Journal of Manufacturing Systems 43 (2017) 225–234

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper

Identification of key features using topological data analysis for accurate prediction of manufacturing system outputs

Wei Guo[a], Ashis G. Banerjee[a,b,*]

[a] Department of Industrial & Systems Engineering, University of Washington, Seattle, WA 98195, USA
[b] Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA

ARTICLE INFO

Article history:
Received 1 October 2016
Received in revised form 22 February 2017
Accepted 24 February 2017
Available online 10 March 2017

Keywords:
Topological data analysis
Feature selection
Yield prediction
Fault detection

ABSTRACT

Topological data analysis (TDA) has emerged as one of the most promising approaches to extract insights from high-dimensional data of varying types such as images, point clouds, and meshes, in an unsupervised manner. To the best of our knowledge, here, we provide the first successful application of TDA in the manufacturing systems domain. We apply a widely used TDA method, known as the Mapper algorithm, on two benchmark data sets for chemical process yield prediction and semiconductor wafer fault detection, respectively. The algorithm yields topological networks that capture the intrinsic clusters and connections among the clusters present in the data sets, which are difficult to detect using traditional methods. We select key process variables or features that impact the system outcomes by analyzing the network shapes. We then use predictive models to evaluate the impact of the selected features. Results show that the models achieve at least the same level of high prediction accuracy as with all the process variables, thereby, providing a way to carry out process monitoring and control in a more cost-effective manner.

© 2017 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Sensors play an essential role in carrying out product feasibility assessment, yield enhancement, and quality control in modern manufacturing systems such as vehicle assembly, microprocessor fabrication, and pharmaceuticals development [1]. A large number of sensors of many different types are typically employed in such systems to measure a variety of process variables ranging from operating conditions and equipment states to material compositions and processing defects over extended time periods. Thus, the volume of acquired data is so vast and heterogeneous that the contribution of individual sensor measurements in predicting the overall system outputs gets obscured. This prediction is made more challenging by the fact that the measurements are often noisy and replete with missing or outlier values. Furthermore, there is significant redundancy among the sensor measurements, leading to the presence of numerous false correlations in the recorded data. It is, therefore, necessary to perform an analysis using statistical methods that are specifically suited to identifying and filtering out existing correlations in erroneous, heterogeneous, and high-dimensional data sets.

Historically, multivariate statistical process control (MSPC) methods, such as principal component analysis (PCA) and partial least-squares (PLS), have served as the dominant mode of addressing this problem [2]. The common idea behind these methods is to define a new set of variables (known as *latent variables*) through linear combinations of the original variables that describe the sensor measurements. The set of latent variables may be reduced in some cases by performing subsequent dimensionality reduction techniques. However, these methods do not work particularly well when there are a large number of input process variables, and they share highly non-linear relationships with the system outputs that cannot be modeled using Gaussian distributions. The methods also encounter difficulties in removing the false correlations among the measurements particularly when they are erroneous. More recently, several non-linear prediction methods have been developed based on response surface fitting as well as kernelized and robust variants of the MSPC techniques [3,4]. While these methods may achieve high prediction accuracy, they do not provide any direct way of quantifying the contribution or impact of the individual process variables.

Here, we present an alternative method that leverages the emerging topic area of topological data analysis (TDA) [5] to select the important variables that are subsequently used in both linear and non-linear prediction models. More specifically, we employ a well-established TDA method known as the Mapper algorithm developed by Singh et al. [6]. It is based on the core idea of understanding the unknown topology of the high-dimensional manifold

* Corresponding author at: Department of Industrial & Systems Engineering, Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA.
E-mail addresses: weig@uw.edu (W. Guo), ashisb@uw.edu (A.G. Banerjee).

http://dx.doi.org/10.1016/j.jmsy.2017.02.015
0278-6125/© 2017 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

in which the data resides to extract hidden patterns. In particular, it clusters all the level sets of the data (defined using a projection of the high dimensional data to a lower dimensional space) to generate a topological network that represents the inherent clusters and connections among the clusters in the actual data.

This Mapper algorithm has already enjoyed immense popularity in fields such as bioinformatics and machine vision. For example, it has been used to reveal unique and subtle aspects of the folding patterns of RNA [7] and to unlock previously unidentified relationships in immune cell reactivity between patients with type-1 and type-2 diabetes [8]. Another influential example occurs in personalized breast cancer diagnosis, in which a novel subgroup of tumors with a unique mutational profile and 100% survival rate has been discovered [9]. Additionally, its deformation invariant property has been used to detect 3D objects from point cloud data with intrinsically different shapes [6].

Despite the potential of TDA in general and the Mapper algorithm in particular, there has been no prior application in the manufacturing domain to the best of our knowledge. Inspired by the success in biomedical and vision problems, we employ the Mapper algorithm and show that it facilitates the analysis of the impact of each process variable on system outputs through direct visualization. It also determines whether particular subgroups of the data are selectively responsive to different process variables, which helps to monitor and diagnose processes effectively.

We first apply the Mapper algorithm on a benchmark chemical processing data set to predict product yield [10]. Specifically, the shape of the generated topological network is used to select key features that explain the observed differences in the process measurements in a statistically significant manner. Second, we investigate the role of individual process variables in causing wafer failures in another publicly available semiconductor manufacturing data set. Although it has been recognized that k-nearest neighbor methods can identify faulty wafers effectively [11–14], the actual process variables that result in the wafer anomalies have never been identified. To this end, we demonstrate how the Mapper algorithm rapidly traces the causality hidden in this high-dimensional data set.

The rest of the paper is organized as follows. Section 2 gives an overview of the general characteristics of manufacturing data and the types of predictor (feature) and response variables that are of interest to us. In Section 3, we review the Mapper algorithm and its application in feature selection. We demonstrate the applicability of the Mapper algorithm for feature selection on two benchmark manufacturing data sets in Section 4. The effectiveness of the selected features is further assessed through predictive models. We conclude the paper with remarks and future research topics.

2. Problem formulation

In real-world manufacturing systems, data is collected using a large number of sensors that are affixed to or embedded within different machines and equipment, resulting in a high-dimensional body of heterogeneous data. The data is usually in the form of *time series measurements* of different process variables such as temperature, pressure, density, humidity, voltage, chemical or material composition including the relative proportions of various constituents of alloys or mixtures, material removal or deposition rate, number and severity of processed part flaws and defects, and so on. The sensors, thus, come in myriad forms ranging from thermocouples, pressure gauges, hydrometers, hygrometers, and voltmeters to optical cameras, spectrometers, laser scanners, and ultrasonic transducers.

Consequently, manufacturing sensor data is prone to noise terms, missing values, and outliers. These measurement errors depend on the sensitivity of the sensors to the operating conditions based on their underlying physical principles of actions. For example, it is not at all uncommon for temporary sensor hardware malfunction to result in missing values. A further problem is that of co-linearity, which is usually caused by partial redundancy in the sensor arrangement such as the placement of multiple sensors in close proximity to one another. The net result of these complications is that manufacturing systems are often “data-rich but information-poor”.

