Automotive Innovation (2023) 6:89–115
https://doi.org/10.1007/s42154-022-00205-0

Review of Clustering Technology and Its Application in Coordinating Vehicle Subsystems

Caizhi Zhang[1] · Weifeng Huang[1] · Tong Niu[1] · Zhitao Liu[2] · Guofa Li[1] · Dongpu Cao[3]

Received: 6 June 2022 / Accepted: 9 November 2022 / Published online: 17 January 2023

Abstract

Clustering is an unsupervised learning technology, and it groups information (observations or datasets) according to similarity measures. Developing clustering algorithms is a hot topic in recent years, and this area develops rapidly with the increasing complexity of data and the volume of datasets. In this paper, the concept of clustering is introduced, and the clustering technologies are analyzed from traditional and modern perspectives. First, this paper summarizes the principles, advantages, and disadvantages of 20 traditional clustering algorithms and 4 modern algorithms. Then, the core elements of clustering are presented, such as similarity measures and evaluation index. Considering that data processing is often applied in vehicle engineering, finally, some specific applications of clustering algorithms in vehicles are listed and the future development of clustering in the era of big data is highlighted. The purpose of this review is to make a comprehensive survey that helps readers learn various clustering algorithms and choose the appropriate methods to use, especially in vehicles.

Keywords Unsupervised learning · Clustering · Similarity measures · Vehicle

Abbreviations

ACO	Ant colony optimization
AMI	Adjusted mutual information
ART	Adaptive resonance theory
CH	Calinski-Harabasz index
CLARA	Clustering large applications
CLIQUE	Clustering in quest
CURE	Clustering using representative
DBI	Davies-Bouldin index
DBSCAN	Density-based spatial clustering of applications with noise
DTW	Dynamic time warping
DVI	Dunn validity index
EV	Electric vehicles
FCM	Fuzzy C-means
FCV	Fuel cell vehicles
GMM	Gaussian mixture model
IoV	Internet of Vehicles
ITS	Intelligent transportation system
NMI	Normalized mutual information
PAM	Partitioning around medoid
RI	Rand index
SOM	Self-organizing mapping
SSE	Within-cluster sum of squared error
STING	Statistical information grid
VANETs	Vehicular ad hoc networks

1 Introduction

Daily objects are often categorized according to their characteristics. For instance, competition can have junior and senior groups divided by age. Whenever an object with a label is found, it can be put into groups with the same label. This is commonly referred to as supervised learning [1, 2].

Academic Editor: Quan Zhou

Caizhi Zhang and Weifeng Huang have contributed equally to this work

✉ Caizhi Zhang
czzhang@cqu.edu.cn

✉ Dongpu Cao
dp_cao2016@163.com

[1] Chongqing Automotive Collaborative Innovation Centre, The State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China

[2] State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

[3] School of Vehicle and Mobility, Tsinghua University, Beijing, China

There has been much research on supervised learning, and the technology has reached a mature stage with wide applications. Neural networks are the most common methods for supervised learning [3, 4]. Fuzzy mathematics [5] and decision trees [6] are both satisfactory classification methods. Many researchers have improved the multi-objective particle swarm optimization (PSO) algorithm [7] and the Bayesian classifier [8] to achieve higher accuracy of classification. But in many cases, when no groups have the pre-assigned label information, data can be put into different groups according to the similarity principle, therefore, unsupervised learning is required [9, 10]. For unsupervised learning, the label information of training samples is unknown, and the goal is to reveal the intrinsic nature and objective law of data by learning unlabeled training samples, which provides the basis for further data analysis [11, 12]. One of the best-known methods is clustering analysis, which attempts to partition the samples from a dataset into several disjoint subsets with each of them called a "cluster." All similar points are divided into a cluster, which is not like others. This division process is performed using proximity measures, density measures, and other similar measures [13]. In the current big data era, the clustering algorithm is widely used in regression prediction, data mining, image recognition, and other fields.

Data clustering was first mentioned in an article dealing with anthropological data which was published in 1954 [14], and then, it was widely studied and applied. Sisodia et al. [15] and Dave et al. [16] summarized some of the most basic clustering methods. K-means [17, 18] is the earliest and simplest clustering method that has been used for decades. To date, it is still the most popular algorithm. Many scholars are still trying to improve clustering in combination with other data processing methods. Graph-based clustering in collaboration with neural networks is a promising clustering method due to its high robustness [19, 20]. Multi-objective clustering can improve the accuracy of clustering and is widely used in forecasting biogenetics and financial trends [21]. In the twenty-first century, there are many kinds of clustering algorithms such as clustering algorithms based on kernel [22] and clustering algorithms for streaming data [23]. These methods are more suitable for processing web data. As similarity is the core factor of clustering analysis, clustering results are expected to show high intra-cluster similarity and low inter-cluster similarity [24], but there are many difficulties in practical applications. For example, the speed data of a vehicle is a set of time series data, if the clustering or classification algorithm wants to divide these different varieties of samples into similar clusters. The preprocessing of high-dimensional data, the definition of the similarity measure [25, 26] in high dimension, and the imprecise matching of similarity models are the problems that must be taken into account.

To solve the above-mentioned problems and improve the accuracy of cluster analysis, clustering algorithms can be developed as well as methods of similarity measures according to the dataset size and shape and the expected target. For example, for the time series of non-convex large-scale datasets, the density-based clustering [27] can be used and dynamic time warping (DTW) [28] can measure similarity, and the following sections explain this in detail.

In view of the above problems, this paper analyzes and compares several clustering techniques in Sect. 2. Then, Sect. 3 defines the classification patterns according to similarity measures. Several evaluation indicators of clustering accuracy are illustrated in Sect. 4. The applications of the clustering methods in the automotive field are summarized in Sect. 5. The future development of the clustering algorithm is discussed in Sect. 6.

2 Clustering Technologies

Clustering technologies are broadly divided into six categories: hierarchical clustering, partition clustering, density-based clustering, model-based clustering, grid-based clustering, and modern clustering. As shown in Fig. 1, there are many specific algorithms in each category [26, 29–32]. Various clustering algorithms are introduced in detail in this section, including principles, advantages, and disadvantages. There is a wide amount of literature on all kinds of clustering technologies and extensive applications, but no algorithm can be perfect under all circumstances. Some methods perform great with some specific nature of data but are not suitable for other types of data. Therefore, for specific tasks (goals), specific strategies should be adopted and tested in real-world applications.

2.1 Hierarchical Clustering

The hierarchical nested clustering tree is created by calculating the similarity between different categories of data points. In the clustering tree, the original data points for different categories are at the bottom layer of the tree and a root node of a cluster is at the top layer of the tree, as shown in Fig. 2 [33]. The clustering is performed on each hierarchy.

Hierarchical clustering can be categorized into agglomerative hierarchical clustering and divisive hierarchical clustering. In the former, each object is a cluster. The same objects are found according to the linkage, and finally, a new cluster is formed. The latter is just the opposite. There are three ways to verdict the same cluster according to the linkage [31]: (1) Single Linkage—some people also call it the minimal method or the nearest neighbor method. The same clusters are chosen by comparing the nearest samples of the two clusters. (2) Complete Linkage—some people

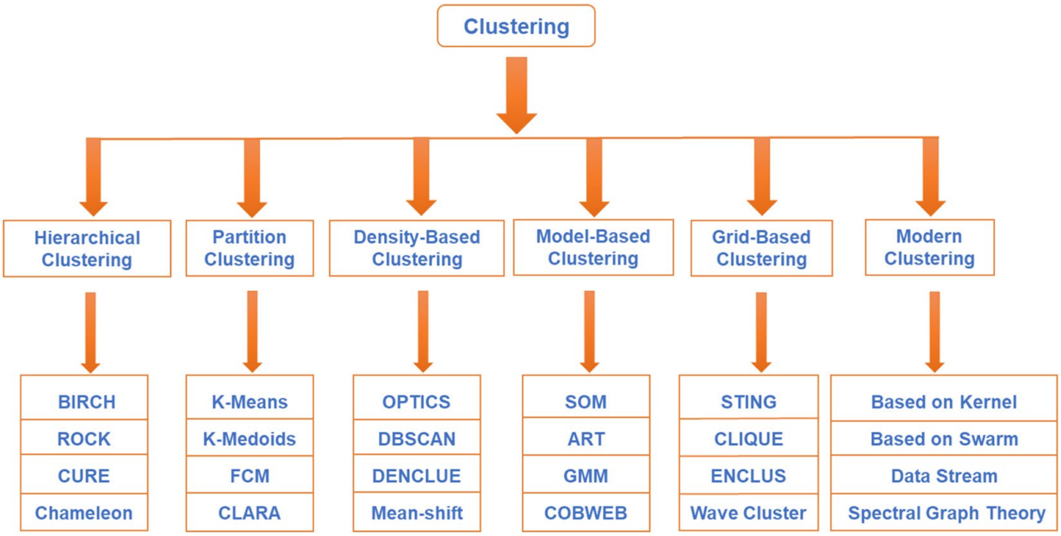

Fig. 1 Classification of clustering algorithms

also call it the maximum method or the furthest neighbor method. The same clusters are chosen by comparing the furthest samples of the two clusters. (3) Average Linkage—it is also known as the minimum variance method. All samples from the clusters need to be compared and chosen. It is the most used and best method because of its good monotony and a moderate degree of space expansion or concentration.

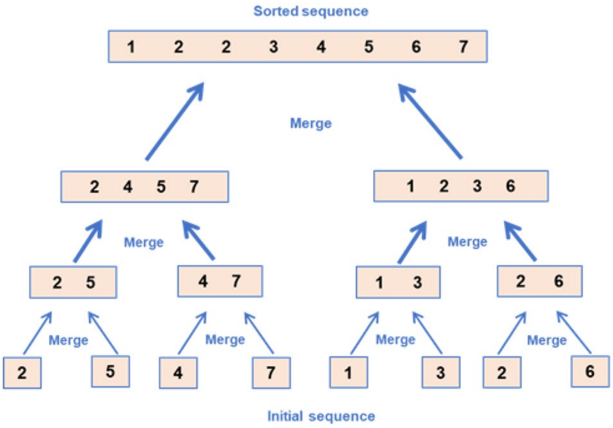

Fig. 2 Flowchart of hierarchical clustering [33]

Advantages: (1) Distance and rule similarity are easy to define, and it has fewer restrictions. (2) It does not need to preset cluster numbers. (3) The hierarchical relationship of the cluster can be explored. (4) It can be clustered into other shapes.

Disadvantages: (1) The complexity of the calculation is too high. (2) Singular values can also have a bad impact. (3) They cannot move in other clusters in a hierarchy when two samples have been linked.

Several typical hierarchical clustering algorithms are CURE (Clustering Using Representative), ROCK (Robust clustering links), BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies), and Chameleon. A detailed comparison of the advantages and disadvantages of the four algorithms is summarized in Table 1 [16, 30, 31, 34], where n stands for the number of total objects/data points, k stands for the number of clusters, s stands for the number of sample

Table 1 Comparison of hierarchical clustering algorithms

Algorithm	Advantages	Disadvantages	Time complexity	Dataset
BIRCH	Good robust for outliers	Limit the number of CF	Low O(n)	Large-scale Low dimension
CURE	Able to identify non-spherical categories	Insensitive to outliers	High O(n^2logn)	Small-scale High dimension
ROCK	For a variety of types of data	The similarity threshold needs to be given in advance	High O(n^2logn)	Large-scale High dimension
Chameleon	Capture the neighborhood dynamically	Many parameters need to be set in advance	High O(n^2)	Small-scale Low dimension

objects/data points, and t stands for the number of iterations. The same variables are used for the other tables.

(1) BIRCH was proposed by Zhang et al. in 1996 [35]. At first, it scans the database and creates a Clustering Feature-tree CF-tree [36], then the leaf nodes of the CF-tree are clustered by a selected clustering algorithm. The goal is to delete the sparse clusters as outliers and merge the denser clusters into larger clusters.

(2) CURE combines two methods of random sampling and partitioning [37]. At first, the scattered objects in a cluster are selected, and then, these can shrink or be moved based on a specific score or shrink factor. At each step of the algorithm, two clusters with a pair of nearest points (each point comes from a different cluster) are merged [38].

(3) ROCK is developed based on the CURE [39]. This algorithm takes a more global view and introduces the concept of "link." The relationship between two objects (sample points/clusters) is determined by taking the number of their common neighbors (similar sample points) into account, and no distance function is required [40].

(4) Chameleon is a multi-stage hierarchical clustering algorithm proposed by Karypis et al. in 2001 [41]. It uses dynamic modeling to determine the similarity between a pair of clusters. Then, clusters can be merged automatically and adapted into a variety of strange data shapes [42]. Many researchers regard it as one of the best clustering algorithms.

2.2 Partition Clustering

Partition clustering is the most common and basic clustering method whose basic principle is to take the center of the data points as the centroid of the corresponding clustering. For a data set containing multiple samples, the partitional algorithm is used to divide the samples into mutually exclusive partitions, and each partition represents a cluster [43, 44]. Clustering results in this classification need to meet two criteria: 1) each partition must contain at least one sample and 2) each sample belongs to only one partition.

Advantages: (1) The principle and the implementation are easy. (2) The convergence speed is high.

Disadvantages: (1) The results are only locally optimal because of iterative methods. (2) When the dataset is very large, the algorithm process needs a long time. (3) If the cluster contains abnormal points, there will be a serious deviation from the mean. (4) It is difficult to converge for data sets that are not convex [30, 43].

Several typical partition clustering algorithms are given below, such as K-means, K-medoids, FCM (Fuzzy C-means), and CLARA (Clustering Large Applications). In addition, PAM, CLARANS, and ISODATA are all improvements based on the previous algorithms, so no more details are presented in this section. A detailed comparison of the advantages and disadvantages of these four algorithms is given in Table 2 [16, 30, 31, 43].

(1) K-means. The term "K-mean" was first used by MacQueen et al. in 1967 [45]. It can find k different clusters, and the center of each cluster is calculated by using the mean of the values contained in the cluster. The number of clusters k is chosen by users, and each cluster is described by its centroid, which is the center of all points in the cluster [29, 46]. The K-means algorithm is the most used clustering method in decades. The steps are as follows:

Step 1. Choose the initial centroid: The initial centroids and the number of clusters k are selected.

Step 2. Sample clustering: The distance between each sample and each centroid is calculated, and then, put the sample into the cluster of the nearest centroid.

Step 3. Recalculate the centroid: The property value of the new centroid for each cluster is equal to the average property value for all samples in this cluster.

Step 4. End criteria: It stops when the number of loops is greater than the maximum number of loops or the maximum

Table 2 Comparison of partition clustering algorithms

Algorithm	Advantages	Disadvantages	Time complexity	Dataset
K-means	Simple principle and good effect	Sensitive to noise, local optimal	Low O(n)	Large-scale Low dimension
K-medoids	Good robustness for noise	The distance among points must be calculated	High O(kn^2)	Small-scale Low dimension
FCM	Be closer to global optimization	The membership parameter m needs to be chosen, sensitive to noise	Low O(n)	Large-scale Low dimension
CLARA	Larger datasets	Local optimal	Middle O($k(n-k-1)(n-k)$)	Large-scale Low dimension

value in the sequence composed of distances with all centroid changes is less than the maximum distance with all centroid changes.

