Profile Abstract: An Optimization-Based Subset Selection and Summarization Method for Profile Data Mining

Feng Zhu, Jianshe Feng, Min Xie, *Fellow, IEEE*, Lishuai Li, *Senior Member, IEEE*, Jingzhe Lei, and Jay Lee

***Abstract*—Nowadays, profile data mining techniques facilitate effective process monitoring, quality control, fault diagnosis, etc., with considerable benefits to manufacturing industry. However, regarding the complex system in modern manufacturing industry, there are two significant challenges for application development based on profile data mining. First, the staggering data volume leads to high memory and computational requirements. Second, the noisy signals in collected data may deteriorate useful information and model performance. This article proposes a novel algorithm for profile data mining called profile abstract, which simultaneously enables profile data compression and segmentation. The proposed algorithm mainly considers the scenario of fault diagnosis and can be utilized as a pre-processing step to address the above challenges. Profile abstract seeks to find a subset of raw data or a group of models as representatives that preserve the essential characteristics of raw data. Finding the data representatives helps reduce data redundancy while maintaining the model performance. Model representatives assist in describing the complex pattern of the profile, which can be used for pattern-based data segmentation. After data segmentation, information gain is adopted to determine the critical primitives for model improvement. In this article, validation of the proposed method's superiority is performed with two datasets from a real production line and one simulation dataset.**

***Index Terms*—Data compression, fault diagnosis, optimization, primitive extraction, profile data, subset selection.**

Manuscript received 4 September 2022; revised 14 October 2022; accepted 17 November 2022. Date of publication 8 December 2022; date of current version 24 July 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 71971181 and Grant 72032005, in part by the Research Grant Council of Hong Kong under Grant 11203519, Grant 11200621 and Grant 11215119, and in part by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA). Paper no. TII-22-3780. *(Corresponding author: Jianshe Feng.)*

Feng Zhu and Jingzhe Lei are with the Department of Advanced Design and Systems Engineering, City University of Hong Kong Hong Kong 999077, China (e-mail: fenzhu2-c@my.cityu.edu.hk; jingzhlei2-c@my.cityu.edu.hk).

Jianshe Feng is with the CRF Union Digital Technology Co. Ltd., Shenzhen 518071, China (e-mail: fengje@mail.uc.edu).

Min Xie is with the Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong, and also with the School of Data Science, City University of Hong Kong, Hong Kong 999077, China (e-mail: minxie@cityu.edu.hk).

Lishuai Li is with the School of Data Science, City University of Hong Kong Hong Kong 999077, China (e-mail: lishuai.li@cityu.edu.hk).

Jay Lee is with the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati, Cincinnati, OH 45221 USA (e-mail: lj2@ucmail.uc.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3227642.

Digital Object Identifier 10.1109/TII.2022.3227642

I. Introduction

With the ever-accelerating progress of sensing technology and introduction of Internet of Things into manufacturing industry, modern manufacturing facilities are becoming data-rich environments with massive amounts of data streams from various in-process sensors [1]. The *in situ* sensor measurements collected from multistage manufacturing operational cycles are usually spatial and time-dependent functional curves, which are also referred to as profile data [2]. Analysis of such large-scale sensor data can yield valuable information such as identification of patterns indicative of abnormal processes and appropriate corrective action suggestions, which facilitates effective process monitoring, quality control and predictive maintenance, with considerable benefits to manufacturing industry [3], [4], [5].

Profile-based method carves out the approaches to locate defects or anomalies in a full trace-wise perspective, of which time series classification (TSC) is an important problem for fault diagnosis in industrial scenarios such as semiconductor manufacturing [6], [7], [8], additive manufacturing [9], [10], and machining process [11], etc. While TSC is well-studied in the machine learning (ML) area, solving it for the complex physical system in modern manufacturing industry yet still involves two significant challenges. The first challenge arises from the staggering volume of profile data, so the algorithms usually suffer from increased memory and computational requirements. Furthermore, huge data volume challenges data transmission and model deployment in shop floor environments [12]. Second, raw profile data usually contain useless, redundant, or noisy signals, swamping the critical information. Take Fig. 1 as an example. The evident differences between normal and abnormal profiles only exist in some subsequences. It is desirable to extract critical primitives related to profile status (normal and abnormal), such as structural features or subsequences [13].

Data compression has not been thoroughly investigated with regard to profile data mining in industrial scenarios. Data

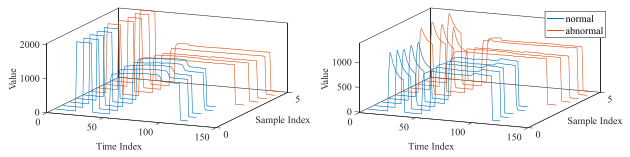


Fig. 1. Example of normal and abnormal profiles for the 405 nm parameter (left) and the 520 nm parameter (right) in the wafer database [14].

compression techniques represent a resource with the potential to reduce the cost associated with the transmission, storage, and computation of data series by obtaining a reduced or compressed representation of the data which is much smaller in volume than the original data [15]. They are generally grouped into three categories, i.e., model-driven algorithms, signal processing tools, and data-driven ML algorithms. Compared to model-driven algorithms like compressed sensing and signal processing tools like Fourier transform and wavelet transform, ML and emerging deep learning algorithms compress data adaptively based on the data properties [16], [17]. However, when applying these methods to profile data, they are mainly adopted to reduce variable numbers rather than instance numbers of trace, so it is hard to track sensor channels with faulty information after multisensor information fusion. Furthermore, it is challenging to combine subject matter expertise with these methods to improve interpretability.

Primitive extraction is another critical problem in TSC, which aims to locate vital information and improve model robustness. Time series shapelets is a well-known time series data mining primitive [18]. By extracting time series subsequences that are maximally representative of a type as shapelets, a time series shapelet classification method has been applied in various industrial scenarios [19], [20], [21]. However, time series shapelets learning methods are time-consuming and incapable of considering functional patterns or within-profile correlation in time series data, which weakens the interpretability of the results obtained. Another frequently applied technique for primitive extraction is data segmentation, which aims to segment time series data whenever the data pattern changes and then extract the representative values to depict each segment pattern. However, the data pattern does not always change when the process step changes. Some researchers attempt to use ML method for pattern-based data segmentation [22], [23]. Wilson [24] gave a detailed review of the existing approaches for data segmentation. To the author's experience, data segmentation is a more flexible approach for primitive extraction, as we can evaluate the information of each segment to identify the critical subsequences as the primitives. Furthermore, it is more intuitive to incorporate expert knowledge into the feature design and extraction process in the case [25]. However, the existing data segmentation methods, usually rule-based scenario-specific approaches, are sensitive to outliers. They tend to give more breakpoints for highly volatile data, which weakens the interpretability of critical segments. Besides, most data segmentation methods only consider handling the univariate signal, leading to the high computational and time cost when dealing with multivariate signals.

This article aims to develop a novel algorithm for time series data compression and primitive extraction simultaneously. It is worth noting that, while primitives can significantly reduce data volume and computational cost, data compression is still necessary for data storage or other purposes that require the characteristics of the entire set to be preserved. In this article, subset selection is used to overcome the above challenges and limitations of the existing methods by finding a representative subset from a large number of raw data points. A typical example of subset selection is keyframe selection for video summarization. Similarly, this article aims to select a smaller, easy-to-manage set of representatives, which acts as a summary of subsequences within the profile. Generally, the selected representatives can be a set of data points or models.

Subset selection has remained a research blind spot in the industrial area. Inspired by Elhamifar et al. [26], we consider the problem of finding representatives, given pairwise dissimilarities between the elements of a given source set $\mathbb{X}$ and a target profile $\mathbb{Y}$, in an unsupervised framework. To find a few representatives from $\mathbb{X}$ that depict the target profile $\mathbb{Y}$, we propose an optimization algorithm based on simultaneous sparse recovery called profile abstract. We formulate the problem as a linear programming model with constraints to consider the time-series characteristic. The solution determines representatives and the mapping relationship between each point to its representative. Profile Abstract algorithm can simultaneously be used for data compression and segmentation, which can serve as a preprocessing step before fault diagnosis modeling. Finding data representatives help eliminate redundant or repeated data points within signals and reduces storage and computation overhead while maintaining the model performance. Model representatives can be used for pattern-based data segmentation considering the within-profile correlation. After data segmentation, we propose an information gain-based algorithm to evaluate the importance of each segment and determine the critical primitives for TSC modeling with improved accuracy. The reader will immediately see that this method has many potential advantages over current methods, as well as its intended contributions.