Consequently, there is a strong need to effectively select a minimal number of process variables that primarily affect the output variables of interest such as product quality and yield of a manufacturing system comprising several processes of varying types. As discussed earlier in Section 1, this form of selection facilitates process monitoring and diagnostics through targeted sensor data acquisition, storage, and processing. Even if it is cheap or convenient to manage data from all the sensors, knowing which measurements of what variables matter the most makes it feasible to rapidly regulate out-of-control processes or adapt them to manufacture high quality products at desired rates.

To formulate the problem mathematically, we suppose there are m process variables (features) and N sensor measurements recorded at different time instants. Each measurement is, thus, represented by an m-dimensional vector $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, 2, \ldots, N$. The data is then assembled into a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times m}$. Each column denotes a process variable, which is measured by one sensor operating alone or by the concurrent operation of several sensors that function in unison. The latter case is known as data fusion [15], which provides a wide range of sensed parameters, and is, hence, more reliable for data analysis.

In a batch process with batch length L, a 3-D data array $\bar{\mathbf{X}} \in \mathbb{R}^{N\times m\times L}$ is often unfolded batch-wise into a 2-D matrix $\mathbf{X} \in \mathbb{R}^{N\times mL}$. In this case, each measurement $\mathbf{x}_i \in \mathbb{R}^{mL}$ is a batch and each process variable is measured L times throughout the batch, hence, corresponding to L columns. For each row, the measurement is either spatially-sampled or temporally-sampled. For instance, in the semiconductor manufacturing environment, electronic wafer map data collected from in-line measurements are sampled spatially across the surface of the wafer for defect inspection [16]. Usually, there will also be one or more response variables to reflect the output quality or quantity. We write the output with r response variables into a matrix $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N\times r}$, where each response variable is represented by one column. Response variables are commonly seen as continuous variables denoting production yields or binary variables indicating pass or fail outcomes.

3. Technical approach

We now present the framework of the Mapper algorithm and outline the typical pipeline of feature selection using the Mapper algorithm. For more details about the Mapper algorithm and concrete examples of real applications, we refer the reader to [6,17].

3.1. Mapper algorithm

The Mapper algorithm can be considered as a partial clustering algorithm inspired by the classical discrete Morse theory [6]. In topology, discrete Morse theory enables one to characterize the topology of high dimensional data via some functional level sets [18]. More specifically, given a topological space $\mathcal{X}$, when $h : \mathcal{X} \to \mathbb{R}$ is a smooth real-valued function (Morse function), topological information of $\mathcal{X}$ is inferred from the level sets $h^{-1}(c)$ for some real c.

The Mapper algorithm extends this inference to incorporate standard clustering methods for the analysis of high dimensional data sets. Given a data matrix **X**, the setup of the Mapper algorithm includes:

1. Set resolution parameters: a number of intervals l and overlap percentage p, where $p \in (0, 100)$.
2. Compute the pairwise distance matrix $\mathbf{D} = [d(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{N \times N}$ based on the distance metric chosen.
3. Select a filter function $f : \mathcal{X} \to \mathbb{R}^n$ to stratify the data.

The most crucial step in the Mapper algorithm is the selection of the filter function to "guide" a clustering algorithm on the high-dimensional data. A few common filter functions include Gaussian kernel density estimator, eccentricity filter, principal metric SVD filter, and eigenvectors of graph Laplacians. Moreover, we can take the projection found by dimensionality reduction/manifold learning techniques that maps the high-dimensional data to a low-dimensional space as the filter function. For example, in the chemical manufacturing process study, our choice of the filter function is the 2-D projection found by the multidimensional scaling (MDS) method. MDS in this case attempts to embed the data such that the pairwise distances in the high-dimensional space are preserved in the 2-D Euclidean space. Accordingly, the 2-D embedding coordinates denoted by $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N$, are the minimizers of a loss function, σ, defined as

$$\sigma(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} (||\mathrm{x}_i - \mathrm{x}_j||_2 - d(\mathbf{x}_i, \mathbf{x}_j))^2. \quad (1)$$

Therefore, the filter function is specified as

$$f : \mathcal{X} \to f_1 \times f_2, \quad (2)$$

where f_1 and f_2 are coordinates of $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N$ along the 1st and 2nd dimension, respectively. For the study of fault detection in the semiconductor manufacturing processes, we employ the 2-D projection found by the t-distributed stochastic neighboring (t-SNE) algorithm as the filter function [19]. t-SNE aims to preserve the joint probabilities p_{ij} that measure similarities between $\mathbf{x}_i$ and $\mathbf{x}_j$, i, $j = 1, 2, \ldots, N$, as much as possible in the 2-D space. Specifically, p_{ij} is defined as

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \quad (3)$$

where the conditional probability $p_{j|i}$ that represents the similarity of $\mathbf{x}_j$ to $\mathbf{x}_i$ is given by

$$p_{j|i} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||\mathbf{x}_i - \mathbf{x}_k||^2/2\sigma_i^2)}. \quad (4)$$

Herein the variance of the Gaussian σ_i centered at $\mathbf{x}_i$ is determined by a predefined perplexity. On the other hand, the joint probability q_{ij} that reflects the similarity between 2-D embedding coordinates $\mathrm{x_i}$ and $\mathrm{x_j}$ is defined based on a heavy-tailed Student's t-distribution with one degree of freedom:

$$q_{ij} = \frac{(1 + ||\mathrm{x}_i - \mathrm{x}_j||^2)^{-1}}{\sum_{k \neq l} (1 + ||\mathrm{x}_k - \mathrm{x}_l||^2)^{-1}}, \quad (5)$$

such that dissimilar measurements in the m-D space are mapped far apart in the 2-D space. $\mathrm{x_i}$, $i = 1, 2, \ldots, N$ are then determined by minimizing the Kullback–Leibler divergence between the joint probability distribution P in the m-D space and the joint probability distribution Q in the 2-D space,

$$D_{\mathrm{KL}}(P||Q) = \sum_{j=2}^{N} \sum_{i=1}^{j-1} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (6)$$

Likewise, the filter function in this case is given by

$$f : \mathcal{X} \to g_1 \times g_2, \quad (7)$$

where g_1 and g_2 are coordinates of $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N$ along the 1st and 2nd dimension, respectively. In addition, it should be noted that when N is too large, numerical optimization techniques are used.

The algorithm is summarized as a flow chart in Fig. 1. After setup, the first step is to divide the filter range and cover it with overlapped intervals so that the clustering algorithm in the ensuing step focuses on the local information of the data that is likely to be ignored by the clustering over the entire data. The second step is to cluster the data in the original high dimensional space for every level set (subset). The Mapper algorithm is not tied to any particular clustering algorithm. However, it is always required to estimate certain parameters (thresholds) in order to determine the number of clusters in every level set. The last step of the algorithm is to link any two clusters from neighboring level sets together if they have one or more common data points.

