neighbor for d_i.
$\Gamma_{k,i}$	The weighted neighbor importance (WNI) of d_i's kth nearest neighbor for d_i.
ρ_i	The local density of d_i.

2.1. Temporal Representation Module

Definition 1. For a set of time series $P = \{T_1, T_2, ..., T_n\}$, each element T_i contains m ordered real values denoted as $T_i = \{v_{1,i}, ... v_{j,i}, ... v_{m,i}\}$, where $1 \le i \le n, 1 \le j \le m$. $v_{j,i}$ represents the jth time series point in T_i.

Definition 2. Key Change Point. For time series T_i, a key change point (KP) in time series T_i is a data point that satisfies one of the conditions outlined in Equation (1).

$$KP = \begin{cases} v_{j-1,i} < v_{j,i} > v_{j+1,i} \\ v_{j-1,i} < v_{j,i} = v_{j+1,i} \\ v_{j-1,i} = v_{j,i} > v_{j+1,i} \\ v_{j-1,i} > v_{j,i} < v_{j+1,i} \\ v_{j-1,i} > v_{j,i} = v_{j+1,i} \\ v_{j-1,i} = v_{j,i} < v_{j+1,i} \end{cases} \quad (1)$$

According to Definition 2 and the conditions specified in Equation (1), there are six different types of KPs in the time series T_i, including all local extremes and inflection points. During the flow of the sequence into the temporal characterization module, all KPs in T_i will be fully identified. However, these KPs contribute differently to the main temporal characterization of T_i. It is necessary to evaluate the importance of these identified KPs and rank them in descending order of their criticality, the more critical KPs ranked higher, the more significant contribution to the identification of T_i. The criticality of a KP can be quantified using the Criticality Assessment Value (CAV), as defined in Definition 3.

Definition 3. Criticality Assessment Value (CAV). The criticality of the KPs was assessed by calculating the vertical distance between the KPs and the mean of the sequence $\overline{T}_i$. The importance weight of the pth ($1 \le p \le m$) trend turning point KP_p is denoted as CAV_p. The calculation formula is shown in Equation (2).

$$CAV_p = \left|\overline{T}_i - KP_p\right|. \quad (2)$$

The KPs undergo a process of criticality assessment, where each point is assigned a CAV. They are subsequently arranged in descending order based on these values, granting higher priority to points with higher criticality assessment value. Once we establish the desired number of time series segments, the segmentation points for each T_i are chosen from the KPs according to their prioritization.

Definition 4. According to Definitions 2 and 3, all the KPs have been identified completely and sorted in the descending order of their CAVs, to form an ordered KP set named S_{kp}. Depending on the needs of the problem, a different number of points from the S_{kp} will be selected as the KP for the MDR_{plr}.

Definition 5. Assuming that k points are selected from the S_{KP}, the time series T_I will be divided into k-1 segments, we sequentially link the points, for each connecting line segment, we compute its respective gradient. The bth segment is calculated as shown in formula (5), denoted as g_b. The result after the time series T_i is represented by the MDR_{plr} is $T_{i-plr} = \{g_1, ... g_b, ... g_{k-1}\}$, where $1 \le b \le k - 1$..

For the time series D, the characterization after MDR_{plr} is shown in Equation (3).

$$MDR_{plr} = \begin{cases} T_{1-plr} = \{g_1, ... g_b, ... g_{k-1}\} \\ T_{2-plr} = \{g_1, ... g_b, ... g_{k-1}\} \\ ... \\ T_{n-plr} = \{g_1, ... g_b, ... g_{k-1}\} \end{cases} \quad (3)$$

Definition 6. According to the above definition, the representation part of MDR_{plr} is completed. Similarly, if we average the time series T_i into q–1 segments, the MDR_{paa} result for the cth segment is denoted as a_c, representing the average of all data in the cth segment. Then the average value of each segment is used as a representation in MDR, and the representation result is $T_{i-paa} = \{a_1, ... a_c, ..., a_{q-1}\}$, where $1 \le c \le q - 1$. For the time series P, the characterization after MDR_{paa} is represented as shown in Equation (4).

$$MDR_{paa} = \begin{cases} T_{1-paa} = \{a_1, ... a_c, ... a_{q-1}\} \\ T_{2-paa} = \{a_1, ... a_c, ... a_{q-1}\} \\ ... \\ T_{n-paa} = \{a_1, ... a_c, ... a_{q-1}\} \end{cases} \quad (4)$$

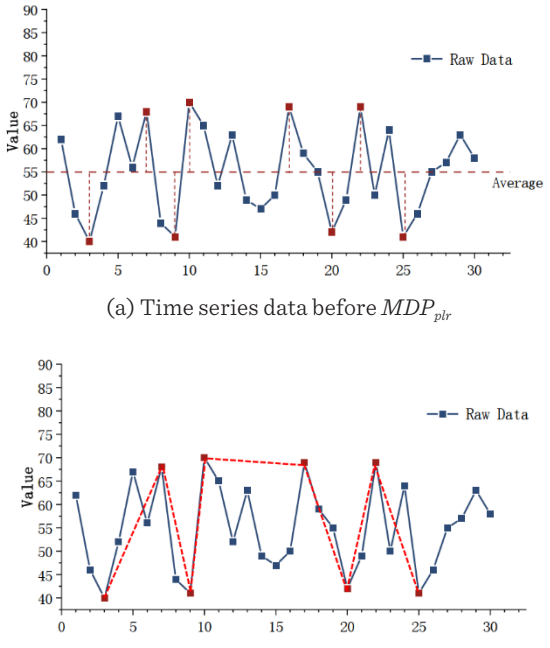
The final time series obtained after MDR is shown in Equation (5).

$$MDR = \begin{cases} T_1 = \{g_1, ... g_b, ... g_{k-1}, a_1, ... a_c, ... a_{q-1}\} \\ T_2 = \{g_1, ... g_b, ... g_{k-1}, a_1, ... a_c, ... a_{q-1}\} \\ ... \\ T_n = \{g_1, ... g_b, ... g_{k-1}, a_1, ... a_c, ... a_{q-1}\} \end{cases} \quad (5)$$

The before and after changes after MDR_{plr} processing are shown in Figure 2. Figure 2(a) represents the original data, and the characterization results are shown

as the red dashed line in Figure 2(b). In this figure, 8 KPs are taken to characterize the data.

