Applied Intelligence (2023) 53:21723–21742
https://doi.org/10.1007/s10489-023-04594-5

Metaheuristic-based time series clustering for anomaly detection in manufacturing industry

Woong Hyun Suh[1] · Sanghoun Oh[2] · Chang Wook Ahn[1]

Accepted: 26 March 2023 / Published online: 10 June 2023

Abstract
Nowadays time series clustering is of great importance in manufacturing industries. Meanwhile, it is considerably challenging to achieve explainable solution as well as significant performance due to computation complexity and variable diversity. To efficaciously handle the difficulty, this paper presents a novel metaheuristic-based time series clustering method which can improve the effectiveness and logicality of existing clustering approaches. The proposed method collects candidate cluster references from hierarchical and partitional clustering through shape-based distance measure as well as dynamic time warping (DTW) on manufacturing time series data. By applying metaheuristics highlighting estimation of distribution algorithms (EDA), such as extended compact genetic algorithm (ECGA), on the collected candidate clusters, advanced cluster centroid combinations with minimal distances can be achieved. ECGA employs the least complicated and the most closely related probabilistic model structure regarding population space during generation cycle. This feature strengthens the comprehension of clustering results in how such optimal solutions were achieved. The proposed method was tested on real-world time series data, open to the public, from manufacturing industry, and showed noticeable performances compared to well-established methods. Accordingly, this paper demonstrates that obtaining both comprehensible result as well as prominent performance is feasible by employing metaheuristic techniques to time series data clustering methods.

Keywords Metaheuristics · Time series clustering · Anomaly detection · Manufacturing industry · Evolutionary algorithm · Estimation of distribution algorithm · Extended compact genetic algorithm · Metaheuristic clustering · Explainable artificial intelligence

1 Introduction

Artificial intelligence and digital transformation are the megatrends shaping the future of the manufacturing industry. Rapid development of both information technology and operational technology are leading fast transition from traditional manufacturing process to future smart factory. Such transition shares the progress with technological improvements in system automation, sensor technology, cloud computing, big data processing and analysis [1, 2]. Ultimately, future smart factories will aim for remarkable advances in manufacturing productivity as well as production quality through comprehensive utilization of digital data and artificial intelligence techniques.

Especially, detecting anomalous cases in production, by analyzing massive data collected from smart sensors during the manufacturing process, is a fashionable area of research soaking up both academic and industrial interests. Recent technical advances in data processing and analysis enable detecting production losses that were unable to capture in the past. This progressive approach is where mass digital data and artificial intelligence methods converge effectively and improve the overall manufacturing productivity [3, 4]. These kinds of anomaly detectionmethods through artificial

Corresponding authors are Sanghoun Oh and Chang Wook Ahn.

✉ Sanghoun Oh
oosshoun@knou.ac.kr

✉ Chang Wook Ahn
cwan@gist.ac.kr

[1] AI Graduate School, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Korea

[2] Department of Computer Science, Korea National Open University, Seoul 03087, Korea

intelligence techniques can be divided into supervised and unsupervised learning methods.

Data clustering techniques are representative unsupervised learning methods. It is specialized in grouping data that have similar features and patterns without any given data labels or tags. They have been utilized in various industries for the use of image classification, recommender systems, and anomaly detection. Deep learning, one of the representative supervised learning methods, is a pattern recognizing data analysis method that employs neural networks. Deep learning shows powerful performances in many real-world industrial domains specialized in image and video based data pattern recognitions. While neural network based methods show outstanding performances, they require a high amount of computing load and yield outcomes which have deficient analytical basis. This is because feature vectors and variables inside neural networks are tremendously huge that tracing analytical bases of the computation results is exceptionally challenging [5, 6].

This paper presents an effective time series clustering method highlighting estimation of distribution algorithm (EDA) for manufacturing time series data. The proposed method collects sub-clusters from both hierarchical and partitional clustering results through shape-based distance measure as well as dynamic time warping (DTW) on time series data. On the collected sub-cluster references, metaheuristic algorithms search for the most optimal cluster group. With the use of search algorithms derived from evolutionary computations, this paper proposes an explainable data clustering method for anomaly detection on manufacturing time series data with prominent performance.

This paper has three contributions. First, a novel metaheuristic-based time series clustering method for anomaly detection in manufacturing is suggested. Compared to existing time series clustering techniques, the proposed method shows noticeable performance on time series data clustering problems. Secondly, this paper elaborated the explainability of time series data clustering methods for anomaly detection. Grouping solutions with linkage information regarding the sub-cluster references are provided. Through metaheuristic-based clustering with the use of EDA, such as ECGA, the proposed algorithm derives dependent and explainable solutions which deliver logical relationships regarding how the collected clustering references have been achieved. Lastly, in order to demonstrate the applicability of the proposed method to real-world problems, this paper tested the proposed algorithm on real-world manufacturing time series data and showed the validity of its technique. This paper deems that the research in this paper can facilitate the reliable use of artificial intelligence technology as well as mass digital data in real-world industry settings.

The rest of this paper is organized as follows. Section 2 contains a brief overview and literature review on time series clustering methods as well as metaheuristic algorithms in order to demonstrate the subtle domain of metaheuristic time series clustering. Section 3 contains the proposed method of this paper which highlights the use of ECGA. Section 4 contains the experimental results and interpretations, and Section 5 includes conclusions and policy implications.

2 Background and related works

The scope of concepts that time series clustering as well as metaheuristics cover is extensive. In this Section, a succinct overview is provided regarding where the proposed method is placed among the two large techniques along with detailed descriptions on the methodologies that are used.

2.1 Time series clustering

Time series clustering techniques can be analyzed based on characteristics such as data representation, clustering techniques, evaluation criteria, and similarity measures. In respect to the huge scope that time series clustering covers, it can be classified as Fig. 1 [7, 8]. Representative time series clustering techniques, which this paper focuses on, can be divided into three methods: hierarchical clustering, partitional clustering, and model-based clustering. Being a domain of research with broad applicability, time series clustering is utilized in a large scope of academic and industrial domains, where the majority of applications are focused on time series classification as well as anomaly detection [7]. Diverse industries, which yield meaningful time series data, show active application of time series clustering, where those industries include astronomy, biology, finance, medicine, or robotics [1–4].

Hierarchical clustering, for instance, assumes initial data points as individual clusters. It compares the distance between each data and successively merge or split clusters based on given distance measures. Hierarchical clustering is powerful when there is no prior knowledge about the data that it does not require a predefined number of clusters for training data. However, when time series data is large and complex, computational load increases sharply. In such cases, organized small-scale data is required in order to train within an estimable amount of computational capability [7]. Partitional clustering, on the other hand, shows rapid computation time compared to hierarchical clustering. It can be combined with fuzzy logic based algorithms for powerful performances as well. As partitional clustering requires a predefined number of clusters for data training, it lacks

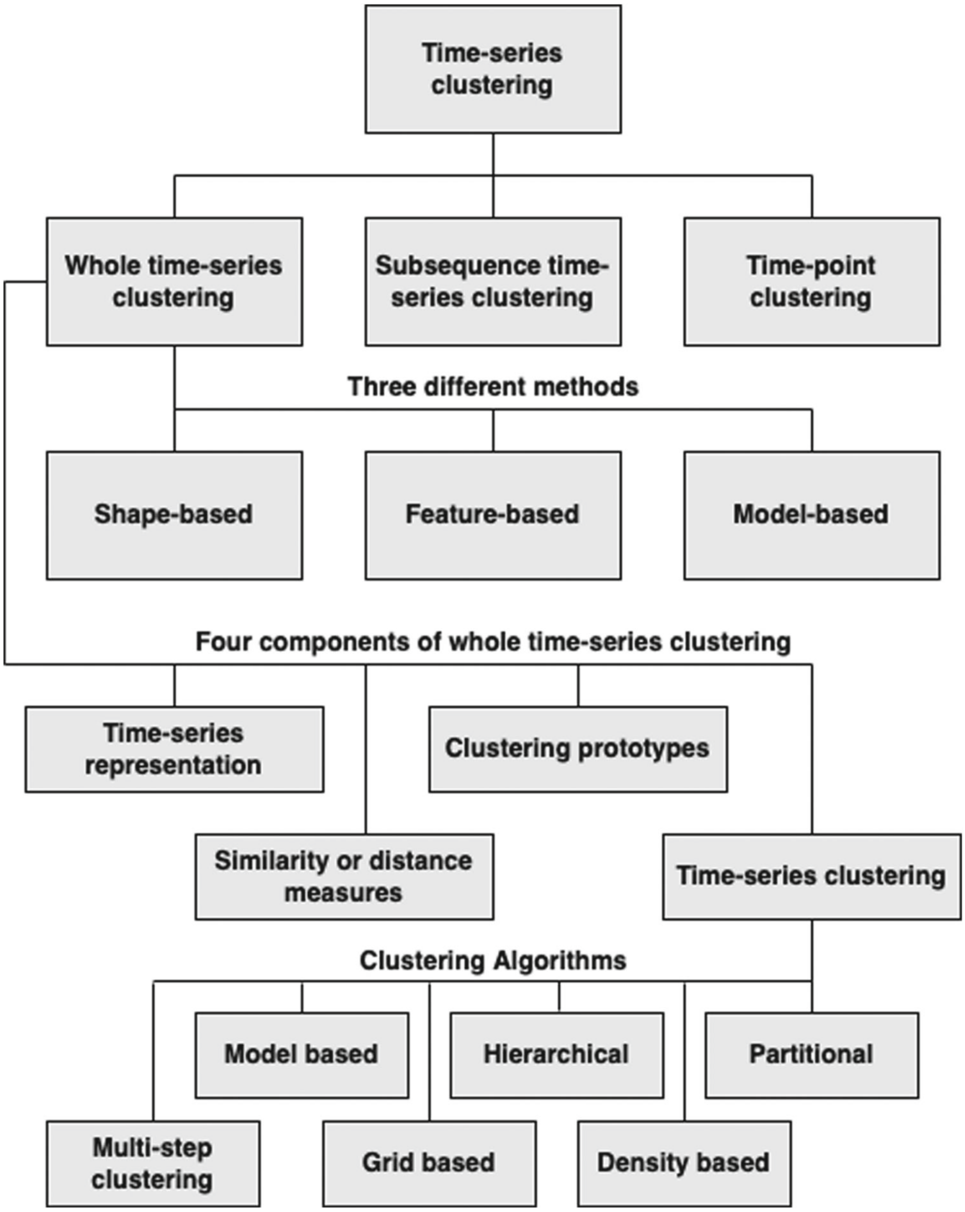
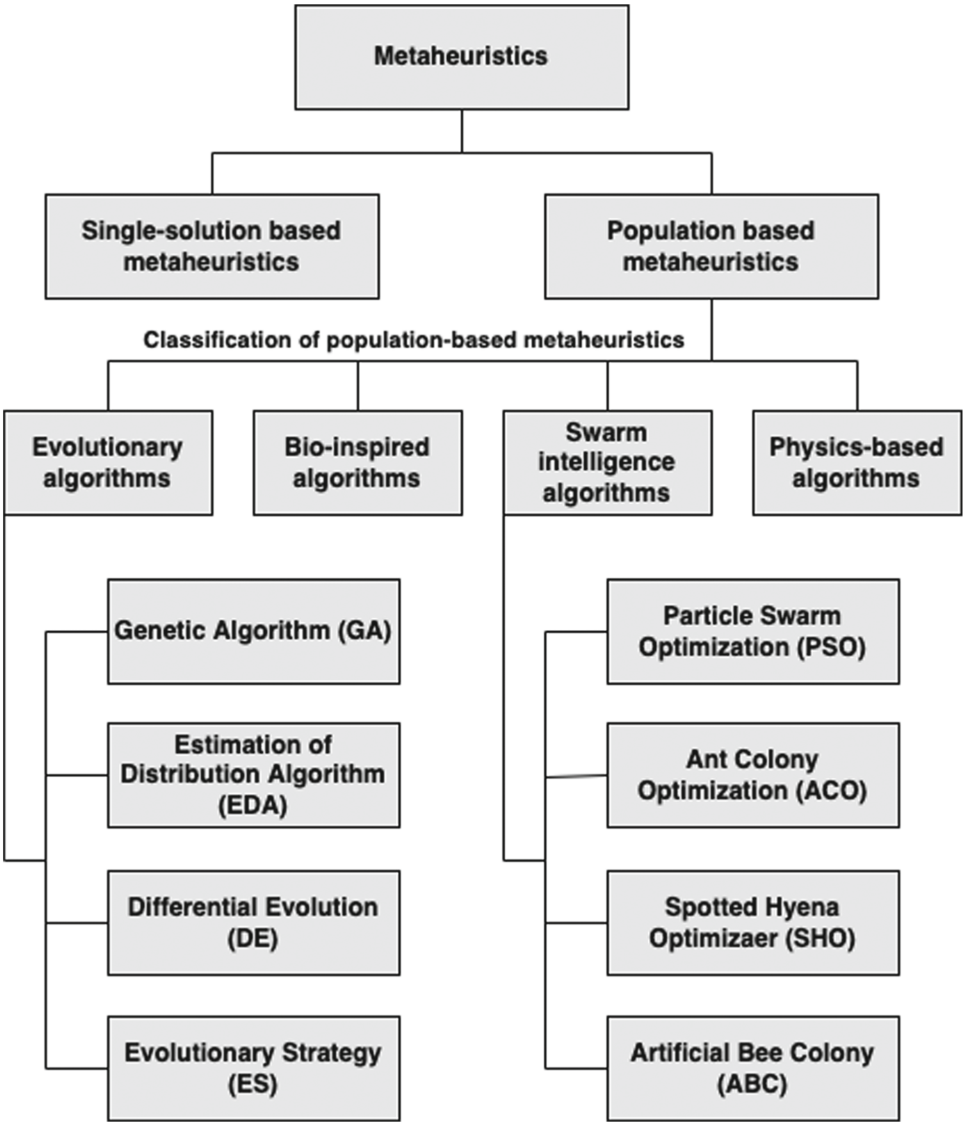