In the 1-D Mapper case, the output is a 1-D simplicial complex that comprises only vertices (0-simplex) and edges (1-simplex). More generally, if the target space is $\mathbb{R}^n$, higher simplices may appear in the output simplicial complex, such as triangular faces (2-simplex) whenever three clusters from neighboring level sets have nonempty intersections. The compressed representation of the simplicial complex allows us to obtain a qualitative understanding of how the data are organized on a large scale through direct visualization. Additionally, the resolution of the complex changes from coarse to fine as the number of intervals l increases. This change of resolution reflects the change in topology of the data set.

It is worth mentioning that the filter range is not necessarily covered by l overlapped intervals of equal length. In fact, the Mapper algorithm is highly parallelizable. To improve the efficiency of parallel computation, it is more convenient to decompose the filter range into l overlapped intervals wherein each interval contains the same number of points so that the running times would be similar for all the level sets.

3.2. Application of Mapper algorithm to feature selection

The output graph of the Mapper algorithm contains the information of clusters in the data at the local level, as well as their positions relative to one another and to the remainder of the data set. Therefore, the principle of applying the Mapper algorithm to feature selection is to recognize shapes in the resulting graph that encode the essential structural information of the data. Typical shapes of interest found in a graph are subgroups of clusters that display distinct patterns such as "loops" (continuous circular segments) and "flares" (long linear segments), as opposed to portions of the graph within which the local environment of each cluster is roughly identical.

Aside from shapes of interest, we also discern the trends in terms of the *output values* associated with each cluster in the graph rendered by the Mapper algorithm, such as which clusters contain several measurements from faulty samples in the case of anomaly detection. Furthermore, we are able to distinguish the fundamental subgroups from artifacts by observing whether the shapes of the given subgroups remain consistent when the resolution parameters are varied over a wide range of values. After the fundamental

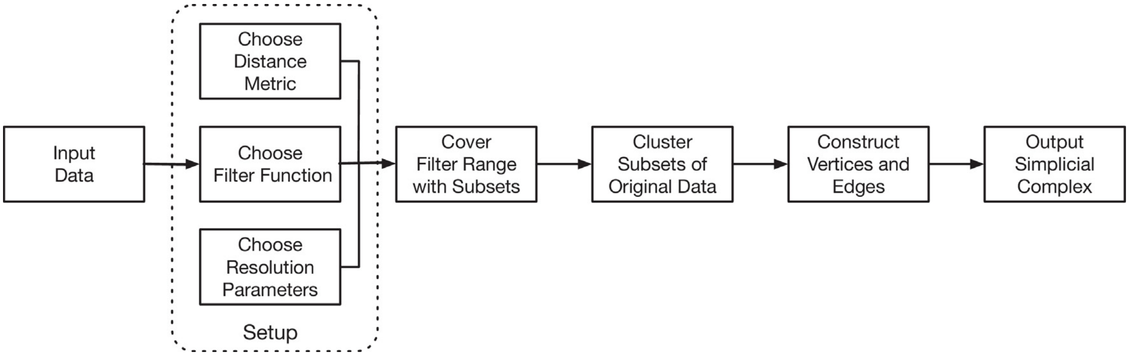


Fig. 1. Framework of the Mapper algorithm for generating topological networks.

subgroups of interest are detected, standard statistical tests, such as the Kolmogorov–Smirnov test and Student's *t*-test, are performed to identify the features that best distinguish the subgroups from one another. The final set of features thus selected are then fed into classification or regression models to perform a desired prediction task.

Thus, we end up addressing two main challenges in applying the Mapper algorithm to identify key features from manufacturing data. The first one pertains to a suitable selection of the filter function so as to map the high-dimensional data to a low-dimensional space where the data can be conveniently stratified. Unlike in the case of point clouds, meshes, or images, there is no well-established function, and we select the MDS projection method based on final output prediction quality. The second challenge is on varying the resolution parameters appropriately so that the fundamental subgroups are correctly distinguished from artifacts in the generated topological networks. Choice of a coarse granularity of variation leads to the appearance and disappearance of subgroups, whereas the use of very fine granularity makes the process time-consuming. We vary the parameters in a simple way such that a majority of the subgroups, which are identified at a particular resolution, remain intact as the parameters change (the other subgroups appear and disappear enabling us to characterize them as artifacts).

4. Results

In this section, we conduct two studies to show how to achieve feature selection using the Mapper algorithm. With selected features, the first study obtains accurate predictions of productivity for a chemical processing benchmark, and the second study reaches a high accuracy in fault classification for a semiconductor etch process.

4.1. Prediction of manufacturing productivity

The data is for a chemical process plant that is described in [20] and can be obtained from the R package "AppliedPredictiveModeling". The data set contains 176 measurements of biological materials for which 57 variables are measured, where there are 12 biological starting materials and 45 manufacturing process parameters (predictors). The starting material is generated from a biological unit and has a wide range of quality and characteristics. The manufacturing process parameters include temperature, drying time, washing time, and concentrations of by-products at various steps. The biological variables are used to gauge the quality of the raw material before processing but cannot be changed, whereas the manufacturing process parameters can be changed during operations. The measurements are not independent since partial measurements are produced from the same batch of biological starting materials. We aim to investigate the relationships between the predictors and the final pharmaceutical product yield, and develop a model to estimate the percentage yield of the manufacturing process.

4.1.1. Data preprocessing

As we want to maximize the level of automation in predicting manufacturing productivity for industrial applications, the data is preprocessed with a minimum amount of work. First, the outliers in the data set are marked as missing values and the features with near-zero variances are discarded. During this step, BiologicalMaterial07 is removed. Second, we apply Box-Cox transformation to the data to eliminate distributional skewness, and scale each column of the data to zero mean and unit variance. The last step is to impute the missing values by the *k*-NN method with $k = 5$. Note that all of these steps can be handled automatically in the production environment.

4.1.2. Feature selection

To begin with, we choose Euclidean distance as the metric to represent the similarity between the measurements. In this work, much effort is spent on the suitable selection of the filter function due to the complex underlying structure of the data. Some commonly considered filter functions include the eccentricity function, linear and nonlinear projections such as PCA and Isomap. Regarding the quantity of interest and the purpose of the filter function, we use the response variable to "supervise" the stratification of the data. The output of the MDS method that reduces the data set to 2 dimensions is shown to provide the smoothest variations of the response values over the embedding coordinates, and is eventually chosen as the filter function.

In the next stage, each dimension is covered by 14 intervals of equal length with 80% overlap between any two successive intervals, leading to the filter range being divided into 196 level sets in all. Density-based spatial clustering of applications with noise (DBSCAN) method is subsequently employed for clustering in each level set, where the number of clusters is determined by the minimum number of measurements in a cluster and the maximum distance between two measurements in the same cluster [21].