Step5. If Step 4 does not end, then Step 2 and Step 3 are repeated. If Step 4 ends, the clusters and centroids are printed (or drawn).

(2) K-medoids. K-means and K-medoids algorithms are very similar. The only difference is that the K-medoids use the sample as the reference point, but K-means method uses the mean value. The centroids of the former can be any value in the continuous space, while the centroids of the latter can only be the sample in the data samples [47]. The reason is that K-means has high requirements for data samples and requires all data samples to be in a European space, which may cause errors for data with a lot of noise. For non-numerical data samples, real variables such as the mean value cannot be calculated [48]. Therefore, it can be said that K-medoids is an improvement of K-means. The difference between the two algorithms can be seen in Fig. 3 [49, 50].

(3) FCM. In 1973, Bezdek et al. proposed the algorithm as an improvement on the early hard C- means clustering (HCM) method [51]. FCM is a fuzzy clustering algorithm based on the objective function that uses membership degree to make the value of each given data point between 0 and 1. And the sum of membership degrees of all data must be 1. The advantage of introducing the concept of a fuzzy algorithm is that each input vector is no longer subordinate to a particular cluster, which may belong to several clusters. Then, clusters are partitioned according to the degree of similarity [52].

(4) CLARA. This algorithm is developed based on K-means and PAM algorithms [53]. PAM is the same as K-means, but the update process is achieved by replacing the centroid with each point of the cluster in Step 3. CLARA makes random sampling in the big dataset and then uses the PAM algorithm for each sample [54]. Finally, among the optimal center points of each sample clustering, CLARA looks for a clustering center with the lowest cost as the optimal clustering of the current big dataset.

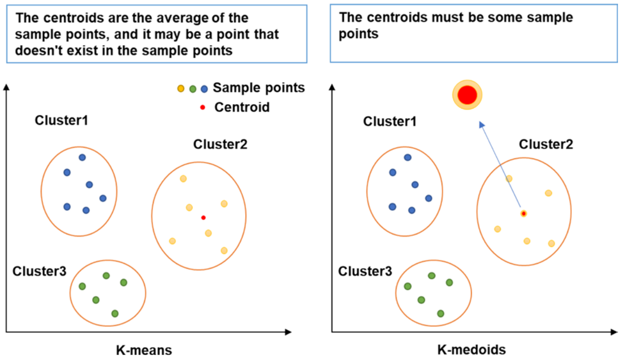
The centroids are the average of the sample points, and it may be a point that doesn't exist in the sample points
Sample points
Centroid
Cluster1
Cluster2
Cluster3
K-means
The centroids must be some sample points
Cluster1
Cluster2
Cluster3
K-medoids

Fig. 3 Comparison between the K-means and K-medoids

2.3 Density-Based Clustering

One fundamental difference between density-based clustering and others is that it is not based on a variety of distances, but on density [27]. The guiding idea of this method is that a cluster is defined as the largest collection of points with similar density. The cluster keeps growing as long as the density of points in the dataset is greater than a certain threshold value, and the outside points with lower density are recognized as noise points. This means that arbitrary shape clustering that can be found in the data with “noise,” convex, concave, and polygonal datasets can all be clustered [55].

Advantages: (1) The algorithm can get non-spherical clustering results and represent the data distribution well. (2) The complexity level of the density-based algorithm is lower than the K-means algorithm. (3) It only considers the distance among points without the need of mapping points to a vector space.

Disadvantages: (1) It needs to calculate the distance among all points in advance. (2) If the volume of sample data is too large, a spatial index needs to be established and requires a large memory space of the whole distance matrix.

Several typical density-based clustering algorithms are given as follows, such as OPTICS (Ordering Points to Identify the Clustering Structure), DBSCAN (Density-Based Spatial Clustering of Applications with Noise), DENCLUE (Density Clustering), and Mean-shift. A detailed comparison of the advantages and disadvantages of these four algorithms is given in Table 3 [18, 29, 31, 55].

(1) DBSCAN was first proposed by Ester et al. in 1996 [56]. Two new parameters are introduced in DBSCAN, i.e., eps—the radius of the neighborhood around a point, minPts—the number of points contained at least in the neighborhood. Then, according to these two parameters, the data points are divided into three categories: (1) Core point: The radius eps of an object is given and the number of samples in the neighborhood exceeds the threshold minPts. (2) Border point: The number of points within the eps is less than minPts, but the point still falls in the neighborhood of the core point. (3) Outliers: A sample that is neither a core point nor a border point [54, 57].

Table 3 Comparison of density-based clustering algorithms

Algorithm	Advantages	Disadvantages	Time complexity	Data set
DBSCAN	Good robustness to outliers and can identify clusters of arbitrary shape	Require a lot of memory and I/O space	Middle O(*n*logn)	Large-scale Low dimension
OPTICS	Overcome DBSCAN shortcomings	An ordered list needs constant maintenance	Middle O(*n*logn)	Large-scale Low dimension
DENCLUE	Fast, good robustness to outliers	A large number of parameters affects the results	Low O(logn)	Large-scale High dimension
Mean-shift	Good robustness to outliers	The width of the window function cannot be changed	High O(kernel)	Small-scale Low dimension

Figure 4 illustrates the process of the DBSCAN clustering algorithm [31, 58, 59], where A is a randomly selected point in the sample points. First, set the radius and the minimum number of sample points contained within the circle and draw a circle around A. Then, if there are enough sample points within the circle, the center of the circle is transferred to these inner sample points, such as B, C, M, and N. At last, the circles have been drawn repeatedly until the number of circled sample points is less than the minPts.

(2) OPTICS was first proposed by M.Ankerst et al. in 1999 [60], and it was an improvement of the DBSCAN algorithm. It has overcome the shortcoming of the DBSCAN algorithm, which is sensitive to neighborhood radius and neighborhood minimum number of points. This algorithm does not directly display clusters after grouping, but only sorts the objects in the dataset to get an ordered list. Then, a decision graph can be obtained, through which datasets with different eps parameters can be processed centrally [61].

(3) DENCLUE. Density Clustering [62] uses a total density function to describe the distribution of data. The contribution of each point to the total density function is represented by an “Influence Function” or “Kernel Function,” and the total density value of a point in the data space is the sum of the influence density functions of each point associated with the point. DENCLUE uses a local density function that considers only the data points, which contribute to the overall density function [62]. There are local peaks (local density maximum) and local valleys (local density minimum), in which each peak corresponds to a cluster centroid, and the clusters are separated from one another by valleys.

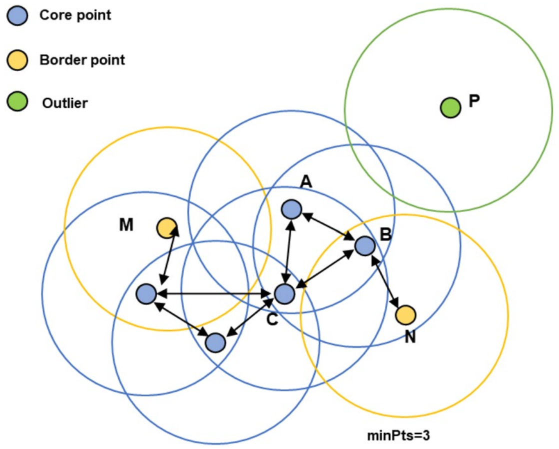

Fig. 4 Principle of DBSCAN algorithm

(4) Mean-shift. Kernel function and weight coefficient have been introduced into the Mean-Shift algorithm [30]. At first, the algorithm assumes that the dataset of different clusters has different probability density distributions, then needs to find the direction where the density of any sample points grows fastest. These sample points eventually converge to the maximum value of local density, and the same local density points are considered to be members of the same cluster [63]. The most important step is to calculate the shift mean of each point and then update the location of the points according to the calculated shift mean.

2.4 Model-Based Clustering

Model-based clustering uses mathematical models to group the given data by solving optimization problems. The basic idea is to select an assumption model for each cluster and find the cluster that fits the assumption model best. There are mainly two kinds of model-based clustering algorithms, one is based on statistical learning and the other is based on neural network learning [30, 64].

The algorithms based on statistical learning use probability distribution of attributes to describe the clustering. In other words, the data set is generated through some statistical process and described by using the best-fit statistical model of the samples. The most typical example is the GMM (Gaussian Mixture Model) and COBWEB.

The algorithms based on neural network learning use a neuron to represent a cluster, and the input data are

Table 4 Comparison of model-based clustering algorithms

Algorithm	Advantages	Disadvantages	Time complexity	Data set
GMM	Understand easily, fast	Not suitable for non-convex data sets	High O($kt*n^2$)	Small-scale Low dimension
COBWEB	The number of clusters is updated automatically	Assumption probability distributions are some- times not true	Low (distribution)	Large-scale Low dimension
SOM	Identify hidden patterns in data easily	No target functions for comparing	High (layer)	Small-scale High dimension
ART	Better flexibility and stability for different input modes	Too much variability for time complexity	Middle (type + layer)	Large-scale Low dimension

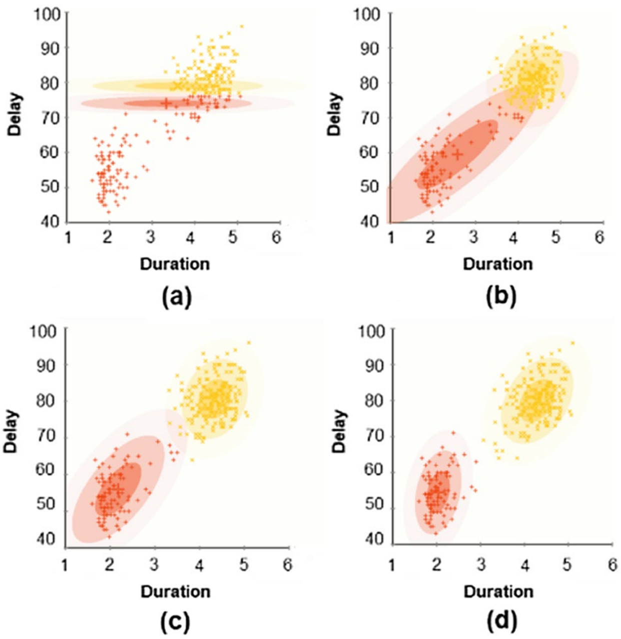

Fig. 5 Process of GMM

represented by neurons that are connected to the prototype neurons. Each connection has a weighting coefficient that is randomly initialized prior to the model learning. Two popular neural clustering algorithms are SOM (self-organizing mapping) and ART (Adaptive Resonance Theory). A detailed comparison of the advantages and disadvantages of the above-mentioned four algorithms (i.e., GMM, COBWEB, SOM, and ART) is given in Table 4 [30, 31].

Advantages: (1) Considering noise and outliers, the number of clusters can be determined automatically. (2) They have high accuracy in various clustering methods.

Disadvantages: (1) They have a poor clustering effect on large data sets and high-dimensional data. (2) They are not suitable for complex Web data.

The four algorithms are described in detail as follows:

(1) GMM. It is similar to K-means clustering. The idea is to find a mixed representation of the probability distribution of the multi-dimensional Gaussian model, then fit the data distribution of any shape by increasing the number of models. The whole process can be seen visually in Fig. 5 [65, 66]. Each GMM consists of K Gaussian distributions, each of which is called a "component" and these components are linearly added together to form the probability density function of the GMM.

(2) COBWEB. The core idea of COBWEB is to build a classification tree, based on some heuristic criteria, to realize hierarchical clustering on the assumption that the probability distribution of each attribute is independent [30, 67]. There are many conditional probability models of classes based on feature space partition. What we need is a decision tree with less contradiction with training data and good generalization ability. Decision tree learning expresses this objective in terms of a loss function that is usually a regularized maximum likelihood function.

(3) SOM. The core idea of SOM is to establish a mapping of dimension reduction from a high-dimensional space to a two-dimensional or three-dimensional feature space on the assumption that there is a topology in the input data [68]. In essence, it is a neural network with only the input layer and the competitive layer. The neurons in the input layer and neurons in the competition layer relate to each other by a weight coefficient. Every node in the competing layer represents a cluster. The training process is done in a "competitive learning" manner, where each input sample point finds a node in the competition layer that matches it best, which is called its activation node and is also known as the "winning neuron." Then, the parameters of the activated nodes are updated by stochastic gradient descent. At the same time, the points adjacent to the active node also update the parameters appropriately according to their distance from the active node. SOM network structure is shown in Fig. 6 [69, 70].

(4) ART is an incremental algorithm whose core idea is to dynamically generate a new neuron to match a new pattern and thus create a new cluster in the case of an insuf-

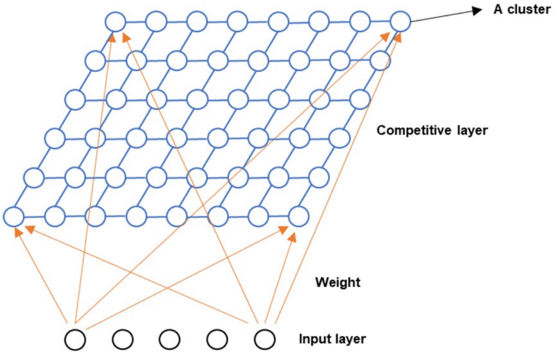

Fig. 6 SOM network structure

ficient number of neurons at present. ART algorithm has three forms: ARTI [71] can process bipolar or binary signals; ART II [72] is an extended form of ART for processing continuous analog signals; ART III [73] is a hierarchical search model, which is compatible with the functions of the neural network and expands the two-layer neural network to multilayer neural network.

2.5 Grid-Based Clustering

This method uses the idea of finite element analysis to divide the dimension into a finite number of cells that form a grid structure. And all clustering operations are performed on this grid structure [29]. The basic process of the grid-based clustering algorithm is as follows: First, the data space, W, is divided into grid cells. Then, the data object set, O, is mapped into grid cells, and the density of each cell is calculated. Finally, according to the density threshold, minPts, input by the user, whether each grid cell is a high-density cell is determined, and clusters are formed by neighboring dense cell groups.

Depending on the strategy of searching subspace, the clustering is mainly based on bottom-up meshing algorithms and top-down meshing algorithms [74].

The bottom-up meshing method is based on the division of user input parameters, and the data space is evenly divided into equal parts according to the size of the grid cell. If all the data points fall into the same grid cell that belongs to the same cluster, each grid cell holds statistics that fall into the data within it. This grid cell containing a certain number of data points is called a high-density grid cell. Wave Cluster, CLIQUE, and ENCLUS are representative algorithms that use the bottom-up meshing method.

The top-down meshing method adopts the divide and conquers principle to recursively divide the data space to continuously reduce the scale of the problem. First, the original data space is divided into several larger regions, where the partitioning process is repeated until each region contains data points belonging to the same cluster. Then, these regions are the final grid cells. This method directly identifies the high-density grid cells or the connected high-density grid cells as a cluster. However, the bottom-up meshing method requires only one linear scan of the data set and describes high-accuracy clusters [75]. The representative algorithm is STING.