1) A subset selection-based algorithm, called profile abstract, is specially designed for multivariate profile summarization and segmentation after revealing the challenges in fault diagnosis modeling.
2) Profile abstract can perform data compression by reducing data redundancy significantly while maintaining the model performance. It can ease the data transmission, storage, and computational requirement, improving the efficiency of model development and deployment.
3) Profile Abstract offers a more effective and reliable data segmentation approach that can detect and handle outliers.
4) We design an alternative, more straightforward segment importance evaluation method for primitive extraction, improving the model performance.
5) Profile Abstract is formulated as a convex optimization problem, which can be efficiently solved by most standard convex solvers, such as CVX, Gurobi, etc.

The rest of this article is organized as follows: Section I introduces related works and highlights the problem encountered in manufacturing process. Sections II and III elaborate the proposed Profile Abstract algorithm for data compression and primitive extraction. In Section IV, two case studies from the real manufacturing industry are demonstrated to validate the proposed method. Finally, Section V concludes this article.

II. Profile Abstract

In this section, we will first introduce our proposed profile abstract algorithm, which aims to find data representatives from the raw profiles as compressed data. First, we formulate the problem as a linear programming model. Then, the solution finds a set of data points as representatives from the profile along with the membership of each point to each representative.

A. Problem Statement

Suppose there is a multivariate profile $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M]$ of length M, which is a discrete time series. The pairwise dissimilarity between every two points $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ in the profile is denoted as $\{d_{ij}\}_{j=1,\ldots,M}^{i=1,\ldots,M}$, which represents how well $\boldsymbol{t}_i$ represents $\boldsymbol{t}_j$, i.e., $\boldsymbol{t}_i$ can be the representative of $\boldsymbol{t}_j$ if d_{ij} is small enough. We can arrange the pairwise dissimilarity into a matrix of the form

$$\boldsymbol{D} \triangleq \begin{bmatrix} \boldsymbol{d}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{d}_M^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} d_{11} & \cdots & d_{1M} \\ \vdots & \ddots & \vdots \\ d_{M1} & \cdots & d_{MM} \end{bmatrix} \in R^{M\times M} \tag{1}$$

where $\boldsymbol{d}_i \in R^M$ denotes the ith row of $\boldsymbol{D}$. Given $\boldsymbol{D}$, we aim to find a few points from $\boldsymbol{T}$, representing the whole profile to achieve data compression. It is noticed that we also can work with similarity $\{s_{ij}\}_{j=1,\ldots,M}^{i=1,\ldots,M}$, by simply setting $d_{ij} = -s_{ij}$. Thus, some kernel functions can also be used to construct the pairwise dissimilarity.

To guarantee the representatives are capable of well presenting the whole profile, the following requirements should be met: each point can only be represented by itself or the points shown before, called causality constraint; each point can only correspond to one representative, called uniqueness constraint; and the representative can only present a continuous segment, called continuity constraint.

B. Profile Abstract Algorithm

Given $\boldsymbol{D}$, the goal of the profile abstract is to select a subset of the raw profile data as representatives. Inspired by the framework of the DS3 algorithm [26], an assignment variable z_{ij} is introduced as an indicator of whether $\boldsymbol{t}_j$ can be represented by $\boldsymbol{t}_i$, which equals 1 when it is true and equals 0 otherwise. We denote the matrix of all assignment variables by

$$\boldsymbol{Z} \triangleq \begin{bmatrix} \boldsymbol{z}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{z}_M^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} z_{11} & \cdots & z_{1M} \\ \vdots & \ddots & \vdots \\ z_{M1} & \cdots & z_{MM} \end{bmatrix} \in \mathbb{R}^{M\times M} \tag{2}$$

where $\boldsymbol{z}_i \in \mathbb{R}^M$ denotes the ith row of $\boldsymbol{Z}$. In this optimization problem, two goals are considered, which are the same as the DS3 algorithm. First, a lower encoding cost is desired, which means that the subset can represent the whole profile well. If $\boldsymbol{t}_i$ is selected as a representative of $\boldsymbol{t}_j$, the cost of encoding $\boldsymbol{t}_j$ via $\boldsymbol{t}_i$ is defined as $z_{ij}d_{ij} \in \{0, d_{ij}\}$. Hence, the cost of encoding the profile by a subset can be quantified as $L_{\text{cost}} = \sum_i^M \sum_j^M z_{ij}d_{ij}$. Secondly, the number of representatives is supposed to be as small as possible, which equals the number of non-zero rows in $\boldsymbol{Z}$ matrix. It can be further formulated as $L_{\text{number}} = \sum_{i=1}^M \mathrm{I}(\|\boldsymbol{z}_i\|_p)$, where $\|\cdot\|_p$ is the p-norm and $\mathrm{I}(\cdot)$ is an indicator function that equals 1 when its argument is positive and equals 0 otherwise.

According to these two goals, the subset selection can be taken as a tradeoff between representative ability, which corresponds to a small L_{cost}, and small subset, which corresponds to a small L_{number}. Given that a small L_{cost} means a big L_{number}, and vice versa, a regularization parameter $\lambda > 0$ is used to set the trade-off between these two terms, and then the following optimization program can be obtained

$$\begin{aligned} &\min_{\{z_{ij}\}} \sum_{i=1}^M \sum_{j=1}^M z_{ij}d_{ij} + \lambda \sum_{j=1}^M \mathrm{I}\left(||\boldsymbol{z}_i||_p\right) \\ &\text{s.t. } \; z_{ij} \in \{0,1\}, \; \forall i, \; j. \end{aligned} \tag{3}$$

However, the minimization in (3) is nonconvex as it involves counting the number of non-zero rows in $\boldsymbol{Z}$. For the efficient implementation, the objective function (3) is further formulated as follows:

$$\begin{aligned} &\min_{\{z_{ij}\}\{b_i\}} \sum_{i=1}^M \sum_{j=1}^M z_{ij}d_{ij} \\ &\text{s.t. } \sum_{j\,=\,1}^M z_{ij} - A_1 b_i \le 0 \;\; \forall i; \; \sum_{i=1}^M b_i \le \tau \\ &z_{ij} \in \{0,1\}, \forall i; \; j, \; b_i \in \{0,1\} \;\; \forall i \end{aligned} \tag{4}$$

where b_i is an auxiliary variable used to count the number of representatives or non-zero rows in $\boldsymbol{Z}$, and A_1 is a big number, e.g., if $\boldsymbol{t}_i$ is selected as a representative, $\sum_{j=1}^M z_{ij} \ge 1$ and $b_i = 1$. Thus, we can get the number of representatives by directly calculating the sum of vector $\boldsymbol{b} = [b_1, \ldots, b_M]$. Compared with the optimization problem in (3), the optimization in (4) does not balance the encoding cost and number of representatives via λ, instead it directly enforces a specific number τ of representatives, which makes it a linear programming problem can be solved easily.

To guarantee the representatives capable to present the profile well, there are three main constraints that should be considered as follows.

1) Considering the profile data is a type of sequential data, each point can only be represented by itself or the points shown before, called causality constraint, that is

$$z_{ij} = 0, \; \text{if}, i > j. \tag{5}$$

2) Each point can only correspond to one representative, called uniqueness constraint, that is

$$\sum_{i=1}^{M} z_{ij} = 1 \ \forall j \tag{6}$$

3) The representative can only present a continuous segment, called continuity constraint, that is

$$z_{i'j'} = 0 \ \forall i' < i, j' \geq j, if, z_{ij} = 1. \tag{7}$$

Uniqueness constraint has already been considered in the DS3 algorithm, and the other two constraints are proposed in this article. However, compared with classic allocation or transportation problems, causality and continuity constraints bring challenges to solving the optimization problem in practice. Therefore, some transformations to the original forms of these two constraints are performed. For causality constraint (5), instead of considering a constraint to variable z_{ij}, we add an extra penalty to the elements in the dissimilarity matrix $\boldsymbol{D}$ in the following way

$$d'_{ij} = \begin{cases} d_{ij} & if \ i \geq j \\ d_{ij} + A_2 & if \ i < j \end{cases} \tag{8}$$

where A_2 is a big number.