Figure 2
Schematic diagram of MDP_{plr} characterization method



(a) Time series data before MDP_{plr}

(b) Time series data after MDP_{plr}

2.2. Temporal Clustering Module

The results obtained from MDR are concatenated to generate the n-dimensional latent representation d_i of the time series T_i. Subsequently, the data are clustered using the potential representation $P_d = \{d_1, d_2, \ldots d_n\}$ of all the data..

Definition 7. Nearest Neighbor Proximity (NNP). Within a given dataset denoted as D_d, the Nearest Neighbor Proximity (NNP) of any data point d_i represents the mean distance between d_i and its K-Nearest neighbors. The calculation is shown in Equation (6). We denote the NNP of d_i as NNP_i. The NNP serves as a valuable indicator in data analysis, offering insights into the inherent structure and clustering tendencies within a dataset. Specifically, it reflects the degree of proximity or closeness between data points by quantifying the distances between each data point and its nearest neighbors. In essence, NNP illuminates the local density variations within the dataset, shedding light on regions where data points are densely clustered as well as areas where they are more dispersed. This proximity metric aids in identifying potential clusters, outliers, and the overall distribution of data points within the feature space.

$$NNP_i = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\sum_{x=1}^{m}(v_{x,e} - v_{x,g})^2}, \quad (6)$$

where K represents the number of nearest neighbors, $v_{x,e}$ represents the eth data in the time series d_x, and m denotes the dimensional size of d_i.

Definition 8. Nearest Neighbor Estimate (NNE). In the context of data point d_i within the dataset P_d, the Nearest Neighbor Estimate (NNE) is represented as Δ_i. It stands in inverse proportion to the NNP. The calculation is shown in Equation (7).

The NNE serves as a reflection of the local data density, revealing the extent to which data points are densely packed or sparse in localized areas of the feature space. By calculating the density of data points within a defined neighborhood or region around each data point, NNE effectively identifies regions of high data density, helping to uncover clusters or groups of similar data points. Conversely, it highlights regions with lower density, indicating potential outliers or areas of reduced data concentration. Based on NNE, we can deduce the Weighted Neighbor Importance (WNI).

$$\Delta_i = \frac{1}{NNP_i}. \quad (7)$$

Definition 9. Weighted Neighbor Importance (WNI). The importance of each K-Nearest neighbor $N_{k,i}$ relative to data point d_i is determined by multiplying its NNE by the reciprocal of its distance from d_i , which is denoted as $\Theta_{k,i}$. After obtaining $\Theta_{k,i}$, the Weighted Neighbor Importance (WNI) of $N_{k,i}$ is calculated as the product of the NNE $\Delta_{k,i}$, denoted as $\Gamma_{k,i}$.

$$\Gamma_{k,i} = \Delta_{k,i} \times \Theta_{k,i} = \Delta_{k,i} \times \frac{1}{\frac{1}{\Delta_{k,i}} \times d_{k,i}}. \quad (8)$$

Definition 10. Local Density. The Local Density of d_i is determined by combining the NNE of d_i with the average importance weights $\overline{\Gamma_{k,i}}$ of its K-Nearest neighbors, denoted as ρ_i.

$$\overline{\Gamma_{k,i}} = \frac{\sum_{k\in K}\Delta_{k,i} \times \frac{1}{\frac{1}{\Delta_{k,i}} \times d_{k,i}}}{K} \quad (9)$$

$$\rho_i = \Delta_i \times \overline{\Gamma_{k,i}} = \Delta_i \times \frac{\sum_{k \in K} \Delta_{k,i} \times \frac{1}{\frac{1}{\Delta_{k,i}} \times d_{k,i}}}{K}. \quad (10)$$

The Local Density values of all time series are computed and then sorted in descending order. Subsequently, based on the desired number of clusters C, the top C points from the Local Density ranking are selected as the final cluster centers.

The Local Density serves as a valuable metric that encapsulates the density characteristics of data points within a given region. It reflects the concentration of data points around a particular point of interest, providing insights into the local data distribution. In essence, Local Density helps identify the prominence of specific data points within their local neighborhoods, highlighting their significance in the context of clustering and pattern recognition tasks.

3. Experiments

The model proposed in this paper designs the Temporal Representation Module and the Temporal Clustering Module mainly for the problems of efficiency and accuracy that exist in time series clustering, and therefore experiments are carried out for the clustering accuracy and clustering time. We first summarize the necessary experimental environment settings in Table 2.

Table 2
Details of the experimental environment settings

Name	Configuration
CPU	24 vCPU AMD EPYC 7642 48-Core Processor
GPU	Nvidia GeForce RTX 2080Ti
Memory	32GB DDR4 RAM
Hard Disk	1TB NVMe SSD
Operating System	Ubuntu 20.04 LTS
Python Framework	PyTorch 1.11.0

Datasets. To evaluate the performance of our DTC method, we utilized the benchmark UCR Time series Classification Archive datasets [5]. Table 3 presents a brief summarization of the characteristics of these datasets. Since we are using these datasets as unlabeled data, we combine the training and test datasets for all UCR datasets for all experiments in this study.