Fig. 1 Time series clustering overview

analytical consistency and, similar to hierarchical clustering, requires sharp increase in computation time for analyzing huge data [7, 9]. Model-based clustering attempts to find the most optimal mathematical-fit on time series data. Model-based clustering is a statistical, as well as neural network based, approach which operates with the rationale that there are finite mixture of underlying probability distribution that constructs the data [10]. It fundamentally assumes that time series data were previously generated by mathematical or probabilistic models.

Various research works approached to improve time series clustering techniques. Lai re-represented time series data into sub-sequences while preserving the fundamental characteristics of the data [11]. The overall sequence and sub-sequence of time series data are considered in two different stages of time series clustering. The given algorithm used symbolic aggregate approximation to reduce data dimension, and used cluster affinity search technique for data clustering algorithm. For a similarity measure of given time series data, Lai used dynamic time warping when lengths of time series data are different and used Euclidean distance measure when the distances of time series data are identical. Munir, for instance, proposed a novel deep learning approach for shape-based time series data clustering [12]. The proposed method analyzes time series data through initially created networks as well as the closest neighboring network. Then, it creates a time series network based on Euclidean distance for similarity distance measure. Meanwhile, deep learning for time

Fig. 2 Metaheuristic overview


Metaheuristics
Single-solution based metaheuristics
Population based metaheuristics
Classification of population-based metaheuristics
Evolutionary algorithms
Bio-inspired algorithms
Swarm intelligence algorithms
Physics-based algorithms
Genetic Algorithm (GA)
Estimation of Distribution Algorithm (EDA)
Differential Evolution (DE)
Evolutionary Strategy (ES)
Particle Swarm Optimization (PSO)
Ant Colony Optimization (ACO)
Spotted Hyena Optimizaer (SHO)
Artificial Bee Colony (ABC)

series classification is one of the most trending research in time series clustering. Fawaz divided deep learning methodologies for time series clustering into generative models and discriminative models; and conducted comprehensive comparison tests of deep learning models on time series data [13]. Those deep learning algorithms encompass models such as multi layer perceptrons (MLP), ResNet, Encoder, convolutional neural networks (CNN) and other prevalent algorithms. By testing such models on diverse real-world time series data - both univariate and multivariate - from devices, sensors, images, or motions; Fawaz demonstrates significant competitiveness of deep learning models in time series classification problems.

This paper employs univariate as well as raw data, which has different lengths of sequences, with shape-based distance and dynamic time warping distance measure. In order to collect candidate cluster references, partitional and hierarchical clustering algorithms are used, which are long-standing and popular clustering algorithms not only in generic data clustering but also in time series clustering. Then, metaheuristics are utilized to search for optimal cluster centroids as well as a statistical approach to facilitate anomaly detection application. This paper compares the experiment results with other machine learning and deep learning classification methods, namely model-based clustering, to demonstrate the effectiveness of the proposed method.

2.2 Metaheuristics

Algorithms highly motivated by nature to solve complex real-world problems account for the substance of metaheuristics. Many related algorithms support the large concept of metaheuristics such as evolutionary algorithms, bio-inspired algo-

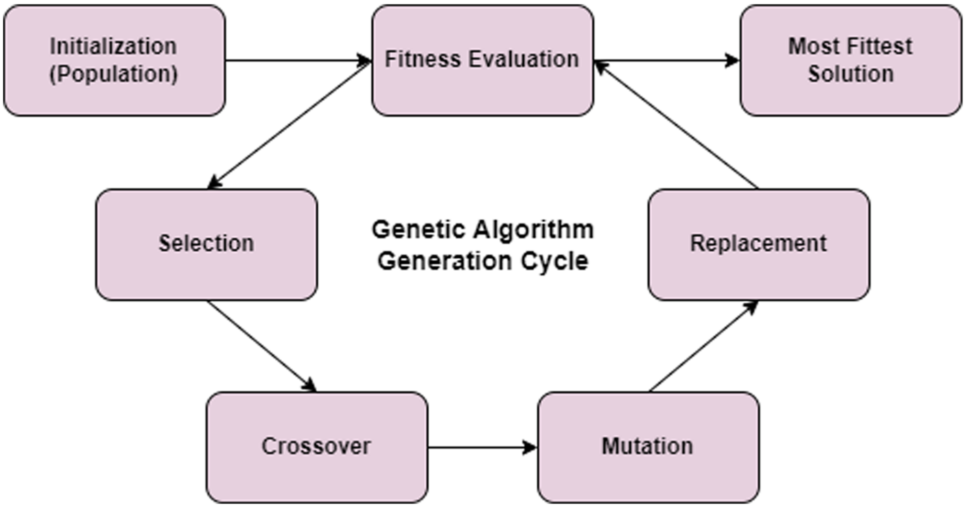


Fig. 3 Simple genetic algorithm (GA) generation cycle and operators

rithms, swarm intelligence algorithms, and physics-based algorithms. As depicted in Fig. 2, the massive concept of metaheuristics can be divided into two categories: single-solution based and population-based approaches [14, 15]. Single-solution based metaheuristic algorithms, such as tabu search (TS) or microcanonical annealing (MA), concentrate on improving single candidate solutions through local search. This relates to the main weakness of such algorithms that the best solution may be stuck in local optima. Population-based metaheuristic algorithms, on the other hand, utilize multiple candidate solutions in order to search for the most optimal solution. This feature makes population-based methods, such as genetic algorithm (GA) or particle swarm optimization (PSO), much more advantageous in searching for global optima compared to single-solution based approaches.

Inspired by Darwin's natural selection theory, genetic algorithm (GA) is one of the most well-known evolutionary algorithms which emulates Darwin's idea of survival of the fittest [16, 17]. GA functions with elements such as chromosome representation, fitness selection, and bio-inspired operators. The bio-inspired operators can be specified into gene (or chromosome) encoding, selection, crossover and mutation, like the process of reproducing next generation offspring in nature. In GA, population is the search space, namely the set of candidate chromosomes, to find the most optimal solution. Here, GA, as well as other evolutionary algorithms, requires specific techniques in each component such as binary encoding, tournament selection, uniform crossover or displacement mutation; and predefined parameters such as fitness function, mutation and crossover rates. The overall cycle of simple genetic algorithms and operators is as described in Fig. 3 [18].

One of the representative swarm intelligence algorithms is particle swarm optimization (PSO) algorithm. Simulating the social behaviors of organisms such as birds or fishes, PSO aims to search for the most optimal solution by employing particles, similar to chromosomes in GA, on population space according to stochastic formulas on their position and velocity [19–21]. Each particle position is considered as a candidate solution for the target problem and the rate of change in position is treated as velocity. The position giving the best fitness value is stored as the best particle for the most optimal solution. Binary particle swarm optimization (BPSO) algorithm is PSO operating with binary variables,

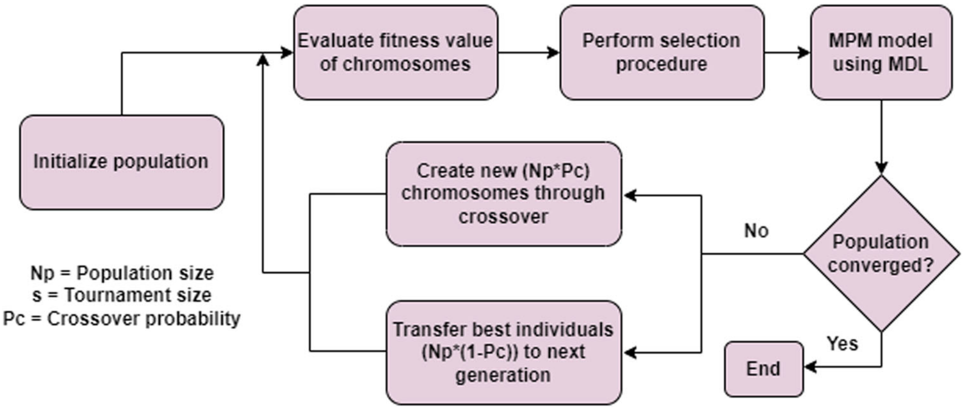

Evaluate fitness value of chromosomes
Perform selection procedure
MPM model using MDL
Initialize population
Create new (Np*Pc) chromosomes through crossover
No
Population converged?
Yes
End
Transfer best individuals (Np*(1-Pc)) to next generation
Np = Population size
s = Tournament size
Pc = Crossover probability

Fig. 4 Extended compact genetic algorithm (ECGA) outline

namely binary encoding, on population space where individuals are represented by binary bits [22, 23]. PSO, as well as BPSO, has advantages of computational efficiency, robust control parameters, and easy computation [20, 24]. At the same time, PSO algorithms may have low convergence rate and may fall into local optima easily for high-dimensional problems.

Estimation of distribution algorithm (EDA) is a probabilistic optimization approach that searches for best solutions in population space through stochastic models [25]. The main characteristic of EDA that differs from other meta-heuristics is that EDA attempts to build probabilistic models towards promising solutions based on trending distribution of selected solutions from previous generations. While EDA possesses high efficiency and applicability with trade-offs in computational complexities depending on the target problem that one aims to solve; as Hauschild and Pelikan suggested, EDA algorithms have strengths in not only solving difficult problems, but also in showing how such problems were solved [26]. This paper refers to the described aspects of EDAs to the artificial intelligence domain in order to enhance logical comprehension of data training outcomes.