As a result, we implement the steps above in Python[1] and obtain a topological network in the form of a simplicial complex as shown in Fig. 2. Each cluster is represented by a node, and the node size is proportional to the number of measurements in the cluster based on a logarithmic scale. An edge is generated between any two nodes from neighboring level sets that have at least one measurement in common. We normalize the value of the product yield within the range 0–1, and color each node based on the average normalized yield value for the measurements in the node. As is seen in

[1] Code adapted from https://github.com/MLWave/kepler-mapper.

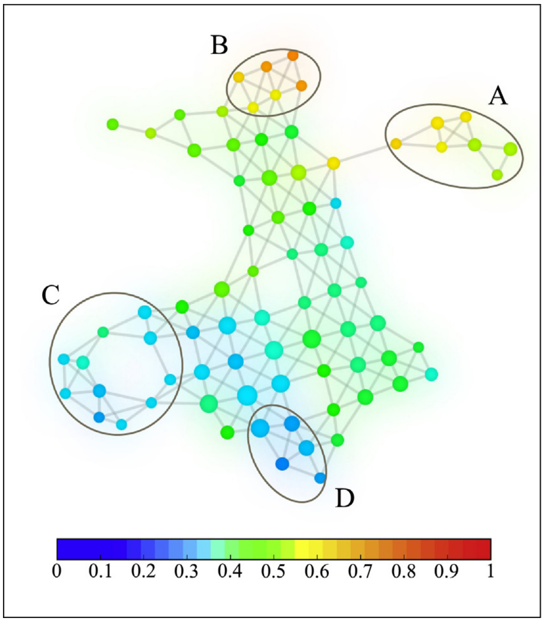


Fig. 2. Topological network derived from the chemical processing data at a specified resolution. Each node is colored based on the average normalized yield value for the measurements in the node, where the normalized yield varies between 0 and 1. High and low yield subgroups are isolated from the rest of the network, where A and C are extracted as outer flares and B and D are extracted from the periphery of the network as suggested in [17]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Fig. 2, the shape of the data is captured by the generated topological network after iterating through multiple times at various resolution scales. The resolution is set at a large number of intervals and a high overlap percentage. A large number of intervals help to uncover subtle aspects of the shape of the data rather than a blob, and a high overlap percentage seeks to have all nodes connected as a single network if possible. Thus, we are able to find out the subgroups of interest and acquire an overall structural information of how the data is distributed within the network. In this problem, we are interested in the difference in patterns between the measurements with high and low yields. Notice that the high yields are separated into two subgroups, and the low yields are also bifurcated into two subgroups with different patterns. Therefore, two subgroups of measurements with high yield and two subgroups of measurements with low yield are extracted from the data as encircled in Fig. 2.

Two-sample Kolmogorov–Smirnov (KS) test, which is sensitive to the difference in both location and shape of the empirical cumulative distributions of two groups, is then performed between subgroups A and C, A and D, B and C, and B and D over all the columns in the data matrix to identify the features that best discriminate between them. We record the largest KS-score and the associated *p*-value as well as the adjusted *p*-value among the four tests for each feature. The results are presented in Table 1 and further visualized in Fig. 3. The *p*-values are adjusted using the well-established Benjamini–Hochberg (B–H) procedure [22,23] that is commonly used to reduce the false discovery rate (FDR) when multiple features or variables are evaluated for statistical significance. The B–H adjustment provides greater flexibility at the expense of somewhat higher FDR as compared to the traditional Bonferroni correction method. This adjustment is, thus, better suited for our purpose as we want to identify *all* the process variables that may have an impact on the manufacturing system outputs. The most salient features are selected based on high KS-scores (>0.9) and low adjusted *p*-values (<0.05), where 11 of them are the measurements of manufacturing processes that can be controlled. Thus, the product yield should be improved by altering these steps in the process to have higher or lower values. We also note that the selection of the most salient features are not affected by the B–H procedure in this case.

Fig. 4 examines the effects of the features on the product yield and probes the relationships between them. We color the same network nodes based on normalized feature values. The color of each

Table 1
Kolmogorov–Smirnov test to identify features that best differentiate between the subgroups.

Feature	KS-score	*p*-value	Adj. *p*-value	Feature	KS-score	*p*-value	Adj. *p*-value
B01[a]	0.882	5.53e−7	2.21e−6	M18	0.882	1.93e−7	7.20e−7
B02[b]	1	7.57e−8	1.06e−6	**M19**	1	1.95e−9	2.18e−8
B03	1	7.57e−8	1.06e−6	M20	0.778	1.12e−4	3.49e−4
B04	0.917	1.16e−6	9.28e−6	M21	0.598	0.002	0.004
B05	0.739	2.36e−5	6.01e−5	M22	0.203	0.821	0.901
B06	1	7.57e−8	1.06e−6	M23	0.369	0.142	0.204
B08	1	7.55e−9	8.46e−8	M24	0.539	0.007	0.012
B09	0.417	0.054	0.082	M25	0.787	5.22e−6	1.39e−5
B10	0.728	3.32e−5	8.09e−5	**M26**	0.941	2.08e−8	1.17e−7
B11	0.886	4.95e−7	2.22e−6	M27	0.717	4.64e−5	1.04e−4
B12	0.952	1.34e−8	9.39e−8	**M28**	1	1.95e−9	2.18e−8
M01	0.533	0.008	0.013	**M29**	1	1.95e−9	2.18e−8
M02	1	7.55e−9	8.46e−8	M30	0.768	2.15e−5	6.35e−5
M03	0.650	0.001	1.23e−3	**M31**	0.944	6.14e−8	3.82e−7
M04	1	1.88e−7	1.75e−6	**M32**	0.941	2.08e−8	1.17e−7
M05	0.647	5.95e−4	1.28e−3	M33	0.894	1.28e−7	5.50e−7
M06	0.722	4.32e−4	1.15e−3	M34	0.238	0.718	0.855
M07	0.261	0.521	0.635	M35	0.501	0.011	0.017
M08	0.314	0.259	0.345	M36	0.787	5.22e−6	1.39e−5
M09	0.944	1.08e−6	6.05e−6	M37	0.317	0.284	0.379
M10	0.833	2.64e−5	1.06e−4	M38	0.381	0.167	0.253
M11	0.886	4.95e−7	2.22e−6	M39	0.294	0.371	0.472
M12	0.667	0.001	0.003	M40	0.278	0.560	0.713
M13	1	1.88e−7	1.75e−6	M41	0.262	0.601	0.783
M14	0.692	9.71e−5	2.01e−4	M42	0.488	0.034	0.064
M15	0.905	8.40e−8	4.28e−7	M43	0.846	7.12e−7	2.21e−6
M16	0.690	0.001	0.002	M44	0.291	0.342	0.426
M17	0.833	2.64e−5	1.06e−4	M45	0.222	0.819	0.936

[a] B: BiologicalMaterial; M: ManufacturingProcess.
[b] Key features characterized with high KS-score (>0.9) and low adjusted *p*-value (<0.05) are written in bold.