Baseline Method. We compare the results of our DCMD model with the following clustering methods:

1. K-means [4]: This model first randomly generates the initial clustering centers and divides the class clusters using the distance as the criterion for similarity measure; subsequently, iteratively updates the clustering centers and the allocation of the remaining points using the average value of each class cluster.
2. K-means++ [33]: This model improves the initial clustering center selection for K-means by using the adaptive sampling scheme of D^2 *–Sampling*, which makes the initial clustering center more reasonable; the clustering is subsequently completed by using the same iterative scheme as K-means.
3. AFK–MC2 [2]: This model uses the Markov Chain Monte Carlo (MCMC) method to approximate the D^2 *–Sampling* process of K-means++, which does not need to recalculate the distance matrix repeatedly and improves the clustering efficiency while guaranteeing good clustering results.
4. DPC [42]: This model proposes a density-based clustering method, which considers that the cluster center should have the following characteristics: a higher density than its neighbors; and a relatively large distance from the point with higher density; the cluster center is determined based on the above two characteristics, and subsequently each non-center point is assigned to the cluster in which the point nearest to it and with a higher density than it is located.
5. DADC [3]: This model proposes a domain adaptive density clustering method based on KNN density, which can effectively detect the peaks of domain density of data points in different density regions, and determine the clustering centers based on the decision parameters formed by the domain density and the distance to the points with higher density.
6. Extreme clustering [59]: This model proposes a new density clustering criterion: clustering based on extreme values, i.e., the density extreme value point in the neighborhood is identified as the clustering center. The main difference between peak clustering and extreme value clustering is that the extreme value clustering center has the largest density in the neighborhood, which can ensure that there is no cross-region between the neighbors of the clustering center.

Table 3
Statistics of the used time series datasets.

Dataset	Type	Train	Test	Length	Classes
Beef	Spectro	30	30	471	5
CBF	Simulated	30	900	128	3
ChlorineConcentration	Sensor	467	3840	166	3
CinCECGTorso	Sensor	40	1380	1639	4
DistalPhalanxOutlineAgeGroup	Image	139	400	81	3
DistalPhalanxTW	Image	400	139	80	6
ECG5000	ECG	500	4500	140	5
FaceFour	Image	24	88	350	4
Haptics	Motion	155	308	1092	5
LargeKitchenAppliances	Device	375	375	720	3
MedicalImages	Image	381	760	99	10
MiddlePhalanxTW	Image	399	154	80	5
MiddlePhalanxOutlineAgeGroup	Image	400	154	80	3
Plane	Sensor	105	105	144	7
ScreenType	Device	375	375	720	3
SmallKitchenAppliances	Device	375	375	720	3
StarLightCurves	Sensor	1000	8236	1024	3
Trace	Sensor	100	100	275	4
Worms	Motion	181	77	900	5

Evaluation Metrics. We employ Receiver Operating Characteristics (ROC) [10] and the area under the curve (AUC) as our evaluation metrics.

4.1. Clustering Accuracy Comparison Results

We conduct comparison experiments between our method DCMD and 6 baseline competitors on 19 benchmark datasets. The corresponding experimental results are listed in Table 4 and Figure 3. According to these results, we can find our proposed DCMD model achieves the highest accuracy on 16 selected datasets, the highest number of optimal times among all the compared models, and achieves a significant improvement relative to other methods on many datasets such as CBF, Trace, and Worms, which are usually characterized by high complexity and noise, and the superior performance of the DCMD model further demonstrates its superiority in handling complex time series data. The reason is listed as our proposed multi-dimensional representation fusion can retain essential information from the original time series to strengthen temporal representation learning capability, therefore significantly improving clustering accuracy. Overall, these result comparisons clearly show that the superior performance of the DCMD model in terms of clustering accuracy makes it the best choice for current time series clustering tasks.

4.2. Clustering Efficiency Comparison Results

The running times of our method and baseline competitors on the benchmark datasets are listed in Table 5. Moreover, we visualize corresponding experimental results in Figure 4. According to these results, we can clearly find that our DCMD model has a lower running time than the other models and a higher overall running efficiency. On datasets such as LargeKitchenAppliances and SmallKitchenAppliances, the running efficiency has a clear advantage. Although the run efficiency is close to Extreme Clustering, the DCMD model outperforms Extreme Clustering in terms of clustering accuracy. The reasons are listed as follows: 1) our proposed feature fusion strategy can reduce original data dimensions to accelerate temporal representation and 2) our proposed neighbor weighted clustering can efficiently reflect the regional distribution relationship

Table 4
The comparison of clustering accuracy between our model and other algorithms on 19 datasets

Dataset	K-means	K-means++	AFK-MC^2	DPC	DADC	Extreme Clustering	DCMD
Beef	0.293	0.300	0.313	0.333	0.433	0.233	0.442
CBF	0.480	0.562	0.493	0.500	0.567	0.467	0.667
ChlorineConcentration	0.468	0.379	0.396	0.458	0.465	0.465	0.567
CinCECGTorso	0.405	0.455	0.510	0.400	0.475	0.650	0.600
DistalPhalanxOutlineAgeGroup	0.643	0.715	0.535	0.538	0.572	0.803	0.805
DistalPhalanxTW	0.533	0.600	0.536	0.565	0.480	0.745	0.735
ECG5000	0.707	0.628	0.670	0.548	0.526	0.916	0.814
FaceFour	0.467	0.625	0.608	0.583	0.417	0.583	0.625
Haptics	0.289	0.341	0.353	0.297	0.297	0.265	0.355
LargeKitchenAppliances	0.365	0.411	0.378	0.467	0.464	0.413	0.483
MedicalImages	0.336	0.317	0.349	0.283	0.320	0.517	0.517
MiddlePhalanxTW	0.400	0.527	0.528	0.398	0.541	0.517	0.556
MiddlePhalanxOutlineAgeGroup	0.593	0.702	0.689	0.690	0.505	0.613	0.725
Plane	0.350	0.560	0.600	0.419	0.500	0.686	0.705
ScreenType	0.347	0.371	0.364	0.392	0.393	0.350	0.405
SmallKitchenAppliances	0.342	0.373	0.383	0.507	0.515	0.395	0.515
StarLightCurves	0.570	0.570	0.570	0.605	0.530	0.577	0.667
Trace	0.542	0.551	0.544	0.338	0.539	0.540	0.660
Worms	0.307	0.282	0.293	0.376	0.320	0.326	0.470