Continuity constraint (7) is re-formulated by introducing a big number A_3, which is

$$z_{i'j'} \leq (1 - z_{ij}) \times A_3 \ \forall i' < i, j' \geq j \tag{9}$$

which equals to the following constraint form:

$$z_{ij} + \frac{1}{A_3} \sum_{i'=1}^{i-1} \sum_{j'=j}^{M} z_{i'j'} \leq 1 \ \ \forall i > 1, \ \ j \leq i. \tag{10}$$

Thus, we obtain the following optimization program:

$$\min_{\{z_{ij}\}\{b_i\}} \sum_{i=1}^{M} \sum_{j=1}^{M} z_{ij} d'_{ij}$$

$$\text{s.t.} \sum_{j=1}^{M} z_{ij} - A_1 b_i \leq 0 \ \ \forall i, \sum_{i=1}^{M} b_i \leq \tau,$$

$$\sum_{i=1}^{M} z_{ij} = \ 1 \ \ \forall j$$

$$z_{ij} + \frac{1}{A_3} \sum_{i'=1}^{i-1} \sum_{j'=j}^{M} z_{i'j'} \leq 1 \ \forall i > 1, \ j \leq i$$

$$z_{ij} \in \{0, 1\} \quad \forall i, \ j, \ b_i \in \{0, 1\} \quad \forall i. \tag{11}$$

The above optimization program keeps the general form of the linear programming problem, which can be efficiently solved by most standard optimization solvers. One can get representative indices from the non-zero rows of the solution $\boldsymbol{Z}^*$. It is essential to note the following details for the implementation of the profile abstract algorithm.

1) The number of representatives τ is a user-defined parameter that determines how much information is preserved in the representatives. The optimal objective value obtained from (11), which is the cost of encoding the whole profile with representatives, quantifies the information loss. Smaller encoding cost means that the selected representatives retain more information from the original set. Thus, the smaller τ is favorable while keeping the encoding cost lower. A sensitivity analysis is presented in the experiments with details for parameter effect.
2) Since multiplying $\boldsymbol{D}$ by the same scalar does not affect the solution of (11), we scale dissimilarities in the experiments so that the maximum absolute entry of $\boldsymbol{D}$ is 1.
3) For online monitoring, we will only process a baseline sample from the datasets, deriving a template for streaming-in time series signals.
4) In the case study, all the optimization programs are solved by Gurobi optimizer version 9.1.12 with MATLAB R2020b.

Remark 1: The optimal solution $\boldsymbol{Z}^*$ in (11) not only indicates the elements of $\mathbb{X}$ that are selected as representatives, but also contains information about the membership of $\mathbb{Y}$ to representatives. z^*_{ij} corresponds a hard assignment of $\boldsymbol{t}_j$ to representatives. Hence, we can obtain a clustering of the target profile $\mathbb{Y}$, e.g., a partitioning of $\mathbb{Y}$ into τ groups corresponding to τ representatives. Moreover, clustering algorithms such as K-means, DBSCAN, assume that all the samples in the set are independent of each other. Profile abstract ensures that the elements in one cluster are adjacent on the timeline as it considers time-series characteristics by causality and continuity constraints. This characteristic brings an opportunity to solve the problem of modeling and segmentation of time-series data from a clustering perspective, which has important applications, such as learning and segmentation of human activities [27], learning nonlinear dynamic models, and inverse modeling of complex motor control systems [28], and working regimes identification in the manufacturing process, etc. Thus, Profile Abstract can also be used for pattern-based data segmentation for time series, which will be discussed in the following section.

III. Primitive Extraction

In this section, we will use model representative extracted by the profile abstract algorithm to achieve pattern-based segmentation. Then, the critical primitives are determined by assessing the importance of each segment. Besides, we show that our algorithm can deal with outliers within the profile.

A. Data Segmentation With Model Representatives

Considering the clustering guarantees of the proposed profile abstract algorithm, it can perfectly solve the problem of pattern-based data segmentation. Model representative is used as it can recognize and describe more complex structural patterns. We aim to find a few models representing the whole profile to achieve data segmentation.

Let us denote by $\boldsymbol{T} = [\boldsymbol{t}_1, \ \boldsymbol{t}_2, \ldots, \boldsymbol{t}_M]$ the given baseline profile for the derivation of data segmentation template. $\boldsymbol{t}_i \in \mathbb{R}^N$ denotes the vector of the observation recorded at ith time point. We use t_i^n to denote the nth variable reading at the ith point. Then, $\boldsymbol{s}_n = \{t_i^n\}_{i=1,\ldots,M}$ represents the nth variable of the

Algorithm 1: Profile Abstract With Model Representatives for Data Segmentation.

Input: 1) A baseline profile data $\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_M]$
2) User-defined parameters f, K, τ
Output: The optimal solution $\boldsymbol{Z}^*$

(I)
for each $\boldsymbol{s}_n = \{t_i^n\}_{i=1,\cdots,M}$ **do**
  for $i \leftarrow 1$ **to** M **do**
    $\{y_k\}_{k=1,\ldots,K} \leftarrow knnSearch(t_i^n, \boldsymbol{s}_n)$ // Finding the K nearest neighbours of t_i^n in $\boldsymbol{s}_n$
    $\{x_k\}_{k=1,\ldots,K} \leftarrow TimeIndex(y_k)$ // Getting the time index of data points
    $\boldsymbol{x} \leftarrow \{x_k\}_{k=1,\ldots,K}$
    $\boldsymbol{y} \leftarrow \{y_k\}_{k=1,\ldots,K}$
    $\boldsymbol{\theta}_i^n \leftarrow CurveFitting(\boldsymbol{x}, \boldsymbol{y}, f)$ // Learning a model with parameter $\boldsymbol{\theta}_i^n$
  end for

(II)
  for $i \leftarrow 1$ **to** M **do**
    for $j \leftarrow 1$ **to** M **do**
      $x_j \leftarrow TimeIndex(t_j^n)$
      $d_{ij}^n \leftarrow ErrorFunction(x_j, t_j^n, \boldsymbol{\theta}_i^n)$
    end for
  end for
  $\boldsymbol{D}_n \leftarrow \{d_{ij}^n\}_{j=1,\cdots,M}^{i=1,\cdots,M}$
end for

(III)
$\boldsymbol{D} \leftarrow \frac{\sum_{n=1}^{N} \boldsymbol{D}_n}{N}$
$\boldsymbol{Z}^* \leftarrow ProfileAbstract(\boldsymbol{D}, \tau)$ \\ *Section II. B.*
return $\boldsymbol{Z}^*$

$\boldsymbol{T}$. Based on the notation, the proposed algorithm for data segmentation can be described as follows.

In Algorithm 1, the pseudocode in Section I generates the source model set $\mathbb{X}$ as candidates. For each data point t_i^n in $\boldsymbol{s}_n$ and its K nearest neighbours, we learn a model with parameter $\boldsymbol{\theta}_i^n$ by using their time index $\boldsymbol{x}$ and data value $\boldsymbol{y}$. The model type f can be selected based on the data properties or subject matter expertise. For simplicity, we suppose to use the one-dimensional affine model with parameter $\boldsymbol{\theta}_i^n = (\alpha_i^n, \beta_i^n)$ here. Section II constructs the dissimilarity matrix for subsequent selection. For each variable $\boldsymbol{s}_n$, we set the source set $\mathbb{X}_n = \{\boldsymbol{\theta}_i^n\}_{i=1,\ldots,M}$ and the target set $\mathbb{Y}_n = \{t_j^n\}_{j=1,\ldots,M}$. Then, we compute the pairwise dissimilarity between each model $\boldsymbol{\theta}_i^n$ and each data point (x_j, t_j^n) by using the loss function, as $d_{ij}^n = \updownarrow_{\boldsymbol{\theta}_i^n}(t_j^n) = \|(\alpha_i^n x_j + \beta_i^n) - t_j^n\|$, corresponding to the affine model here. The dissimilarity matrix for each variable can be obtained. In Section III, we take the mean of all the dissimilarity matrix $\boldsymbol{D}_n$ for information fusion. Then, the profile abstract algorithm proposed in Section II.B can be used to determine the model representatives through the optimal solution $\boldsymbol{Z}^*$, and a continuous subsequence represented by a model can be regarded as a segment.

It is essential to note the following details for the implementation of the data segmentation algorithm: K represents the number of the nearest neighbors for source model learning. From the author's experience, K can be determined in the range $[3, 10]$, because each model only needs to describe the local pattern surrounding each data point; and τ represents the number of representatives, which has been discussed in Section II-B.