Additionally, EDAs can be categorized into univariate models, which assume problem variables are independent of any other variables, and multivariate models, which capture correlated interactions among problem variables. Compact genetic algorithm (CGA) is an incremental univariate EDA which resembles population-based incremental learning (PBIL) [27–29]. Unlike other EDAs bequeath population space successively to the next generation, incremental EDAs replace population based on stochastic models derived from the previous generation. CGA and PBIL use probability vectors in order to represent the entire solutions in population space where fixed-length binary strings are used for encoding. While PBIL changes entire probability vectors in each generation, CGA changes probability vectors when only there exists a winning solution in each generation.

Extended compact genetic algorithm (ECGA), one of multivariate EDAs, was proposed by Harik with a key idea that choosing a good probability distribution over population space shall be equivalent to linkage learning in genetic algorithm (GA) [30–33]. Linkage learning in GA highlights building blocks of chromosomes that should be conserved under crossover operation for effective search of optimal solution. Here, quantifying the quality of a distribution is determined by minimum description length (MDL) models. When all things are equal, MDL models assume that simpler distributions are better than complex ones. Using MDL models, both inaccurate and complex models are penalized which results in an optimal probability distribution. This probability distribution is one of a class of marginal product models (MPM). MPMs in CGA and PBIL are measured as a product of marginal distributions on singular gene partition; on the other hand, MPMs in ECGA represent the probability distributions for multiple genes simultaneously. The overall process of ECGA is depicted in Fig. 4, and simple pseudo-outline of ECGA is as below [34]:

1. Initialization: create population randomly or use other initialization methods.
2. Fitness: evaluate fitness values of individuals in the population based on fitness function.
3. Selection: choose individual chromosomes through methods such as tournament selection or other selection procedures.
4. Build the probabilistic model: ECGA searches for both the structure and the parameters of probabilistic models. ECGA selects the model with lowest combined complexity and replaces it if there is a better one compared to previously selected model.
5. Create new individuals: produce new individuals by sampling probabilistic model on population.
6. Replace the parental population with the offspring population.
7. Repeat steps 2-6 until a convergence criteria is met.

In ECGA, marginal product models (MPMs) using minimum description length (MDL) models as well as creating new populations based on the probabilistic model are the core elements. This can be described as below.

$$\text{Minimize:} \qquad C_m + C_p \tag{1}$$

$$\text{Subject to:} \quad 2^{l_{bb,i}} \leq N_p \quad \forall i \in [1, N_{bb}] \tag{2}$$

In equations from (1) to (4), C_m represents model complexity, namely the cost of a complex model, and C_p is the compressed population complexity, namely the cost of using a simple model compared to using a complex model [31].

$$C_m = \log_2\left(N_p + 1\right)\sum_{i=1}^{N_{bb}}\left(2^{l_{bb,i}} - 1\right) \tag{3}$$

$$C_p = \sum_{i=1}^{N_{bb}}\sum_{j=1}^{2^{l}bb,i} N_{ij}\log_2\left(\frac{N_p}{N_{ij}}\right) \tag{4}$$

N_p is the size of population and N_{bb} represents the number of building blocks. $l_{bb,i}$ is the length of building blocks, where $i \in [1, N_{bb}]$, and N_{ij} is the number of chromosomes in the current population space possessing bit-sequence $j \in \left[1, 2^{l_{bb,i}}\right]$ for building block i. With optimal MPM calculated, a new population is filled with best individuals by size of $N_p{*}(1 - P_c)$, where P_c is the crossover probability. These subsets are the groups identified by the MPM models and the rest individuals are generated randomly.

This paper highlights the use of EDA, especially ECGA, on time series clustering techniques. While application of metaheuristics on time series clustering is considered to be very few, evaluating time series clustering techniques based on the use of optimization methods has been concentrated on algorithms based on GA and PSO [8, 35]. The use of a probabilistic model, such as ECGA, in optimizing time series clustering techniques is considered to be unprecedented. This paper expects that the proposed method will enhance the understanding of data analysis results in the artificial intelligence domain regarding real-world problem solving.

2.3 Metaheuristic-based clustering

In this paragraph, the subtle domain of metaheuristic-based clustering techniques is briefly examined. Efficient search and effective optimization are the main aspects that data clustering methods adopt from metaheuristics, in order to overcome limitations such as inefficiency in distance measure or computational complexity [8]. Recent algorithms which advance the technical capacity of metaheuristics include red fox optimization algorithm, adopting the model of red fox habits, as well as polar bear algorithm, adopting polar bear's survival mechanism [48, 49]. There are various convergences betweenmetaheuristic algorithms and

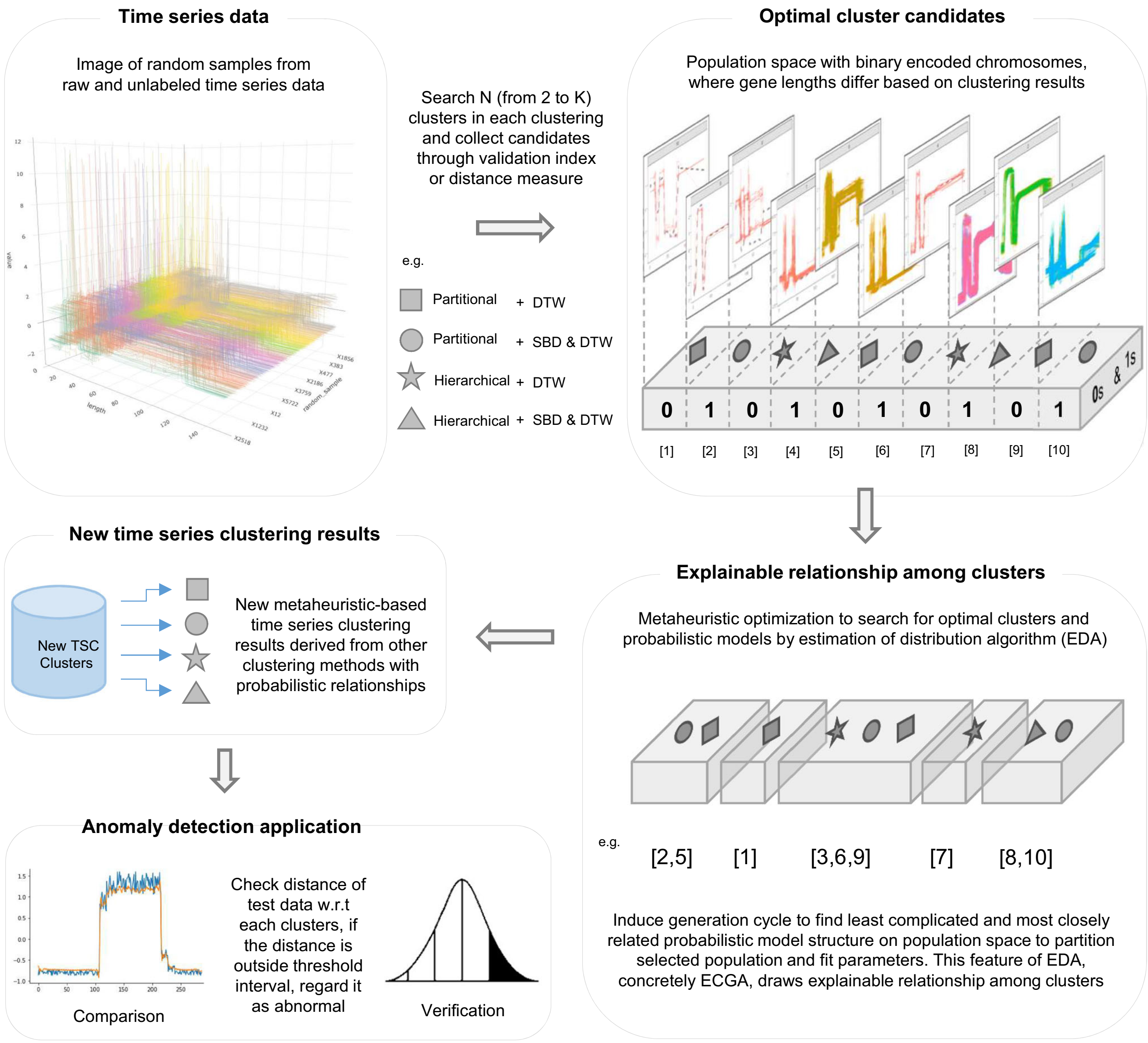

Time series data
Image of random samples from raw and unlabeled time series data
value
length
random_sample
Search N (from 2 to K) clusters in each clustering and collect candidates through validation index or distance measure
e.g.
Partitional + DTW
Partitional + SBD & DTW
Hierarchical + DTW
Hierarchical + SBD & DTW
Optimal cluster candidates
Population space with binary encoded chromosomes, where gene lengths differ based on clustering results
0 1 0 1 0 1 0 1 0 1
0s & 1s
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
Explainable relationship among clusters
Metaheuristic optimization to search for optimal clusters and probabilistic models by estimation of distribution algorithm (EDA)
e.g.
[2,5] [1] [3,6,9] [7] [8,10]
Induce generation cycle to find least complicated and most closely related probabilistic model structure on population space to partition selected population and fit parameters. This feature of EDA, concretely ECGA, draws explainable relationship among clusters
New time series clustering results
New TSC Clusters
New metaheuristic-based time series clustering results derived from other clustering methods with probabilistic relationships
Anomaly detection application
Check distance of test data w.r.t each clusters, if the distance is outside threshold interval, regard it as abnormal
Comparison
Verification

Fig. 5 Illustration of the proposed metaheuristic-based time series clustering method

clustering techniques. For instance, metaheuristics can be used to search for the most appropriate fixed-number of cluster centers by optimizing the initialization process, and to enhance the effectiveness of distance measures, or to reduce dimensions in time series data representation [36, 37]. Accordingly, deep learning methods, such as deep autoencoder networks, can be optimized with genetic algorithms for time series data clustering [38]. Recent relevant research includes short-term forecasting models with hybrid use of machine learning methods and support vector regression with grey catastrophe and random forest [50]. Real-world applications of metaheuristic clustering include distributed denial of service (DDoS) detection in order to protect network in cyber security, healthcare data collection in mobile wireless multimedia sensor networks, hybrid time series clustering based on evolutionary algorithms with pearson distance measure for manufacturing time series data, and evolutionary computation based manufacturing big data analysis [39–42]. The proposed method of this paper intends to expand both the theoretical utilization as well as the range of application of metaheuristics for time series clustering techniques.

3 Proposed method

The proposed framework can be divided into two steps: time series clustering stage and metaheuristic optimization stage. The main idea of the proposed method is to find the most optimal cluster combinations from the cluster candidate pool, with analytical bases, for the purpose of applying clustering results to anomaly detection. The overall process is depicted in Fig. 5 and the flowchart is described in Fig. 6. The algorithm aims to find optimal cluster centroids with shorter distances, namely advancing the existing clustering results, on time series data. Also, The proposed method derives how such a solution has been achieved, by highlighting ECGA, to deliver analytical substance for clustering results.