Figure 3
Clustering accuracy of our model vs other models on 10 datasets

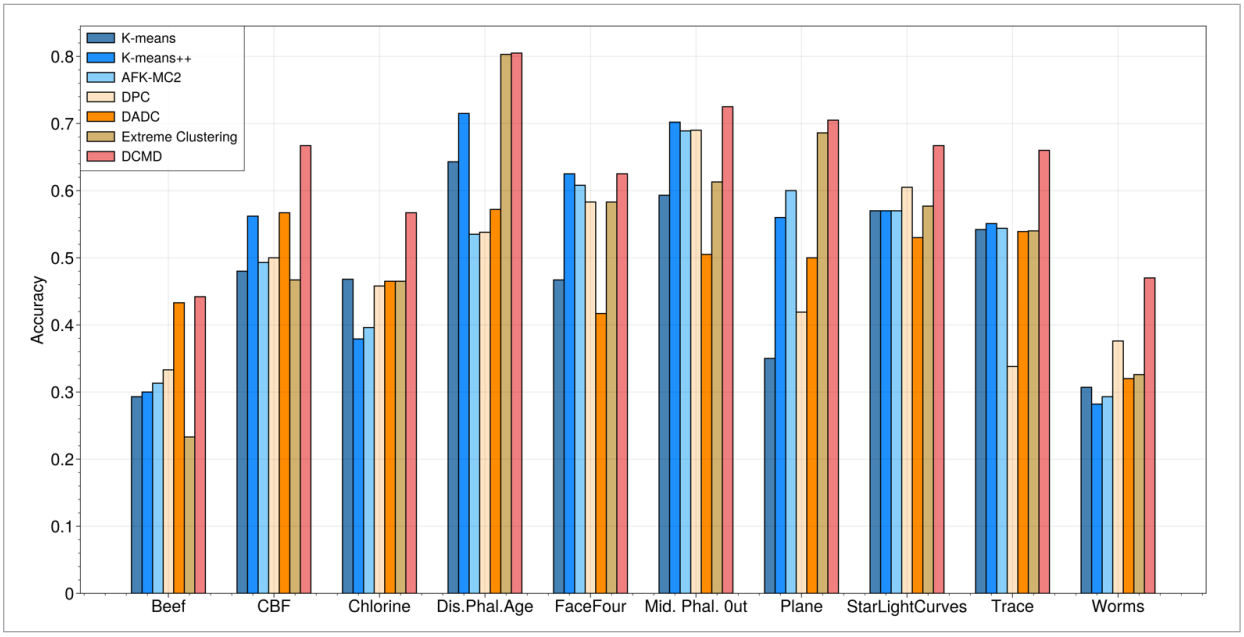
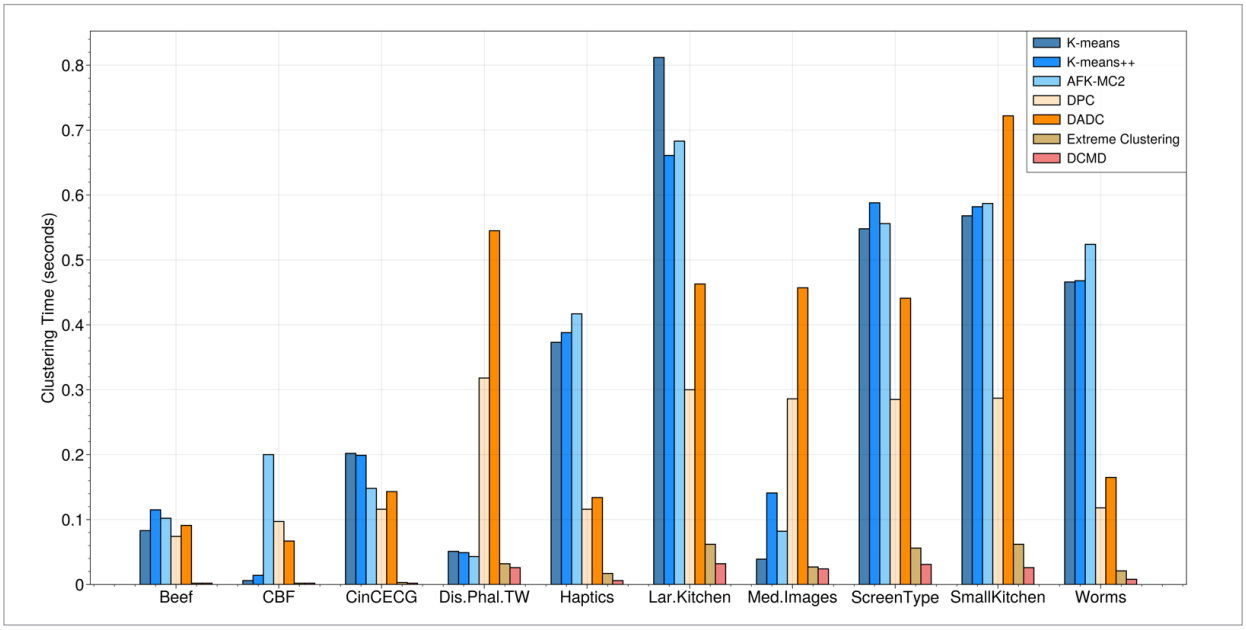


Table 5
The comparison of clustering efficiency between our model and other algorithms on 19 datasets