B. Dealing With Outliers

We will demonstrate in this section that our data segmentation method can effectively deal with outliers in profile data. Generally, there will always be outliers within the profile, which can also be regarded as a kind of pattern. The outliers cannot be explained efficiently by any of the models in the source set $\mathbb{X}$, e.g., having a large representation error. However, the optimization program (11) requires every element in the profile to be encoded, enforcing the outliers to be represented by $\mathbb{X}$ often results in the selection of undesired representatives. Then, the interpretability of each segment will be weakened. In such cases, we aim to detect outliers and allow the optimization not to encode outliers via representatives to ensure that segmentation results are more reasonable. The detected outliers and selected model representatives can be combined to describe the overall structural pattern of the profile. To achieve this goal, we introduce a new optimization variable $e_j \in \{01\}$, whose value indicates if $\boldsymbol{t}_j$ is outlier. We propose to solve

$$\min_{\{z_{ij}\}\{b_i\}\{e_j\}} \sum_{i=1}^{M}\sum_{j=1}^{M} z_{ij} d'_{ij} + \sum_{j=1}^{M} \omega_j e_j$$

$$\text{s.t.} \sum_{j=1}^{M} z_{ij} - A_1 b_i \le 0 \quad \forall i; \sum_{i=1}^{M} b_i \le \tau$$

$$\sum_{i=1}^{M} z_{ij} + e_j = 1 \; \forall j$$

$$z_{ij} + \frac{1}{A_3} \sum_{i'=1}^{i-1} \sum_{j'=j}^{M} z_{i'j'} \le 1 \; \forall i > 1, \; j \le i$$

$$z_{ij} \in \{0,1\} \quad \forall i, \; j, \; b_i \in \{0,1\} \quad \forall i, \; e_j \in \{0,1\} \quad \forall j. \tag{12}$$

When $e_j = 0$, we have $\sum_{i=1}^{M} z_{ij} = 1$. Thus, $\boldsymbol{t}_j$ is an inlier and must be encoded by the candidate models. On the other hand, if $e_j = 1$, we have $\sum_{i=1}^{M} z_{ij} = 0$. Hence, $\boldsymbol{t}_j$ is an outlier and will not be encoded by the model. The weight $\omega_j > 0$ puts a penalty on the selection of $\boldsymbol{t}_j$ as an outlier.

One possible choice for the weight is to set ω_j as a constant for all j, which results in one additional regularization parameter with regard to (11). Another choice for the outlier weights is to set

$$\omega_j = e^{-\zeta \frac{\min_i d_{ij}}{\gamma}} \tag{13}$$

for nonnegative parameter ζ and γ. In other words, when there exists a model that can well represent $\boldsymbol{t}_j$, the likelihood of $\boldsymbol{t}_j$ being an outlier should decrease, i.e., ω_j should increase, and vice versa.

C. Segment Importance Evaluation

After the segmentation, we must evaluate the importance of each segment based on some metrics to determine the critical primitive for classification modeling. Information gain is used to evaluate the importance of each segment.

Algorithm 2: Segment Importance Evaluation Via Information Gain.

Input: Segment set $\mathbf{SG} = \{\boldsymbol{ST}_i\}_{i=1,\ldots,Q}$
Label $\boldsymbol{c} = \{c_i\}_{i=1,\ldots,Q}$
Output: Information gain IG
for each $\boldsymbol{ST}_i$ in $\boldsymbol{SG}$
$dist_i = \text{EuclideanDist}(\boldsymbol{ST}_i, \boldsymbol{ST}_{\text{baseline}})$
end for
$d_{th} \leftarrow \text{OptimalSplitPoint}(\boldsymbol{dist}, \boldsymbol{c})$
$\mathbf{SG}_1 \leftarrow \emptyset,\ \mathbf{SG}_2 \leftarrow \emptyset$
for each dist_i in $\boldsymbol{dist}$
if $dist_i <$ SplitDist
$\mathbf{SG}_1 \leftarrow \mathbf{SG}_1 \cup \boldsymbol{ST}_i$
else
$\mathbf{SG}_2 \leftarrow \mathbf{SG}_2 \cup \boldsymbol{ST}_i$
end if
end for
return $IG = I(\mathbf{SG}) - \frac{Q_1}{Q} I(\mathbf{SG}_1) - \frac{Q_2}{Q} I(\mathbf{SG}_2)$ // (15)

Let us denote by $\mathbf{DS} = \{\boldsymbol{T}_i, c_i\}_{i=1,\ldots,Q}$ the given TSC dataset. Q is the total number of samples and $c_i \in \{12\}$ represents the class label for the ith sample considering the binary classification problem. After the segmentation, the set of each segment can be denoted by $\mathbf{SG} = \{\boldsymbol{ST}_i, c_i\}_{i=1,\ldots,Q}$, and the $\boldsymbol{ST}_i$ represents the subsequence corresponding to this segment in the ith sample. Given that the proportion of samples in classes 1 and 2 is p_1 and p_2 separately, the entropy of $\mathbf{SG}$ is defined as

$$I(\mathbf{SG}) = -p_1 \log_2 p_1 - p_2 \log_2 p_2. \tag{14}$$

If we divide the $\mathbf{SG}$ into two subsets $\mathbf{SG}_1$ and $\mathbf{SG}_2$ based on a certain splitting strategy and the number of samples in $\mathbf{SG}_1$ and $\mathbf{SG}_2$ are Q_1 and Q_2, respectively. The information gain from this splitting strategy is defined as

$$IG = I(\mathbf{SG}) - \frac{Q_1}{Q} I(\mathbf{SG}_1) - \frac{Q_2}{Q} I(\mathbf{SG}_2). \tag{15}$$

Based on the above notation, the proposed algorithm for segment importance evaluation can be described as

First, the algorithm calculates the Euclidean distance between each sample $\boldsymbol{ST}$ in the set $\mathbf{SG}$ and the baseline sample $\boldsymbol{ST}_{\text{baseline}}$. Based on the set of distances $\mathbf{dist} = \{\text{dist}_i\}_{i=1,\ldots,Q}$ and the label vector $\boldsymbol{c}$, we can choose an optimal distance threshold d_{th} as the best splitting strategy with the highest information gain by using a histogram. Then, it can split $\mathbf{SG}$ into $\mathbf{SG}_1$ and $\mathbf{SG}_2$, such that for every subsequence $\boldsymbol{ST}_{1,i}$ in $\mathbf{SG}_1$, $\text{EuclideanDist}(\boldsymbol{ST}_i, \boldsymbol{ST}_{\text{baseline}}) < d_{th}$ and for every subsequence $\boldsymbol{ST}_{2,i}$ in $\mathbf{SG}_2$, $\text{EuclideanDist}(\boldsymbol{ST}_i, \boldsymbol{ST}_{\text{baseline}}) \geq d_{th}$. Finally, we can get the information gain IG for this segment set $\mathbf{SG}$, and the segments with the higher information gain can be considered as the critical primitives. This evaluation method can also be used for the multiclass problem using the one versus rest mechanism. Then more distance thresholds can be determined to split the original set.

IV. Experimental Case Studies

To verify the performance of the Profile Abstract algorithm for data compression and primitive extraction, experiments are conducted on two datasets, including dataset I from University of Cincinnati and dataset II from Carnegie Mellon University (CMU). We design a flow chart for TSC modeling with data compression and primitive extraction using the profile abstract algorithm, as shown in Fig. 2.

For data compression, one baseline profile is selected to derive a compression template first. Typically, the normal profiles after maintenance activity are regarded as the candidates of the baseline profile. Using the profile abstract algorithm, we can get the indexes of the selected representative points in the raw profile, which is used as a compression template. It can be applied to all other input profiles. Dynamic time warping (DTW) is used to handle the time variation, aligning the time steps between the baseline and input profiles. The time complexity of computing DTW is $O(m * n)$, where m and n represent the length of each sequence. Then, the indexes of the representative points in each input profile can be obtained. After the data compression, the compressed multivariate profiles will be flattened to feature vectors. The last step is fault diagnosis modeling based on TSC. TSC modeling is well discussed in prognostics and health management. We consider using common models for the present case because it can simplify the experiment setup and verify the effectiveness of the proposed algorithm. In this article, support vector machine (SVM) with Gaussian Kernel is used to build the TSC model for profile-based fault diagnosis.

The flow of the TSC modeling with primitive extraction is similar. With the selected baseline profile, a segmentation template can be derived, and then DTW is used to generalize the segmentation template to all the input profiles. After segmentation, information gain is used to evaluate the importance of each segment to determine the primitives for TSC modeling. At last, the selected primitives are flattened to the vectorized input for SVM modeling. Meanwhile, other typical algorithms for data segmentation are also implemented for benchmarking to illustrate the performance.