The initial stage of the proposed method begins with processing various types of time series clustering techniques on raw and unlabeled time series data. Primarily, it expands the pool of potentially best clusters by collecting time series clustering results. In the proposed framework, diverse time series clustering techniques as well as distance or similarity measures can be used to provide potential cluster references. In this paper, the mixture of long-standing and effective time series clustering techniques are used: k-means and hierarchical clustering with shape-based distance (SBD) and dynamic time warping (DTW).

One of the most popular distance measures in time series sequences shall be DTW [43, 44]. DTW can be thought of as an extended approach from Euclidean distance (ED). DTW is effective when time series sequences vary in time or speed.

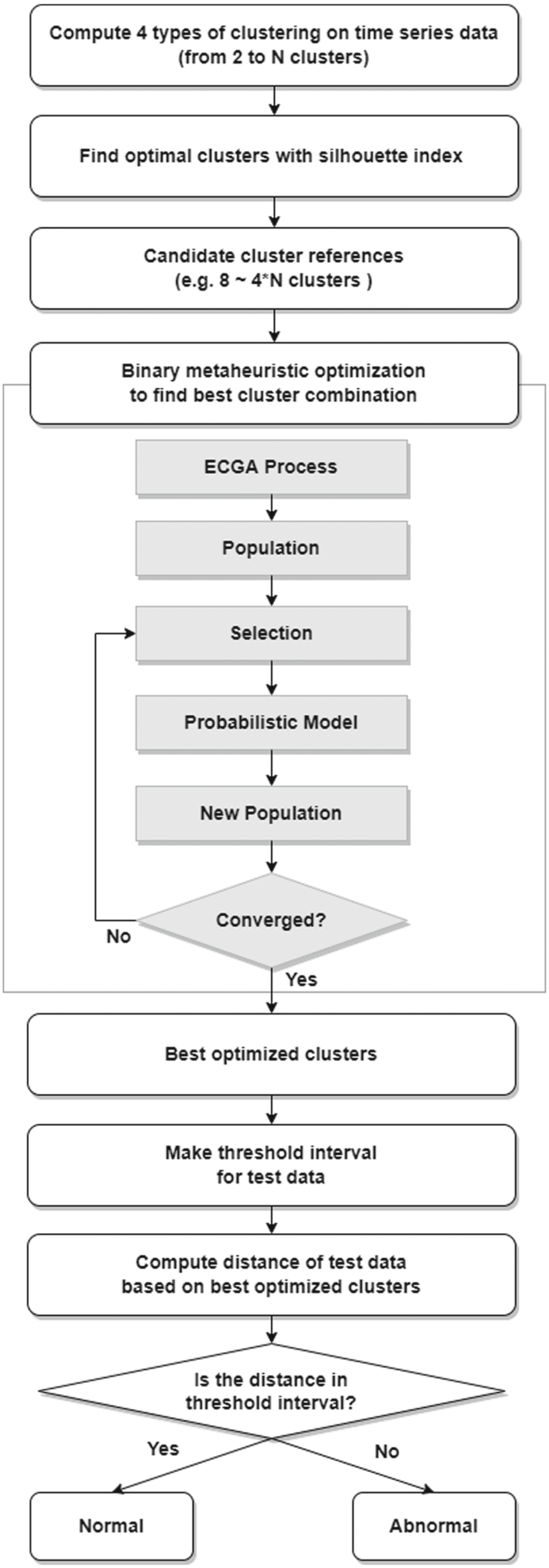

Compute 4 types of clustering on time series data
(from 2 to N clusters)
Find optimal clusters with silhouette index
Candidate cluster references
(e.g. 8 ~ 4*N clusters)
Binary metaheuristic optimization
to find best cluster combination
ECGA Process
Population
Selection
Probabilistic Model
New Population
Converged?
No
Yes
Best optimized clusters
Make threshold interval
for test data
Compute distance of test data
based on best optimized clusters
Is the distance in
threshold interval?
Yes
No
Normal
Abnormal

Fig. 6 Flowchart of the proposed metaheuristic-based time series clustering algorithm

Through warping path $W = w_1, w_2, ..., w_k$, with $k >= m$, achieved from m by m matrix M with ED between any two points of two time series sequence vectors; DTW maps two sequence vectors in the case where lengths of two sequences differ. This path is evaluated through the following recurrent equation of evaluation, $\gamma(i, j)$, which can be computed through dynamic programming. In many cases, the model can be constrained to visit only a limited subset of the warping path on matrix M for practical computation.

$$ED(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{5}$$

$$DTW(\vec{x}, \vec{y}) = \min \sqrt{\sum_{i=1}^{k} w_i} \tag{6}$$

$$\gamma(i, j) = ED(i, j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{7}$$

SBD is a distance measure based on normalized cross-correlation proposed by Paparrizos and Gravano for time series clustering method called k-shape clustering algorithm [45]. SBD uses distance measure through cross-correlation which can be thought of as time lagged correlation. For time series sequences that differ in length, SBD adds zero paddings to match the length of sequences. Then, SBD calculates ED to find maximal similarity by cross-correlation. In order to enhance the performance of SBD, employing z-normalization is a common case to match the scale of time series sequences. With time series sequence vectors $\vec{x}_{(s)}$, where $s \in [-m, m]$, SBD creates cross-correlation $CC_w(\vec{x}, \vec{y}) = (c_1, \ldots, c_w)$, which the length is $2m - 1$. The cross-correlation is defined as $CC_w(\vec{x}, \vec{y}) = R_{w-m}(\vec{x}, \vec{y})$, $w \in \{1, 2, \ldots, 2m - 1\}$, where $R_{w-m}(\vec{x}, \vec{y})$ is computed as:

$$R_k(\vec{x}, \vec{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k} \cdot y_l, & k \geq 0 \\ R_{-k}(\vec{y}, \vec{x}), & k < 0 \end{cases} \tag{8}$$

Then, SBD can be derived as:

$$SBD(\vec{x}, \vec{y}) = 1 - \max_{w} \left(\frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}) \cdot R_0(\vec{y}, \vec{y})}} \right) \tag{9}$$

After computing clustering results, the proposed method finds the optimal number of clusters in each given clustering result using Silhouette analysis as cluster validity index [46]. Silhouette index is a method used for validating consistency within clusters of data. The silhouette value is a measure of how similar an object is to its own cluster in comparison to other clusters. This can be achieved by calculating silhouette index for each sample, average silhouette index for each cluster, and overall average index for a dataset. The coefficient is calculated by utilizing mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample.

$$\text{Silhoutte index} = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{10}$$

The silhouette index is defined as equation (10) which $a(i)$ is the average dissimilarity of i^{th} sample to all other samples in the same cluster and $b(i)$ is the average dissimilarity of i^{th} sample with all samples in the closest cluster. The index value lies between $[-1, 1]$. If the value is close to 1, the sample is well-clustered; if the value is close to 0, the sample can be assigned to a different cluster; and if the value is close to -1, the sample is in the wrong cluster. By using silhouette width as cluster validity index, the proposed method finds and collects cluster centroids in order to make candidate references for optimal clusters.

Based on the collected candidate cluster references, the proposed method searches for the most optimal cluster centroid combinations through metaheuristics. By considering given candidate centroids for the population pool, metaheuristic algorithms search for the best cluster combination through a binary encoded chromosome. In other words, the size of the population is 2^n, where n is the length of cluster references that may differ based on previous clustering results. Accordingly, clusters represented as “1” are considered for fitness value calculation, and clusters represented as “0” are neglected for fitness value calculation. Both the length of the chromosome and the number of each clustering method in the chromosome vary according to the clustering results from the previous step.

For metaheuristic optimization, this paper employed GA, CGA, BPSO, and ECGA based on binary encoded chromosomes with tournament selection and average-max fitness function to find the most well-clustered centroid combinations. As mentioned in Section 2.2, many uses of metaheuristics in time series clustering techniques are focused on GA and PSO algorithms, and the use of EDAs such as CGA and ECGA in time series clustering is considered to be very few. Compared to other algorithms, ECGA holds a distinctive characteristic delivering how the algorithm has achieved such solutions by grouping population through probabilistic models which highlight trending as well as relevant candidates in population pool.

ECGA induces the search process in every generation cycle to find least complicated and most closely related probabilistic model structure on population space to partition selected population and fit parameters. In other words, as the chromosome is considered to be the collected references of diverse time series clustering results, by searching optimal combinations and providing logical bases, enhancing performances and having explainable relationships among clusters are feasible simultaneously. This feature of EDA, concretely ECGA, draws logical solutions for data analysis.

Through metaheuristics, newly optimized time series clustering centroids derived from previously computed time series clustering methods are achieved. With given new time series clustering results, this paper applies them for anomaly detection. Each test data input is checked with the distance in respect to each cluster. If the distance is outside the threshold interval, it is considered as abnormal, and if the distance is within the threshold interval it is considered as normal. This paper refers to the statistical approach for building threshold intervals. More detailed descriptions on the overall framework can be found in Algorithms 1 and 2 as well as Section 4

This paragraph summarizes the main features of the proposed method. One of the main characteristics of the proposed research is that the input data for this framework is raw and unlabeled. Data does not need to be labelled to discern clusters among data, unlike supervised learning. Also, as long-standing clustering approaches require iterative experiments for optimal solutions or prior knowledge to set a predefined number of clusters, this paper proposes metaheuristic-based clustering method which collects clustering references to find the most optimal clusters. In addition, to the knowledge of this paper, there are no existing approaches that link or group candidate clusters by probabilistic models to find dependent and explainable solutions. These features enable analyzing time series data without prior knowledge as well as delivering explainable solutions, while maintaining outstanding performances.

Algorithm 1 Metaheuristic-based time series clustering framework

```
Require: maximum cluster range number $K$, population rate $r$, fitness function $f$, crossover rate $c$, tournament size $t$, selection size $s$

Initialization: compute various types of time series clustering
(optional) normalize time series data, if needed
1. partitional clustering 2:K with DTW
2. hierarchical clustering 2:K with DTW
3. partitional clustering 2:K with SBD and DTW
4. hierarchical clustering 2:K with SBD and DTW
find tentatively optimal clusters with Silhouette index
collect cluster centroids as candidate references

Optimization: metaheuristic search by ECGA
population size = r * (cluster candidate length)
max generation = 1.5 * sqrt(population)
chromosome size = length of candidate references
tournament size = t
selection size = s
initialize population
set fitness value of population
calculate initial model with beginning population
while $i$ < max generation do
    model = marginal_product_model(population, model)
    ▷ Algorithm 2
    new_model = link_group(model)
    if new_model has converged then
        break;
    else
        model = new_model;
    end
    offspring = shuffle(population, model)
    population_pool = tournament_selection(t, s, offspring)
    evaluate = fitness_function(population_pool)
    i += 1
end

Application: anomaly detection test
set threshold interval based on optimization results
interval = mean clustering result +− 2 or 3 sigma values
if test data is within interval then
    normal;
else
    abnormal;
end
```

Algorithm 2 Marginal product model

```
Minimum Description Length (MDL): compute model and population complexity
$N_p$ = size of population
$L$ = list of building block lengths
$C_m = log_2(N_p + 1) * \text{sum}(2^L - 1)$
for $i$ in number of building blocks do
    $l_i$ = length of building block $i$
    for $j$ in $2^{l_i}$ do
        $N_{i_j}$ = number of chromosomes in current population
        where $j \in [1, 2^{l_i}]$
        $C_p = \text{sum}(N_{i_j} * log_2(N_p / N_{i_j}))$
    end
end
mdl = $C_m + C_p$
```