Dataset	K-means	K-means++	AFK-MC^2	DPC	DADC	Extreme Clustering	DCMD
Beef	0.083	0.115	0.102	0.074	0.091	0.002	0.002
CBF	0.006	0.014	0.200	0.097	0.067	0.002	0.002
ChlorineConcentration	0.019	0.021	0.025	0.397	0.658	0.049	0.035
CinCECGTorso	0.202	0.199	0.148	0.116	0.143	0.003	0.002
DistalPhalanxOutlineAgeGroup	0.027	0.016	0.016	0.333	0.517	0.031	0.026
DistalPhalanxTW	0.051	0.049	0.043	0.318	0.545	0.032	0.026
ECG5000	0.029	0.119	0.319	0.452	0.751	0.053	0.036
FaceFour	0.054	0.066	0.065	0.063	0.127	0.002	0.002
Haptics	0.373	0.388	0.417	0.116	0.134	0.017	0.006
LargeKitchenAppliances	0.812	0.661	0.683	0.300	0.463	0.062	0.032
MedicalImages	0.039	0.141	0.082	0.286	0.457	0.027	0.024
MiddlePhalanxTW	0.138	0.098	0.137	0.314	1.446	0.026	0.026
MiddlePhalanxOutlineAgeGroup	0.039	0.161	0.022	0.336	0.527	0.031	0.033
Plane	3.632	4.468	4.260	0.081	0.384	0.005	0.004
ScreenType	0.548	0.588	0.556	0.285	0.441	0.056	0.031
SmallKitchenAppliances	0.568	0.582	0.587	0.287	0.722	0.062	0.026
StarLightCurves	15.369	22.926	20.137	1.591	4.785	0.536	0.119
Trace	2.508	2.802	2.678	0.080	0.130	0.005	0.003
Worms	0.466	0.468	0.524	0.118	0.165	0.021	0.008

Figure 4
Clustering time of our model vs other models on 10 datasets



of similar features, thus boosting clustering efficiency. In general, these impressive runtime efficiency results not only emphasize the efficiency of the DCMD model in processing largescale time series data but also provide strong support for processing large-scale data in a limited time.

5. Conclusion

In this paper, we propose a novel temporal density clustering model (DCMD), which integrates a temporal representation module and a temporal clustering module to enhance both efficiency and accuracy. Concretely, our temporal representation module effectively retains essential feature information during dimensionality reduction, while the clustering module can accelerate regional density calculation efficiently. Comprehensive evaluations on benchmark datasets demonstrate the effectiveness and superiority of our approach, highlighting its potential applications across various domains.

In the future, we plan to deepen and widen our work from the following four aspects: 1) we intend to integrate the necessary Implicit Scale Correspondence Learning [56], information fusion strategies [54], and Fine-grained feature fusion network [36, 39] into our model to improve clustering accuracy. 2) We will incorporate the graph convolutional metric learning [73, 74], temporal state perception [27], and online KNN join processing [18] [67] into our model to accelerate temporal representation, whereby boosting the overall efficiency. 3) We plan to introduce automatic component identification [6, 24], variational rectification inference [53] into our model to enhance the precision of clustering, especially in the face of noisy data [49]. And 4) we desire to adopt our model as a useful data preprocessing tool into a wide range of time series application scenarios, such as time series similarity search [65, 69], temporal anomaly detection [70, 66], and sequential cross-domain recommendation [62, 72].

References

1. Aghabozorgi, S., Shirkhorshidi, A. S., Wah, T. Y. Time-Series Clustering-A Decade Review. Information Systems, 2015, 53, 16-38. https://doi.org/10.1016/j.is.2015.04.007
2. Bachem, O., Lucic, M., Hassani, H., Krause, A. Fast and Provably Good Seedings for K-Means. Advances in Neu-ral Information Processing Systems, 2016, 29. https://doi.org/10.1145/3404835.3462835
3. Chen, J., Philip, S. Y. A Domain Adaptive Density Clus-tering Algorithm for Data with Varying Density Distribu-tion. IEEE Transactions on Knowledge and Data Engi-neering, 2019, 33, 2310-2321. https://doi.org/10.1109/TKDE.2019.2954133
4. Cui, M. Introduction to the K-Means Clustering Algo-rithm Based on the Elbow Method. Accounting, Auditing and Finance, 2020, 1, 5-8. DOI: 10.23977/accaf.2020.010102
5. Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Keogh, E. The UCR Time Series Archive. IEEE/CAA Journal of Auto-matica Sinica, 2019, 6, 1293-1305. https://doi.org/10.1109/JAS.2019.1911747
6. Du, M., Wei, Y., Hu, Y., Zheng, X., Ji, C. Automatic Component Identification Based on Time Series Classifi-cation for Intelligent Devices. IEEE International Confer-ence on Data Mining, 2023, 540-549. https://doi.org/10.1109/ICDMW60847.2023.00077
7. Du, M., Wei, Y., Hu, Y., Zheng, X., Ji, C. Multivariate Time Series Classification Based on Fusion Features. Ex-pert Systems with Applications, 2024, 248, 123452-123462. https://doi.org/10.1016/j.eswa.2024.123452
8. Farahani, A., Voghoei, S., Rasheed, K., Arabnia, H. R. A Brief Review of Domain Adaptation. Advances in Data Science and Information Engineering, 2021, 877-894. https://doi.org/10.1007/978-3-030-71704-9_65
9. Gan, J., Tao, Y. Dynamic Density-Based Clustering. Pro-ceedings of the 2017 ACM International Conference on Management of Data, 2017, 1493-1507. https://doi.org/10.1145/3035918.3064050
10. Han, H. The Utility of Receiver Operating Characteristic Curve in Educational Assessment: Performance Predic-tion. Mathematics, 2022, 10, 1493. https://doi.org/10.3390/math10091493
11. Hisaharo, S., Nishimura, Y., Takahashi, A. Optimizing LLM Inference Clusters for Enhanced Performance and Energy Efficiency. Authorea Preprints, 2024. https://doi.org/10.36227/techrxiv.172348951.12175366/v1
12. Hu, L., Wang, L., Chen, Y., Hu, N., Jiang, Y. Bearing Fault Diagnosis Using Piecewise Aggregate Approxima-tion and Complete Ensemble Empirical Mode Decomposi-tion with Adaptive Noise. Sensors, 2022, 22, 6599. https://doi.org/10.3390/s22176599

13. Hu, Y., Ji, C., Jing, M., Ding, Y., Kuai, S., Li, X. A Contin-uous Segmentation Algorithm for Streaming Time Series. International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2017. https://doi.org/10.1007/978-3-319-59288-6_13

14. Hu, Y., Ji, C., Zhang, Q., Chen, L., Zhan, P., Li, X. A Novel Multi-Resolution Representation for Time Series Sensor Data Analysis. Soft Computing, 2020, 24, 10535-10560. https://doi.org/10.1007/s00500-019-04562-7