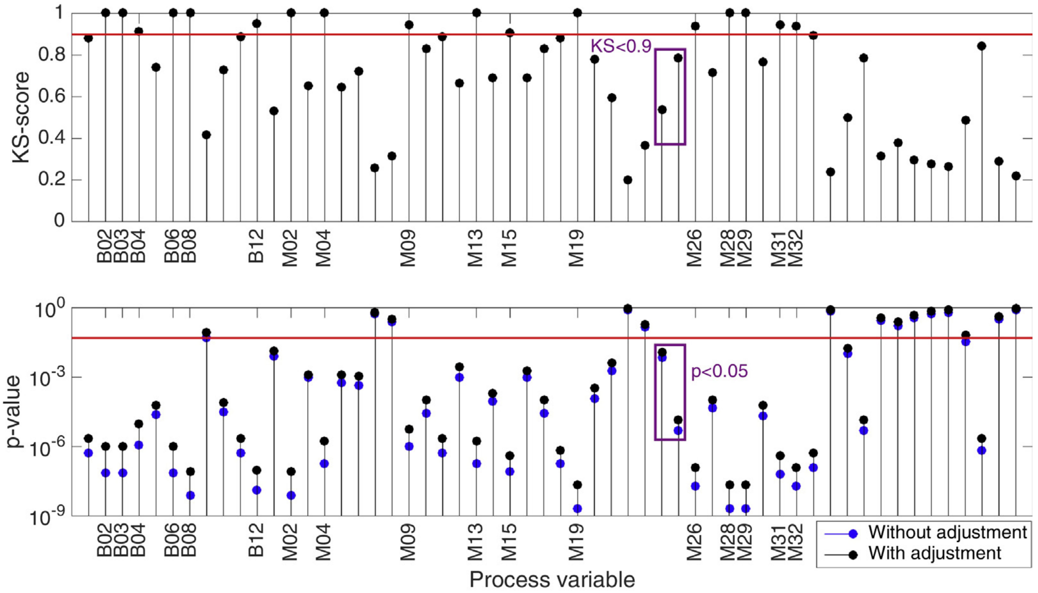
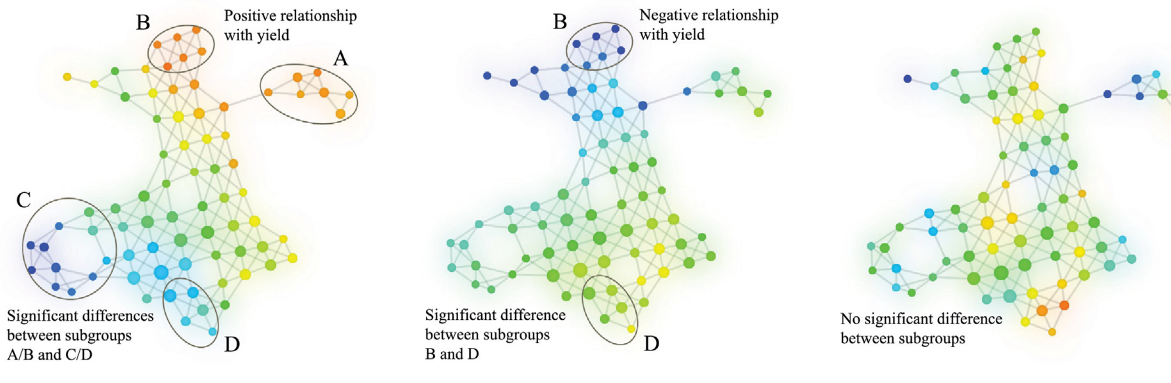


Fig. 3. Key features (marked by *x*-axis tick labels) that best differentiate between the subgroups are identified by Kolmogorov–Smirnov tests as those which yield a high KS-score (>0.9) and a low corresponding adjusted *p*-value (<0.05).


B
Positive relationship with yield
A
C
Significant differences between subgroups A/B and C/D
D
B
Negative relationship with yield
Significant difference between subgroups B and D
D
No significant difference between subgroups

(a) BiologicalMaterial06 (b) ManufacturingProcess13 (c) ManufacturingProcess22

Fig. 4. Topological networks colored based on different selection of features at the same resolution as in Fig. 2. For every network, each node is colored based on the average normalized feature value of all the measurements included in the node, where the normalized feature value varies between 0 and 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

node encodes the normalized feature value averaged across all the measurements in the node, with blue denoting a low value and red indicating a large value. We see that significant differences between the subgroups exist both for BiologicalMaterial06 and ManufacturingProcess13, both of which are selected in Table 1. Contrary to Fig. 4(a) and (b), an unselected feature ManufacturingProcess22 shows no significant difference between any of the subgroups in Fig. 4(c). Meanwhile, on comparing with Fig. 2, BiologicalMaterial06 shows a positive relationship with the yield, whereas ManufacturingProcess13 displays a negative relationship with the yield.

Table 2
Estimation errors and computation times for different models with all features and selected features.

	Method	Errors (RMSE)		Computation times (s)	
		Training	Testing	Training	Testing
All features	PLS	1.20 ± 0.05	1.29 ± 0.10	1.33 ± 0.30	0.005 ± 0.001
	RF	1.13 ± 0.06	1.15 ± 0.15	130 ± 2.40	0.006 ± 0.001
	Cubist	1.00 ± 0.07	1.15 ± 0.13	58.5 ± 4.11	0.025 ± 0.004
	kGP	1.21 ± 0.04	1.25 ± 0.11	8.14 ± 0.43	0.002 ± 9.4e-4
Selected features	PLS	**1.13** ± **0.05**	**1.25** ± **0.09**	**1.02** ± **0.16**	**0.002** ± **3.8e−4**
	RF	**1.11** ± **0.06**	**1.13** ± **0.15**	**91.2** ± **2.53**	**0.005** ± **8.9e−4**
	Cubist	1.05 ± 0.10	1.20 ± 0.08	**24.7** ± **1.54**	**0.008** ± **0.002**
	kGP	**1.19** ± **0.05**	**1.22** ± **0.11**	**6.24** ± **0.33**	**0.001** ± **6.3e−4**

4.1.3. Predictive modeling

Four regression models, PLS, random forest (RF), cubist and Gaussian process with a Gaussian kernel (kGP), are chosen to predict the yield of the chemical manufacturing process. These models represent a linear model, a tree-based model, a rule-based model and a kernelized technique, respectively. We randomly split the entire data set into a training set and a testing set in 7:3 ratio. Parameters in each trained model are tuned to be optimal using 25 iterations of 10-fold cross-validated search over the parameter set. The trained models are then adopted to predict the percentage yield for the testing set.

Table 2 compares the prediction results and computation times between all the features and just the selected features for the models based on 30 runs. The prediction accuracy is evaluated by the root mean squared error (RMSE) and computation times are

Table 3
Top 17 important features identified by different methods[a, b, c].