Advantages: (1) It does not need to calculate distance and its processing time is independent of the number of data objects, but only relies on the number of dimension units in the quantitative space. (2) It can determine which clusters are adjacent easily. (3) It has high scalability and is suitable for parallel processing and increment updating.

Disadvantages: (1) Only clusters with horizontal or vertical boundaries can be found, but clusters with oblique boundaries cannot be detected. (2) When processing high-dimensional data, the number of grid cells will increase exponentially with the increase in attribute dimension. (3) Input parameters have a great influence on clustering results and are difficult to choose. (4) When there is noise in the data, the clustering result of the algorithm is poor if there is no special treatment.

Several typical grid-based clustering algorithms are given below, mainly STING (Statistical Information Grid), CLIQUE (Clustering in Quest), ENCLUS, and Wave Cluster. A detailed comparison of these four algorithms is given in Table 5 [30, 31]:

Table 5 Comparison of grid-based clustering algorithms

Algorithm	Advantages	Disadvantages	Time complexity	Data set
STING	The calculation is independent of the search	Boundaries can only be horizontal or vertical	Low O(n)	Large-scale High dimension
CLIQUE	Combine the advantages of grid-based and density-based clustering	The best value for all subspaces cannot be found	Low O($n+k^2$)	Small-scale High dimension
ENCLUS	The method of searching for subspaces is entropy-based	Many parameters, affect each other	Low Similar to CLIQUE	Small-scale High dimension
Wave Cluster	Belongs to undirected clustering	Low accuracy	Low O(n)	Large-scale Low dimension

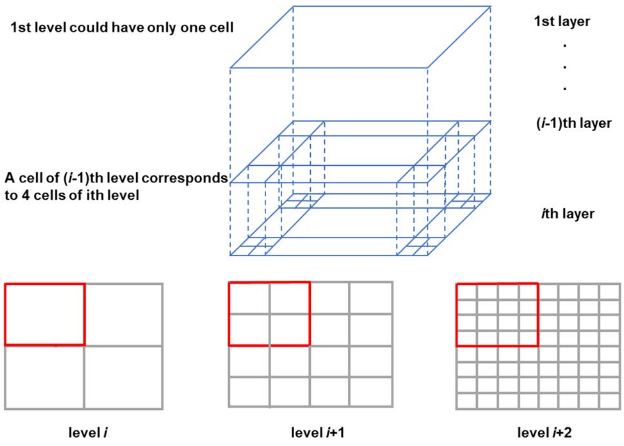
1st level could have only one cell
1st layer
(i-1)th layer
A cell of (i-1)th level corresponds to 4 cells of ith level
ith layer
level i
level i+1
level i+2

Fig. 7 Hierarchical Structure of STING

(1) STING is a grid-based multi-resolution clustering technique [76], which divides the spatial region of the input object into rectangular units, and the space can be divided by hierarchical and recursive methods. Each cell of a multilayer rectangular corresponds to different resolutions and forms a hierarchy. Statistics about the attributes of each grid cell (such as mean) are calculated and stored as statistical parameters in advance, and data points within a grid are a cluster. The hierarchical structure is shown in Fig. 7 [31].

(2) CLIQUE was developed by Agrawal et al. in 1998 [77]. It divides each dimension into nonoverlapping partitions, and the whole embedded space is divided into units of data objects. Then, it uses a density threshold to identify dense units and finds the largest area that covers the cell, and the remaining dense cells that have not yet been covered are processed until all dense cells are covered. The algorithm has two core parameters: The mesh step size determines the partition of the space, and the density threshold is used to define the dense mesh [78].

(3) In ENCLUS, a technique is used to find clustering subspaces: According to the specified entropy value, an effective subspace is found from the bottom to the top (starting from one dimension) [79]. Based on the search for effective subspace technology of the CLIQUE algorithm, a method of searching for effective subspace based on entropy is proposed, and its entropy value is calculated for each subspace. If the value is smaller than the specified entropy value, the unit is effective.

(4) Wave Cluster is a multi-resolution clustering algorithm that aggregates data by imposing a multi-dimensional grid structure on the data space and then uses a wavelet transform to upgrade the original feature space and find the dense region in the transformed space. Wavelet transform is a signal processing technique that decomposes a signal into sub-bands of different frequencies. The data are transformed to preserve the relative distance among objects at different resolution levels. Clustering can be determined by seeking high-density regions in new space.

2.6 Modern Clustering

With the progress of the times and the development of science and technology, there are no longer limitations to using a single method of clustering in many cases. Researchers often combine several algorithms, including some methods of neural network and deep learning (DL), to get more comprehensive modern clustering technologies. The standard unsupervised learning method (such as clustering) requires selecting the relevant features of objects manually, while deep learning can automatically extract the relevant features. In addition, deep learning is end-to-end learning in which the network is given tasks such as raw data and classification and can be done automatically. But deep learning is often complex and requires high-performance GPUs and large amounts of labeled data. Therefore, combining DL with clustering algorithms can reinforce complementary advantages.

This paper mainly introduces four kinds of modern clustering which are easy to understand and widely applied, including those based on kernel, based on swarm intelligence, based on spectral graph theory, and data stream.

Table 6 Comparison of four modern clustering algorithms

Algorithm	Time complexity	Scalability	Dataset
SM	High (eigenvector + heuristics)	Middle	Small-scale; High dimension
NJW	High (eigenvector)	Middle	Small-scale; High dimension
SVC	High (kernel)	Low	Small-scale; Low dimension
Kernel K-means	High (kernel)	Middle	Small-scale; Low dimension
PSO-based	High (iterations)	Low	Small-scale; Low dimension
ACO-based	High (iterations)	Low	Small-scale; Low dimension
STREAM	Low O(*kn*)	Middle	Large-scale; Low dimension
CluStream	Low (Online + Offline)	High	Large-scale; Low dimension

Table 6 [30, 80–82] presents the specific comparison of four common modern clustering methods:

(1) Clustering Algorithm Based on Kernel. The kernel function is introduced into clustering, and a new clustering target function on the basis of the classical clustering algorithm can be obtained [22]. Then, by mapping the data of the input space to the high-dimensional space, the high-dimensional space is clustered linearly. In this way, the data are mapped to increase the difference of data and expand the linear division of data. The typical algorithms of this kind of clustering include kernel K-means [83], kernel FCM [84], SVC [85], and MMC [86]. The first two algorithms add kernel function to the original clustering method. The core idea of SVC (Support Vector Clustering) is to use kernel functions to map data points from data space in high-dimensional feature space. In feature space looking for the smallest sphere that encloses all data points, it forms a contour that encloses the data points. These contours are interpreted as cluster boundaries. MMC (Maximum Margin Clustering) tries to find the hyperplane with the maximum margin to cluster and it can be promoted for the multi-label clustering problem.

Advantages: (1) It is suitable for high-dimensional data. (2) It can analyze noise and separate overlapping clusters. (3) It is suitable for any shape of clustering [30, 80].

Disadvantages: Because of the existence of kernel functions, it is not suitable for large-scale data sets and has high time complexity.

(2) Clustering Algorithm Based on Swarm Intelligence. The method is to combine the idea of the biological population-changing process in swarm intelligence and clustering, the most famous of which is ACO-based [87] and PSO-based [88]. The main idea of ACO (Ant Colony Optimization) is to build a model for searching for a minimum cost path in a graph. Therefore, in the clustering process, data are distributed on the two-dimensional grid randomly, and then, further operation can be conducted based on the decision of an ant to select data or not. At last, the process can be iterated to obtain satisfactory clustering results. PSO(Particle Swarm Optimization)-based clustering is suited for processing data points, which can be regarded as a particle. First, another clustering algorithm is used to get the initial cluster of particles. Then, based on the center, the cluster, the position, and the speed of each particle are constantly updated until satisfactory clustering results are obtained.

Advantages: It is equipped with a great ability to cope with local optimal by maintaining, recombining, and comparing several candidate solutions simultaneously [82].

Disadvantages: Algorithms are too complex to be used for high-dimension and large-scale data sets because of the high time complexity.

(3) Clustering Algorithm Based on Spectral Graph Theory. The basic idea of this kind of clustering algorithm is to regard the sample points as the vertex and the similarity among the data as the weighted edge. Then, through a graph division method that can make the connection weight among different groups as small as possible and the connection weight among the edges in the same group as large as possible, in order to turn the clustering problem into the division problem of the graph [89].

The success of this method is mainly that there is no high requirement on the form of the clusters. In addition, as long as making sure that a similar graph is sparse, they can effectively implement spectral clustering. But choosing a good similarity graph is not so easy, and it is unstable when using different parameters. Therefore, this method cannot be used as a "black box algorithm," so it is not possible to automatically detect the right cluster of any given dataset. There are recursive spectral and multipath spectral, and the typical algorithms of those two categories are SM [90] and NJW [91], respectively. The two algorithms are clustered in feature space by processing feature vectors in different ways.

Advantages: (1) It can get the global optimal solution. (2) It only needs a similar matrix as input. (3) It is suitable for high-dimensional arbitrary shape data.

Disadvantages: (1) It needs to set a large number of parameters in advance. (2) It has high time complexity.

(4) Clustering Algorithm for Data Stream. Data stream shares the characteristics of arriving based on sequence, large in scale, and limited frequency of reading [30]. In data stream environment, since large volumes of data arrive in a stream and these data points unfold with time, most of the conventional clustering algorithms are not sufficiently efficient. Therefore, some new types of data stream clustering algorithms have emerged. And STREAM [81], CluStream [92], and D-Stream [23] are three representative algorithms. STREAM organizes ordered data into a hierarchical clustering structure. CluStream is an incremental algorithm. Dynamically changing data can be processed online or offline through it. D-Stream is a framework for clustering stream data using a density-based approach. The algorithm uses an online component that maps each input data record into a grid and an offline component which computes the grid density and clusters the grids based on the density. The algorithm adopts a density

Table 7 Comparison of six types of clustering algorithms

Clustering algorithm	Applicable scene
Hierarchical clustering	The hierarchical relationship of the cluster can be explored, and distance and similarity are easy to define. But they cannot move in other clusters in a hierarchy when two samples have been linked, and singular values can also have a bad impact
Partition clustering	The principle and the implementation are easy and the convergence speed is high. But results are only locally optimal because of iterative methods, and it is more applicable to small and converged datasets
Density-based clustering	The algorithm can get non-spherical clustering results and the complexity of the algorithm is lower than others. But it needs to calculate the distance among all points in advance
Model-based clustering	It has high accuracy and the number of clusters can be determined automatically. But it has a poor clustering effect on large data sets, high-dimensional data, and complex Web data
Grid-based clustering	It does not need to calculate distance and its processing speed is fast. It also has high scalability and is suitable for parallel processing and increment updating. But it needs to set input parameters in advance, and only clusters with horizontal or vertical boundaries can be found, but clusters with oblique boundaries cannot be detected
Modern clustering	It is more suitable for high-dimensional data than traditional clustering algorithms. It can get the global optimal solution and generate and adjust clusters automatically in real time. But it needs to set a large number of parameters in advance and has high time complexity

decaying technique to capture the dynamic changes of a data stream.

Advantages: (1) It can generate and adjust clusters efficiently in real time. (2) The system's efficiency of space and time can be improved. (3) It can process data streams.

Disadvantages: It has a low clustering effect for time-varying data of mixed types.

The comparison of six types of clustering algorithms is shown in Table 7, it is a summary of Sect. 2.

3 Similarity Measures

The similarity of objects within a cluster is critical in the clustering process. A good cluster finds the greatest similarity among its objects [93]. The measure of similarity in the cluster is mainly decided by the distance among its members. Similar data points are clustered into the same cluster in two-dimensional or three-dimensional space, while different or distant data points are placed in different clusters. In general, the similarity between objects a and b is $Sim(a, b)$, and the measured distance of objects a and b is $d(a, b)$, so the similarity is generally obtained by adopting $Sim(a, b) = 1/(1 + d(a, b))$. A valid distance measure should be symmetric, i.e., $d(a, b) = d(b, a)$ and obtain its minimum value (ideally zero) in the case of identical vectors. Typical distance calculation methods include [29, 31, 93–95]:

3.1 Minkowski Distance

Minkowski distance is a very common method to measure the distance among numerical points [96, 97]. Assuming that the coordinates of numerical points P and Q are as follows: $P = (x_{i1}, x_{i2}, \cdots, x_{ik})$ and $Q = (x_{j1}, x_{j2}, \cdots, x_{jk}) \in R^n$. Then, Minkowski distance is defined as

$$d_{\min} = \left(\sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right|^p \right)^{1/p} \tag{1}$$

where $p \geq 1$, x_{ik} is the value of the kth variable for the entity i, and x_{jk} is the value of the kth variable for the entity j. For $p=1$, it becomes Manhattan distance, and for $p=2$, it becomes Euclidean distance. As p approaches infinity, Minkowski distance is transformed into Chebyshev distance.

Minkowski distance is intuitive and performs well when data clusters are isolated or compressed. However, it is independent of the distribution of data and has certain limitations [93]. If the value of the x direction is much larger than that of the y direction, this distance formula will over-amplify the effect in the x dimension. Therefore, the normalization of continuous features is the way to solve this problem.

3.2 Manhattan Distance

This is similar to walking on a street in a city, which only follows one of the x and y axes at a time and cannot walk diagonally between two points. This measure is defined as follows [29, 98]:

$$d_{\text{man}} = \sum_{k=1}^{n} \left| x_{ik} - x_{jk} \right| \tag{2}$$

3.3 Euclidean Distance

Euclidean distance [99] can be simply described as the geometric distance between any two points in multi-dimensional space. Usually, original data are used instead of normalized data. For example, if an attribute has a value within 1–100, it can be used directly instead of being normalized to the interval of [0,1]. In this way, the original meaning of Euclidean distance is eliminated, so the advantage is that a new object does not affect the distance between any two objects. However, if the measurement criteria of the object attributes are different, such as using a scale of 10 and a scale of 100 when measuring fractions, the results may be greatly affected [100]. This distance is defined as follows:

$$d_{\text{euc}} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{3}$$

3.4 Chebyshev Distance

Chebyshev distance [29, 101] is mainly regarded as the minimum distance between two objects in a multi-dimensional space, which can be simply described as determining which cluster an object belongs to with a one-dimensional attribute. It is usually used for a specific case, such as an object moves to another object in a coordinate system. Chebyshev distance is calculated as follows:

$$d_{\text{che}} = \max(\left|x_{ik} - x_{jk}\right|) \tag{4}$$

3.5 Cosine Similarity

Cosine similarity can be simply described as the size of the angle among the vectors formed by the attributes of two objects in space. Cosine similarity is applicable to sparse datasets [102]. For example, in document clustering, the document is usually composed of many zeros, which makes the data set sparse. Cosine similarity is very suitable for judging whether different documents are of the same cluster. The normalized inner product for the Cosine measure is defined as

$$d_{\cos} = \frac{x_i^{\mathrm{T}} \cdot x_j}{\|x_i\| \cdot \left\|x_j\right\|} \tag{5}$$

3.6 Jaccard Similarity Coefficient

In data mining, attribute values are often binarized. By calculating the Jaccard similarity, the similarity of two objects can be obtained simply and quickly [32, 103]. However, for different attributes, the degree of binary type is not the same. It measures the similarity as the ratio of the intersection and the union of two sets (say S_1 and S_2):

$$J(S_1, S_2) = \frac{\left|S_1 \cap S_2\right|}{\left|S_1 \cup S_2\right|} \tag{6}$$

3.7 Mahalanobis Distance

This is the covariance distance of the data. It differs from the Euclidean distance in that it considers the relationship among attributes [29, 93]. Such as gender information is accompanied by height information, because the two pieces of information have a certain degree of correlation, and the measurement scale is independent. The calculation formula is as follows:

$$d_{\text{mah}} = \sqrt{(x_i - x_j)^{\mathrm{T}} S^{-1} (x_i - x_j)} \tag{7}$$

3.8 Pearson Correlation

The correlation coefficient was first discovered by Bravais [104] and later shown by Pearson [105]. It is used to indicate the degree to which different objects deviate from the center line of fitting. First, many objects' attributes are fitted into a straight line or curve, and then, the deviation degree of each object's attributes relative to this line is calculated. The disadvantage of Pearson correlation is that it is sensitive to outliers. The normalized Pearson correlation for two vectors $\bar{x}_i$ and $\bar{x}_j$ is defined as

$$d_{\text{pea}} = \frac{(x_i - \bar{x}_i)^2 \cdot (x_j - \bar{x}_j)^2}{\|x_i - \bar{x}\| \cdot \left\|x_j - \bar{x}_j\right\|} \tag{8}$$

where $\bar{x}_i$ denotes the average feature value of x in all dimensions.