15. Hu, Y., Jiang, Z., Zhan, P., Zhang, Q., Ding, Y., Li, X. A Novel Multi-Resolution Representation for Streaming Time Series. Information and Knowledge in the Internet of Things, 2018, 178-184. https://doi.org/10.1016/j.procs.2018.03.069

16. Hu, Y., Ren, P., Luo, W., Zhan, P., Li, X. Multi-Resolution Representation with Recurrent Neural Networks Appli-cation for Streaming Time Series in IoT. Computer Net-works, 2019, 152, 114-132. https://doi.org/10.1016/j.comnet.2019.01.035

17. Hu, Y., Yang, C., Ji, C., Xu, Y., Li, X. Efficient Snapshot KNN Join Processing for Large Data Using MapReduce. IEEE International Conference on Parallel and Distribut-ed Systems, 2016, 713-720. https://doi.org/10.1109/ICPADS.2016.0098

18. Hu, Y., Yang, C., Zhan, P., Zhao, J., Li, Y., Li, X. Efficient Continuous KNN Join Processing for Real-Time Recom-mendation. Personal and Ubiquitous Computing, 2021, 25, 1001-1011. https://doi.org/10.1007/s00779-019-01282-5

19. Hu, Y., Zhan, P., Xu, Y., Zhao, J., Li, Y., Li, X. Temporal Representation Learning for Time Series Classification. Neural Computing and Applications, 2021, 33, 3169-3182. https://doi.org/10.1007/s00521-020-05179-w

20. Huang, S., Kang, Z., Xu, Z., Liu, Q. Robust Deep K-Means: An Effective and Simple Method for Data Clus-tering. Pattern Recognition, 2021, 117, 107996. https://doi.org/10.1016/j.patcog.2021.107996

21. Ji, C., Du, M., Hu, Y., Liu, S., Pan, L., Zheng, X. Time Series Classification Based on Temporal Features. Applied Soft Computing, 2022, 128, 109494-109455. https://doi.org/10.1016/j.asoc.2022.109494

22. Ji, C., Du, M., Wei, Y., Hu, Y., Liu, S., Pan, L., Zheng, X. Time Series Classification with Random Temporal Features. Journal of King Saud University - Computer and Information Sciences, 2023, 35, 101783-101793. https://doi.org/10.1016/j.jksuci.2023.101783

23. Ji, C., Hu, Y., Liu, S., Pan, L., Li, B., Zheng, X. Fully Con-volutional Networks with Shapelet Features for Time Series Classification. Information Sciences, 2022, 612, 835-847. https://doi.org/10.1016/j.ins.2022.09.009

24. Ji, C., Hu, Y., Wang, K., Zhan, P., Li, X., Zheng, X. Identi-fiable Temporal Feature Selection via Horizontal Visibility Graph Towards Smart Medical Applications. Interdisci-plinary Sciences: Computational Life Sciences, 2021, 13, 717-730. https://doi.org/10.1007/s12539-021-00460-5

25. Ji, C., Zou, X., Hu, Y., Liu, S., Lyu, L., Zheng, X. XG-SF: An XGBoost Classifier Based on Shapelet Features for Time Series Classification. International Conference on Identification, Information and Knowledge in the Internet of Things, 2019, 24-28. https://doi.org/10.1016/j.procs.2019.01.179

26. Jiang, W., Wu, J., Li, F., Wang, G., Zheng, H. Trust Eval-uation in Online Social Networks Using Generalized Network Flow. IEEE Transactions on Computers, 2015, 65, 952-963. https://doi.org/10.1109/TC.2015.2435785

27. Jin, Y., Wang, S., Liu, F., Fan, H., Hu, Y., Li, X., Liu, S. Deep Temporal State Perception Towards Artificial Cyber-Physical Systems. IEEE Internet of Things Journal, 2023, 10, 18782-18789. https://doi.org/10.1109/JIOT.2023.3239413

28. Karthigeyan, C. M. T., Rani, C. Optimizing Parkinson's Disease Diagnosis with Multimodal Data Fusion Techniques. Information Technology and Control, 2024, 53, 262-279. https://doi.org/10.5755/j01.itc.53.1.34718

29. Krawczyk, B. Learning from Imbalanced Data: Open Challenges and Future Directions. Progress in Artificial Intelligence, 2016, 5, 221-232. https://doi.org/10.1007/s13748-016-0094-0

30. Krishnan, N., Raja, K., Divakaran, S. Maximization of WSN-Based IoT Systems Lifetime by Minimized Intra-Cluster Transmission Distance Clustering Protocol. In-formation Technology and Control, 2023, 52, 140-154. https://doi.org/10.5755/j01.itc.52.1.32199

31. Li, G., Wang, X., Li, Y., Li, Z. Adaptive Clustering Object Detection Method for UAV Images Under Long-Tailed Distributions. Information Technology and Control, 2024, 52, 1025-1044. https://doi.org/10.5755/j01.itc.52.4.33460

32. Li, H., Liu, T., Wu, X., Chen, Q. A Bearing Fault Diagno-sis Method Based on Enhanced Singular Value Decompo-sition. IEEE Transactions on Industrial Informatics, 2020, 17, 3220-3230. https://doi.org/10.1109/TII.2020.3001376

