Journal of Intelligent Manufacturing (2021) 32:823–836
https://doi.org/10.1007/s10845-020-01591-0

Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing

Chia-Yu Hsu[1] · Wei-Chen Liu[2]

Received: 22 January 2020 / Accepted: 14 May 2020 / Published online: 18 May 2020

Abstract

The development of information technology and process technology have been enhanced the rapid changes in high-tech products and smart manufacturing, specifications become more sophisticated. Large amount of sensors are installed to record equipment condition during the manufacturing process. In particular, the characteristics of sensor data are temporal. Most the existing approaches for time series classification are not applicable to adaptively extract the effective feature from a large number of sensor data, accurately detect the fault, and provide the assignable cause for fault diagnosis. This study aims to propose a multiple time-series convolutional neural network (MTS-CNN) model for fault detection and diagnosis in semiconductor manufacturing. This study incorporates data augmentation with sliding window to generate amounts of subsequences and thus to enhance the diversity and avoid over-fitting. The key features of equipment sensor can be learned automatically through stacked convolution-pooling layers. The importance of each sensor is also identified through the diagnostic layer in the proposed MTS-CNN. An empirical study from a wafer fabrication was conducted to validate the proposed MTS-CNN and compare the performance among the other multivariate time series classification methods. The experimental results demonstrate that the MTS-CNN can accurately detect the fault wafers with high accuracy, recall and precision, and outperforms than other existing multivariate time series classification methods. Through the output value of the diagnostic layer in MTS-CNN, we can identify the relationship between each fault and different sensors and provider valuable information to associate the excursion for fault diagnosis.

Keywords Fault detection and diagnosis · Time series classification · Deep learning · Convolutional neural network · Smart manufacturing

Introduction

Early fault detection and quick diagnosis of faulty wafer are important to ensure controlling process operations and reduce yield losses in semiconductor manufacturing (Hsu et al. 2020). Advanced in sensing and information technology have enabled the automatic collection and recording of the massive data generated by the production and testing equipment during complicated semiconductor manufacturing processes. In a wafer manufacturing factory, changes in the equipment status (e.g., temperature, humidity, pressure, flow rate, and chemical gas flow) during the manufacturing process may adversely affect the process. Equipment sensor data are real-time recordings in chronological order which are primarily recordings of the signals of machinery condition. Effective equipment monitoring is crucial to cost reduction and yield improvement by avoid unexpected downtime (Dalpiaz and Rivola 1997; Han and Song 2003).

With the growth of Industry 4.0 and intelligent manufacturing, amounts of sensors have been installed in equipment and machine and these sensors are used to automatically accumulate various time-series information (Oztemel and Gursev 2020). To improve production yield, automated production has begun to evolve into intelligent production, in which machinery and equipment sensor data are analyzed to obtain the important data recorded by each sensor and to determine the root cause of wafer faults to help the engineers adjust the machinery parameters. This multivariate time

✉ Chia-Yu Hsu
chiayuhsu@mail.ntut.edu.tw

[1] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan

[2] Department of Information Management, Yuan Ze University, Taoyuan, Taiwan

series data in semiconductor manufacturing are also called Fault detection and classification (FDC) data in semiconductor manufacturing which are consisted of three dimensional information including wafer, status variable identification (SVID), recorded time. The SVID represents the status of equipment or machine such as temperature, pressure, and gas flow. Through analyzing FDC data analysis can obtain behind important information to detect faults and determine the relations between the sensors and faulty wafers (Chien et al. 2013). Fault detection involved multivariate statistical fault detection methods (Cherry and Qin 2006; Yu 2011; Chien et al. 2013; Rostami et al. 2018) and machine learning models (He and Wang 2007; Mahadevan and Shah 2009; Fan et al. 2016; Wang et al. 2018) were used to analyze equipment sensor data and detect the abnormality at early stage of wafer processing. In addition, fault diagnosis is to identify excursion and to provide critical information for analyzing root cause of abnormality (Gertler 1998; Chien et al. 2013; Lee et al. 2017b). In practice, the diagnosis of excursion is mainly relied on experience of domain expert to perform investigation on the sensor data of abnormal sensor. It's time-consuming and subjective to remove assignable causes.