3.9 Dice Coefficient

It was first proposed by Dice [106]. The Dice coefficient measure is similar to the extended Jaccard measure, which is used to measure the similarity of two sets. Since a string can be understood as a set, the Dice distance can also be used to measure the similarity of strings [107]. It is defined as

$$d_{\text{dic}} = \frac{2x_i^{\mathrm{T}} \cdot x_j}{\|x_i\|^2 + \left\|x_j\right\|^2} \tag{9}$$

3.10 Hamming Distance

The above measures are suitable for the data samples being clustered into the static numeric variable, but for the sequences of characters, the similarity is used to measure the following two metrics. Hamming first introduced this concept in the basic paper on error detection and correction code [108, 109]. It is used to measure the similarity between two binary codes.

Hamming distance represents the number of different characters in the corresponding position of two (same length) strings. For example, the Hamming distance between 1,101,101 and 1,001,001 is 2.

3.11 Levenshtein Distance

For equal-length strings, which can be encoded in each letter by one-hot, the Mahalanobis distance or European distance can be used to measure the similarity. However, for the similarity of two non-equal-length strings or arrays, they need to be calculated by edit distance, which is often called the Levenshtein distance [110]. It is defined as the minimum number of edits required for one of two strings to be converted to another [111]. The edit operation includes replacing a character with another character, inserting a character, or deleting a character. In general, the smaller the edit distance, the greater the similarity between the two strings.

The Levenshtein distance equation is defined as follows:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j), \text{if } \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1 \end{cases}, \text{ otherwise} \end{cases} \tag{10}$$

where a and b are two arrays (string), i and j are the array subscript label.

3.12 Dynamic Time Warping

Dynamic Time Warping (DTW) [28] is an algorithm to measure the similarity among time series. In general, this method, by stretching and compressing a time series, allows a computer to find an optimal match method between two given sequences within certain constraints. It has the advantage of allowing a point to map to multiple points in another sequence. Thus, the DTW distance allows for a more intuitive distance measure for time series that have similar shapes but are not aligned in time. Given two time series: $T = \{t_1, t_2, \cdots, t_N\}, N \in R$ and $S = \{s_1, s_2, \cdots, s_M\}, M \in R$. The warping path is denoted as $W = w_1, w_2, \dots, w_k, \dots, w_n$, and $w_l = (p_1, p_2, \cdots p_K), K \in [\max(N,M), N+M-1]$, the elements $p_k = (a_l, b_l) \in [1:N] \times [1:M], k \in K$. Three other conditions should be met [112]:

(1) The boundary conditions:
$p_1 = (1,1), p_K = (N,M)$. This condition constrains the starting and ending points of the path.
(2) The monotonicity condition:
$a_1 \le a_2 \le \cdots \le a_K, b_1 \le b_2 \le \cdots \le b_K$. This condition constrains the chronological order of the points.
(3) Step length conditions:

$p_{k+1} - p_k \in \{(1,0),(0,1),(1,1)\}, k \in [1:K-1]$. This condition limits the jump step size of the algorithm when looking for a warping path.

DWT path and condition restrictions are shown in Fig. 8 [113]. The DWT distance is calculated as follows:

$$d_{\mathrm{DWT}} = \min\left\{\sqrt{\sum_{k=1}^{n} w_k / n}\right\} \tag{11}$$

The comparison of different similarity measures is shown in Table 8, it is a summary of Sect. 3.

4 Evaluation Indicator

With more and more clustering methods, users need to choose a relatively better one, so it is necessary to determine the validity and accuracy of a clustering algorithm. According to the evaluation criteria to judge whether the similarity among the objects in the same cluster is the maximum, the similarity among the objects in a different cluster is the minimum. At present, the validity evaluation indicator of the clustering algorithm can be divided into three categories: internal evaluation indicator, external evaluation indicator, and relative evaluation indicator [29–31, 93]. Among them,

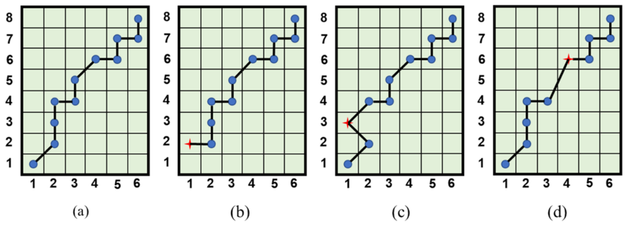

Fig. 8 Graphs of the DTW wrapping path between time series with lengths of 8 and 6: **a** a path that meets all conditions, **b** a path that does not meet the boundary condition, **c** a path that does not meet the monotonicity condition, **d** a path that does not meet the step size condition

Table 8 Comparison of different similarity measures

Similarity measures	Applicable scene
Minkowski distance	It is a very common method to measure the distance among numerical points. As the p value changes, it becomes other measures
Manhattan distance	$p=1$, as p increases, the larger value of the fractional vector will have a greater impact on distance. It is applicable to low-dimensional data sets which have discrete or binary properties
Euclidean distance	$p=2$, it is applicable to low-dimensional data, the advantage is that a new object does not affect the distance between any two objects
Chebyshev distance	$p=3$, it is usually used for a specific case, such as an object moves to another object in a coordinate system
Cosine similarity	It is applicable to sparse datasets, such as document clustering
Jaccard similarity coefficient	It is used to measure the similarity of two objects whose attribute values are binarized
Mahalanobis distance	It is different from the Euclidean distance in that it considers the relationship among attributes
Pearson correlation	The objects are in linear relation and positive distribution
Dice coefficient	It is used to measure the similarity of two sets
Hamming distance	It is used to measure the similarity between two binary codes
Levenshtein distance	It is used to measure the similarity of two not equal-length strings or arrays
Dynamic Time Warping	It is an algorithm to measure the similarity among time series data

the relative evaluation indicator is tested for different parameter settings of the clustering algorithm according to the evaluation criteria, and the optimal parameter settings and clustering mode are finally selected.

4.1 Internal Evaluation Indicator

The internal evaluation indicator uses the attributes of a dataset to evaluate the merits of the clustering algorithm. Clustering quality is evaluated by calculating the overall similarity, the average inter-cluster similarity, or the average intra-cluster similarity. There is the within-cluster sum of squared error (SSE), compactness (CP), separation (SP), Silhouette coefficient, Calinski-Harabasz index (CH), Davies-Bouldin Index (DBI), Dunn validity index (DVI), and so on.

4.1.1 Within-Cluster Sum of Squared Error

For each cluster, the distance between the samples in the cluster and the center point of the cluster is calculated, and then, the sum is obtained [114, 115]. In theory, the smaller the value, the better the validity and accuracy. The limitation of this index is that it only considers the intra-cluster similarity and does not consider the relationship among different clusters.

The formula is as follows:

$$SSE = \sum_{i=1}^{r} \sum_{j=1}^{n_i} (X_{ij} - \overline{X}_i)^2, \overline{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} \tag{12}$$

4.1.2 Compactness

For a single cluster, the average distance between the samples in the cluster and the center point is calculated. Finally, the index can be calculated by taking the average value of all the clusters [116, 117]. It is like SSE, only the similarity intra-cluster is considered. The smaller the value, the better the clustering effect. It is defined as

$$\overline{CP}_i = \frac{1}{|\Omega_i|} \sum_{x_i \in \Omega_i} \|x_i - w_i\|, \overline{CP} = \frac{1}{K} \sum_{k=1}^{K} \overline{CP}_k \tag{13}$$

where Ω_i is the set of instances in cluster i, and w_i is the central point of the cluster i.

4.1.3 Separation

The final value is obtained by calculating the distance between the center points of pairwise clusters [117, 118]. In contrast to the compactness, this index only considers the inter-cluster distance. The larger the value, the better the clustering effect. It is defined as

$$\overline{SP} = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \left\|w_i - w_j\right\|^2 \tag{14}$$

where w denotes the central point of cluster, and k is the number of clusters.

4.1.4 Silhouette Coefficient

For a sample point, the average distance between the sample point and other sample points in the cluster is defined as the cohesion of the cluster, and the average distance between the sample point and all sample points in the nearest clusters is defined as the separation degree of the cluster [119, 120]. Then, the calculation formula of the Silhouette coefficient of the sample is as follows:

$$s = \frac{b - a}{\max(a, b)} \tag{15}$$

where a is the cohesion of the cluster, and b is the separation degree of the cluster.

For the set of all samples, the Silhouette coefficient is the average of the Silhouette coefficient of each sample. The value of this index ranges from -1 to 1. When the degree of separation among clusters is larger than the degree of cohesion, the value of the Silhouette coefficient is approximately 1. Therefore, if the value of this index is close to 1, the clustering effect is good, and if the value is negative, the clustering effect is poor.

4.1.5 Calinski-Harabasz Index

The inter-cluster distance and intra-cluster distance are considered in this index [121]. The intra-cluster distance is represented by the distance between the sample point in the cluster and the center point of the cluster, and the inter-cluster distance is represented by the distance between the sample point and the center point of other clusters. The larger the value of CH, the closer the inter-cluster distance, the farther the intra-cluster distance, and the better the clustering effect [122]. The calculation formula is as follows:

$$CH = \frac{SS_B}{SS_W} \times \frac{N - K}{K - 1} = \frac{\sum_{i=1}^{K} n_i \cdot d(v_i - \overline{v})}{K - 1} \Big/ \frac{\sum_{i=1}^{K} \sum_{j=1}^{n} d(x_j - v_i)}{N - K} \tag{16}$$

where SS_B is the inter-cluster distance, SS_W is the intra-cluster distance, N is the sample size, and K is the number of clusters.

4.1.6 Davies-Bouldin Index

In the DB index proposed by Davies et al. [123], the distance between the sample point and the cluster center point is used to estimate the tightness within the cluster, and the distance among the cluster center points is used to represent the degree of separation among clusters. The farther the inter-cluster distance, the closer the intra-cluster distance, the smaller the value of the DB index, and the better the clustering performance. The disadvantage is the use of Euclidean distance, which has a poor clustering effect on the circular distribution sample points [124]. Definitions are as follows:

$$DB = \frac{1}{K} \sum_{i=1}^{K} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right) \tag{17}$$

where K stands for the number of clusters, c_i is the center of the cluster i, σ_i is the average distance between any data in cluster i and c_i, and d is the distance between c_i and c_j.

4.1.7 Dunn Validity Index

Dunn index [125] is based on the idea of identifying cluster sets that are compact and well separated. It defines the ratio between the minimal inter-cluster distances to the maximal intra-cluster distance. It is computed as

$$DVI = \frac{\min\limits_{0<m \neq n<K} \left\{ \min\limits_{\substack{\forall x_i \in \Omega_m \\ \forall x_j \in \Omega_n}} \left\{ \left\| x_i - x_j \right\| \right\} \right\}}{\max\limits_{0<m<K} \max\limits_{0<n<K} \left\{ \left\| x_i - x_j \right\| \right\}} \tag{18}$$

The farther the inter-cluster distance is, the closer the intra-cluster distance is, the larger the DVI index is, and the better the clustering performance is. The number of clusters with maximal DVI is taken as the optimal number of clusters and indicates that the clusters are compact and well separated [18]. But the disadvantage is that it only applies to discrete datasets.

4.2 External Evaluation Indicator

The external evaluation indicator is evaluated based on a dataset that has a known group label so that the original labeled data can be compared with the clustering output results. The ideal clustering results of external quality assessment indicators are as follows: Data with different labels are aggregated into different clusters, and data with the same labels are aggregated into the same cluster. Typical external evaluation indicators are the Rand index (RI), Jaccard index, F-measure (FM), normalized mutual information (NMI), adjusted mutual information (AMI), Fowlkes–Mallow's index, and so on.

4.2.1 Rand Index

RI is applicable for clustering where the results are divided into two categories [126]. Usually, results are divided into two categories according to whether they are consistent or not, so as to list the 2 × 2 statistics tables. Rand index is calculated according to the table, and the formula is as follows:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{19}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. It reflects the percentage of consistent results in two kinds of clustering results, and its value range is 0–1. The closer it is to 1, the better the clustering effect is. But the false positives and false negatives are equally weighted and this may cause the RI to be used only in certain application scenarios. At present, there is also a new index adjusted Rand index (ARI) [127], which is an improvement of RI.

4.2.2 Jaccard Index

The Jaccard index is considered to identify the equivalency between two datasets. This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets [128, 129]. It is defined as follows:

$$J(A,B) = \left|\frac{A \cap B}{A \cup B}\right| = \frac{TP}{TP + FP + FN} \tag{20}$$

It is similar to RI, $0 \leq J(A,B) \leq 1$, and the closer the Jaccard coefficient is to 1, the more similar the two data sets are.

4.2.3 F-measure

The F-measure is a series of indicators, and the specific F-measure is defined by introducing the parameter β. It is commonly used in information retrieval. The F-measure is also adopted in the classification model evaluation, and the clustering results are converted into classification results for evaluation through external labels [130, 131]. The formula is as follows:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{21}$$

where P is the precision rate and R is the recall rate. When β=0, the recall rate is also 0, and the increase in β value may increase the weight of the recall rate in the final F-measure. It is also an improved index of the Rand index.

4.2.4 Fowlkes–Mallow's Index

The Fowlkes–Mallows index is defined as the geometric mean of the pairwise precision rate and recall rate [132, 133]:

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}} \tag{22}$$

Its value range is also 0 to 1, and the closer it is to 1, the better the clustering effect is, and it does not need any assumptions about the clusters in advance.