A. Boring Process in Automotive Manufacturing Industry (Univariate Analysis)

1) Data Description: The boring process in the automotive manufacturing industry is investigated in the first case study. The dataset is collected from the real boring machine in a major, domestic automotive facility, which includes the spindle load signals with 3000 boring process cycles within two weeks. The equipment degradation is mainly caused by the boring tool wear, and 25 boring tools were worn out during these two weeks. To create a manageable dataset for method validation, we randomly selected 100 cycles right after the tool change, which are labeled as normal, and another 100 cycles priors to the tool change that are labeled as faulty, indicating that the boring tool is completely worn out. The in-between degradation stage is represented by the cycles from 100 cycles in the middle of the tool life, which are also included as the third class. It represents the medium

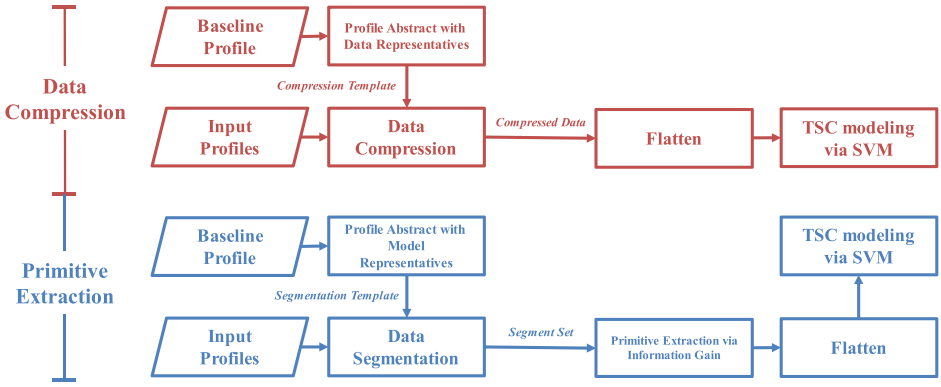


Fig. 2. Flow chart of the TSC modeling in the experiments.

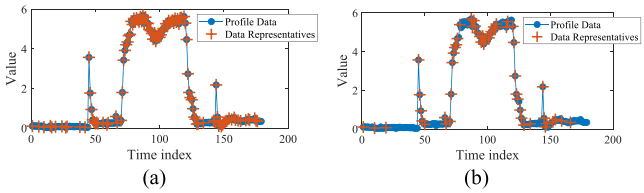


Fig. 3. Data compression results for the baseline run. (a) $\eta = 60\%$. (b) $\eta = 30\%$.

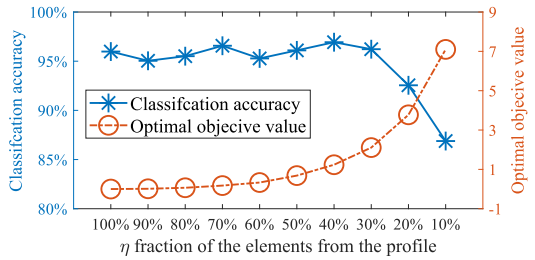


Fig. 4. Mean classification accuracy rates on test folds, using a three-fold three-repeat cross-validation procedure and the optimal objective value obtained from the optimization program for each fraction.

TABLE I
SUMMARY OF TIME SERIES SEGMENTATION APPROACHES

Segmentation Principle	Algorithm	Pros	Cons
Piecewise Approximation	PAA	a) Simple	a) Divide time series into equal-length segments b) Poor performance
	PLA [23], [30]	a) Pattern-based segmentation based on linear approximation	a) Tends to give more breakpoints for highly volatile data b) Need to pre-define the segment number
Identification of Important Points	PIP [31]	a) Simple b) Pattern-based segmentation	a) Need to pre-define the number of important points
	PELT [32]	a) Effective b) Pattern-based segmentation	a) Only can pre-define the maximum number of important points
Clustering	Profile Abstract	a) Pattern-based segmentation b) Robust to outliers c) Multivariate analysis	\

status of the boring tools. More details about this dataset can be found in [29].

2) Data Compression: Fig. 3 shows a typical example of the spindle load signal from one operation cycle, which is used as the baseline profile for data compression. Its corresponding data representatives selected by our algorithm are also given in the figure. η represents the fraction of selected points from the profile, which is used to determine the number of data representative points, i.e., $\tau = \text{int}(\eta M)$. We can see that the subsequences with greater fluctuations require more data representatives to encode, but fewer data representatives are needed for the flat segment.

Fig. 4 shows the mean classification accuracy rates on test folds from three-fold cross-validation and the optimal objective value obtained from the optimization program, i.e., the cost of encoding the profile with representatives. We experimented with changing the fraction η of the representatives extracted from the profile. 100% fraction of the elements means the raw profile without compression. As the results show, decreasing the value of η, i.e., having fewer representatives from the profile data, has no significant effect on the classification results. As long as the encoding cost is small, which means that the information retained by the representatives is sufficient, the classification model performance will be maintained. It demonstrates that the data representatives help to compress the data redundancy effectively by reducing the data length. However, if the encoding cost increases, the representatives will lose part of critical information, leading to the deterioration of the classification model performance. Thus, the encoding cost can be used to determine the optimal fraction η for data compression. In this case, we can compress the dataset to 60% of its original size based on the encoding cost.

3) Primitive Extraction: Because data representatives cannot describe the more sophisticated structural patterns, model representatives are used to segment the data and extract the critical primitives for accuracy improvement.

Some typical algorithms for data segmentation are given in Table I to show the superiority of the profile abstract algorithm. Apart from the proposed algorithm, data segmentation algorithms are generally grouped into two categories based on the segmentation principle. Piecewise aggregate approximation (PAA) is the simplest to implement, which divides the data into

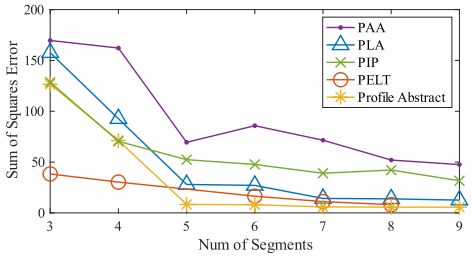


Fig. 5. Overall fitting error of the different time series segmentation algorithms with the changing number of segments.

equal lengths. Piecewise linear approximation (PLA) divides the data into several segments by fitting a linear model for each segment. Perceptually important points (PIP) and pruned exact linear time (PELT) segment the data by finding the important points sequentially. Profile abstract achieves pattern-based segmentation based on a novel principle, with many potential benefits.

The work in [23] and [30] and considering most data segmentation approaches are developed based on linear models, we can perform a linear fitting to each segment after segmentation and check if the segmentation results from each algorithm are reasonable based on the overall fitting error, which is defined as

$$\text{Error} = \sum_{j=1}^{N_{\text{seg}}} \text{SSE}\left(T^j, \widehat{T}^j\right) \tag{16}$$

where T^j is the jth segment and $\widehat{T}^j$ is the corresponding fitted curve. N_{seg} refers to the total number of segments for the profile data, and SSE represents the sum of squared error. We have experimented with various numbers of segments for each algorithm because we are unable to determine the optimal setting in advance. The fitting error will keep decreasing as the number of segments increases. Then, the number of segments at the "breakpoint" in the error curve will be the optimal one because the error converges here.

Fig. 5 shows the overall fitting error of the different algorithms. We can see that the proposed algorithm has the lowest error when the error converges. The main reason is that outliers within the signals can be detected and kept from interfering with the segmentation process using the proposed Profile Abstract algorithm. In comparison, we can see that PLA and PELT both tend to give more breakpoints to achieve a lower fitting error. The results demonstrate the effectiveness and superiority of Profile Abstract for data segmentation. It has high robustness to outliers, thus ensuring that a reasonable segmentation result can be generated. Another main advantage of profile abstract is that it can be used for multivariate analysis, which will be investigated in the second case.

Fig. 6 shows the segmentation results for the baseline run by using profile abstract. The selected model representatives are shown, which can describe the fundamental pattern of the baseline profile. Some points are regarded as outliers. It is noticed there are two categories of points that are identified as outliers by our algorithm. The first type of outliers exists in the flat segment, which is an observation that lies at an abnormal

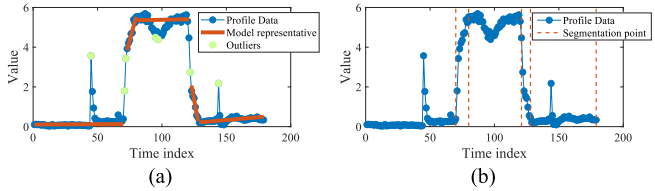


Fig. 6. Segmentation results for the baseline run.