Because of the massive volume and nonstationary patterns, most of fault detection and diagnosis studies in semiconductor manufacturing extract the features from each SVID by calculating the summery statistics at each time step such as mean, standard deviation, maximum, minimum, skewness, and kurtosis (Park et al. 2014; Rato et al. 2017), which is defined based on the engineers' past experience, and determine the upper and lower limit specifications for analysis. However, considering the variety of products and the difficulty in stipulating specifications, handcraft features could lose the important information and result in a high false alarm rate (Lee et al. 2017b). There are a large number of measurement variables associated with equipment, machinery and complex product combinations. Thousands of variables are recorded that are highly correlated with one another. As a result, equipment sensory analysis relies on the determination of key steps. Constructing the indicators of these steps has become complicated and difficult. While the statistics of these sensor data indirectly provide information on product quality and machinery status, the practical difficulty is to acquire large amounts of engineers' knowledge to facilitate the determination of meaningful steps. Even with the determined steps, false detection can also occur as a result of delay. The use of conventional FDC methods alone can no longer effectively locate important features in sensor data and can easily lead to false detection or omissions.

With the development of parallel computing and graphic processing unit (GPU) technique, deep learning has gradually become an effective technique for tasks requiring heavy computation and automatically feature extraction from raw time series data. In most research related to equipment sensory data, deep learning methods based on convolutional neural network (CNN) and autoencoder are primarily used for important feature extraction and fault detection (Zheng et al. 2014; Yang et al. 2015; Lee et al. 2017a, b). To compare with principal component analysis (PCA), support vector machine (SVM), k-nearest neighbor (kNN), decision tree-based methods, CNN-based models can perform better results of fault detection without handcraft features in advance. However, the existing deep-learning fault detection models are difficult to determine the correlation between faults and the collected equipment sensor variables. When a fault occurs, efforts should be made to find the most probable root cause, provide information to help prevent the occurrence of wafer faults in advance and make adjustments based on the data collected by the sensor that detected the fault to avoid unnecessary loss from machinery and equipment shutdown. Moreover, the architectures of existing fault detection model by CNN need fixed-length in SVID data (Kim et al. 2019). But, the recorded time period of each SVID could be different in semiconductor manufacturing.

To bridge the gap of real setting in practice, this paper aims to propose a multiple time-series convolution neural network (MTS-CNN) model for multivariate time-series classification, in which the fault detection and identification of the key SVID for root cause analysis are performed simultaneously. To maintain the sensor data patterns and avoid scale errors, the MTS-CNN model in the first stage performs standardization of the original sensor data into the same scale with zero mean and unit variance. To different wafers collected during the manufacturing process are varied in length, this study facilitates the subsequent determination of the temporal features by sliding window-based subsequence extraction. The argumentation of subsequences increase the total volume and the diversity of the data for training and improve the accuracy of the entire MTS-CNN model. The MTS-CNN in the second stage, extract the data features of each SVID using a convolutional neural network (CNN) along each subsequence in conjunction with a backpropagation algorithm as well as detects product faults. Faulty and normal wafers are then identified using the MTS-CNN model. In addition, the correlations between individual sensors and the faulty wafers are determined based on the diagnostic layer to provide reference information to the maintenance engineers. To validate the proposed MTS-CNN, we conducted an experiments by a real-world data collected from a semiconductor manufacturing process. The MTS-CNN has better classification performance among other multivariate time series classification models.

The contribution of this paper is mainly in three aspects. First, slide window is used to split original time-series into various subsequence for dealing with variable-length sensor data. Second, effective feature are extracted for fault detection via multivariate time series classification through the stacked convolution and pooling layers. Third, the related SVIDs to faulty wafer can be identified based on the output of diagnostic layer after the deep convolutional-pooling layers. While detect the abnormality of observed wafers, the MTS-CNN can identify the correlated SVIDs and provides useful information for further root cause diagnosis.

The remainder of this paper is organized as follows. Section 2 introduces the fundamentals of time series data and multivariate time series data classification approaches to be used in this paper. Section 3 presents the proposed MTS-CNN model for fault detection and diagnosis. In Sect. 4, an empirical from a semiconductor company was conducted to validate our proposed MTS-CNN and compare the performance among the other multivariate time series classification approaches. Lastly, Sect. 5 concludes the paper by listing its contributions and discussing future research directions.

Fundamental

Time series data

There are two primary types of time series analysis. The first type directly calculates the degrees of similarity between different series and labels the classified series based on the order of degrees of similarity using a k-nearest neighbor (k-NN) algorithm. The second type extracts the key features of a time series, finds the important features by analyzing the subsequences of the time series using an algorithm, and then inputs the key features to construct a classification model.