4 Experimental results

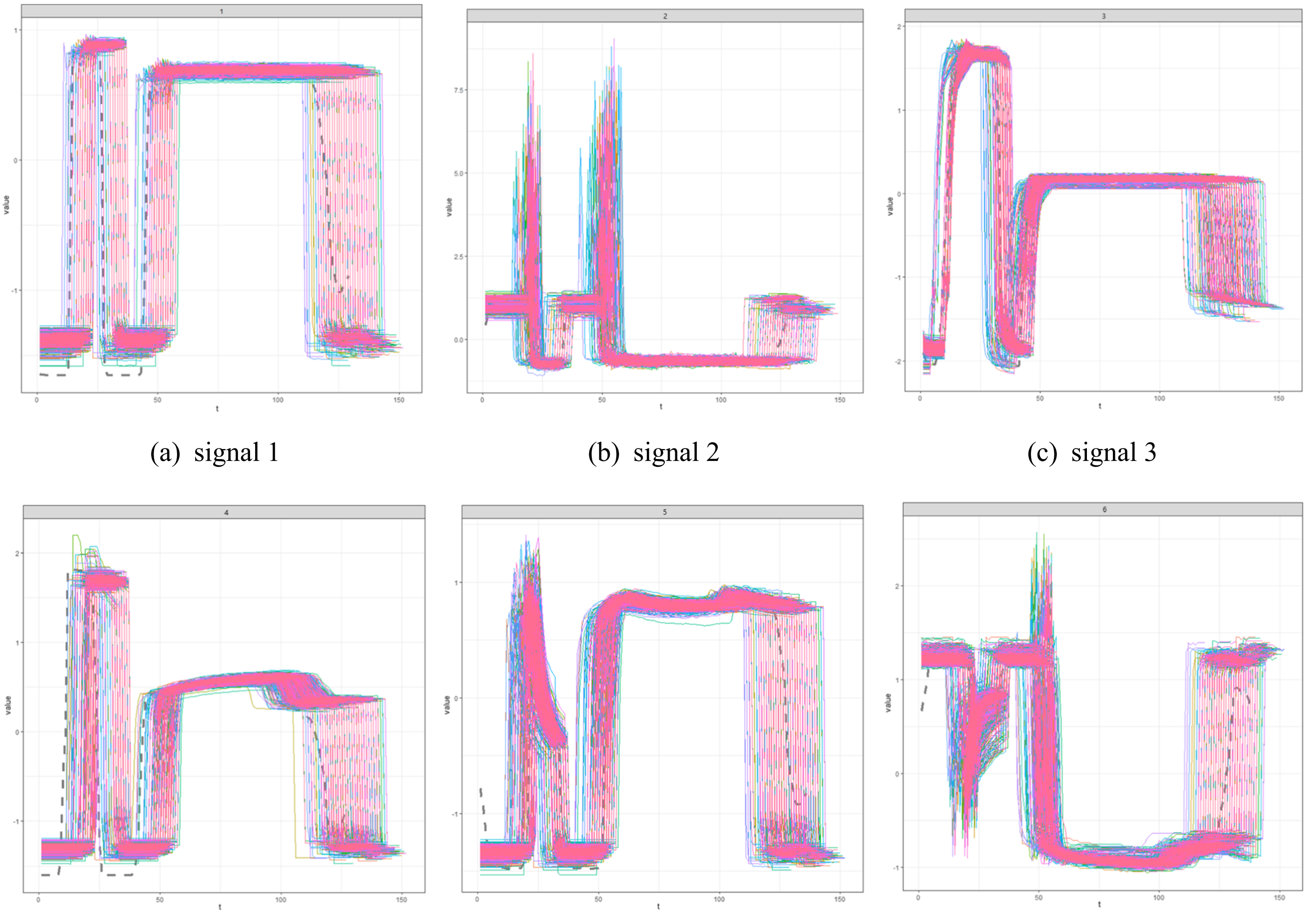
4.1 Experiment setup

The proposed clustering framework is demonstrated on real-world collected time series data from the manufacturing process. The data, contributed by Olszewski, comprises a collection of in-line process-control measurements recorded from various sensors during etch processing of silicon wafers for semiconductor fabrication [47]. There are six data types, each set in the database contains measurements by one sensor during processing one wafer. Those sensors include radio frequency forward power sensor, radio frequency reflected power sensor, chamber pressure sensor, 405 nm emission sensor, 520 nm emission sensor, and direct current bias sensor. Each data is treated as signal one to six respectively in this paper, and each wafer in the dataset has an assigned classification as normal or abnormal. The sample cluster shapes of six signal data can be found in Fig. 7. In order to investigate the validity of clustering results among time series clustering techniques, the proposed clustering method is demonstrated by using classified normal data only in each signals; and, for anomaly detection application, an undifferentiated whole time series data without distinction of signals is used.

The experiment environment is as follows. Computing power used in this paper is AMD Ryzen 5 5600G Radeon Graphics 3.90 GHz with 16.0 GB of RAMs. Main code language used in this paper is R, with Python assisting for partially efficient computations. One NVIDIA GeForce GTX 1070 Ti graphic card is used in computing other deep learning algorithms for anomaly detection result comparison. For metaheuristic algorithms, the proposed method had population size rate, r, population size, p, and max generation. Max generation as well as population size can be chosen with no limitation with the cost of computational time. In this paper, r is 15, p is $candidate\ length * r$, and max generation is $1.5 * \sqrt{p}$. Also, the proposed method utilizes tournament selection for selection strategy, mutation rate of 0.8 and crossover rate of 0.1. Each type of data was sampled with both 75% and 25% random data sample rate and this paper verified that the method works properly regardless of data sampling bias. For clustering validity experiment, the data were normalized using z-score, which is $(x - \mu) \ / \ \sigma$, within given time series data set, when μ is mean and σ is standard deviation. On the other hand, the whole data were not normalized for anomaly detection in order to compare with other time series classification methods.

4.2 Experimental results and interpretation

On univariate and raw time series data, four time series clustering techniques as well as four metaheuristic algorithms are used. First, as described in Section 3, hierarchical and partitional clustering with DTW, and two additional hybrid clustering approaches which include SBD prior to DTW are



(a) signal 1

(b) signal 2

(c) signal 3

(d) signal 4

(e) signal 5

(f) signal 6

Fig. 7 Cluster shape samples of six signals

obtained for experimental comparison. Four types of clustering techniques are abbreviated as H, P, HH and HP; which represent hierarchical (H), partitional (P), hybrid hierarchical (HH) and hybrid partitional (HP). In order to propose a fundamental framework for metaheuristic-based time series clustering, this paper used K-means and hierarchical clustering techniques as referential time series clustering methods as they are one of the most common, basic, and efficient methods used widely. The time series clustering techniques in the proposed method can be altered freely to other techniques based on a researcher's preferences or insights. Maximum cluster number used in this experiment is 5. Each clustering methods searches for clusters from 2 to 5, and maximum cluster candidate length will be $4 * 5$, which is 20. The proposed method assumes there is no previous knowledge regarding time series data so that max number of clusters can be adjusted freely and it does not have to be strictly determined before implementation. Four metaheuristics used in this paper include GA, BPSO, CGA and ECGA. The proposed method optimizes four types of clustering results by searching for better, or the best, cluster centroid combinations. In other words, the proposed framework performs various clustering techniques to expand the optimal cluster candidate pool then optimizes cluster centroid combinations using metaheuristics.

Tables 1, 2, 3, 4 and 5 show the experimental results. The experiment dataset is divided into two parts by data sample rate. In the beginning of each experiment iterations, the training data is divided into two subsets with random sample rate 25%, and 75%. This paper experimented the entire method 100 times for both sample rates, consequently a total of 200 iterations. Results are measured in the total average out of 100 iterations in each sample rate. The numerical values are DTW distances of each test data calculated with cluster centroids in each method. The smaller values are the better ones. Signals 2 and 6 have spiking fluctuations, and this characteristic results in longer distances and less variation. In general, this paper demonstrates that four metaheuristic methods are capable of searching for optimal clusters. While there were subtle differences in performance among metaheuristic methods, CGA and BPSO tended to converge late resulting in slightly higher distance compared to GA and ECGA.

One of the weaknesses of the proposed approach, with ECGA, is the computational burden. In Table 1, the average of total computation time of each metaheuristics from 100 iterations is described. Although time measurement can differ by coding language or code syntax, there are distinct differences in runtime among metaheuristic methods. It can be said that the computational time is fastest in the order of CGA, BPSO, GA and ECGA. With respect to the results shown in Tables 4 and 5, while CGA is the fastest metaheuristic method, there are many cases where CGA fails to find optimal solutions compared to other methods. While ECGA is considered to be the slowest method for searching optimal solutions, this paper highlights ECGA due to its advantages of effective search capability as well as logical search techniques. Comparing the convergence rate of metaheuristics algorithms, this paper suggests that CGA showed deficiency in performance. Comparing the performance of metaheuristics may differ by the problem one aims to solve or by the characteristics of the data one attempts to use. Within the scope of optimizing the clustering result using time series data originated from the manufacturing process, GA, BPSO and ECGA showed near performance with negligible amount of difference. With strict comparison, the performance is better in the order of GA, ECGA and BPSO which can also be seen in Tables 4 and 5. While CGA, one of EDA, shows rapid computation time

Table 1 Metaheuristics computation time average (in seconds) of 100 iterations

		GA	BPSO	CGA	ECGA
Sample rate 25%	signal 1	51.192	64.885	0.998	76.347
	signal 2	32.713	43.871	0.819	54.739
	signal 3	40.700	52.230	0.912	62.877
	signal 4	42.104	54.736	0.909	66.065
	signal 5	47.769	61.241	0.975	75.473
	signal 6	43.875	57.058	0.928	70.516
	mean	43.059	55.670	0.924	67.669
Sample rate 75%	signal 1	128.420	167.573	2.761	195.367
	signal 2	98.155	132.073	2.454	162.813
	signal 3	126.696	166.298	2.752	197.311
	signal 4	108.122	144.319	2.562	174.948
	signal 5	129.809	169.913	2.756	206.869
	signal 6	122.967	162.145	2.700	199.320
	mean	119.028	157.053	2.664	189.438

Table 2 ECGA model samples

Signal 1	[2]	[5]	[6,9]	[3,11]	[0,8]	[7,10]	[1,4]	
	[6]	[2,7]	[0,3]	[5,8]	[1,4]			
	[2]	[8]	[4,6]	[7,9]	[0,5]	[1,3]		
	[1]	[3]	[5]	[6]	[2,7]	[0,4,8]		
	[0]	[5]	[8]	[2,7]	[3,4]	[1,6]		
Signal 2	[7]	[3,5]	[4,6]	[0,1,2]				
	[4]	[0,6]	[1,7]	[3,2,5]				
	[6]	[1,7]	[2,0,3]	[4,5]				
	[6]	[1,7]	[4,5]	[3,0,2]				
	[6]	[1,2]	[4,7]	[0,3,5]				
Signal 3	[0]	[8,9]	[1,4,6]	[2,7]	[3,5]			
	[4,7]	[2,6]	[0,1]	[3,5]				
	[4]	[2,7]	[5,8]	[0,3]	[1,6]			
	[0]	[5]	[8]	[2,7]	[3,4]	[1,6]		
	[2]	[4]	[6]	[1,7]	[3,0,5]			
Signal 4	[3]	[5]	[4,7]	[0,1]	[2,6]			
	[12]	[0,2]	[9,11]	[3,7]	[1,6]	[8,10]	[4,5]	
	[0]	[2]	[4]	[1,7]	[3,5,6]			
	[5]	[6]	[8]	[2,7]	[0,3]	[1,4]		
	[6,8]	[5,7]	[1,2]	[4,0,3]				
Signal 5	[6]	[4,7]	[2,5]	[3,0,1]				
	[9]	[10]	[5,7]	[1,3]	[8,4,6]	[0,2]		
	[2]	[5]	[7]	[9]	[10]	[1,3]	[4,8]	[0,6]
	[5]	[4,7]	[3,6]	[1,0,2]				
	[1]	[5]	[8]	[2,7]	[4,6]	[0,3]		
Signal 6	[7]	[5,8]	[0,3]	[1,6]	[2,4]			
	[7]	[5,8]	[0,3]	[2,6]	[1,4]			
	[1,6]	[7,9]	[8,0,10]	[3,5]	[2,4]			
	[8,4,6]	[7,9]	[2,3]	[1,0,5]				
	[4]	[8]	[0,3]	[5,7]	[6,1,2]			

compared to the other three algorithms, this paper suggests that CGA holds a weakness that it requires more time to converge or may fail to find the most optimal solution compared to the metaheuristics used in this paper. In Figs. 8 and 9, this can be seen clearly. The figures show the initial 15 generation cycle to compare the converging performances. ECGA and GA tend to search for better initial fitness values compared to the other two.