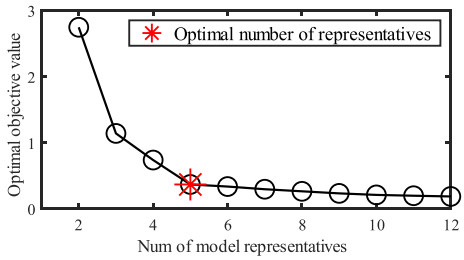


Fig. 7. Optimal objective value obtained from the optimization program for the different number of model representatives.

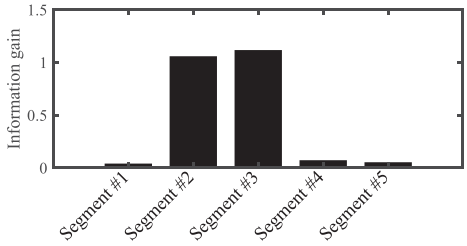


Fig. 8. Information Gain of each segment.

distance from other surrounding points. The first few points during a pattern change are the second type of outliers, which always have an abrupt shift. We can describe the overall structural pattern of the profile based on the detected outliers and selected model representatives. Using model representatives, the profile is reasonably segmented, where the segmentation points shown in the figure are the ending points of each segment. It should be noted that the outliers will be involved in further analysis, which will not lead to information loss. The optimal objective value in Fig. 7 is the same as that shown in Fig. 4. It is the cost of encoding the whole profile with representatives obtained from the solution of the optimization program (12). Similar to data compression, the optimal segment number is determined by the optimal objective value (encoding cost) obtained from the optimization program. As the number of representatives increases, we can check whether the encoding cost converges and thus decide the optimal number of representatives. Fig. 7 indicates that the optimal segment number for this case is 5.

Fig. 8 visualizes the information gain of the obtained five segments. One can easily see that segments #1, #4, and #5 cannot contribute any valuable information for classification modeling, which can be directly excluded in the data analysis. Thus, segments #2 and #3 are selected as the critical primitives, which can be used for further classification modeling.

Table II gives that the critical primitives determined by the proposed method can significantly improve the classification

TABLE II
MEAN CLASSIFICATION ERROR RATES ON TEST FOLDS, USING A THREE-FOLD THREE-REPEAT CROSS-VALIDATION PROCEDURE

η	Mean classification error rate	
	Whole profile data input	Primitive input
100%	4.02%	0.47%
90%	4.96%	0.35%
80%	4.49%	0.35%
70%	3.43%	0.35%
60%	4.73%	0.47%

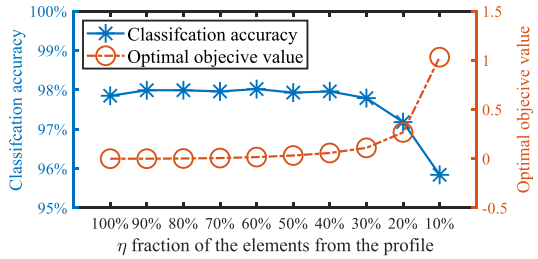


Fig. 9. Mean classification accuracy rates on test folds, using a three-fold three-repeat cross-validation procedure and the optimal objective value obtained from the optimization program for each fraction.

model performance. We also attempted to extract primitives from the data representatives after data compression and then perform the same classification verification. The results show that the primitive extracted from the data representative also can achieve high classification accuracy, which further proves the effectiveness of the proposed method for data compression and critical primitive extraction.

B. Wafer Fabrication Process in Semiconductor Industry (Multivariate Analysis)

1) Data Description: The wafer fabrication process in the semiconductor manufacturing industry is investigated in the second case study to verify the effectiveness of our proposed method for processing multivariate profiles. This dataset is a collection of in-line process control measurements recorded from an etching process that is donated by CMU. Six sensor channels are given by domain experts as being critical for process monitoring, including: radio frequency forward power; radio frequency reflected power; chamber pressure; 4.5 nanometer (nm) emission; 520 nanometer (nm) emission; and direct current bias. Considering the low correlation of chamber pressure with the other sensor channels, chamber pressure is excluded from the original data set in this article. The faulty samples in this dataset refer to the wafer fabrication cycles with abnormalities. More details about this dataset can be found in the research of [14]. In this case, we will verify the effectiveness of our proposed algorithm on multivariate analysis.

2) Data Compression: Fig. 9 verifies the effectiveness of our proposed profile abstract algorithm for data compression on multivariate profile data. The mean classification accuracy on test folds and the cost of encoding the profile with representatives are given. Similar results to the first case study can be obtained. As long as the encoding cost is small enough and the information retained by the representatives is sufficient, the classification

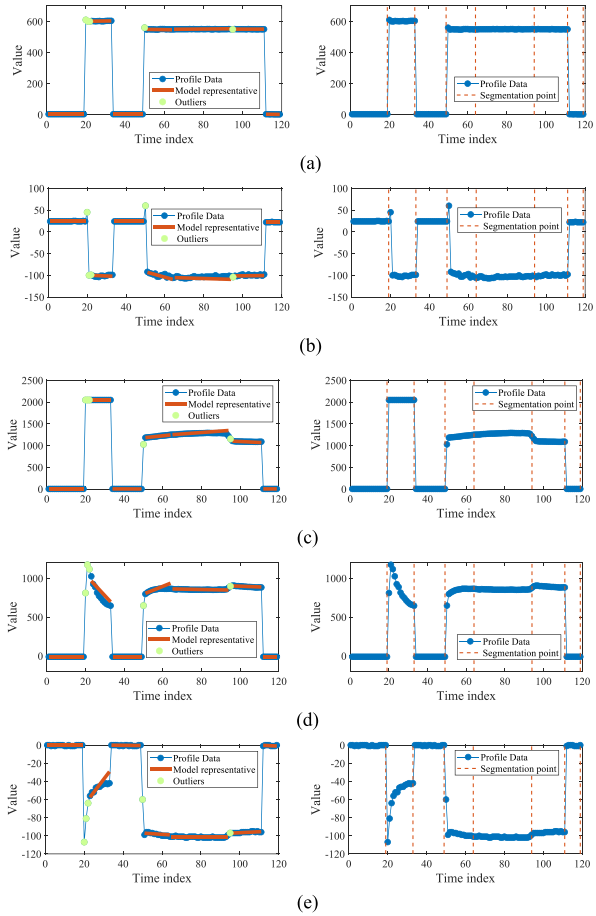


Fig. 10. Segmentation results for the golden wafer run. (a) Radio frequency forward power. (b) Radio frequency reflected power. (c) 4.5 nanometer (nm) emission. (d) 520 nanometer (nm) emission. (e) Direct current bias.

model performance will be maintained. In this case, we can compress the dataset to 30% of its original size based on the encoding cost.

3) Primitive Extraction: Fig. 10 verifies the effectiveness of the proposed method for multivariate profile data segmentation with model representatives. The number of representatives is determined as 7 based on the optimal objective value. The segmentation results for each sensor channel of the baseline wafer run are shown. The multivariate profile is also reasonably segmented based on the structural pattern. It should be noted that sensor #1 may need fewer representatives to depict its whole patterns. In this case, however, we attempted to divide the manufacturing process into several stages using multivariate analysis, and we treated the multivariate profile as a whole. Thus, we need to determine the representative number based on all the sensor channels. In real-world applications, the users can decide the segmentation way (univariate or multivariate) based on the scenarios and demands. Besides, outliers within each sensor channel can be accurately located so as to avoid affecting segmentation results.

Fig. 11 visualizes the information gain of each segment. It is noticed that each segment is also a multivariate profile. Obviously, segments #1, #2, #3, and #7 are useless for

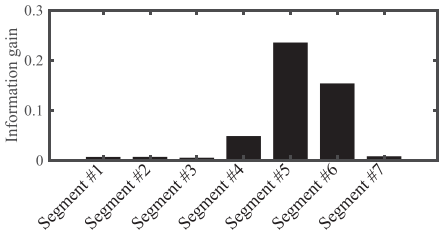


Fig. 11. Information gain of each segment.

TABLE III
MEAN CLASSIFICATION ERROR RATES ON TEST FOLDS, USING A THREE-FOLD THREE-REPEAT CROSS-VALIDATION PROCEDURE

η	Mean classification error rate	
	Whole profile data input	Primitive input
100%	2.15%	0.25%
90%	2.01%	0.59%
80%	2.01%	0.59%

classification modeling, which can be directly excluded in the data analysis. Thus, segments #4, #5, and #6 are selected as the critical primitives, which can be used for further classification modeling.