4.2.5 Normalized Mutual Information

The mutual information between two variables is used to measure the correlation between two pieces of information. If the two variables are completely independent, the mutual information is zero [134]. Therefore, NMI is often used to measure the level of fit between two clustering results π_K and π_L. The formula is as follows:

$$NMI(\pi_K, \pi_L) = \frac{\sum\limits_{k=1}^{K}\sum\limits_{l=1}^{L} n_l^k \log(\frac{nn_l^k}{n^k n^l})}{\sqrt{(\sum\limits_{k=1}^{K} n^k \log(\frac{n^k}{n}))(\sum\limits_{l=1}^{L} n^l \log(\frac{n^l}{n}))}} \tag{23}$$

where n^k and n^l are the number of samples corresponding to the kth cluster obtained by the clustering algorithm and the lth cluster in the actual classification, respectively. n_l^k is the common sample points owned in the cluster obtained by the clustering algorithm and the actual classification. NMI can also be expressed by the following equation:

$$MI(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} p_{ij} \log(\frac{p_{ij}}{p_i \cdot p_j}) \tag{24}$$

$$NMI = \frac{2MI(U,V)}{H(U) + H(V)} \tag{25}$$

and

$$p_{ij} = \frac{|U_i \cap V_j|}{N}, pi = \frac{|U_i|}{N}, pj = \frac{|V_j|}{N} \tag{26}$$

$$H(U) = -\sum_{i=1}^{R} p_i \log p_i, H(V) = -\sum_{j=1}^{C} p_j \log p_j \tag{27}$$

where U is a group of data obtained by actual classification, and V is a group of data obtained by clustering.

The values of mutual information are normalized to between 0 and 1 so that comparisons can be made among different datasets [135]. The closer the value of standardized mutual information is to 1, the better the clustering effect is.

4.2.6 Adjusted Mutual Information

The values of mutual information and normalized mutual information are both affected by the number of clustering categories K, while AMI is not, and the range of value is -1

Table 9 Comparison of different evaluation indicators

Evaluation indicators	Applicable scene
The sum of squared error	It is very suitable for K-means. The limitation of this index is that it only considers the intra-cluster similarity and does not consider the relationship among different clusters
Compactness	For a single cluster, it is like SSE, only the similarity intra-cluster is considered
Separation	In contrast to the compactness, this index only considers the inter-cluster distance
Silhouette coefficient	It is used to determine the relative size of "intra-cluster distance" and "intra-cluster distance"
Calinski-Harabasz index	The inter-cluster distance and intra-cluster distance are considered in this index. It cannot be used to compare different clustering algorithms and can only be used to select better parameters in one algorithm
Davies-Bouldin index	This method is used in space–time sequence clustering. It can be used only if the similarity measure is used in European distance, which has a poor clustering effect on the circular distribution sample points
Dunn validity index	It only applies to discrete datasets
Rand index	It is applicable for clustering where the results are divided into two categories
Jaccard index	It is considered to identify the equivalency between two datasets, especially the text dataset
F-measure	It is an improved index of the Rand index. And it is commonly used in information retrieval
Fowlkes–Mallow's index	It does not assume anything for the clustering process. But sample categories need to be known
Normalized mutual information	The method has a complex process and high precision, which can be compared between different datasets
Adjusted mutual information	The values of MI and NMI are both affected by the number of clustering categories K, while AMI is not

to 1 [136]. The larger the value is, the closer the results of the two kinds of clustering are. Definitions are as follows:

$$AMI = \frac{MI - E(MI)}{\mathrm{mean}(H(U), H(V)) - E(MI)} \tag{28}$$

The comparison of different evaluation indicators is shown in Table 9, it is a summary of Sect. 4.

5 Applications in Vehicles

Clustering technology is useful in plenty of applications. This section analyzes the application of clustering in the automotive industry in detail. Because the vehicle has a complex structure, it requires the coordinated operation of different systems, and many control strategies are applied to it. As an excellent method, clustering has been widely studied by many researchers in the automotive field. The specific applications are shown in Table 10 [137–143].

5.1 Conventional Vehicles

The clustering algorithm has been studied in engines, suspension, vehicle crashes, noise and vibration level assessment, fault identification, and so on, see Fig. 9 [143–145].

5.1.1 Vehicle Crash and Knock

In order to explore the knock threshold in automotive engines, a Gaussian mixture model (GMM) clustering-based method was proposed by Shen et al. [146, 147] to learn the optimal threshold of the knock intensity metric in automotive engines, which minimized the probability of the judgment error of the knock event. The problem was formulated based on the assumption that the probability distribution model of knock intensity was a two-component GMM. It was a clustering problem in which parameters of the two-component GMM were optimized to maximize the likelihood of obtaining the measurements. One of the major challenges with vehicle crash frequency studies was how to deal with the unobserved heterogeneity in crash data. Gong and Wang [148] processed crash frequency data with hierarchical clustering to capture more unobserved heterogeneity. In this study, the data of single crash accidents were first clustered into subgroups using a hierarchical clustering method, and then, a Random Effects Negative Binomial model was applied to each subgroup with crash counts at an intersection as observations. A model with no data clustering was also estimated to serve as the comparison benchmark. With the collision data, the injury degree of the occupant or pedestrian in the accident could be analyzed by clustering regression [139, 149], and the correlation between them could be obtained.

5.1.2 Vehicle Vibration and Acoustical Comfort

Vehicle vibration and acoustical comfort are crucial criteria that may attract customers when purchasing a vehicle, a good suspension system and noise processing ability could make passengers more comfortable. Nguyen et al. [150] used fuzzy clustering to process historical road load data, based on the FISs, the desired force values were calculated according to the status of the road at each time. Then, the B-ANFIS was used to build ANFISs for inverse dynamic models of the suspension system (I-ANFIS) to improve suspension

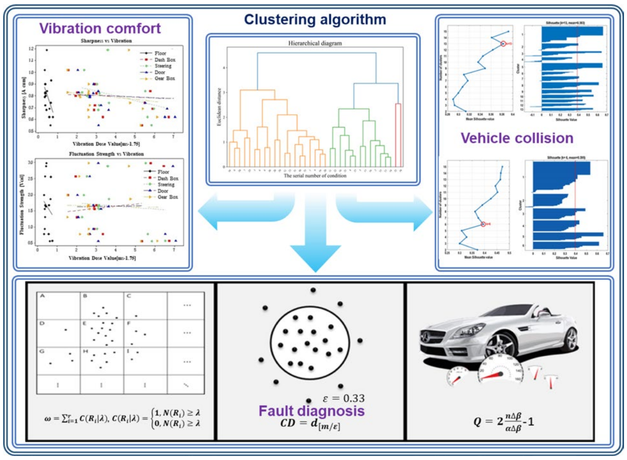

Fig. 9 Application of clustering method in traditional vehicles [143–145]

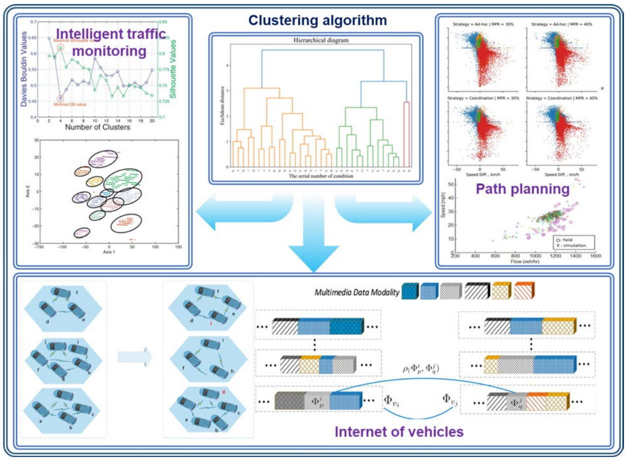

Fig. 10 Application of clustering in intelligent vehicles [153–155]

dynamics models and the control of dampers to reduce vibration levels. In addition, many references [140, 143] tried to determine the correlation between noise sources and the level of significance of their noise generation and classified them into empirical evaluation models. The objective was to propose an approach that clusters the level of sound and vibration into a few categories and classified them into those categories without implementing the subjective test that normally involves human assessment, then solves and avoids every noise problem effectively.

5.1.3 Vehicle Fault Monitoring

For vehicle fault monitoring, some researchers performed vehicle data clustering (such as speed, engine torque, temperature, wheel rotation speed, and so on), and the clustering result could be analyzed by using vehicle outlier information caused by complex correlation of vehicle components [144, 151]. The cluster data of the representative attributes was sampled and the cluster characteristics were determined according to the relationship between the vehicle data and state. The vehicle outliers were monitored by the complex vehicle state, even on this basis, it was not difficult to get an accident prediction model. In addition, K-means clustering of the axle load spectrum of different vehicles in different time periods was used, and the corresponding road surface conditions were inferred according to the different clusters obtained. It could significantly improve the prediction accuracy of pavement performance in a certain area, so that the relevant departments could improve the conditions of the road [152]. In general, clustering processed any data in a vehicle well and provided the basis for further control and optimization.

5.2 Intelligent and Connected Vehicles

The decision and control of intelligent and connected vehicles need a large amount of historical data as support, and clustering, as a big data mining method, is very suitable for path planning, Internet of vehicles communication, and intelligent traffic monitoring. And the specific application is shown in Fig. 10 [153–155].

5.2.1 Path Planning

For autonomous vehicles, the core is to automatically identify the surrounding obstacles and decide the best driving route. The shape of the car and the distance between them can be analyzed and evaluated through clustering, to get a better geometric path. Ewbank et al. [156] processed vehicle position parameters through fuzzy clustering, and it used unsupervised fuzzy clustering as the cornerstone of a proposed heuristic, an assignment algorithm redistributed the demand points among the clusters based on their membership grades, observing the vehicle capacity to save computational time presenting optimal results. C. Besse et al. [157] compared a new distance(symmetrized segment-path distance) to the others according to their corresponding clustering results obtained using both the hierarchical clustering and affinity propagation methods. And they tackled the issue of clustering trajectories of geo-localized observations based on the distance between trajectories. The above methods were all offline-based and could not store and update the historical data in real time. Therefore, the path planning method of cluster-based real-time online sharing was proposed in Refs. [158, 159]. The proposed trajectory clustering algorithm took advantage of the network topology and time–space distance measurements within vehicle trajectory data. The time–space distance between two locations was

Table 10 Application of clustering method in vehicles

Type of vehicles	Component	Specific application	Analysis
Traditional vehicles	Engine Suspension	The data of oil and gas, knock intensity Load data, noise processing, stiffness, and strength	The basic data and parameters in the car are processed It is not widely used
	Others	Fault locating or forecasting based on vehicle data	
Intelligent vehicles	Path planning Internet of vehicles communication	Identify obstacles and choose the best route Vehicle network multimedia data processing and sharing	The decision-making process and Internet of Vehicles communication need to store a large amount of historical data Pattern recognition needs to be used, and clustering can better realize data processing and online sharing
	Intelligent traffic monitoring	Image segmentation, pattern recognition, location detection	
New energy vehicles	HEV BEV FCV	Energy management, power distribution Charging and consistency of lithium batteries Fuel cell fault identification, power distribution, hydrogen refueling station selection, vehicle configuration	The core is the battery and fuel cell. Charging or hydrogenation, microscopic analysis inside the battery and cell, and constant switching between different power sources, lots of data needs to be processed, so clustering is more suitable for it

dependent on the time-dependent shortest-path (TDSP) distance in the network, and the gap between the time stamps when the object was detected at each targeted location. In the future, the method combining clustering with deep learning may be the best solution to path planning problems [160, 161].

5.2.2 Intelligent Transportation System (ITS)

Traffic jam reduces the efficiency of transportation infrastructure usage and increases travel time, air pollution as well as fuel consumption. Then, Intelligent Transportation System (ITS) came as a solution to this problem by implementing information technology and communications networks. One classical method of ITS was video camera technology [162]. Particularly, it has been applied to collect traffic data including vehicle detection and analysis. A new clustering method based on 3D information in real traffic video was used to classify the vehicle trajectory points, and vehicle type could be estimated to realize vehicle behavior analysis and vehicle classification. However, this application still had limitations when it had to deal with complex traffic and environmental condition [163]. Therefore, clustering analysis was used in image segmentation [164] and pattern recognition [153] to overcome these limitations. Important information about dynamic traffic processes, such as the instantaneous number of vehicles, their weight, speed, and distance among vehicles, was available for precise positioning detection. These methods led to meaningful labels that could be automatically retrieved from large databases and also led to more efficient separation of the resulting feature space. At present, the autonomous coordinated transportation system based on the potential impact between driverless and manned vehicles is still being explored [154]. Thus far, the majority of the CACC studies have been focusing on the overall network performance with limited insights on the potential impacts of connected and autonomous vehicles (CAVs) on human-driven vehicles (HVs). This paper aims to quantify such impacts by studying the high-resolution vehicle trajectory data obtained from microscopic simulation.

5.2.3 The Internet of Vehicles

In recent years, the Internet of Vehicles (IoV) has provided promising solutions to upgrade traditional vehicular ad hoc networks (VANETs) to the next level. The information was exchanged between vehicles and the infrastructure of the vehicle (VANET) (manet or mobile self-networking), in order to improve efficiency and ensure traffic safety [165]. However, there were also some problems, such as too much data stored in the network, dynamic changes in vehicle nodes which caused frequent changes in the topology, and high restrictions on available resources. Therefore, the clustering method was usually used in VANETS. Dutta et al. [166] and Senouc et al. [167] made a study of the large-scale data, using hierarchical clustering, where vehicles were connected to the Internet through road-side unit gateways. Each vehicle collected information about its neighboring nodes and updated the appropriate cluster heads in real time, while avoiding instantaneous failures caused by excessive data accumulation at a certain moment. Metaheuristic dragonfly-based clustering algorithm (CAVDO) was used for cluster-based packet route optimization to make a stable topology [138]. Considerable important parameters involved in the clustering process were the number of un-clustered nodes as a re-clustering criterion, clustering time, re-clustering delay,

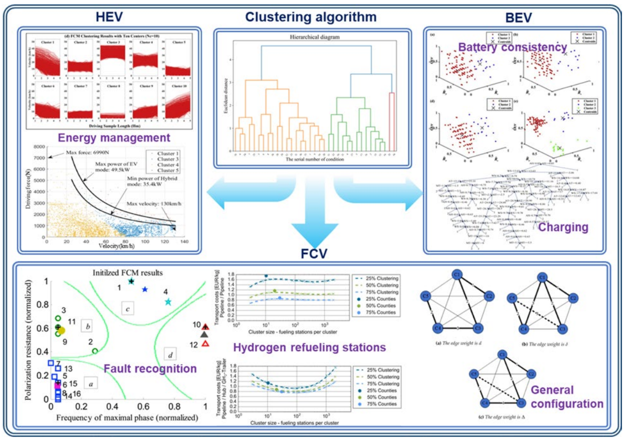

Fig. 11 Application of clustering in new energy vehicles [168–174]

dynamic transmission range, direction, and speed. According to these parameters, results indicated that CAVDO outperformed ACO-based clustering and CLPSO in various network settings. In order to solve the problem of resource limitation, IoV clustering needed to explore the correlation between multimedia data from vehicles. Vehicles with similar multimedia content were classified into the same cluster according to multimedia data analysis results. In addition, considering the limitation of the real-time problem and the vehicle location, the communication range in each clustering was considered. It might be a solution to maximize the stable sharing of the same resource [155].