PLS	RF	Cubist	kGP	TDA
M32	M32	M32	M32	B02
M36	B06	M17	B06	**B03**
M17	M17	M31	M13	**B06**
M13	M31	B06	M17	**B08**
M09	B03	M13	M36	**M02**
M33	M13	M04	B03	**M04**
M06	M01	M21	M31	**M13**
B06	B08	B03	M33	M19
M12	B11	M09	M09	**M28**
B03	M39	M01	B04	**M29**
B04	B04	M20	M06	B12
B08	M20	M39	M29	**M09**
B01	B09	B04	M04	**M31**
B11	M06	M33	B11	M26
M31	M18	M02	M02	**M32**
M04	M11	M05	B01	**B04**
M28	M33	B10	M27	M15

[a] B02, B12, M30, M40 are excluded from the PLS, RF, Cubist or kGP model since these features are removed before models being trained due to their high correlation with other features.
[b] The important features given by PLS, RF, Cubist, kGP and the TDA method are ranked based on the weighted sums of the absolute regression coefficients, average impurity reduction, usage in the rule conditions, and KS-score in Table 1, respectively. Features with the same KS-score are ordered by their feature names. For the kGP method, a LOESS smoother is fitted to assess the relationship between each feature and the outcome. The importance of the features is ranked by their R^2 statistics.
[c] The ranking of feature importance varies somewhat with the training samples and the results in Table 3 are reported based on a certain choice of the samples.

Table 4
Process variables for semiconductor wafer fault detection.

No.	Variables	No.	Variables
1	BCl_3 flow	10	RF power
2	Cl_2 flow	11	RF impedance
3	RF bottom power	12	TCP tuner
4	Endpoint A detector	13	TCP phase error
5	Helium chuck pressure	14	TCP impedance
6	Pressure	15	TCP top power
7	RF tuner	16	TCP load
8	RF load	17	Vat valve
9	RF phase error		

Table 5
Induced faults and experiments associated with each faulty wafer.

No.	Exp.	Fault names	No.	Exp.	Fault names
1	29	TCP power +50[a]	11	31	Cl_2 flow +5
2	29	RF power −12	12	31	BCl_3 flow −5
3	29	RF power +10	13	31	Pressure +2
4	29	Pressure +3	14	31	TCP power −20
5	29	TCP power +10	15	33	TCP power −15
6	29	BCl_3 flow +5	16	33	Cl_2 flow −10
7	29	Pressure −2	17	33	RF power −12
8	29	Cl_2 flow −5	18	33	BCl_3 flow +10
9	29	Helium chuck pressure	19	33	Pressure +1
10	31	TCP power +30	20	33	TCP power +20

[a] The addition term in each fault name represents an offset of the process variable from its normal baseline value during the batch. For example, "TCP power +50" means that the induced fault is an increase of 50 units in the TCP power.

measured on a laptop with an Intel Core i5 (1.7 GHz) CPU and 4 GB RAM. We find that the models with selected features achieve comparable performance as the models with all the features. Especially, in the case of the PLS, RF and kGP models, selected features outperform all the features in terms of both training and testing errors, which highlights the efficacy of the selected features based on the Mapper algorithm. Meanwhile, the training times are reduced by about 30%~60% for the RF and Cubist models using the selected features.

Table 3 compares the top features identified by different methods. Since there is almost no dominant feature due to the complexity of the data, the features identified by each method vary from each other. For the Mapper algorithm, the feature that overlaps with the features identified by at least one of the other methods is highlighted. In fact, even though the other four methods have the ability to detect significant features, it is difficult for them to interpret how the yield is affected by these features. In contrast, the Mapper algorithm is well capable of unraveling the relationships between the features and product yield through easy and rapid visualization as shown in Fig. 4.

4.2. *Fault detection of semiconductor manufacturing process*

In this study, the data set[2] is collected from an Al stack etch process performed on a commercial-scale Lam 9600 plasma etch tool at Texas Instrument Inc. [24]. The data consists of 108 normal wafers and 21 faulty wafers from three separate experiments (denoted as experiment numbers 29, 31, and 33) with 19 process variables for monitoring. Since two of the process variables, RF reflected power and TCP reflected power, remain almost zero during the entire batch, only 17 process variables are used for fault detection and diagnosis, as tabulated in Table 4. Moreover, one normal wafer and one faulty wafer are removed from the data set due to a large amount of missing values. Finally, because the experiments were run several weeks apart from one another, the process shift and drift lead to different means and covariance structures in the data gathered in each of the three experiments.

The faulty wafers were intentionally induced through the modification of several of the process variables: TCP power, RF power, pressure, BCl_3 or Cl_2 flow rate, and Helium chuck pressure. To simulate an actual sensor failure, readings from the corresponding sensor were intentionally adjusted using a bias term so that its mean value was equal to the original baseline value of the relevant process variable. For example, if the TCP power was modified from its normal baseline value of 350 W to a value of 400 W, the values of TCP power in the data set would be reset to a mean of 350 W by adding a constant bias of −50 W. Table 5 lists the induced faults associated with each faulty wafer in the three experiments. In general, the modification of any one of the process variables may generally be expected to result in changes to the remainder of them because of correlations which may exist between the process variables. In this work, our goal is to determine the process variables which are most affected by the induced faults and use this information to construct a classification model for fault detection.

4.2.1. *Data preprocessing*

We follow a similar data preprocessing step as the aforementioned study. First, we remove the first five records to eliminate effects which due to initial fluctuations. To accommodate shorter batches, we retain 85 records in each batch to ensure that each batch record is of equal length. Next, the 3-D data array is unfolded batch-wise to a 2-D matrix, resulting in a total of 1445 features, i.e. each feature is considered to be a pairwise combination of a process variable and a batch record. Finally, each column of the 2-D matrix is scaled to zero mean and unit variance.

4.2.2. *Feature selection*

The etching process reflected in our data set is a typical nonlinear, multimodal process. For this reason, the filter function used to identify a 2-D embedding of the data is taken to correspond to that of t-SNE, a nonlinear dimensionality reduction method which, as previously mentioned, tends to map dissimilar measurements

[2] Available at http://software.eigenvector.com/Data/Etch/index.html.

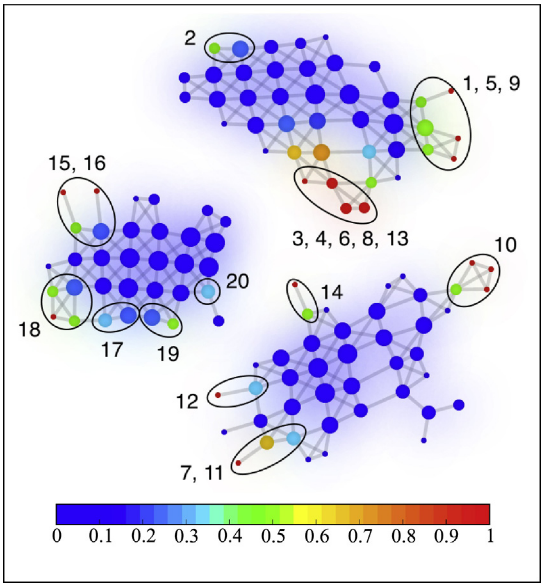


Fig. 5. Topological network derived from the semiconductor data at a specified resolution. Each node is colored based on the average output for the measurements included in the node, where the output of a faulty wafer is 1 while the output of a normal wafer is 0. Subgroups that consist of nodes containing measurements of faulty wafers are extracted from each subnetwork. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