Finally, Table III gives the classification error using profile data input and primitive input, respectively. The results are very similar to the first case study, demonstrating that the proposed primitive extraction method also works well on multivariate profiles. In summary, these two cases can draw the following conclusion.

1) Profile abstract can work well on both univariate profiles and multivariate profiles.
2) Profile abstract using data representatives is effective for data compression, which can significantly reduce the data redundancy while maintaining the model performance.
3) Profile abstract using model representative provide a robust data segmentation approach.
4) The proposed method for primitive extraction can significantly improve the classification model performance.

C. Simulation Data Analysis

The experimental signals from dataset I and dataset II are similar and not representative enough, as both are rectangular waves. Therefore, we consider more different waveforms in this part to further test the performance of the proposed data segmentation approach based on profile abstract. Three types of waveforms are considered: Sawtooth wave; triangle wave; and sine wave. For each type of waveform, we generate seven periods of a wave with a fundamental frequency of 50 Hz. The sample rate is 1 kHz, and two noise levels are considered. Then, profile abstract is used to summarize their structural patterns and perform data segmentation.

Figs. 12–14 verify the effectiveness of the proposed method for these three waveforms. The results demonstrate that the selected models can effectively describe their structural pattern and perform reasonable segmentation. Even with relatively high levels of signal noise, the algorithm remains effective.

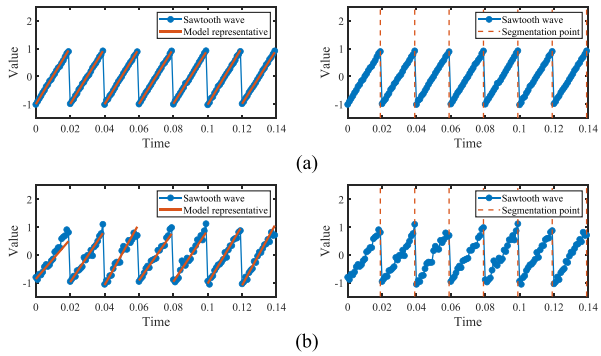


Fig. 12. Segmentation results for Sawtooth waves. (a) SNR: 30db. (b) SNR: 10db.

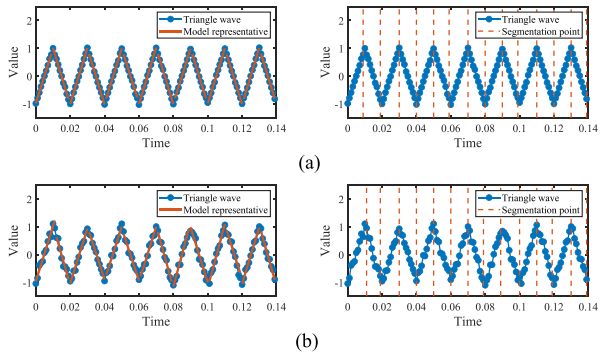


Fig. 13. Segmentation results for triangle waves. (a) SNR: 30db. (b) SNR: 10db.

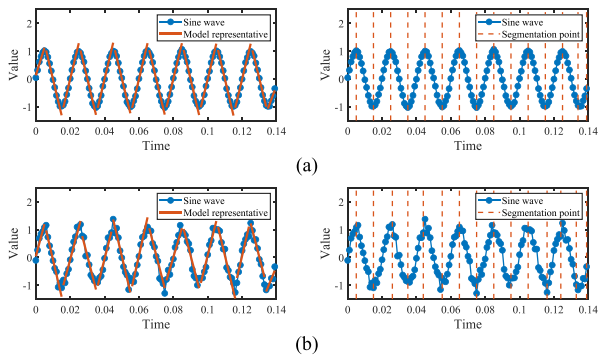


Fig. 14. Segmentation results for sine waves. (a) SNR: 30db. (b) SNR: 10db.

However, if the noise level is too high, the signal patterns will be partially swamped. Then, the selected model representatives may not describe the basic patterns of the signal well. The reason is that profile abstract is a subset selection-based algorithm. Subset selection attempted to preserve all the essential information in the representative set, including noise components. In this article, the data representatives are directly selected from the raw signals, so the original noise is not filtered. The model representatives are also generated using the raw data points with noises. If the noise level is high, we may not be able to generate high-quality model representatives. This should be a limitation of profile abstract, and we recommend that data cleaning and

filtering are conducted first before data compression and segmentation to remove the noise component.

V. Conclusion

This article proposes a novel profile data mining algorithm called Profile Abstract, which can simultaneously be used for data compression and segmentation. The proposed algorithm was developed based on subset selection, which aims to find representatives that can efficiently encode the profile data. Given pairwise dissimilarities, we formulate the problem as a linear programming model, which was solved by any common solvers efficiently. Finding data representatives can effectively compress the data redundancy by reducing the data length of each profile in the dataset. Compared with the common data compression techniques, profile abstract does not need to fuse the sensor information, so we can still track each sensor channel individually after compression. Finding model representatives can be used for pattern-based data segmentation. After data segmentation, an information gain-based algorithm was used to evaluate the importance, determining the critical primitives for model improvement. We also show that our algorithm has strong robustness on the outliers within the profiles. The effectiveness of the proposed method is demonstrated in case studies 1 and 2 by using public and proprietary data. Based on these case studies, the following conclusions are reached.

1) Profile abstract using data representatives is effective for data compression, significantly reducing the data volume while maintaining the model performance. The datasets in case studies 1 and 2 can be compressed to 60% and 30% of their original sizes, respectively.
2) Compared to other typical approaches, profile abstract has the lowest overall fitting error, demonstrating its robustness and superiority in data segmentation.
3) The proposed segment importance evaluation method with information gain can locate the critical primitives after data segmentation. Primitive input can increase TSC accuracy by about 4% and 2%, respectively, in case studies 1 and 2.
4) Profile abstract also works with different types of waveforms.

However, profile abstract is an unsupervised learning algorithm. The selected data representatives preserve the essential information rather than critical information of the raw data set. For primitive extraction, an additional step by using information gain is required to locate the important segments for primitive extraction. In the future investigation, we plan to develop a supervised profile abstract that can directly select the representatives with critical information. It can serve as a novel feature extraction approach.

References

[1] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Inform.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

[2] W. Guo, J. Jin, and S. J. Hu, "Profile monitoring and fault diagnosis via sensor fusion for ultrasonic welding," *J. Manuf. Sci. Eng.,* vol. 141, no. 8, 2019, Art. no. 081001.

[3] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Rel. Eng. Syst. Saf.,* vol. 215, 2021, Art. no. 107938.

[4] Y. Lin, Y. Li, X. Yin, and Z. Dou, "Multisensor fault diagnosis modeling based on the evidence theory," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 513–521, Jun. 2018, doi: 10.1109/TR.2018.2800014.

[5] Y. Jiang, S. Yin, and O. Kaynak, "Performance supervised plant-wide process monitoring in industry 4.0: A roadmap," *IEEE Open J. Ind. Electron. Soc.*, vol. 2, pp. 21–35, 2021, doi: 10.1109/OJIES.2020.3046044.

[6] C. Zhang, H. Yan, S. Lee, and J. Shi, "Multiple profiles sensor-based monitoring and anomaly detection," *J. Qual. Technol.,* vol. 50, no. 4, pp. 344–362, 2018.

[7] S.-K. S. Fan, C. - Y. Hsu, D. - M. Tsai, F. He, and C. - C. Cheng, "Data-driven approach for fault detection and diagnostic in semiconductor manufacturing," *IEEE Trans. Autom. Sci. Eng.,* vol. 17, no. 4, pp. 1925–1936, Oct. 2020, doi: 10.1109/TASE.2020.2983061.

[8] F. Zhu et al., "Methodology for important sensor screening for fault detection and classification in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.,* vol. 34, no. 1, pp. 65–73, Feb. 2021, doi: 10.1109/TSM.2020.3037085.

[9] J. Liu, Y. Hu, B. Wu, and Y. Wang, "An improved fault diagnosis approach for FDM process with acoustic emission," *J. Manuf. Process.,* vol. 35, pp. 570–579, 2018.

[10] V. Mahato, M. A. Obeidi, D. Brabazon, and P. Cunningham, "Detecting voids in 3-D printing using melt pool time series data," *J. Intell. Manuf.*, vol. 33, pp. 1–8, 2020.

[11] Y. Zerehsaz, C. Shao, and J. Jin, "Tool wear monitoring in ultrasonic welding using high-order decomposition," *J. Intell. Manuf.*, vol. 30, no. 2, pp. 657–669, 2019.