33. Li, H., Wang, J. Collaborative Annealing Power K-Means++ Clustering. Knowledge-Based Systems, 2022, 255, 109593. https://doi.org/10.1016/j.knosys.2022.109593
34. Li, H., Wu, X., Wan, X., Lin, W. Time Series Clustering via Matrix Profile and Community Detection. Advanced Engineering Informatics, 2022, 54, 101771. https://doi.org/10.1016/j.aei.2022.101771
35. Li, H., Xun, L., Zheng, G. Piecewise Linear Strain Cosserat Model for Soft Slender Manipulator. IEEE Transactions on Robotics, 2023, 39, 2342-2359. https://doi.org/10.1109/TRO.2023.3236942
36. Lin, H., Wen, H., Song, X., Liu, M., Hu, Y., Nie, L. Fine-Grained Textual Inversion Network for Zero-Shot Com-posed Image Retrieval. Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, 240-250. https://doi.org/10.1145/3626772.3657831
37. Liu, H., Shao, M., Li, S., Fu, Y. Infinite Ensemble for Im-age Clustering. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 1745-1754. https://doi.org/10.1145/2939672.2939813
38. Liu, J., Yao, J., Wang, Z., Wang, Z., Huang, L. Small Sample Time Series Classification Based on Data Aug-mentation and Semi-Supervised Learning. Information Technology and Control, 2024, 53, 470-491. https://doi.org/10.5755/j01.itc.53.2.35797
39. Liu, Y., Xu, M., Xiao, T., Tang, H., Hu, Y., Nie, L. Hetero-geneous Feature Collaboration Network for Salient Object Detection in Optical Remote Sensing Images. IEEE Trans-actions on Geoscience and Remote Sensing, 2024, 62, 1-14. https://doi.org/10.1109/TGRS.2024.3439401
40. Raykov, Y. P., Boukouvalas, A., Baig, F., Little, M. A. What to Do When K-Means Clustering Fails: A Simple Yet Principled Alternative Algorithm. PLoS One, 2016, 11, 1-28. https://doi.org/10.1371/journal.pone.0162259
41. Ren, H., Liao, X., Li, Z., Al-Ahmari, A. Anomaly Detec-tion Using Piecewise Aggregate Approximation in the Amplitude Domain. Applied Intelligence, 2018, 48, 1097-1110. https://doi.org/10.1007/s10489-017-1017-x
42. Ren, W., Zhang, J., Di, X., Lu, Y., Zhang, B., Zhao, J. Anomaly Detection Algorithm Based on CFSFDP. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2020, 24, 453-460. https://doi.org/10.1109/CIAPP.2017.8167057
43. Ren, Y., Wang, N., Li, M., Xu, Z. Deep Density-Based Image Clustering. Knowledge-Based Systems, 2020, 197, 105841. https://doi.org/10.1016/j.knosys.2020.105841
44. Ruan, S., Mehmood, R., Daud, A., Dawood, H., Alowibdi, J. S. An Adaptive Method for Clustering by Fast Search-and-Find of Density Peaks: Adaptive-DP. Proceedings of the 26th International Conference on World Wide Web Companion, 2017, 119-127. https://doi.org/10.1145/3041021.3054148
45. Sampath, P., Jomy, E., Kalyanaraman, R., Shamuga-nathan, V., Crespo, R. G., Chakrabarti, P. Detecting the Medical Plant Association from PubMed Using Hyper-graph-Based Clustering With Dominating Set. Infor-mation Technology and Control, 2024, 53, 758-771. DOI: 10.5755/j01.itc.53.3.35101
46. Sardar, T. H. Reflecting on a Decade of Evolution: MapReduce-Based Advances in Partitioning-Based, Hier-archical-Based, and Density-Based Clustering (2013-2023). Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2024, e1566. https://doi.org/10.1002/widm.1566
47. Shayboub, M., Reda, E., Al-Mahdi, H., Nassar, H. An Ef-ficient Framework for Sensor Data Collection by UAV Based on Clustering, Two-Fold Ant Colony Optimization and Node Grouping. Information Technology and Con-trol, 2024, 53, 865-887. https://doi.org/10.21203/rs.3.rs-3079277/v1
48. Shen, M., Wen, P., Song, B., Li, Y. An EEG-Based Real-Time Epilepsy Seizure Detection Approach Using Discrete Wavelet Transform and Machine Learning Methods. Biomedical Signal Processing and Control, 2022, 77, 103820. https://doi.org/10.1016/j.bspc.2022.103820
49. Shi, H., Liu, M., Mu, X., Song, X., Hu, Y., Nie, L. Breaking Through the Noisy Correspondence: A Robust Model for Image-Text Matching. ACM Transactions on Information Systems, 2024, 42, 1-26. https://doi.org/10.1145/3662732
50. Shokrzade, A., Ramezani, M., Tab, F. A., Mohammad, M. A. A Novel Extreme Learning Machine-Based KNN Clas-sification Method for Dealing with Big Data. Expert Sys-tems with Applications, 2021, 183, 115293. https://doi.org/10.1016/j.eswa.2021.115293
51. Shu, W., Li, Y. A Novel Demand-Responsive Customized Bus Based on Improved Ant Colony Optimization and Clustering Algorithms. IEEE Transactions on Intelligent Transportation Systems, 2022, 24, 8492-8506. https://doi.org/10.1109/TITS.2022.3145655
52. Singh, P., Meshram, P. A. Survey of Density-Based Clus-tering Algorithms and Its Variants. 2017 International Conference on Inventive Computing and Informatics (ICICI). IEEE, 2017, 920-926. https://doi.org/10.1109/ICICI.2017.8365272

53. Sun, H., Wei, Q., Feng, L., Hu, Y., Liu, F., Fan, H., Yin, Y. Variational Rectification Inference for Learning with Noisy Labels. International Journal of Computer Vision, 2024. https://doi.org/10.1007/s11263-024-02205-5

54. Tang, H., Hu, Y., Wang, Y., Zhang, S., Xu, M., Zhu, J., Zheng, Q. Listen as You Wish: Fusion of Audio and Text for Cross-Modal Event Detection in Smart Cities. Infor-mation Fusion, 2024, 110, 102460:1-102460:9. https://doi.org/10.1016/j.inffus.2024.102460

55. Wang, G., Song, Q. Automatic Clustering via Outward Statistical Testing on Density Metrics. IEEE Transactions on Knowledge and Data Engineering, 2016, 28, 1971-1985. https://doi.org/10.1109/TKDE.2016.2535209

56. Wang, K., Liu, H., Jie, L., Li, Z., Hu, Y., Nie, L. Explicit Granularity and Implicit Scale Correspondence Learning for Point-Supervised Video Moment Localization. ACM International Conference on Multimedia, 2024. https://doi.org/10.1145/3664647.3680774