Table 2 shows ECGA model examples sampled from the first five iterations of the experiment with data sample rate 25%. The lengths of each chromosomes are different because optimal cluster candidates from the previous step vary based on the time series clustering results. The concept of probabilistic model used in ECGA can be summarized as creating partitions in population space for selecting better chromosomes. For example, if a binary chromosome with length of 8 is partitioned as [1], [0,3], [2,4,6], [5,7]; there are 4 model blocks. The bits inside each block are considered to be jointly distributed, while the sole bit is considered to be independently distributed. In other words, cluster centroids located in the 0th and 3rd place in the binary chromosome are highly related and are jointly distributed in the population space. This probabilistic model is used for selecting a new population, namely chromosomes, for the next generation cycle for searching for better, or the best, gene that returns the highest fitness value.

Table 3 ECGA model dependency index

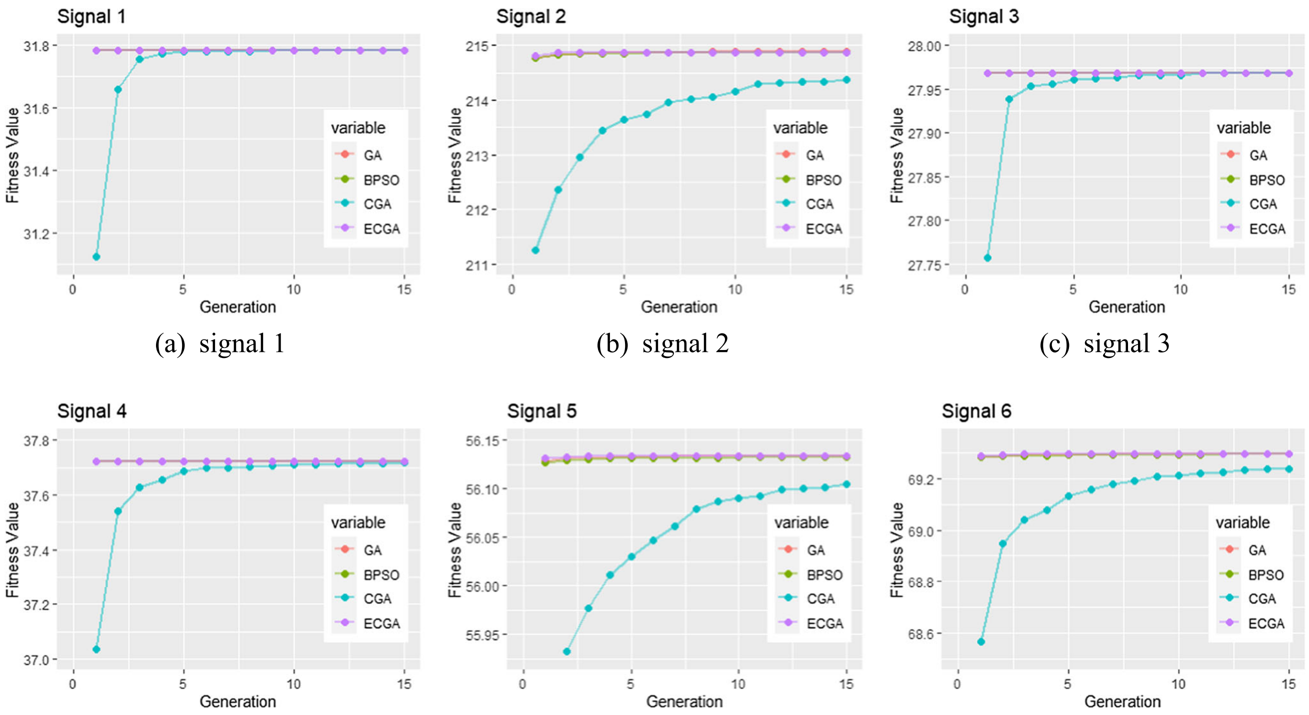
Data	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6
Sample rate 25%	1.650	**2.379**	1.661	1.612	1.737	**1.832**
Sample rate 75%	1.610	**2.243**	1.689	1.591	1.692	**1.931**

Table 4 Total average of 100 iterations with 25% sample rate

	stats	GA	BPSO	CGA	ECGA	HH	HP	H	P
Signal 1	mean	1.911	1.911	1.911	1.911	14.558	15.207	2.063	2.696
	sd	0.694	0.694	0.694	0.694	0.897	0.739	0.765	0.689
	Q1	1.436	1.436	1.436	1.436	13.850	14.731	1.490	2.194
	Q2	1.999	1.999	1.999	1.999	14.603	15.094	2.155	2.797
	Q3	2.279	2.279	2.279	2.279	15.177	15.674	2.497	3.063
	Low	1.906	1.906	1.906	1.906	14.552	15.201	2.058	2.691
	High	1.916	1.916	1.916	1.916	14.565	15.212	2.069	2.701
	CI	0.010	0.010	0.010	0.010	0.013	0.011	0.011	0.010
Signal 2	mean	37.790	37.799	38.260	37.798	49.200	48.837	47.131	42.110
	sd	27.620	27.615	27.772	27.618	25.286	25.244	33.186	33.057
	Q1	22.843	22.851	23.139	22.843	37.328	37.403	28.556	23.271
	Q2	30.596	30.599	30.986	30.601	41.945	41.441	36.658	31.706
	Q3	39.163	39.169	39.736	39.207	48.309	47.469	47.813	42.709
	Low	37.587	37.595	38.055	37.594	49.014	48.651	46.887	41.866
	High	37.994	38.002	38.464	38.001	49.386	49.023	47.376	42.353
	CI	0.407	0.407	0.409	0.407	0.373	0.372	0.489	0.487
Signal 3	mean	3.382	3.382	3.382	3.382	18.197	17.234	3.564	3.967
	sd	0.913	0.913	0.913	0.913	0.979	0.953	0.959	0.902
	Q1	2.795	2.795	2.795	2.795	17.551	16.574	2.925	3.391
	Q2	3.207	3.207	3.207	3.207	18.228	17.213	3.393	3.791
	Q3	3.757	3.757	3.757	3.757	18.839	17.848	3.993	4.318
	Low	3.376	3.376	3.376	3.376	18.190	17.227	3.557	3.961
	High	3.389	3.389	3.389	3.389	18.204	17.241	3.571	3.974
	CI	0.013	0.013	0.013	0.013	0.014	0.014	0.014	0.013
Signal 4	mean	4.241	4.241	4.246	4.241	16.104	13.750	4.613	4.828
	sd	2.708	2.708	2.726	2.708	2.399	2.385	3.014	2.608
	Q1	2.972	2.972	2.972	2.972	14.707	12.466	3.048	3.732
	Q2	3.704	3.704	3.705	3.704	15.686	13.193	3.949	4.239
	Q3	4.667	4.667	4.669	4.667	16.857	14.318	5.238	5.001
	Low	4.221	4.221	4.226	4.221	16.087	13.733	4.591	4.809
	High	4.260	4.260	4.266	4.260	16.122	13.768	4.635	4.847
	CI	0.040	0.040	0.040	0.040	0.035	0.035	0.044	0.038
Signal 5	mean	7.095	7.096	7.120	7.095	16.357	15.354	7.647	7.837
	sd	6.195	6.198	6.250	6.195	5.091	5.503	7.084	6.745
	Q1	3.808	3.808	3.809	3.808	13.722	12.337	3.851	4.642
	Q2	5.117	5.117	5.123	5.117	14.674	13.589	5.379	5.547
	Q3	7.768	7.769	7.783	7.768	16.700	15.906	8.523	8.007
	Low	7.049	7.050	7.074	7.049	16.319	15.313	7.595	7.787
	High	7.140	7.141	7.166	7.140	16.394	15.394	7.700	7.886
	CI	0.091	0.091	0.092	0.091	0.075	0.081	0.104	0.099
Signal 6	mean	10.076	10.078	10.129	10.077	18.571	16.530	11.710	10.607
	sd	8.904	8.908	8.988	8.905	8.376	8.085	10.852	10.362
	Q1	4.877	4.877	4.898	4.877	13.733	11.542	5.532	5.057
	Q2	6.695	6.695	6.704	6.695	15.243	13.770	7.893	6.763
	Q3	11.404	11.404	11.424	11.405	20.111	18.067	13.008	11.517
	Low	10.011	10.012	10.062	10.011	18.510	16.470	11.630	10.531
	High	10.142	10.143	10.195	10.142	18.633	16.589	11.790	10.684
	CI	0.131	0.131	0.132	0.131	0.123	0.119	0.160	0.153