5.3 New Energy Vehicles

New energy vehicles (NEV) are the research hot spots in recent years, and clustering has been widely used in charge and discharge technology, energy management, and so on. The specific applications are shown in Fig. 11 [168–174].

5.3.1 Hybrid Electric Vehicles

Hybrid electric vehicles were very popular in the market. The coordinated operation of the battery and engine of two power sources could adapt to different driving conditions. So clustering technology was usually used to assist energy management and rational switching of power sources to reduce energy consumption [175]. The most commonly used technology was clustering of speed and power information, and Zhou et al. [175]. used FCM to preprocess the driving data, including speed and acceleration sequence and selected three parameters to represent each driving sample, which were the average speed, speed standard deviation, and the average acceleration. Then, the dataset was divided into *n* high-dimensional driving vectors. In addition to the Markov chain method, a new power allocation strategy was obtained for energy management [173]. A further study applied clustering for driving behavior recognition and real-time traffic information prediction. First, the driver style and personal information provided by the driver are analyzed, mainly the factors that could determine the driving behavior and provide the basis for the clustering. Then, the vehicle state and the driving state information were used as the attribute of the samples. The sample points of a period of time were integrated into a point set, as a sample of the final clustering analysis. Adaptive equivalent consumption minimization strategy or optimal rule-based pattern partitioning strategy was achieved [174, 176]. Some applications of clustering within batteries were described in detail in the following session.

5.3.2 Battery Electric Vehicles

The clustering algorithm is widely used in charging technology. Jurjen R. Helmus et al. [177] and Gilanifar et al. [137] constructed the charging demand prediction model based on the data statistics of the charging station. They overcame predefined stereotypical expectations of user behavior by using a bottom-up data-driven two-step clustering approach that clusters charging sessions and thereafter portfolios of charging sessions per user. But the influencing factors were not comprehensively considered. Therefore, many papers [171, 178] analyzed different distribution characteristics of EV charging loads in a period through clustering and predicted EV charging demands by considering actual traffic distribution data and weather conditions. The considered variables were the charging starting time determined by the real-world traffic patterns and the initial state of charge of a battery. And the forecasting processes included a cluster analysis to classify traffic patterns, a relational analysis to identify influencing factors, and a decision tree to establish classification criteria. In addition, clustering was also commonly used for the location selection of charging stations. The multi-level clustering method was usually adopted to cluster the charging demand positions of electric vehicles, and the utilization rate of charging stations and the traveling time of electric vehicles were taken as the objective functions to seek the optimal positions [141, 179].

In addition, the internal structure and chemical properties of lithium-ion batteries also involved a large number of parameters, some people began to use modern clustering methods to evaluate the consistency of lithium-ion batteries [168]. First, the features which reflected the static or dynamic characteristics of batteries are excavated. Second, a weighted method of multi-feature inconsistency was proposed to evaluate pack consistency. Third, an improved Greenwald-Khanna algorithm was developed to cluster batteries. Liu et al. [180] used the SOM clustering algorithm to further equalize the electrochemical performance of the battery to extend the battery life and improve performance. Experiments were conducted by dismantling the pack and measuring the capacity, voltage, and internal resistance data. Clustering analysis based on self-organizing map (SOM) neural networks is then applied to the measured data to form clusters of battery packs.

5.3.3 Fuel Cell Vehicles

As a green power generation device, the fuel cell is widely used in vehicles. In recent years, many automobile manufacturers have been committed to the development of fuel cell vehicles (FCV) [181, 182], whose scale of production and application has expanded rapidly. FCV will be a hot research area in the future. At present, the clustering algorithm is rarely used in fuel cell vehicles.

Since clustering was often used in fault identification and image segmentation in traditional and intelligent vehicles, it could also be used to monitor the internal mechanism and to identify and classify faults in fuel cells. Zheng et al. [170, 183] studied air stoichiometry through electrochemical impedance spectroscopy (EIS), and a double-fuzzy method consisting of fuzzy clustering and fuzzy logic was developed to mine diagnostic rules from the experimental data automatically and to identify and classify the phenomenon of flooding or drying. On this basis, the influence of approximate relative humidity in the membrane, dynamics gases aspect, voltage behavior, and load demand on these two failures was considered in Ref. [184] to improve the performance and life of fuel cells comprehensively. In order to further understand the internal microstructure of the cells, a three-phase segmentation technique of the anode image was usually used, the core of which was the quantum-inspired clustering algorithm [185, 186]. This mixture clustering model combined a quantum-inspired MRF-based fuzzy logic model with GMM. In addition, GMM-based negative log-prior functions were designed as pixel distance metrics in the consequent part.

For the study of FCVs, clustering could also be used for power allocation, hydrogenation station selection, vehicle configuration, and so on. Fuel cells were often used in conjunction with power batteries or ultracapacitors [187]. Fuzzy clustering [188] could achieve global power distribution to avoid hypoxia, enhance transient performance and extend the operating life of the hybrid system. The control employed fuzzy clustering-based modeling, constrained model predictive control, and adaptive switching among multiple models. The reasonable arrangement of multiple cells and other components had a great influence on the transmission rate of hydrogen and water. Therefore, Kang et al. [169] and Bankupalli et al. [189] tried to design the vehicle configuration through hierarchical clustering, and this allowed the efficient generation of decentralized control configuration as well as the entire hierarchy of block decentralized control configurations. In addition, a notion of modularity was used to evaluate the compactness and separation of the resulting clusters, allowing the identification of optimal control configurations, which could improve fuel efficiency greatly.

Finally, it is similar to the location selection of charging piles, and hydrogen refueling stations should also be distributed in all parts of the city as much as possible. As more stations will be added in the future, it will require a spatiotemporal clustering algorithm to better understand how the location of the station contributes to their respective behavior and utilization. Kalai [142] demonstrated an unsupervised temporal clustering approach and a large number of fuel types and hydrogenation time data need to be collected to make a plan for the location distribution of hydrogen refueling stations.

Compared with conventional combustion engine-driven vehicles, new energy vehicles have a great development prospect and are being promoted vigorously. But there are still many problems to be solved. Battery system involves a larger dataset, and cluster analysis can certainly provide more help for it.

6 Future Development of Clustering

In the future, the main development directions of clustering are as follows [190, 191]:

(1) Clustering analysis is not only a process of selecting or designing clustering algorithms but also a process of data preprocessing and feature extraction. For an actual dataset, a large number of detailed mathematical calculations need to be completed during data preprocessing, and the quality of feature extraction may directly affect the final clustering results. Therefore, preprocessing and feature extraction need to be improved to reduce the time cost.

(2) In practical applications, datasets are characterized by complexity and diversity, so any clustering algorithm may not be suitable. Therefore, it is necessary to study the fusion of multiple algorithms based on understanding the advantages and disadvantages of basic clustering algorithms, and the algorithms that can generate clusters of any shape will be the development direction of clustering algorithm research.

(3) In the era of big data, big data clustering algorithms will have a good development prospect, such as GPU-based clustering, SPARK-based clustering, graph computing framework (such as Pregel) clustering, Map Reduce clustering, and other new clustering concepts and technologies. Tasks can be distributed over many servers to perform a task decomposition.

(4) The algorithm can be improved to enhance personalization. Personalization algorithms synthesize information from the data and make appropriate conclusions, such as personal interests, then provide recommendations to users and induce them to perform certain behaviors.

(5) With the increasingly mature application of cloud computing, Internet of Things and big data technology, the complexity of data to be clustered is unprecedented. Therefore, in the context of complex data, integrating quantum computing into the clustering algorithm to speed up data processing is also an important and demanding research hot spot.

7 Conclusions

This paper introduces some basic ideas of various commonly used clustering algorithms and analyzes the advantages and disadvantages of each algorithm as well as their scope of application. Because there are many clustering algorithms, and new algorithms are being proposed all the time, thus to compare the clustering algorithms is necessary. But there are many clustering technologies, some of which are almost unapplied. Therefore, this paper does not cover all the methods, it selects 20 kinds of commonly used traditional clustering algorithms with high practical value and more in-depth research and four kinds of modern new clustering methods and discusses each kind of clustering algorithm in detail. This paper illustrates the basic definition of the most used unsupervised learning method clustering, 6 categories of clustering algorithms, 12 commonly used similarity measures, and 13 clustering evaluation indexes. Finally, applications of clustering methods in the automotive field are illustrated. Since the current clustering technology still has limitations in areas like data dimension, data size and shape, stability and scalability, and parameter selection, more high-performance clustering algorithms need to be proposed and applied in future studies.

Acknowledgements This work is supported in part by the founding of the State Key Laboratory of Industrial Control Technology, Zhejiang University (ICT2021B19), and the Technological Innovation and Application Demonstration in Chongqing (Major Themes of Industry: cstc2019jscx-zdztzxX0033, cstc2019jscx-fxyd0158).

Declarations

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

References

1. Idiart, M.A.P.: Performances in supervised learning. Phys. A **285**(3), 566–578 (2000)
2. LeBourgeois, F., Bouayad, M., Emptoz, H.: Structure relation between classes for supervised learning using pretopology. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318) (1999)
3. Zhang, Y., Huang, Z., Zhang, C., Lv, C., Deng, C., Hao, D., Chen, J., Ran, H.: Improved short-term speed prediction using spatiotemporal-vision-based deep neural network for intelligent fuel cell vehicles. IEEE Trans. Ind. Inform. **17**(9), 6004–6013 (2021)
4. Li, Q., Yin, L., Yang, H., Wang, T., Qiu, Y., Chen, W.: Multiobjective optimization and data-driven constraint adaptive predictive control for efficient and stable operation of PEMFC system. IEEE Trans. Ind. Electron. **68**(12), 12418–12429 (2021)
5. Chen, R., Guo, S.K., Wang, X.Z., Zhang, T.L.: Fusion of multi-RSMOTE with fuzzy integral to classify bug reports with an imbalanced distribution. IEEE Trans. Fuzzy Syst. **27**(12), 2406–2420 (2019)
6. Pang, R., Zhang, C., Dai, H., Bai, Y., Hao, D., Chen, J., Zhang, B.: Intelligent health states recognition of fuel cell by cell voltage consistency under typical operating parameters. Appl. Energy **305**, 117735 (2022)
7. Paul, D., Jain, A., Saha, S., Mathew, J.: Multi-objective PSO based online feature selection for multi-label classification. Knowl. Based Syst. **222**, 106966 (2021)
8. Herp, J., Pedersen, N.L., Nadimi, E.S.: Wind turbine performance analysis based on multivariate higher order moments and Bayesian classifiers. Control Eng. Pract. **49**, 204–211 (2016)
9. Li, G., Chen, Y., Cao, D., Qu, X., Cheng, B., Li, K.: Extraction of descriptive driving patterns from driving data using unsupervised algorithms. Mech. Syst. Signal Process. **156**, 107589 (2021)
10. Nikbakht, R., Jonsson, A., Lozano, A.: Unsupervised learning for parametric optimization. IEEE Commun. Lett. **25**(3), 678–681 (2021)
11. Levada, A.L.M.: Parametric PCA for unsupervised metric learning. Pattern Recognit. Lett. **135**, 425–430 (2020)
12. Yao, J., Mao, Q., Goodison, S., Mai, V., Sun, Y.: Feature selection for unsupervised learning through local learning. Pattern Recognit. Lett. **53**, 100–107 (2015)
13. Hanson, S.J., Bauer, M.: Conceptual clustering, categorization, and polymorphy. Mach. Learn. **3**(4), 343–372 (1989)
14. Higbee, K.: Mathematical classification and clustering. Technometrics **40**(1), 80–80 (1998)
15. Sisodia, D., Singh, L., Sisodia, S., Saxena, K.: Technology: clustering techniques: a brief survey of different clustering algorithms. Int. J. Latest Trends Eng. **1**(3), 82–87 (2012)
16. Dave, M., Gianey, H.: Different clustering algorithms for big data analytics: a review. In: 2016 International Conference System Modeling & Advancement in Research Trends (SMART) (2016)
17. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: a k-means clustering algorithm. J. R. Stat. Soc. C Appl. **28**(1), 100–108 (1979)
18. Arora, S., Chana, I.: A survey of clustering techniques for big data analysis. In: 2014 5th International Conference-Confluence

the Next Generation Information Technology Summit (Confluence) (2014)