A time series of FDC data can be defined as follows. Assuming that there are M wafers being processed through an equipment with K sensors being installed. Thus, the FDC data $\mathbf{X}_p$ for pth wafer are defined as follows:

$$\mathbf{X}_p = \begin{bmatrix} x_{p,1,1} & x_{p,1,2} & \cdots & x_{p,1,n_p} \\ x_{p,2,1} & x_{p,2,2} & \cdots & x_{p,2,n_p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p,K,1} & x_{p,K,2} & \cdots & x_{p,K,n_p} \end{bmatrix} \tag{1}$$

where x_{pqt} represents the sensor reading value for pth wafer of the qth sensor at time t, for $p = 1, \ldots, M$, $q = 1, \ldots, K$, and $t = 1, \ldots, n_p$. Here, n_p denotes the total number of recorded time points for pth wafer and is usually not equal between wafers.

The output $\mathbf{Y}_p$ is one hot label vector $\mathbf{Y}_p \in [0,\ 1]$, in which the $\mathbf{Y}_p =\ 1$ if the class of time series $\mathbf{X}_p$ is belong to fault wafer. Otherwise, the $\mathbf{Y}_p =\ 0$. Therefore, the time series classification problem is to search a function that can map the input $\mathbf{X}_p$ to a possible $\mathbf{Y}_p$ as well.

The degree of similarity between two time series is primarily measured using the Euclidean distance or the dynamic time warping (DTW) distance as the distance between them (Fig. 1). A 1-nearest neighbor (1-NN) algorithm combined with the Euclidean distance (1-NN-EUC) can be employed to classify the time series (Faloutsos et al. 1994). While the Euclidean distance is simple and fast to calculate, the use of the Euclidean distance to calculate the degree of similarity between two time series requires the assumption that the two time series have the same length. The DTW distance can be used to calculate the degree of similarity between time series of different lengths. Compared to 1-NN-EUC algorithms, 1-NN algorithms combined with the DTW distance (1-NN-DTW) are more widely used in time series analyses (Sakoe and Chiba 1978; Xi et al. 2006). While a k-NN algorithm can directly classify time series without training, the calculation of the DTW distance is relatively time consuming, rendering the practical application of 1-NN-DTW algorithms limited.

Machine learning is often used to extract features of time series. Nanopoulos et al. (2001) extracted and classified features of time series using a multilayer perceptron (MLP), which is a type of artificial neural network (ANN). Kampouraki et al. (2009) used a support vector machine (SVM) to classify time-series data and found that the SVM performed better in classifying time-series datasets compared to an ANN. To determine the most important time step in an original time series, the time series is first decomposed into a number of subsequences, and the most important subsequences are located and used as the features of the time series. This approach not only maintains the representation of the original time series but also can be used to analyze time series of various lengths.

Ye and Keogh (2009) first obtained all the subsequences of a time series, calculated the distances between the subsequences of each length and other time series, and then

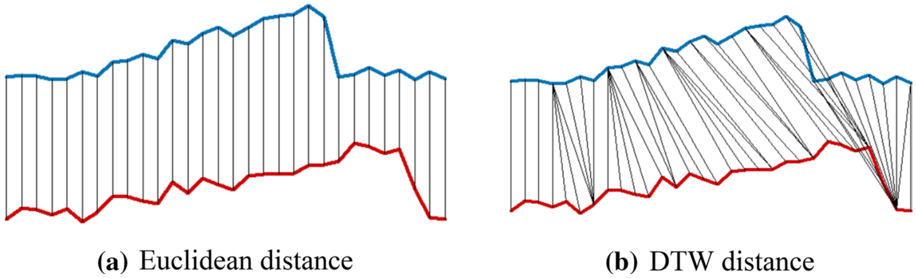


Fig. 1 Degree of similarity between time series

(a) Euclidean distance **(b)** DTW distance calculated the information gain of each subsequence. Time series subsequences that are maximally representative of a type are referred to as shapelets as shown in Fig. 2. Shapelet discovery requires the calculation of all the subsequences of each time series, which is often extremely time consuming. Rakthanmanon and Keogh (2013) proposed a symbolic aggregate approximation (SAX)–random masking-combined approach, which can rapidly discover the most important subsequences, then calculate the information gains, and classify the time series.