Table 5 Total average of 100 iterations with 75% sample rate

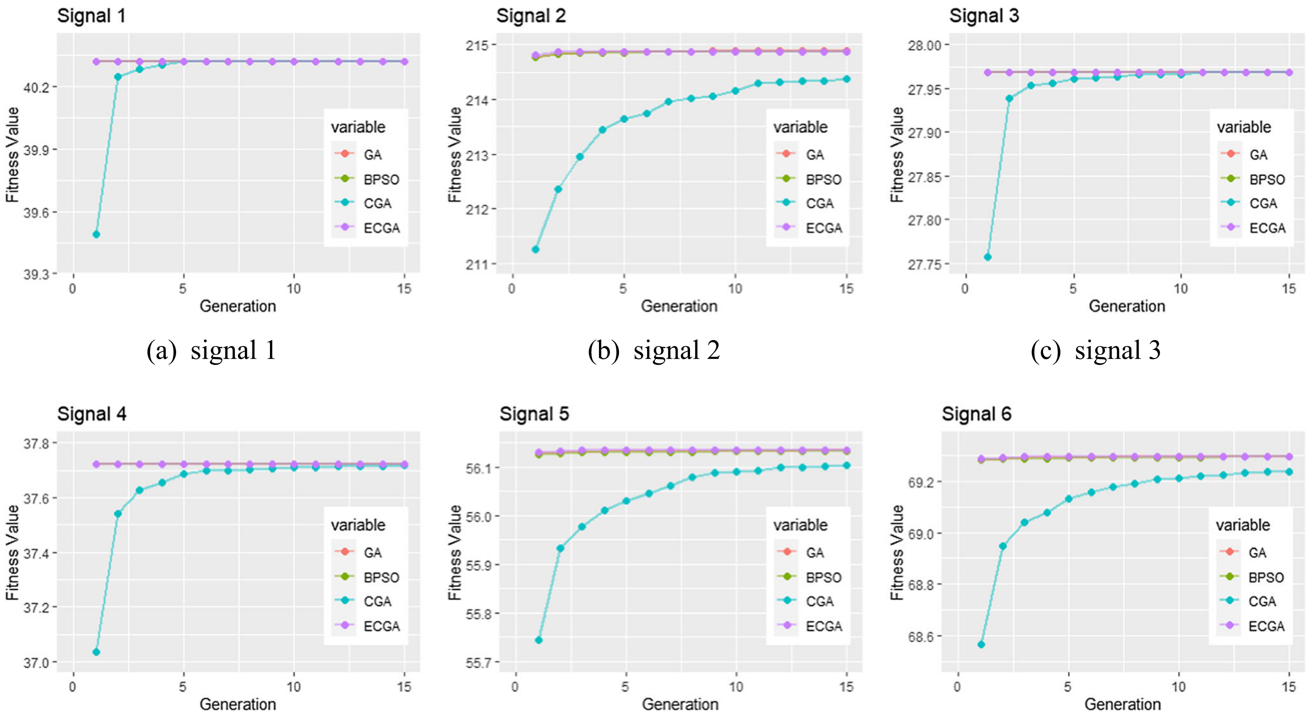
	stats	GA	BPSO	CGA	ECGA	HH	HP	H	P
Signal 1	mean	2.088	2.088	2.088	2.088	18.082	14.436	2.111	3.025
	sd	0.880	0.880	0.880	0.880	0.741	0.737	0.901	0.735
	Q1	1.304	1.304	1.304	1.304	17.578	13.967	1.304	2.455
	Q2	2.288	2.288	2.288	2.288	18.091	14.305	2.299	3.155
	Q3	2.607	2.607	2.607	2.607	18.562	14.937	2.643	3.422
	Low	2.086	2.086	2.086	2.086	18.080	14.434	2.109	3.023
	High	2.090	2.090	2.090	2.090	18.084	14.438	2.113	3.026
	CI	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004
Signal 2	mean	37.747	37.763	38.193	37.754	48.681	49.244	46.609	41.994
	sd	27.484	27.482	27.533	27.486	25.948	24.827	32.937	33.018
	Q1	23.009	23.014	23.267	23.012	36.483	38.193	28.549	23.383
	Q2	30.597	30.605	30.997	30.600	40.848	41.904	36.189	31.376
	Q3	39.208	39.243	39.853	39.232	47.222	47.994	45.748	42.190
	Low	37.680	37.696	38.126	37.687	48.617	49.183	46.529	41.913
	High	37.815	37.831	38.261	37.821	48.744	49.304	46.690	42.075
	CI	0.135	0.135	0.135	0.135	0.127	0.122	0.161	0.162
Signal 3	mean	3.239	3.239	3.240	3.239	20.599	16.692	3.282	3.976
	sd	0.915	0.915	0.916	0.915	0.953	0.956	0.937	0.897
	Q1	2.668	2.668	2.668	2.668	19.978	16.022	2.686	3.416
	Q2	3.061	3.061	3.061	3.061	20.627	16.653	3.105	3.798
	Q3	3.600	3.600	3.601	3.600	21.216	17.310	3.671	4.302
	Low	3.237	3.237	3.237	3.237	20.597	16.690	3.280	3.974
	High	3.242	3.242	3.242	3.242	20.602	16.694	3.284	3.979
	CI	0.004	0.004	0.004	0.004	0.005	0.005	0.005	0.004
Signal 4	mean	4.157	4.157	4.163	4.157	16.803	13.829	4.377	4.957
	sd	2.806	2.806	2.822	2.806	2.459	2.378	3.012	2.687
	Q1	2.811	2.811	2.811	2.811	15.433	12.561	2.827	3.907
	Q2	3.590	3.590	3.590	3.590	16.352	13.256	3.693	4.377
	Q3	4.647	4.647	4.652	4.647	17.522	14.369	4.931	5.075
	Low	4.150	4.150	4.156	4.150	16.797	13.823	4.370	4.951
	High	4.164	4.164	4.170	4.164	16.809	13.835	4.385	4.964
	CI	0.014	0.014	0.014	0.014	0.012	0.012	0.015	0.013
Signal 5	mean	7.142	7.142	7.164	7.142	17.761	15.167	7.643	7.862
	sd	6.321	6.323	6.405	6.323	4.975	5.514	7.167	6.751
	Q1	3.829	3.829	3.829	3.829	15.287	12.135	3.849	4.723
	Q2	5.090	5.090	5.092	5.090	16.102	13.447	5.331	5.579
	Q3	7.824	7.824	7.827	7.824	18.007	15.704	8.448	8.021
	Low	7.126	7.127	7.148	7.126	17.749	15.154	7.625	7.846
	High	7.157	7.158	7.180	7.157	17.773	15.181	7.660	7.879
	CI	0.031	0.031	0.031	0.031	0.024	0.027	0.035	0.033
Signal 6	mean	10.017	10.018	10.073	10.018	19.643	16.428	11.680	10.595
	sd	9.024	9.025	9.122	9.026	8.036	8.166	11.207	10.559
	Q1	4.725	4.725	4.756	4.725	14.923	11.344	5.304	4.959
	Q2	6.586	6.586	6.590	6.586	16.283	13.662	8.001	6.612
	Q3	11.426	11.427	11.442	11.426	21.347	18.009	12.825	11.555
	Low	9.995	9.996	10.051	9.996	19.623	16.408	11.652	10.569
	High	10.039	10.040	10.096	10.040	19.663	16.448	11.707	10.621
	CI	0.044	0.044	0.045	0.044	0.039	0.040	0.055	0.052



(a) signal 1 (b) signal 2 (c) signal 3

(d) signal 4 (e) signal 5 (f) signal 6

Fig. 8 Fitness function convergence graphs of initial 15 generations - 25% data sample rate



(a) signal 1 (b) signal 2 (c) signal 3

(d) signal 4 (e) signal 5 (f) signal 6

Fig. 9 Fitness function convergence graphs of initial 15 generations - 75% data sample rate

As mentioned above, signal 2 and 6 have traits of spiking fluctuations. This can also be inferred from the ECGA model results. Time series data with high variance means less cluster coherence. The length of chromosomes of signal 2 and 6 tends to be shorter than other signals, which means that there has been found less number of cluster candidates among time series clustering results. In addition, both signals show a relatively high number of model blocks compared to other signals. This is because, as there are fewer cluster candidates, there are comparably high number of similar (or jointly related) clusters within the chromosome. In other words, less cluster coherence with a smaller number of cluster candidates resulting into highly related clusters within limited candidate pools.

This logical reasoning can be numerically described as Table 3. Numerical values presented in the table are calculated by dividing the total sum of chromosome lengths with the total sum of number of model blocks in each signal data out of 100 iteration experiments. This paper describes this value as ECGA model dependency index. Higher values can be interpreted as there are relatively more dependent binary bits within chromosomes or, at the same time, shorter lengths of chromosomes have been created. As signal 2 and 6 are time series data which possess spiking traits compared to other signals, both of the signal data show a high number of model dependency index. Because signal 2 and 6 are comparably difficult to cluster the data, less number of promising clusters have been created as candidate clusters, and the cluster candidates show high dependency to each other due to less number of trending cluster centroids within chromosomes. Such numerical analysis as well as analytical reasoning are the unique characteristics of ECGA that differs from other algorithms. This feature of ECGA enhances the comprehension of data analysis results.

4.3 Anomaly detection investigation

Time series clustering, or time series classification, is highly correlated to anomaly detection. In the generic data analysis domain, including machine learning and deep learning, data clustering discovers the features and patterns from the train data. Hence, discerning anomalous cases among test data based on the clustering results is an inseparable twin-like mission to accomplish in data clustering. By inheriting the previous results from metaheuristic based time series clustering, the proposed method is employed to time series data anomaly detection through statistical tests for application purpose. Therefore, the proposed method is compared with other machine learning and deep learning techniques in detecting anomalous cases from time series data. The machine learning and deep learning methods used in this paper for comparison include - logistic regression (LR), XGBoost (XGB), fully connected layer (FCN), multilayer perceptron (MLP), ResNet, and long-short term memory (LSTM). The data used for the anomaly detection comparison test are time series data from semiconductor and automotive industries named as wafer, FordA and FordB.

Data used for anomaly detection is - combined, raw, and whole time series data - identical for all methods. Rather than dividing the time series data by sensors as in the previous clustering experiment, all time series data is gathered as one whole data set with labels in each data row as normal or abnormal. The wafer data has 6,165 rows of train data and 1,000 rows of test data with sequence lengths of 152. FordA has 3,601 rows of train data and 1320 rows of test data with sequence length of 500; and Ford B has 3,636 rows of train data and 810 rows of test data with sequence length of 500. The wafer time series data is identical from the previous sections except they are gathered and labeled as a whole dataset. Unlike machine learning or deep learning methods, the proposed method does not use the labels for data training. The proposed method tests anomaly detection based on the labeled test data after discovering clusters out of raw whole data. This holds the same for FordA and FordB dataset, which the data was originally used in a competition in the IEEE World Congress on Computational Intelligence, 2008. All three datasets can be found on UCR/UEA time series data archive.

Table 6 is the result of anomaly detection comparison of machine learning, deep learning, and the proposed method. Accuracy is calculated as dividing the number of correct predictions - aggregation of true positives and true negatives - by the number of total predictions. The results are the average accuracy from 10 iterations. For anomaly detection, statistical confidence interval test is used in this paper. Based on the optimization results, confidence intervals are calculated based on the three-sigma rule (68-95-99.7 rule). This follows the statistical hypothesis that 68 percent of the data will lie within mean value plus and minus one standard deviation (σ), 95 percent of the data within mean value plus and minus two σ, and 99 percent of the data within mean value plus and minus three σ.

While the wafer dataset is relatively monotonous compared to the FordA and FordB dataset, both Ford dataset possess large amounts of noise in data. The overall average accuracy for the wafer dataset is the highest and the overall average accuracy for the FordB dataset is the lowest. The table shows that, compared to long-standing machine learning algorithms such as regressions or tree search algorithms, metaheuristic time series clustering methods show prominent performances. While the metaheuristic based time series clustering method shows deficient performance compared to prevailing deep learning techniques such as LSTM or ResNet; the proposed approach possesses a distinctive advantage of logical bases on understanding data analysis results

Table 6 Anomaly detection accuracy (%) comparison

	Method	Wafer	FordA	FordB
ML	LR	93.90	49.18	49.01
	XGB	93.90	49.18	49.01
Metaheuristic	GA	69.9 / 88.93 / **90.04**	68.25 / **68.64** / 52.49	**57.15** / 53.94 / 50.98
($1\sigma/2\sigma/3\sigma$)	BPSO	71.7 / 88.92 / **90.05**	68.26 / **68.64** / 52.49	**57.16** / 53.95 / 50.98
	CGA	71.6 / 89.76 / **88.48**	68.41 / **68.56** / 52.42	**56.54** / 52.76 / 50.84
	ECGA	69.9 / 88.8 / **90.00**	68.25 / **68.62** / 52.50	**57.15** / 53.94 / 50.98
DL	FCN	99.71	88.71	70.25
	MLP	99.562	80.76	68.27
	ResNet	99.562	93.1	81.48
	LSTM	99.90	93.79	79.01

with significant level of performance compared to black-box based neural network models.