[12] J. Azar, A. Makhoul, R. Couturier, and J. Demerjian, "Robust IoT time series classification with data compression and deep learning," *Neurocomputing*, vol. 398, pp. 222–234, 2020.

[13] H. Wang, J. Wu, P. Zhang, and Y. Chen, "Learning shapelet patterns from network-based time series," *IEEE Trans. Ind. Inform.*, vol. 15, no. 7, pp. 3864–3876, Jul. 2018, doi: 10.1109/TII.2018.2885700.

[14] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2001.

[15] A. Gómez-Brandón, J. R. Paramá, K. Villalobos, A. Illarramendi, and N. R. Brisaboa, "Lossless compression of industrial time series with direct access," *Comput. Ind.*, vol. 132, 2021, Art. no. 103503.

[16] M. Russell and P. Wang, "Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring," *Mech. Syst. Singal Process.*, vol. 168, 2022, Art. no. 108709.

[17] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive multivariate data compression in smart metering Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 1287–1297, Feb. 2021, doi: 10.1109/TII.2020.2981382.

[18] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–41, 2018.

[19] L. Zhu, C. Lu, and Y. Luo, "Time series data-driven batch assessment of power system short-term voltage security," *IEEE Trans. Ind. Inform.*, vol. 16, no. 12, pp. 7306–7317, Dec. 2020, doi: 10.1109/TII.2020.2977456.

[20] L. Zhu, C. Lu, Z. Y. Dong, and C. Hong, "Imbalance learning machine-based power system short-term voltage stability assessment," *IEEE Trans. Ind. Inform.*, vol. 13, no. 5, pp. 2533–2543, Oct. 2017, doi: 10.1109/TII.2017.2696534.

[21] S. Malinowski, B. Chebel-Morello, and N. Zerhouni, "Remaining useful life estimation based on discriminating Shapelet extraction," *Rel. Eng. Syst. Saf.,* vol. 142, pp. 279–288, 2015.

[22] E. L. Park, J. Park, J. Yang, S. Cho, Y. - H. Lee, and H. - S. Park, "Data based segmentation and summarization for sensor data in semiconductor manufacturing," *Expert Syst. Appl.,* vol. 41, no. 6, pp. 2619–2629, 2014.

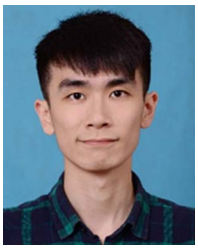
[23] P. Li et al., "A novel method for deposit accumulation assessment in dry etching chamber," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 183–189, May 2019, doi: 10.1109/TSM.2019.2904889.

[24] S. J. Wilson, "Data representation for time series data mining: Time domain approaches," *Wiley Interdiscipl. Rev. Comput. Statist.,* vol. 9, no. 1, 2017, Art. no. e1392.

[25] H. Cai et al., "A framework for semi-automated fault detection configuration with automated feature extraction and limits setting," in *Proc. IEEE 31st Annu. SEMI Adv. Semicond. Manuf. Conf.*, 2020, pp. 1–6.

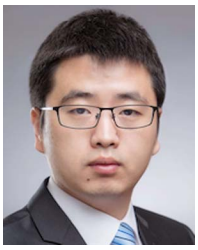
[26] E. Elhamifar, G. Sapiro, and S. S. Sastry, "Dissimilarity-based sparse subset selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2182–2197, Nov. 2016, doi: 10.1109/TPAMI.2015.2511748.

[27] D. A. Adama, A. Lotfi, and R. Ranson, "Adaptive segmentation and sequence learning of human activities from skeleton data," *Expert Syst. Appl.*, vol. 164, 2021, Art. no. 113836.
[28] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *Eur. J. Control,* vol. 13, no. 2-3, pp. 242–260, 2007.
[29] J. Liu, D. Djurdjanovic, J. Ni, N. Casoetto, and J. Lee, "Similarity based method for manufacturing process performance prediction and diagnosis," *Comput. Ind.,* vol. 58, no. 6, pp. 558–566, 2007.
[30] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," in *Data Mining in Time Series Databases*, Singapore: World Scientific, 2004, pp. 1–21.
[31] T.-C. Fu, Y.-K. Hung, and F.-L. Chung, "Improvement algorithms of perceptually important point identification for time series data mining," in *Proc. IEEE 4th Int. Conf. Soft Comput. Mach. Intell.*, 2017, pp. 11–15.
[32] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Amer. Stat. Assoc.,* vol. 107, no. 500, pp. 1590–1598, 2012, doi: 10.1080/01621459.2012.737745.



Feng Zhu received the B.S. degree in mechanical engineering from Chongqing University, Chongqing, China, and University of Cincinnati, Cincinnati, OH, USA, in 2018, respectively, and the M.S. degree in mechanical engineering from University of Cincinnati, Cincinnati, OH, USA, in 2020. He is currently working toward the Ph.D. degree in industrial engineering with the Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong.

His current research interests include prognostic and health management, data mining, and machine learning.



Jianshe Feng received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Tongji University, Shanghai, China, in 2012, Zhejiang University, Hangzhou, China, in 2015, and University of Cincinnati, Cincinnati, USA, in 2020, respectively.

He is currently the Chief Technology Officer of CRF Union, a Company focusing on AI-based smart manufacturing solutions. He has experience in advanced process control, predictive analytics (fault diagnosis, predictive maintenance, and soft measurement), and smart manufacturing fields, such as big data modeling, machine learning and cyber physical system with applications in machine tools, vehicle industry, and semiconductor manufacturing, etc. He has authored or coauthored more than 30 refereed publications and holds 70 patents in each of these areas. He was recognized as Overseas High-Caliber Personnel in Shenzhen.

Min Xie (Fellow, IEEE) received the Ph.D. degree in quality technology from Linkoping University, Linkoping, Sweden, in 1987.

He is currently the Chair Professor with the Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong. He has authored or co-authored about 300 peer-reviewed journal articles and eight books on quality and reliability engineering. More than 50 Ph.D. Students have graduated under his supervision.

Dr. Xie was a Member of the European Academy of Science and Arts in 2022. He was a recipient of the Prestigious Lee Kuan Yew Research Fellowship in 1991. He is currently an Editor and an Associate Editor, and on the editorial board of many established international journals. He has chaired many international conferences and given keynote speeches.

Lishuai Li (Senior Member, IEEE) received the B.Eng. degree in aircraft design and engineering from Fudan University, Shanghai, China, in 2007, and the M.S. and Ph.D. degrees in aeronautics and astronautics from the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, in 2009 and 2013, respectively.

She is currently an Associate Professor with the School of Data Science at City University of Hong Kong.

Her research is in the interdisciplinary field of intelligent transportation systems and data science. She has developed analytical methods (ranging from data analytics to Artificial Intelligence) using large-scale operational data for airline safety management and operations improvement, air traffic management, and health monitoring of train systems. More recently, she is working on traffic management problems and infrastructural challenges in drone delivery service and urban air mobility.

Jingzhe Lei received the B.S. degree in mathematics and applied mathematics from University of Science and Technology of China, Hefei, China, in 2021. She is currently working toward the Ph.D. degree in industrial engineering with the Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong.

Her current research interests include reliability analysis, reliability design, and applied statistics.

Jay Lee received the M.S. degree in mechanical engineering from University of Wisconsin-Madison, Madison, WI, USA, in 1983, and the M.S. degree in industrial engineering from State University of New York at Stony Brook, Stony Brook, NY, USA, in 1987, and the Ph.D. degree in mechanical engineering from George Washington University, Washington, DC, USA, in 1992.

He is an Ohio Eminent Scholar, L.W. Scott Alter Chair Professor, and a Distinguished University Professor with the University of Cincinnati, Cincinnati, OH, USA, and the Founding Director of the National Science Foundation (NSF) Industry/University Cooperative Research Center (I/UCRC) on Intelligent Maintenance Systems (www.imscenter.net) which is a multicampus NSF Industry/University Cooperative Research Center consisting of University of Cincinnati (lead institution), University of Michigan, Missouri University of S&T, and University of Texas-Austin. Since its inception in 2001, the Center has been supported by over 100 global companies and was ranked with the highest economic impact (270:1) by NSF Economics Impacts Report in 2012.

He is also the Founding Director of Industrial AI Center. He was a board Member of Foxconn Technology Group during 2019-2021. He was selected by SME as 30 visionary leaders in smart manufacturing in 2016 and 20 most influential professors in smart manufacturing in 2020.