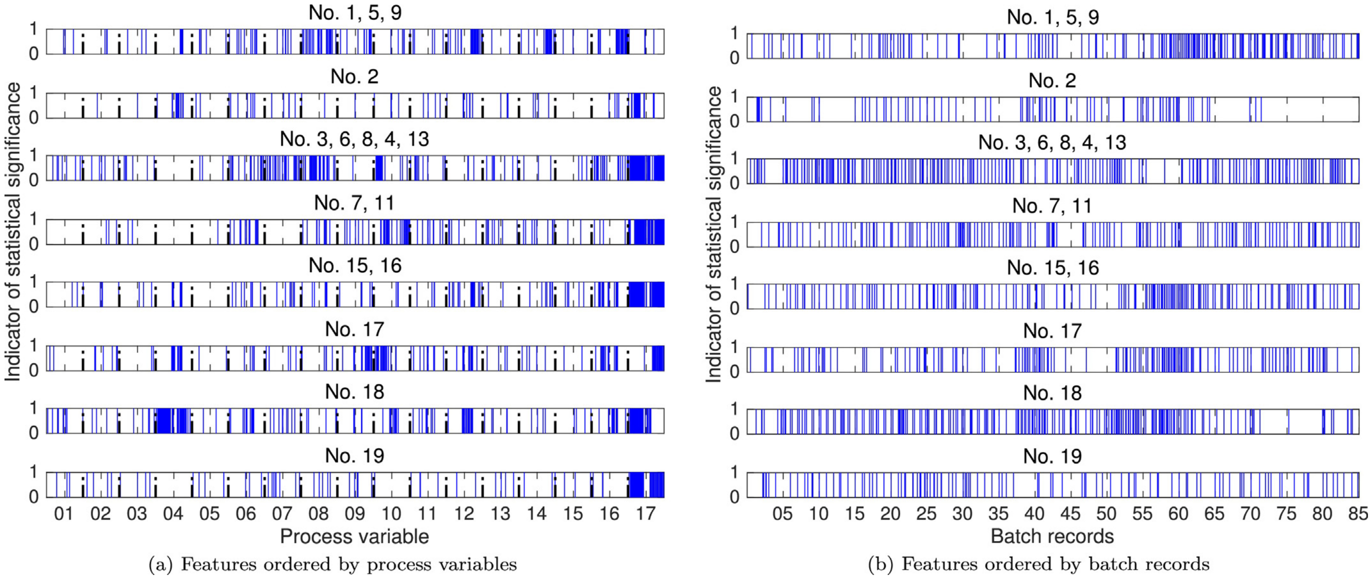
far apart in the low-dimensional space. The distance metric used between a given pair of 1445-dimensional data measurements is, therefore, taken to be the joint probability between the two, as defined in Eq. (3). The resolution is 24 intervals per dimension with 80% overlap between adjacent intervals and the DBSCAN method is once again used for subset clustering. Fig. 5 shows that the generated topological network of the semiconductor data is separated into three subnetworks. This is consistent with the fact that the data sets collected from the three experiments have different means and somewhat different covariance structure. It is worth noting that faulty wafers 7 and 13 are two exceptions in the sense that each one is grouped with other wafers which originated from a different experiment.

We color each node based on the average output values across all the measurements in the node. The output is either 0 or 1, representing a normal or a faulty wafer, respectively. As expected, measurements representing faulty wafers are positioned at the boundary regions of each subnetwork. We conjecture that this is because each faulty wafer was induced differently, giving rise to different behaviors in the wafer processing. We further identify subgroups consisting of nodes containing measurements of faulty wafers in Fig. 5, as indicated by closed elliptical paths. Since the subgroups for faulty wafers 10, 12, 14, and 20 have extremely small sample size, they are excluded from the statistical tests for feature selection. For the rest of the subgroups, the Wilcoxon rank-sum tests are performed across all of the process variables throughout the batch. As a non-parametric alternative to the two-sample Student's t-test, the Wilcoxon rank-sum test is able to handle small sample size for non-normal distributions. These tests are conducted between each subgroup of faulty wafers and the nodes corresponding to normal wafers in the rest of its subnetwork, excluding those which belong to other subgroups of faulty wafers. The results of these tests are shown in Fig. 6, where they are organized by process variable in subfigure (a) and by batch record in subfigure (b).

By comparing the two rankings of the features, we find that statistically significant features ($p < 0.05$) are more concentrated within individual process variables than within individual batch records. For example, it is evident that process variable 17 (Vat valve) is strongly correlated with faulty wafers, while process variables 5 (Helium chuck pressure) have little impact on wafer failure. As in Section 4.1, we perform B–H procedure to adjust the p-values and count the occurrence of each statistically significant feature throughout the batch for every process variable. The results for both raw and adjusted p-values are shown in Fig. 7. It is seen that the relative importance of the process variables remains more or less the same after B–H adjustment, especially for the first eight process variables. Hence, we only select the first eight process variables for fault classification.

4.2.3. *Predictive modeling*

To build a fault detection classifier, we first compute the column means throughout the batch for each variable and use them for the new feature values. The transformed data is then randomly split



(a) Features ordered by process variables

(b) Features ordered by batch records

Fig. 6. Wilcoxon rank-sum test to identify the features that best differentiate between faulty wafers and normal wafers. The features are ordered by (a) process variables and (b) batch records, respectively. Statistically significant features ($p < 0.05$) have values of 1 as represented by the blue lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

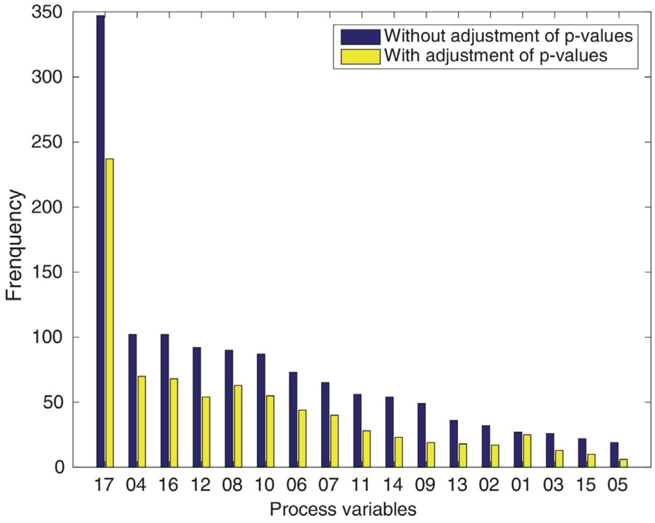


Fig. 7. Counts of statistically significant features in terms of differentiating between faulty and normal wafers for each process variable.