This paper shows that the proposed method shows noticeable performances compared to other time series techniques. With the combined use of long-standing time series clustering techniques, the metaheuristic based method surpassed 90% accuracy with a 3 σ confidence interval for wafer data. With meaningful figures of performance, the proposed approach possesses a significant advantage of logical bases on understanding data analysis results. This feature of the proposed method makes the difference compared to other artificial intelligence techniques which mainly adopt black-box based neural network models as well as a considerable amount of complex variables.

5 Conclusion

This paper proposed a novel time series clustering framework employing metaheuristics, especially the estimation of distribution algorithm (EDA). Most data analysis techniques affiliated with the artificial intelligence domain include characteristics of deriving features and patterns among numerous amounts of data. Meanwhile, reasoning out logical substance from astronomical numbers of variables and parameters in artificial intelligence models is significantly challenging. Accordingly, the recent trend in artificial intelligence stresses about the importance of explainability regarding data analysis results.

With the combined use of time series clustering techniques as well as metaheuristic algorithms, this paper demonstrated logical outcomes from the data analysis results through employing extended compact genetic algorithm (ECGA). By expanding the cluster candidate pool with various types of time series clustering methods, the proposed method searched for the most optimal cluster centroids combination using evolutionary algorithms. Particularly, ECGA is specialized in providing how such a solution is derived based on probabilistic models during the generation cycle. In other words, model blocks enable tracing back the cluster combination outcomes and interpreting the data analysis results as well.

Based on the optimized time series cluster centroid combinations, the proposed method applied them for effective use on anomaly detection in manufacturing time series data. The proposed approach showed meaningful performance compared to other machine learning and deep learning time series classification techniques. We reckon that our proposed research has advantages of not only an outstanding performance in anomaly detection application but also a logical background for data analysis results. In real-world use of data analysis in manufacturing industries, the analytical description as well as logical background regarding data training outcome is essential for robust utilization of mass data and artificial intelligence techniques.

Explainable relationship among clusters is what this paper mainly contributes in advancing the time series clustering techniques as well as extending the use of artificial intelligence methods in real-world settings. In future research, we expect to apply the advantage of metaheuristics, especially estimation of distribution algorithms (EDA), to other complex algorithm-based data analysis models such as reinforcement learning or graph neural networks. As they require a significant amount of computation complexity as well as variable diversity, there is recent research focusing on improving overall performance and efficiency of graph neural networks through various mathematical and statistical approaches [51, 52]. Furthermore, we plan to conduct future research on the proposed method for the performance change based on the change of maximum cluster number as well as the performance change based on the change of time series clustering techniques. We expect metaheuristics such as EDA will be effective not only in optimizing challenging problems but also delivering logical substances regarding causal comprehension on data analysis results.

Funding This work was supported by IITP grant funded by the Korea government (MSIT)(No. 2019-0-01842, Artificial Intelligence

Gradate School Program (GIST)), and the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2021R1A2C3013687)

Declarations

Conflict of Interests The authors declare no conflict of interest

References

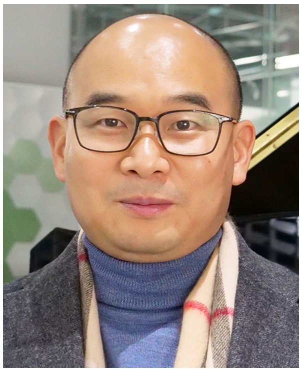
1. Bajgar M, Calligaris S, Calvino F, Criscuolo C, Timmis J (2019) Bits and bolts: The digital transformation and manufacturing. OECD Science, Technology and Industry Working Papers, 2019/01, OECD Publishing, Paris
2. Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Hung Byers A (2011) Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute, Washington, District of Columbia
3. Jones MD, Hutcheson S, Camba JD (2021) Past, present, and future barriers to digital transformation in manufacturing: A review. Journal of Manufacturing Systems 60:936–948
4. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: From big data to big impact. MIS quarterly, 1165-1188
5. Samek W, Wiegand T, Müller KR (2017) Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296
6. Samek W, Müller KR (2019) Towards explainable artificial intelligence. Explainable AI: interpreting, explaining and visualizing deep learning. Springer, Cham, pp 5–22
7. Aghabozorgi S, Shirkhorshidi AS, Wah TY (2015) Time-series clustering-a decade review. Information Systems 53:16–38
8. Mehrmolaei S, Keyvanpour MR, Savargiv M (2020) Metaheuristics on time series clustering problem: Theoretical and empirical evaluation. Evolutionary Intelligence, 1-20
9. Andreopoulos B, An A, Wang X, Schroeder M (2009) A roadmap of clustering algorithms: finding a match for a biomedical application. Briefings in Bioinformatics 10(3):297–314
10. Liao TW (2005) Clustering of time series data-a survey. Pattern Recognition 38(11):1857–1874
11. Leng M, Lai X, Tan G, Xu X (2009) Time series representation for anomaly detection. In 2009 2nd IEEE International Conference on Computer Science and Information Technology . IEEE, pp 628–632
12. Munir M, Siddiqui SA, Dengel A, Ahmed S (2018) DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. Ieee Access 7:1991–2005
13. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33(4):917–963
14. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications 80(5):8091–8126
15. Dhiman G (2021) ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. Engineering with Computers 37(1):323–353
16. David Edward G (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley
17. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press
18. Ahn CW (2006) Advances in evolutionary algorithms. Springer-Verlag, Berlin Heidelberg
19. Kennedy J, Eberhart R (1995) Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks , vol 4. IEEE, pp 1942–1948
20. Eberhart RC, Shi Y (1998) Comparison between genetic algorithms and particle swarm optimization. International conference on evolutionary programming. Springer, Berlin, Heidelberg, pp 611–616
21. Lee S, Soak S, Oh S, Pedrycz W, Jeon M (2008) Modified binary particle swarm optimization. Progress in Natural Science 18(9):1161–1166
22. Khanesar MA, Teshnehlab M, Shoorehdeli MA (2007) A novel binary particle swarm optimization. In 2007 Mediterranean conference on control & automation . IEEE, pp 1–6
23. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, vol. 5. IEEE, pp 4104–4108
24. Shabir S, Singla R (2016) A comparative study of genetic algorithm and the particle swarm optimization. International Journal of electrical engineering 9(2016):215–223
25. Larrañaga P, Lozano JA (Eds.) (2001) Estimation of distribution algorithms: A new tool for evolutionary computation, vol. 2. Springer Science & Business Media
26. Hauschild M, Pelikan M (2011) An introduction and survey of estimation of distribution algorithms. Swarm and Evolutionary Computation 1(3):111–128
27. Harik GR, Lobo FG, Goldberg DE (1999) The compact genetic algorithm. IEEE Transactions on Evolutionary Computation 3(4):287–297
28. Baluja S (1994) Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science
29. Jin Y, Oh S, Jeon M (2010) Incremental approximation of nonlinear constraint functions for evolutionary constrained optimization. In IEEE Congress on Evolutionary Computation. IEEE, pp 1–8
30. Harik G (1999) Linkage learning via probabilistic modeling in the ECGA. IlliGAL report, 99010
31. Sastry K, Goldberg DE (2000) On extended compact genetic algorithm. In Late-Breaking Paper at the Genetic and Evolutionary Computation Conference pp 352–359
32. Lanzi PL, Nichetti L, Sastry K, Voltini D, Goldberg DE (2008) Real-coded extended compact genetic algorithm based on mixtures of models. Linkage in evolutionary computation. Springer, Berlin, Heidelberg, pp 335–358
33. Oh S, Lee S, Jeon M (2009) Evolutionary optimization programming with probabilistic models. In 2009 Fourth International on Conference on Bio-Inspired Computing. IEEE, pp 1–6
34. Sastry K, Goldberg DE, Johnson DD (2007) Scalability of a hybrid extended compact genetic algorithm for ground state optimization of clusters. Materials and Manufacturing Processes 22(5):570–576
35. José-García A, Gómez-Flores W (2016) Automatic clustering using nature-inspired metaheuristics: A survey. Applied Soft Computing 41:192–213
36. Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm and Evolutionary Computation 16:1–18
37. Maulik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recognition 33(9):1455–1465
38. Thinsungnoen T, Kerdprasop K, Kerdprasop N (2018) Deep autoencoder networks optimized with genetic algorithms for efficient ECG clustering. International Journal of Machine Learning and Computing 8(2):112–116
39. Shakil M, Fuad Yousif Mohammed A, Arul R, Bashir AK, Choi JK (2022) A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering. Transactions on Emerging Telecommunications Technologies 33(3):e3622

40. Kadiravan G, Sujatha P, Asvany T, Punithavathi R, Elhoseny M, Pustokhina I, Shankar K (2021) Metaheuristic clustering protocol for healthcare data collection in mobile wireless multimedia sensor networks. Computers, Materials & Continua 66(3):3215–3231
41. Oh S, Ahn CW (2021) Evolutionary Computation-based Hybird Clustring Technique for Manufacuring Time Series Data. Smart Media Journal 10(3):23–30
42. Oh S, Suh WH, Ahn CW (2021) Self-Adaptive Genetic Programming for Manufacturing Big Data Analysis. Symmetry 13(4):709
43. Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. In KDD workshop 10(16):359–370
44. Müller M (2007) Dynamic time warping. Information retrieval for music and motion 69–84
45. Paparrizos J, Gravano L (2015) K-shape: Efficient and accurate clustering of time series. In Proceedings of the 2015 ACM SIGMOD international conference on management of data. pp 1855–1870
46. Sardá-Espinosa A (2017) Comparing time-series clustering algorithms in r using the dtwclust package. R package vignette 12:41
47. Olszewski RT (2001) Generalized feature extraction for structural pattern recognition in time-series data. Carnegie Mellon University
48. Połap D, Woźniak M (2021) Red fox optimization algorithm. Expert Systems with Applications 166:114–107
49. Połap D, Woźniak M (2017) Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. Symmetry 9(10):203
50. Fan GF, Yu M, Dong SQ, Yeh YH, Hong WC (2021) Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling. Utilities Policy 73:101–294
51. Dong W, Wu J, Zhang X, Bai Z, Wang P, Woźniak M (2022) Improving performance and efficiency of Graph Neural Networks by injective aggregation. Knowledge-Based Systems 254:109–616
52. Dong W, Wozniak M, Wu J, Li W, Bai Z (2022) De-Noising Aggregation of Graph Neural Networks by Using Principal Component Analysis. IEEE Transactions on Industrial Informatics

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Woong Hyun Suh received the M.S. degree in Artificial Intelligence from the AI Graduate School at Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2022. In 2018, he received a B.A. in Electrical Engineering and Society, with a concentration in Economics, from the Jacobs School of Engineering, University of California, San Diego (UCSD). He is currently an Investment Manager at the Corporate Venture Capital Team of Hyundai Motor Group in Seoul, Korea. With the combined background of tech and finance, his current interests include discovering open innovation partnerships and strategic investment opportunities, especially in the field of future mobility.

Sanghoun Oh received the Ph.D. degree from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2011. From 2011 to 2018, he worked with the Manufacturing Technology Center of Samsung Electronics, South Korea. He is currently working as a head of Development / Package Test / Management data analytics with the Department of Digital Transformation, SK Hynix. His research interests include genetic algorithms/programming, multi-objective optimization and explainable artificial intelligence. Currently, he concentrates on applying meta-heuristic algorithms into manufacturing industries to find optimized solutions.



Chang Wook Ahn received the Ph.D. degree from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2005. From 2005 to 2007, he worked with the Samsung Advanced Institute of Technology, South Korea. From 2008 to 2016, he was an Assistant/Associate Professor at the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. He is currently working as a Professor with the AI Graduate School at GIST. His research interests include genetic algorithms/programming, multi-objective optimization, deep learning, and quantum machine learning.