57. Wang, K., Wang, C., Wang, Y., Luo, W., Zhan, P., Hu, Y., Li, X. Time Series Classification via Enhanced Temporal Representation Learning. IEEE International Conference on Big Data Analytics, 2021. https://doi.org/10.1109/ICBDA51983.2021.9403177

58. Wang, K., Wen, H., Li, G. Accurate Frequency Estimation by Using Three-Point Interpolated Discrete Fourier Trans-form Based on Rectangular Window. IEEE Transactions on Industrial Informatics, 2020, 17, 73-81. https://doi.org/10.1109/TII.2020.2981542

59. Wang, S., Li, Q., Zhao, C., Zhu, X., Yuan, H., Dai, T. Extreme Clustering-A Clustering Method Via Density Extreme Points. Information Sciences, 2021, 542, 24-39. https://doi.org/10.1016/j.ins.2020.06.069

60. Wei, Y., Wang, Y., Du, M., Hu, Y., Ji, C. Adaptive Shape-let Selection for Time Series Classification. Proceedings of the International Conference on Computer Supported Cooperative Work in Design, 2023, 1607-1612. https://doi.org/10.1109/CSCWD57460.2023.10152664

61. Xia, H., Zhou, Y., Li, J., Yue, X., Li, J. Outlier Detection Method Based on Improved DPC Algorithm and Centrif-ugal Factor. Information Sciences, 2024, 682, 121255. https://doi.org/10.1016/j.ins.2024.121255

62. Xu, Y., Hong, X., Peng, Z., Hu, Y., Yang, G. HOMMIT: A Sequential Recommendation for Modeling Interest-Transferring via High-Order Markov Model. Internation-al Conference on Web Information Systems Engineering, 2017, 372-386. https://doi.org/10.1007/978-3-319-68786-5_30

63. Yu, Z., Kuang, Z., Liu, J., Chen, H., Zhang, J., You, J., Wong, H. S., Han, G. Adaptive Ensembling of Semi-Supervised Clustering Solutions. IEEE Transactions on Knowledge and Data Engineering, 2017, 29, 1577-1590. https://doi.org/10.1109/TKDE.2017.2695615

64. Yu, Z., Luo, P., You, J., Wong, H. S., Leung, H., Wu, S., Zhang, J., Han, G. Incremental Semi-Supervised Cluster-ing Ensemble for High Dimensional Data Clustering. IEEE Transactions on Knowledge and Data Engineering, 2015, 28, 701-714. https://doi.org/10.1109/TKDE.2015.2499200

65. Zhan, P., Hu, Y., Chen, L., Luo, W., Li, X. Efficient Search of the Optimal Prefix and Suffix Invariant Size for Dy-namic Time Warping. IEEE International Conference on High Performance Computing and Communications, 2019, 1971-1978. https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00272

66. Zhan, P., Hu, Y., Chen, L., Luo, W., Li, X. SPAR: Set-Based Piecewise Aggregate Representation for Time Series Anomaly Detection. Science China Information Sciences, 2021, 64, 149101-149103. https://doi.org/10.1007/s11432-020-3021-6

67. Zhan, P., Hu, Y., Luo, W., Xu, Y., Zhang, Q., Li, X. Fea-ture-Based Online Segmentation Algorithm for Streaming Time Series. International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2019, 477-487. https://doi.org/10.1007/978-3-030-12981-1_33

68. Zhan, P., Hu, Y., Zhang, Q., Zheng, J., Li, X. Feature-Based Dividing Symbolic Time Series Representation for Streaming Data Processing. International Conference on Information Technology in Medicine and Education, 2018. https://doi.org/10.1109/ITME.2018.00184

69. Zhan, P., Sun, C., Hu, Y., Luo, W., Zheng, J., Li, X. Feature-Based Online Representation Algorithm for Stream-ing Time Series Similarity Search. International Journal of Pattern Recognition and Artificial Intelligence, 2019, 34, 2050010:1-2050010:25. https://doi.org/10.1142/S021800142050010X

70. Zhan, P., Wang, S., Wang, J., Qu, L., Wang, K., Hu, Y., Li, X. Temporal Anomaly Detection on IIoT-Enabled Manu-facturing. Journal of Intelligent Manufacturing, 2021, 32, 1669-1678. https://doi.org/10.1007/s10845-021-01768-1

71. Zhang, Y., Duan, L., Duan, M. A New Feature Extraction Approach Using Improved Symbolic Aggregate Approx-imation for Machinery Intelligent Diagnosis. Measure-ment, 2019, 133, 468-478. https://doi.org/10.1016/j.measurement.2018.10.045

72. Zhang, Y., Liu, F., Hu, Y., Li, X., Dong, X., Cheng, Z. Du-al Attentive Graph Convolutional Networks for Cross-Domain Recommendation. Journal of Intelligent & Fuzzy Systems, 2023, 44, 7367-7378. https://doi.org/10.3233/JIFS-222411

73. Zhao, X., Hu, Y., Mu, Y., Li, M., Wang, H., Li, X. Graph Convolutional Metric Learning for Recommender Sys-tems in Smart Cities. IEEE Transactions on Consumer Electronics, 2024. https://doi.org/10.1109/TCE.2024.3411704

74. Zhao, X., Liu, F., Liu, H., Xu, M., Tang, H., Li, X., Hu, Y. CoGCN: Co-Occurring Item-Aware GCN for Recom-mendation. Neural Computing and Applications, 2023, 35, 25107-25120. https://doi.org/10.1007/s00521-023-08703-w

75. Zheng, Y., Guo, Q., Tung, A. K., Wu, S. LazyLSH: Ap-proximate Nearest Neighbor Search for Multiple Distance Functions with a Single Index. Proceedings of the 2016 International Conference on Management of Data, 2016, 2023-2037. https://doi.org/10.1145/2882903.2882930

76. Zou, H. Clustering Algorithm and Its Application in Data Mining. Wireless Personal Communications, 2020, 110, 21-30. https://doi.org/10.1007/s11277-019-06709-z