19. Zhong, C., Miao, D., Wang, R.: A graph-theoretical clustering method based on two rounds of minimum spanning trees. Pattern Recognit. **43**(3), 752–766 (2010)
20. Zhang, H., Yang, Z., Oja, E.: Improving cluster analysis by co-initializations. Pattern Recognit. Lett. **45**, 71–77 (2014)
21. Garza-Fabre, M., Handl, J., Knowles, J.: An improved and more scalable evolutionary approach to multiobjective clustering. IEEE Trans. Evolut. Comput. **22**(4), 515–535 (2017)
22. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Netw. **12**(2), 181–201 (2001)
23. Zubaroğlu, A., Atalay, V.: Data stream clustering: a review. Artif. Intell. Rev. **54**(2), 1201–1236 (2020)
24. Kang, Z., Xu, H., Wang, B., Zhu, H., Xu, Z.: Clustering with similarity preserving. Neurocomputing **365**, 211–218 (2019)
25. Guo, Z., Shang, C., Ye, H.: A novel similarity metric with application to big process data analytics. Control Eng. Pract. **113**, 104843 (2021)
26. Nagpal, A., Jatain, A., Gaur, D.: Review based on data clustering algorithms. In: 2013 IEEE Conference on Information & Communication Technologies, 11–12 April 2013 (2013)
27. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
28. Li, M., Zhu, Y., Zhao, T., Angelova, M.: Weighted dynamic time warping for traffic flow clustering. Neurocomputing (2021). https://doi.org/10.1016/j.neucom.2020.12.138
29. Mehta, V., Bawa, S., Singh, J.: Analytical review of clustering techniques and proximity measures. Artif. Intell. Rev. **53**(8), 5995–6023 (2020)
30. Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. Ann. Data Sci. **2**(2), 165–193 (2015)
31. Saxena, A., Prasad, M., Gupta, A., et al.: A review of clustering techniques and developments. Neurocomputing **267**, 664–681 (2017)
32. Rasyid, L.A., Andayani, S.: Review on clustering algorithms based on data type: towards the method for data combined of numeric-fuzzy linguistics. J. Phys. **1097**, 012082 (2018)
33. Wei, W., Liang, J., Guo, X., Song, P., Sun, Y.: Hierarchical division clustering framework for categorical data. Neurocomputing **341**, 118–134 (2019)
34. Sadeghzadeh, K., Fard, N.: Analytical clustering procedures in massive failure data. In: 2017 Annual Reliability and Maintainability Symposium (2017)
35. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. **25**(2), 103–114 (1996)
36. Kovács, L., Bednarik, L.: Parameter optimization for BIRCH pre-clustering algorithm. In: 2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI) (2011)
37. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. Inform. Syst. **26**(1), 35–58 (2001)
38. Lathiya, P., Rani, R.: Improved CURE clustering for big data using Hadoop and Mapreduce. In: 2016 International Conference on Inventive Computation Technologies (ICICT) (2016)
39. Guha, S., Rastogi, R., Shim, K.: ROCK: a robust clustering algorithm for categorical attributes. Inform Syst. **25**(5), 345–366 (2000)
40. Dutta, M., Mahanta, A.K., Pujari, A.K.: QROCK: a quick version of the ROCK algorithm for clustering of categorical data. Pattern Recognit. Lett. **26**(15), 2364–2373 (2005)
41. Karypis, G., Han, E.H., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. Computer **32**(8), 68–75 (1999)
42. Dong, Y., Wang, Y., Jiang, K.: Improvement of partitioning and merging phase in chameleon clustering algorithm. In: 2018 3rd International Conference on Computer and Communication Systems (ICCCS) (2018)
43. Dharmarajan, A., Velmurugan, T.: Applications of partition based clustering algorithms: a survey. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research (2013)
44. Liu, S., Zhao, Q., Wu, X.: Feature selection based on partition clustering. Knowl. Based Syst. **18**(2), 135–142 (2014)
45. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, No. 14 (1967)
46. Filali, A., Jlassi, C., Arous, N.: Dimensionality reduction with unsupervised ensemble learning using k-means variants. In: 2017 14th International Conference on Computer Graphics, Imaging and Visualization (2017)
47. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. Expert Syst. Appl. **36**(2), 3336–3341 (2009)
48. Ushakov, A.V., Vasilyev, I.: Near-optimal large-scale k-medoids clustering. Inform. Sci. **545**, 344–362 (2021)
49. Kanika, Rani, K., Sangeeta, Preeti: Visual analytics for comparing the impact of outliers in k-means and k-medoids algorithm. In: 2019 Amity International Conference on Artificial Intelligence (AICAI) (2019)
50. Purnamasari, K.K.: K-means and K-medoids for Indonesian text summarization. IOP Conf. Ser. Mater. Sci. Eng. **662**(6), 062013 (2019)
51. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2), 191–203 (1984)
52. Dai, B., Wang, F., Chang, Y.: Multi-objective economic load dispatch method based on data mining technology for large coal-fired power plants. Control Eng. Pract. **121**, 105018 (2022)
53. Ng, R.T., Jiawei, H.: CLARANS: a method for clustering objects for spatial data mining. IEEE Trans. Knowl. Data Eng. **14**(5), 1003–1016 (2002)
54. Vukčević, M., Popović-Bugarin, V., Dervić, E.: DBSCAN and CLARA clustering algorithms and their usage for the soil data clustering. In: 2019 8th Mediterranean Conference on Embedded Computing (MECO) (2019)
55. Kamali, T., Stashuk, D.W.: Discovering density-based clustering structures using neighborhood distance entropy consistency. IEEE Trans. Comput. Soc. Syst. **7**(4), 1069–1080 (2020)
56. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd (1996)
57. Niu, T., Huang, W., Zhang, C., Zeng, T., Chen, J., Li, Y., Liu, Y.: Study of degradation of fuel cell stack based on the collected high-dimensional data and clustering algorithms calculations. Energy AI (2022). https://doi.org/10.1016/j.egyai.2022.100184
58. Chen, Y., Zhou, L., Bouguila, N., Wang, C., Chen, Y., Du, J.: BLOCK-DBSCAN: fast clustering for large scale data. Pattern Recognit. **109**, 107624 (2021)
59. Behara, K.N.S., Bhaskar, A., Chung, E.: A DBSCAN-based framework to mine travel patterns from origin-destination matrices: Proof-of-concept on proxy static OD from Brisbane. Transp. Res. C Emerg. **131**, 103370 (2021)
60. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: ordering points to identify the clustering structure. SIGMOD Rec. **28**(2), 49–60 (1999)
61. Wagner, T., Feger, R., Stelzer, A.: Modifications of the OPTICS clustering algorithm for short-range radar tracking applications. In: 2018 15th European Radar Conference (EuRAD). (2018)
62. Hinneburg, A., Keim, D. A.: An efficient approach to clustering in large multimedia databases with noise. In: KDD (1998)

63. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. **24**(5), 603 (2002)
64. Denœux, T.: Calibrated model-based evidential clustering using bootstrapping. Inform. Sci. **528**, 17–45 (2020)
65. Wang, Z., Ritou, M., Da Cunha, C., Furet, B.: Contextual classification for smart machining based on unsupervised machine learning by Gaussian mixture model. Int. J. Comput. Integr. Manuf. **33**(10–11), 1042–1054 (2020)
66. Zhang, Y., Li, M., Wang, S., Dai, S., Luo, L., Zhu, E., Xu, H., Zhu, X., Yao, C., Zhou, H.: Gaussian mixture model clustering with incomplete data. ACM Trans. Multimed. Comput. **17**(1s), 1–14 (2021)
67. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Mach. Learn. **2**(2), 139–172 (1987)
68. Araujo, A.F.R., Antonino, V.O., Ponce-Guevara, K.L.: Self-organizing subspace clustering for high-dimensional and multi-view data. Neural Netw. **130**, 253–268 (2020)
69. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
70. Delgado, S., Higuera, C., Calle-Espinosa, J., Morán, F., Montero, F.: A SOM prototype-based cluster analysis methodology. Expert Syst. Appl. **88**, 14–28 (2017)
71. Carpenter, G.A., Grossberg, S.: The ART of adaptive pattern recognition by a self-organizing neural network. Computer **21**(3), 77–88 (1988)
72. Carpenter, G.A., Grossberg, S.: ART 2: self-organization of stable category recognition codes for analog input patterns. Appl. Opt. **26**(23), 4919–4930 (1987)
73. Carpenter, G.A., Grossberg, S.: ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. Neural Netw. **3**(2), 129–152 (1990)
74. Han, J., Kamber, M., Pei, J.: Cluster analysis: basic concepts and methods. In: Data Mining (3rd Edn.), The Morgan Kaufmann Series in Data Management Systems, Boston, pp. 443–495 (2012)
75. Lee, G.H.: Grid-based dynamic clustering with grid proximity measure. Intell. Data Anal. **20**(4), 853–875 (2016)
76. Wang, W., Yang, J., Muntz, R.: STING: a statistical information grid approach to spatial data mining. In: Vldb (1997)
77. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington (1998)
78. Chrobak, M., Dürr, C., Fabijan, A., Nilsson, B.J.: Online clique clustering. Algorithmica **82**(4), 938–965 (2019)
79. Cheng, C.H., Fu, A.W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (1999)
80. Girolami, M.: Mercer kernel-based clustering in feature space. IEEE Trans. Neural Netw. **13**(3), 780–784 (2002)
81. Callaghan, L.O., Mishra, N., Meyerson, A., Guha, S., Motwani, R.: Streaming-data algorithms for high-quality clustering. In: Proceedings 18th International Conference on Data Engineering, (2002)
82. Figueiredo, E., Macedo, M., Siqueira, H.V., Santana, C.J., Gokhale, A., Bastos-Filho, C.J.A.: Swarm intelligence for clustering—a systematic review with new perspectives on data mining. Eng. Appl. Artif. Intell. **82**, 313–329 (2019)
83. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
84. Wu, Z.D., Xie, W.X., Yu, J.P.: Fuzzy C-means clustering algorithm based on kernel method. In: Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003 (2003)
85. Asharaf, S., Shevade, S.K., Murty, M.N.: Rough support vector clustering. Pattern Recognit. **38**(10), 1779–1783 (2005)
86. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. Proc. Adv. Neural Inf. Process. Syst. **17**, 1537–1544 (2004)
87. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intell. **1**(2), 95–113 (2007)
88. Omran, M., Engelbrecht, A.P., Salman, A.: Particle swarm optimization method for image clustering. Int. J. Pattern Recognit. **19**(03), 297–321 (2005)
89. Sellami, A., Ben Abbes, A., Barra, V., Farah, I.R.: Fused 3-D spectral-spatial deep neural networks and spectral clustering for hyperspectral image classification. Pattern Recognit. Lett. **138**, 594–600 (2020)
90. Fabijanska, A.: Normalized cuts and watersheds for image segmentation. In: IET Conference on Image Processing (IPR 2012) (2012)
91. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Proceedings of the Advances in Neural Information Processing Systems (2002)
92. Sangam, R.S., Om, H.: Equi-clustream: a framework for clustering time evolving mixed data. Adv. Data Anal. Classi. **12**(4), 973–995 (2018)
93. Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y.: A comparison study on similarity and dissimilarity measures in clustering continuous data. PLoS ONE **10**(12), e0144059 (2015)
94. Al-Sultana, K.S., Khan, M.M.: Computational experience on four algorithms for the hard clustering problem. Pattern Recognit. Lett. **17**(3), 295–308 (1996)
95. Zhu, G., Wu, C., Chen, H.: K-way fast approximate spectral clustering. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (2019)
96. Cha, S.H.: Comprehensive survey on distance/similarity measures between probability density functions. City **1**(2), 1 (2007)
97. Xu, D.G., Zhao, P.L., Yang, C.H., Gui, W.H., He, J.J.: A novel Minkowski-distance-based consensus clustering algorithm. Int. J. Autom. Comput. **14**(1), 33–44 (2016)
98. Rui, X., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Netw. **16**(3), 645–678 (2005)
99. Wang, H., Wang, W., Yang, J., Yu, P.S.: Clustering by pattern similarity in large data sets. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of data (2002)
100. Sun, G., Jiang, C., Cheng, P., Liu, Y., Wang, X., Fu, Y., He, Y.: Short-term wind power forecasts by a synthetical similar time series data mining method. Renew. Energy. **115**, 575–584 (2018)
101. Ma, R., Angryk, R.: Distance and density clustering for time series data. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (2017)
102. Mana, S.C., Sasipraba, T.: Research on cosine similarity and pearson correlation based recommendation models. J. Phys. **1770**(1), 012014 (2021)
103. Wu, C., Wang, B.: Extracting topics based on Word2Vec and improved Jaccard similarity coefficient. In: 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC) (2017)
104. Bravais, A.: Analyse Mathématique sur les Probabilités des Erreurs de Situation d'un Point. Impr. Royale (1844)
105. Pearson, K.J.: Containing papers of a mathematical or physical character. VII. Mathematical contributions to the theory of evolution.—III. Regression heredity panmixia. Philos. Trans. R. Soc. Lond. **187**, 253–318 (1896)
106. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)

107. Lin, Y., Jiang, J., Lee, S.: A similarity measure for text classification and clustering. IEEE Trans. Knowl. Data Eng. **26**(7), 1575–1590 (2014)
108. Taheri, R., Ghahramani, M., Javidan, R., Shojafar, M., Pooranian, Z., Conti, M.: Similarity-based Android malware detection using Hamming distance of static binary features. Future Gener. Comput. Syst. **105**, 230–247 (2020)
109. Mrabah, N., Khan, N.M., Ksantini, R., Lachiri, Z.: Deep clustering with a dynamic autoencoder: from reconstruction towards centroids construction. Neural Netw. **130**, 206–228 (2020)
110. Dinler, D., Tural, M.K., Ozdemirel, N.E.: Centroid based tree-structured data clustering using vertex/edge overlap and graph edit distance. Ann. Oper. Res. **289**(1), 85–122 (2020)
111. Zhai, W., Bai, X., Peng, Z.R., Gu, C.: From edit distance to augmented space-time-weighted edit distance: detecting and clustering patterns of human activities in Puget Sound region. J. Transp. Geogr. **78**, 41–55 (2019)
112. Du, S., Wu, M., Chen, L., Cao, W., Pedrycz, W.: Operating mode recognition of iron ore sintering process based on the clustering of time series data. Control Eng. Pract. **96**, 104297 (2020)
113. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognit. **44**(3), 678–693 (2011)
114. Brusco, M.J.: A repetitive branch-and-bound procedure for minimum within-cluster sums of squares partitioning. Psychometrika **71**(2), 347–363 (2006)
115. Martín-Santamaría, R., Sánchez-Oro, J., Pérez-Peló, S., Duarte, A.: Strategic oscillation for the balanced minimum sum-of-squares clustering problem. Inform. Sci. **585**, 529–542 (2022)
116. Paul, D., Saha, S., Mathew, J.: Improved subspace clustering algorithm using multi-objective framework and subspace optimization. Expert Syst. Appl. **158**, 113487 (2020)
117. Li, K., Cao, X., Ge, X., Wang, F., Lu, X., Shi, M., Yin, R., Mi, Z., Chang, S.: Meta-heuristic optimization-based two-stage residential load pattern clustering approach considering intra-cluster compactness and inter-cluster separation. IEEE Trans. Ind. Appl. **56**(4), 3375–3384 (2020)
118. Wu, X.H., Wu, B., Sun, J., Zhao, J.W.: Mixed fuzzy inter-cluster separation clustering algorithm. Appl. Math. Model. **35**(10), 4790–4795 (2011)
119. Gagolewski, M., Bartoszuk, M., Cena, A.: Are cluster validity measures (in)valid? Inform. Sci. **581**, 620–636 (2021)
120. Chang, C., Dai, W.: A grey silhouette coefficient for the small sample forecasting. In: Proceedings of 2013 IEEE International Conference on Grey systems and Intelligent Services (2013)
121. Caliński, T., Harabasz, J.: Methods: a dendrite method for cluster analysis. Commun. Stat. Theory Methods **3**(1), 1–27 (1974)
122. Łukasik, S., Kowalski, P.A., Charytanowicz, M., Kulczycki, P.: Clustering using flower pollination algorithm and Calinski-Harabasz index. In: 2016 IEEE Congress on Evolutionary Computation (2016)
123. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intel. **2**, 224–227 (1979)
124. Lovino, M., Randazzo, V., Ciravegna, G., Barbiero, P., Ficarra, E., Cirrincione, G.: A survey on data integration for multi-omics sample clustering. Neurocomputing (2021). https://doi.org/10.1016/j.neucom.2021.11.094
125. Pakhira, M.K., Bandyopadhyay, S., Maulik, U.: Validity index for crisp and fuzzy clusters. Pattern Recognit. **37**(3), 487–501 (2004)
126. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
127. Zhang, S., Wong, H.S.: ARImp: A generalized adjusted rand index for cluster ensembles. In: 2010 20th International Conference on Pattern Recognition, 2010 of Conference
128. Gupta, A.K., Sardana, N.: Significance of clustering coefficient over jaccard index. In: 2015 Eighth International Conference on Contemporary Computing (IC3) (2015)
129. Lei, Y., Bezdek, J.C., Romano, S., Vinh, N.X., Chan, J., Bailey, J.: Ground truth bias in external cluster validity indices. Pattern Recognit. **65**, 58–70 (2017)
130. Powers, D.M.W.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Int. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
131. Limin, L., Junjie, W., Shiwei, Z.: Implication intensity: randomized F-measure for cluster evaluation. In: 2009 6th International Conference on Service Systems and Service Management, 8–10 June 2009 (2009)
132. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Stat. Assoc. **78**(383), 553–569 (1983)
133. Bihari, A., Tripathi, S., Deepak, A.: Gene expression analysis using clustering techniques and evaluation indices. In: Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) (2019)
134. Estevez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized mutual information feature selection. IEEE Trans. Neural Netw. **20**(2), 189–201 (2009)
135. Amelio, A., Pizzuti, C.: Correction for closeness: adjusting normalized mutual information measure for clustering comparison. Comput. Intell. **33**(3), 579–601 (2017)
136. Newman, M.E.J., Cantwell, G.T., Young, J.G.: Improved mutual information measure for clustering, classification, and community detection. Phys. Rev. E. **101**(4–1), 042304 (2020)
137. Gilanifar, M., Parvania, M.: Clustered multi-node learning of electric vehicle charging flexibility. Appl. Energy **282**, 116125 (2021)
138. Aadil, F., Ahsan, W., Rehman, Z.U., Shah, P.A., Rho, S., Mehmood, I.: Clustering algorithm for internet of vehicles (IoV) based on dragonfly optimizer (CAVDO). J. Supercomput. **74**(9), 4542–4567 (2018)
139. Mohamed, M.G., Saunier, N., Miranda-Moreno, L.F., Ukkusuri, S.V.: A clustering regression approach: a comprehensive injury severity analysis of pedestrian–vehicle crashes in New York, US and Montreal, Canada. Saf. Sci. **54**, 27–37 (2013)
140. Nopiah, Z.M., Junoh, A.K., Ariffin, A.K.: K-means clustering and neural network for evaluating sound level vibration in vehicle cabin. J Vib Control **21**(9), 1698–1720 (2013)
141. Liang, H., You, Y., Yang, L.: Research on electric vehicle cluster model based on scenes simulation. In: International Conference on Renewable Power Generation (RPG 2015), Beijing, China, 17–18 Oct (2015)
142. Ramea, K.: Unsupervised temporal clustering to monitor the performance of alternative fueling infrastructure. In: International Conference on Machine Learning (2019)
143. Nopiah, Z.M., Junoh, A.K., Ariffin, A.K.: Vehicle interior noise and vibration level assessment through the data clustering and hybrid classification model. Appl. Acoust. **87**, 9–22 (2015)
144. Yun, U., Ryang, H., Kwon, O.C.: Monitoring vehicle outliers based on clustering technique. Appl. Soft Comput. **49**, 845–860 (2016)
145. Nitsche, P., Thomas, P., Stuetz, R., Welsh, R.: Pre-crash scenarios at road junctions: a clustering method for car crash data. Accid. Anal. Prev. **107**, 137–151 (2017)
146. Shen, X., Zhang, Y., Sata, K., Shen, T.: Gaussian mixture model clustering-based knock threshold learning in automotive engines. IEEE ASME Trans. Mech. **25**(6), 2981–2991 (2020)
147. Qiu, L., Qian, L., Abdollahi, Z., Kong, Z., Pisu, P.: Engine-map-based predictive fuel-efficient control strategies for a group of connected vehicles. Automot. Innov. **1**(4), 311–319 (2018)
148. Gong, H., Wang, F., Zhou, B., Dent, S.: Application of random effects negative binomial model with clustered dataset for vehicle