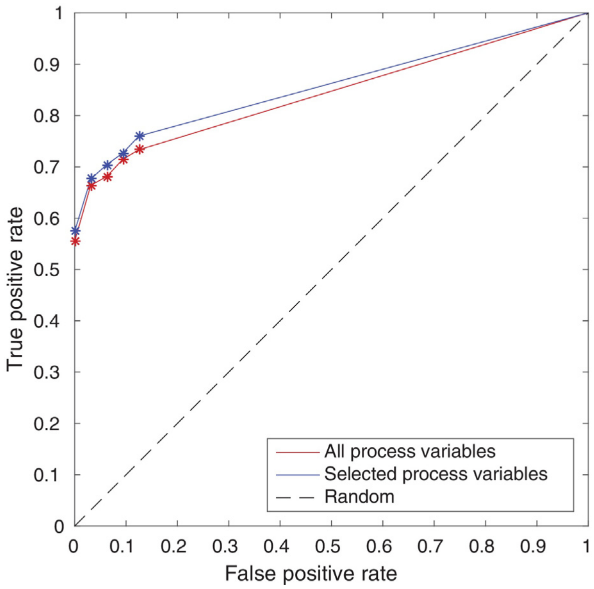


Fig. 8. ROC curves of Gaussian kernel SVM classifiers on the data with all process variables and with selected process variables.

into a training set and a testing set in the ratio of 7:3, where each set maintains the same proportion of normal and faulty wafers. The standard soft margin C-support vector machine (SVM) classifier with a Gaussian kernel, as implemented in LIBSVM [25], is employed for fault classification. The cost factor C and the variance σ of the Gaussian kernel are tuned using 10-fold cross-validation on the training set using an iterative grid search. We start a coarse grid search with exponentially growing sequences of C and γ first, thereafter proceeding with finer grid searches in the vicinity of the optimal region yielded by the previous grid search. Each grid search includes a total of 50 pairs of (C, γ) values which are used to apply the training model. To illustrate the performance of the fault classifiers, receiver operating characteristic (ROC) curves for the testing set with all process variables and with selected variables are reported in Fig. 8. As seen in Fig. 8, the fault classifier with the eight selected process variables outperforms the classifier which uses all process variables, indicating the effectiveness of the former variables in predicting wafer failure. Meanwhile, about 18% reduction in the computational time is achieved from ~1.1 s to ~0.9 s of each run.

5. Conclusion

In this paper, we apply a powerful TDA tool, the Mapper algorithm, for predictive analysis of a chemical manufacturing process data set for yield prediction and a semiconductor etch process data set for fault detection. We show that the Mapper algorithm adds a new perspective to the traditional means of feature selection and provide critical insights hidden in the complex data. Through direct visualization, we generate an abstract view of the data to facilitate a better understanding of the casual relationships between the features and manufacturing system outputs. The contributions of the work are summarized below:

- To the best of our knowledge, we successfully demonstrate the value of any TDA method in the manufacturing systems domain for the first time.
- We effectively detect structural information present in manufacturing systems data, which is highly valuable as it allows identification of subgroups of interest for targeted hypothesis testing with respect to the differences in the observed patterns.
- We show that just using the identified features with the most significant causal relationships provides a similarly high level of prediction accuracy as achieved with the complete set of features but with substantially reduced training times.

Thus, our results open a feasible path for efficient manufacturing process monitoring and control especially in complex systems with a large number of process variables. In the future, we plan to embed the Mapper algorithm in a sparse sensing framework to further reduce the need for measurements in an optimal manner. We further aim to combine the Mapper algorithm with existing machine learning techniques to increase the robustness of our approach and yield a practical method which is more suitable to the context of high-dimensional, heterogeneous manufacturing data in general.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments.

References

[1] Tlusty J. Manufacturing processes and equipment. Prentice Hall; 2000.
[2] MacGregor JF, Kourti T. Statistical process control of multivariate processes. Control Eng Pract 1995;3:403–14.
[3] Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 1998;10:1299–319.
[4] Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel Hilbert space. J Mach Learn Res 2001;2:97–123.
[5] Carlsson G. Topology and data. Bull Am Math Soc 2009;46:255–308.
[6] Singh G, Mémoli F, Carlsson GE. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: Eurographics symposium on point-based graphics. 2007. p. 91–100.
[7] Yao Y, Sun J, Huang X, Bowman GR, Singh G, Lesnick M, et al. Topological methods for exploring low-density states in biomolecular folding pathways. J Chem Phys 2009;130:144115.
[8] Sarikonda G, Pettus J, Phatak S, Sachithanantham S, Miller JF, Wesley JD, et al. CD8 T-cell reactivity to islet antigens is unique to type 1 while CD4 T-cell reactivity exists in both type 1 and type 2 diabetes. J Autoimmun 2014;50:77–82.
[9] Nicolau M, Levine AJ, Carlsson G. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. Proc Natl Acad Sci 2011;108:7265–70.
[10] Guo W, Banerjee AG. Toward automated prediction of manufacturing productivity based on feature selection using topological data analysis. In: Proceedings of IEEE international symposium on assembly and manufacturing. 2016. p. 31–6.

[11] He QP, Wang J. Fault detection using the *k*-nearest neighbor rule for semiconductor manufacturing processes. IEEE Trans Semicond Manuf 2007;20:345–54.

[12] He QP, Wang J. Large-scale semiconductor process fault detection using a fast pattern recognition-based method. IEEE Trans Semicond Manuf 2010;23:194–200.

[13] Li Y, Zhang X. Diffusion maps based *k*-nearest-neighbor rule technique for semiconductor manufacturing process fault detection. Chemom Intell Lab Syst 2014;136:47–57.

[14] Zhou Z, Wen C, Yang C. Fault detection using random projections and *k*-nearest neighbor rule for semiconductor manufacturing processes. IEEE Trans Semicond Manuf 2015;28:70–9.

[15] Famili F, Shen W-M, Weber R, Simoudis E. Data pre-processing and intelligent data analysis. Int J Intell Data Anal 1997;1.

[16] Su C-T, Yang T, Ke C-M. A neural-network approach for semiconductor wafer post-sawing inspection. IEEE Trans Semicond Manuf 2002;15:260–6.

[17] Lum P, Singh G, Lehman A, Ishkanov T, Vejdemo-Johansson M, Alagappan M, et al. Extracting insights from the shape of complex data using topology. Sci Rep 2013;3.

[18] Milnor JW. Morse theory, 51. Princeton University Press; 1963.

[19] Maaten LVD, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9:2579–605.

[20] Kuhn M, Johnson K. Applied predictive modeling. Springer; 2013.

[21] Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd international conference on knowledge discovery and data mining, vol. 96. 1996. p. 226–31.

[22] Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B 1995;57:289–300.

[23] Yekutieli D, Benjamini Y. Discovering the false discovery rate. J Stat Plan Inference 1999;82:171–96.

[24] Wise BM, Gallagher NB, Butler SW, White Jr DD, Barna GG. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. J Chemom 1999;13:379–96.

[25] Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2011;2:27.