crash frequency analysis. Int. J. Trans. Sci. Technol. **9**(3), 183–194 (2020)

149. Yang, H., Wang, Z., Xie, K., Dai, D.: Use of ubiquitous probe vehicle data for identifying secondary crashes. Transp. Res C Emerg. **82**, 138–160 (2017)
150. Nguyen, S.D., Nguyen, Q.H., Choi, S.B.: A hybrid clustering based fuzzy structure for vibration control—Part 2: an application to semi-active vehicle seat-suspension system. Mech. Syst. Signal Process. **56–57**, 288–301 (2015)
151. Li, L., Hansman, R.J., Palacios, R., Welsch, R.: Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring. Transp. Res C Emerg. **64**, 45–57 (2016)
152. Hasan, M.A., Islam, M.R., Tarefder, R.A.: Clustering vehicle class distribution and axle load spectra for mechanistic-empirical predicting pavement performance. Transp. Eng. J. ASCE. **142**(11), 05016006 (2016)
153. Rabbouch, H., Saâdaoui, F., Mraihi, R.: Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing. Neurocomputing **260**, 157–173 (2017)
154. Zhong, Z., Lee, E.E., Nejad, M., Lee, J.: Influence of CAV clustering strategies on mixed traffic flow characteristics: an analysis of vehicle trajectory data. Transp. Res. C Emerg. **115**, 102611 (2020)
155. Lin, K., Xia, F., Fortino, G.: Data-driven clustering for multimedia communication in Internet of vehicles. Future Gener. Comput. Syst. **94**, 610–619 (2019)
156. Ewbank, H., Wanke, P., Hadi-Vencheh, A.: An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem. Neural Comput. Appl. **27**(4), 857–867 (2015)
157. Besse, P.C., Guillouet, B., Loubes, J.-M., Royer, F.: Review and perspective for distance-based clustering of vehicle trajectories. IEEE Trans. Intell. Transp. Syst. **17**(11), 3306–3317 (2016)
158. Hong, Z., Chen, Y., Mahmassani, H.S., Xu, S.: Commuter ride-sharing using topology-based vehicle trajectory clustering: methodology, application and impact evaluation. Transp. Res. Part C Emerg. Technol. **85**, 573–590 (2017)
159. Hong, Z., Chen, Y., Mahmassani, H.S.: Recognizing network trip patterns using a spatio-temporal vehicle trajectory clustering algorithm. IEEE Trans. Intell. Transp. Syst. **19**(8), 2548–2557 (2018)
160. Li, J., Wang, H.: Preface for feature topic on intelligent safety for CAVs. Autom. Innov. **4**(3), 239–240 (2021)
161. Wang, Z., Liang, M., Delahaye, D.: A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area. Transp. Res C Emerg. **95**, 280–294 (2018)
162. Song, H., Wang, X., Hua, C., Wang, W., Guan, Q., Zhang, Z.: Vehicle trajectory clustering based on 3D information via a coarse-to-fine strategy. Soft. Comput. **22**(5), 1433–1444 (2017)
163. Prakoso, P.B., Sari, Y.: Vehicle detection using background subtraction and clustering algorithms. Telecommun. Comput. Electron. Control **17**(3), 1393 (2019)
164. Nguyen, T.T., Krishnakumari, P., Calvert, S.C., Vu, H.L., van Lint, H.: Feature extraction and clustering analysis of highway congestion. Transp. Res. Part C Emerg. Technol. **100**, 238–258 (2019)
165. Wang, W., Yu, S., Cao, W., Guo, K.: Review of in-vehicle optical fiber communication technology. Autom. Innov. (2022). https://doi.org/10.1007/s42154-022-00184-2
166. Dutta, A.K., Elhoseny, M., Dahiya, V., Shankar, K.: An efficient hierarchical clustering protocol for multihop Internet of vehicles communication. Trans. Emerg. Telecommun. Technol. **31**(5), e3690 (2019)
167. Senouci, O., Aliouat, Z., Harous, S.: DCA-DS: a distributed clustering algorithm based on dominating set for internet of vehicles. Wireless Pers. Commun. **115**(1), 401–413 (2020)
168. Tian, J., Wang, Y., Liu, C., Chen, Z.: Consistency evaluation and cluster analysis for lithium-ion battery pack in electric vehicles. Energy **194**, 116944 (2020)
169. Kang, L., Tang, W., Liu, Y., Daoutidis, P.: Control configuration synthesis using agglomerative hierarchical clustering: a graph-theoretic approach. J. Process Control **46**, 43–54 (2016)
170. Zheng, Z., Péra, M.-C., Hissel, D., Becherif, M., Agbli, K.-S., Li, Y.: A double-fuzzy diagnostic methodology dedicated to online fault diagnosis of proton exchange membrane fuel cell stacks. J. Power Sources **271**, 570–581 (2014)
171. Arias, M.B., Bae, S.: Electric vehicle charging demand forecasting model based on big data technologies. Appl. Energy. **183**, 327–339 (2016)
172. Reuß, M., Grube, T., Robinius, M., Stolten, D.: A hydrogen supply chain with spatial resolution: comparative analysis of infrastructure technologies in Germany. Appl. Energy **247**, 438–453 (2019)
173. Zhou, Y., Li, H., Ravey, A., Péra, M.-C.: An integrated predictive energy management for light-duty range-extended plug-in fuel cell electric vehicle. J. Power Sources **451**, 227780 (2020)
174. Zhang, J., Chu, L., Wang, X., Guo, C., Fu, Z., Zhao, D.: Optimal energy management strategy for plug-in hybrid electric vehicles based on a combined clustering analysis. Appl. Math. Model. **94**, 49–67 (2021)
175. Cai, W., Wu, X., Zhou, M., Liang, Y., Wang, Y.: Review and development of electric motor systems and electric powertrains for new energy vehicles. Autom. Innov. **4**(1), 3–22 (2021)
176. Zhang, Y., Chu, L., Fu, Z., Xu, N., Guo, C., Zhang, X., Chen, Z., Wang, P.: Optimal energy management strategy for parallel plug-in hybrid electric vehicle based on driving behavior analysis and real time traffic information prediction. Mechatronics **46**, 177–192 (2017)
177. Helmus, J.R., Lees, M.H., van den Hoed, R.: A data driven typology of electric vehicle user types and charging sessions. Transp. Res. C Emerg. **115**, 102637 (2020)
178. Jahangir, H., Gougheri, S.S., Vatandoust, B., Golkar, M.A., Ahmadian, A., Hajizadeh, A.: Plug-in electric vehicle behavior modeling in energy market: a novel deep learning-based approach with clustering technique. IEEE Trans. Smart Grid. **11**(6), 4738–4748 (2020)
179. Meng, W., Kai, L.: Location of electric vehicle charging station based on spatial clustering and multi-hierarchical fuzzy evaluation. Trans. Nanjing Univ. Aeronaut. Astronaut. **34**(1), 89–94 (2017)
180. Yun, L., Sandoval, J., Zhang, J., Gao, L., Garg, A., Wang, C.-T.: Lithium-Ion battery packs formation with improved electrochemical performance for electric vehicles: experimental and clustering analysis. J. Electrochem. Energy **16**(2), 021011 (2019)
181. Zhang, C., Qiu, Y., Chen, J., Li, Y., Liu, Z., Liu, Y., Zhang, J., Hwa, C.S.: A comprehensive review of electrochemical hybrid power supply systems and intelligent energy managements for unmanned aerial vehicles in public services. Energy AI. **9**, 100175 (2022)
182. Xu, J., Zhang, C., Wan, Z., Chen, X., Chan, S.H., Tu, Z.: Progress and perspectives of integrated thermal management systems in PEM fuel cell vehicles: a review. Renew. Sustain. Energy Rev. **155**, 111908 (2022)
183. Zheng, Z., Petrone, R., Pera, M. C., Hissel, D., Becherif, M., Pianese, C.: Diagnosis of a commercial PEM fuel cell stack via incomplete spectra and fuzzy clustering. In: IECON 2013—39th

Annual Conference of the IEEE Industrial Electronics Society, 10–13 Nov. 2013 (2013)

184. Mammar, K., Saadaoui, F., Laribi, S.: Design of a PEM fuel cell model for flooding and drying diagnosis using fuzzy logic clustering. Renew. Energy Focus. **30**, 123–130 (2019)
185. Fu, X., Xiang, Y., Chen, L., Xu, X., Li, X.: Solid oxide fuel cell anode image segmentation based on a novel quantum-inspired fuzzy clustering. J. Power Sources **300**, 57–68 (2015)
186. Li, X., Xu, X., Guo, C., Fu, X.: Three phases segmentation from Ni/YSZ anode optical microscopy images using quantum-inspired mixture clustering model. Opt. Eng. **57**(07), 073107 (2018)
187. Li, Q., Meng, X., Gao, F., Zhang, G., Chen, W.: Approximate cost-optimal energy management of hydrogen electric multiple unit trains using double Q-learning algorithm. IEEE Trans. Ind. Electron. (2021). https://doi.org/10.1109/tie.2021.3113021,1-1
188. Chen, Q., Gao, L., Dougal, R.A., Quan, S.: Multiple model predictive control for a hybrid proton exchange membrane fuel cell system. J. Power Sources **191**(2), 473–482 (2009)
189. Bankupalli, P.T., Ghosh, S., Sahu, L.K., Dwivedi, A.K.: Piecewise temperature dependent electrical equivalent modeling of PEM fuel cell for power conditioning unit design using fuzzy clustering and hybrid optimization. Energy Sources Part A (2021). https://doi.org/10.1080/15567036.2021.1903619
190. Rozin, B., Pereira-Ferrero, V.H., Lopes, L.T., Guimarães Pedronette, D.C.: A rank-based framework through manifold learning for improved clustering tasks. Inform. Sci. **580**, 202–220 (2021)
191. Clark, J.R., Stanton, N.A., Revell, K.M.A.: Automated vehicle handover interface design: focus groups with learner, intermediate and advanced drivers. Autom. Innov. **3**(1), 14–29 (2020)

Caizhi Zhang received a Ph.D. degree from Nanyang Technological University, in 2016. He joined Chongqing University (CQU) in 2016. He is currently a supervisor of doctoral students and the Leader of the Fuel Cell Vehicle Laboratory, College of Mechanical and Vehicle Engineering, CQU. He is also a researcher with the State Key Laboratory of Mechanical Transmissions and the Chongqing Automotive Collaborative Innovation Centre, CQU. He participated in and took charge of over ten national, provincial, and ministerial-level projects. He has published 45 articles. His research interests are fuel cells, machine learning, and intelligent fuel cell vehicle.

Weifeng Huang received a B.E. degree from the School of Automotive Engineering, Chongqing University in 2018. He is currently pursuing a master’s degree with the College of Mechanical and Vehicle Engineering, Chongqing University. His research interests include machine learning and the performance degradation of fuel cells.

Tong Niu received a B.E. degree from the School of Automotive Engineering, Shandong University of Technology in 2018. She is currently pursuing a Ph.D. degree with the College of Mechanical and Vehicle Engineering, Chongqing University. Her current research interests include machine learning and the aging and life prediction of fuel cells.

Zhitao Liu received a B.S. degree from Shandong University in 2005, and a Ph.D. degree in Control Science and Engineering from Zhejiang University in 2010. From 2011 to 2014, he was a Research Fellow with TUM CREATE, Singapore. Since 2017, he has been a professor at the Institute of Cyber-Systems and Control, Zhejiang University. His current research interests include robust adaptive control, wireless transfer systems, and energy management systems.

Guofa Li received a Ph.D. degree in Mechanical Engineering from Tsinghua University, Beijing, China, in 2016. He is currently a professor with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China. His research interests include environment perception, driver behavior analysis, and human-like decision-making and control based on artificial intelligence technologies in autonomous vehicles and intelligent transportation systems. He has published more than 70 papers in his research areas. He is the recipient of the Young Elite Scientists Sponsorship Program in China, and he receives the best paper awards from the China Association for Science and Technology (CAST) and the Automotive Innovation journal. In addition, he serves as the Associate Editor for IEEE Sensors Journal, as well as the lead guest editor for IEEE Intelligent Transportation Systems Magazine and Automotive Innovation.

Dongpu Cao received a Ph.D. degree from Concordia University in 2008. He is currently a professor at the School of Vehicle and Mobility, Tsinghua University. His current research interests include driver cognition, automated driving, and cognitive autonomous driving. He has contributed more than 200 publications, three books, and one patent. He received the SAE Arch T. Colwell Merit Award in 2012 and three Best Paper Awards from the ASME and IEEE conferences.