Senin and Malinchik (2013) proposed a SAX–vector space model (SAX–VSM) algorithm for time series classification. First, the time series are standardized. Then, all the subsequences of the time series are subjected to piecewise aggregate approximation and subsequently transformed to SAX symbols using the SAX algorithm (Lin et al. 2003). Afterwards, a VSM is constructed. First, the SAX symbols that represent all the subsequences of the same type are converted to a bag-of-words model based on the frequency of occurrence. The term frequency–inverse document frequency (TF–IDF) weight is calculated based on the bag-of-words model constructed for each type (Salton et al. 1975). When predicting the series types, the established TF–IDF is used to calculate the degree of cosine similarity. The highest degree of cosine similarity is used as the classification type. Figure 3 shows the framework of the SAX–VSM algorithm.

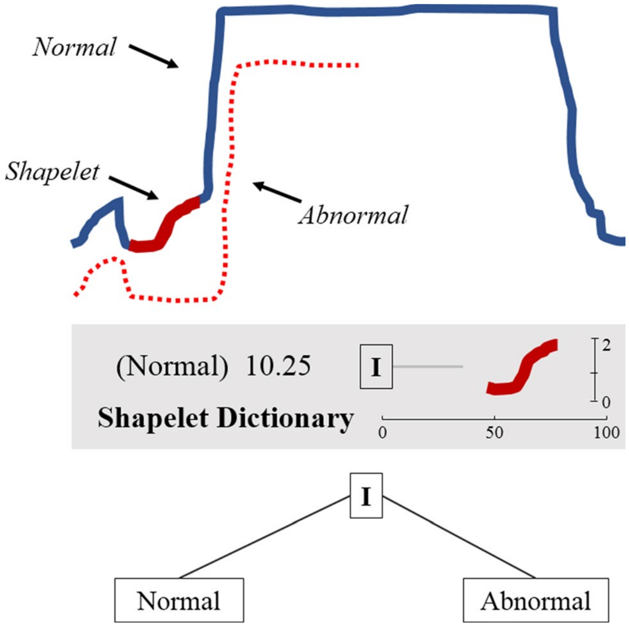


Fig. 2 Schematic diagram of a shapelet

Multivariate time series classification

Patri et al. (2014) designed a shapelet forests approach based on shapelet extraction that can effectively solve multivariate time series classification problems. This approach first extracts shapelets from the data collected by different sensors and then, based on these shapelets, calculates the shortest distances from the subsequences of the time-series data collected by the same sensors. A new feature consisting of these distances is then formed and used to represent the relation between the final product and the shapelets. Afterward, weight learning and classification can be performed using a general machine learning algorithm or a feature selection method. The advantage of this approach lies in that it can compare the contribution of the sensors to fault detection based on their weights.

DCNNs are also widely used in multivariate time series analysis. Zheng et al. (2014) proposed a multi-channel DCNN (MC-DCNN) to solve problems involving multiple sensors. The MC-DCNN addresses the problem that different sensor data have different lengths. Initially, the CNN uses one channel to learn the features of the subsequences of the data collected by each sensor. Consequently, the entire CNN model will learn the features of the different sensors. Then, the features extracted by the CNN are combined and used, in conjunction with a conventional MLP, to classify the time series. Yang et al. (2015) used MC-CNN to identify human movements. They used the MC-CNN to first learn the features from two real datasets and then to classify actual human movements. Huang et al. (2020) adopted multi-domain features in time-domain and fused these features into DCNN model for tool wear prediction in dry milling operations. Similarly, various features in frequency and time–frequency domains were used as input for tool condition monitoring (Cai et al. 2020; Lee et al. 2020).

To effectively analyze multivariate time-series data collected by multiple sensors, the general approach is to first extract features from the data collected by each sensor using a time series analysis method, then find the features that subsequently construct a time series classification model. When a fault occurs in the machinery or equipment, it is necessary to understand the importance of the sensors to correct the parameters of the machinery to improve production yield. While an MC-CNN approach has excellent classification performance, unlike the shapelet forests approach, it is incapable of classifying time series data while simultaneously determining how the sensors affect one another and the contribution of the sensors to the source of the fault. Lee et al. (2017b) propose a FDC-CNN for fault detection, in which a receptive field tailored to multivariate sensor signals slides along the time axis, to extract fault features. The receptive field and the feature map in the first convolutional layer was used

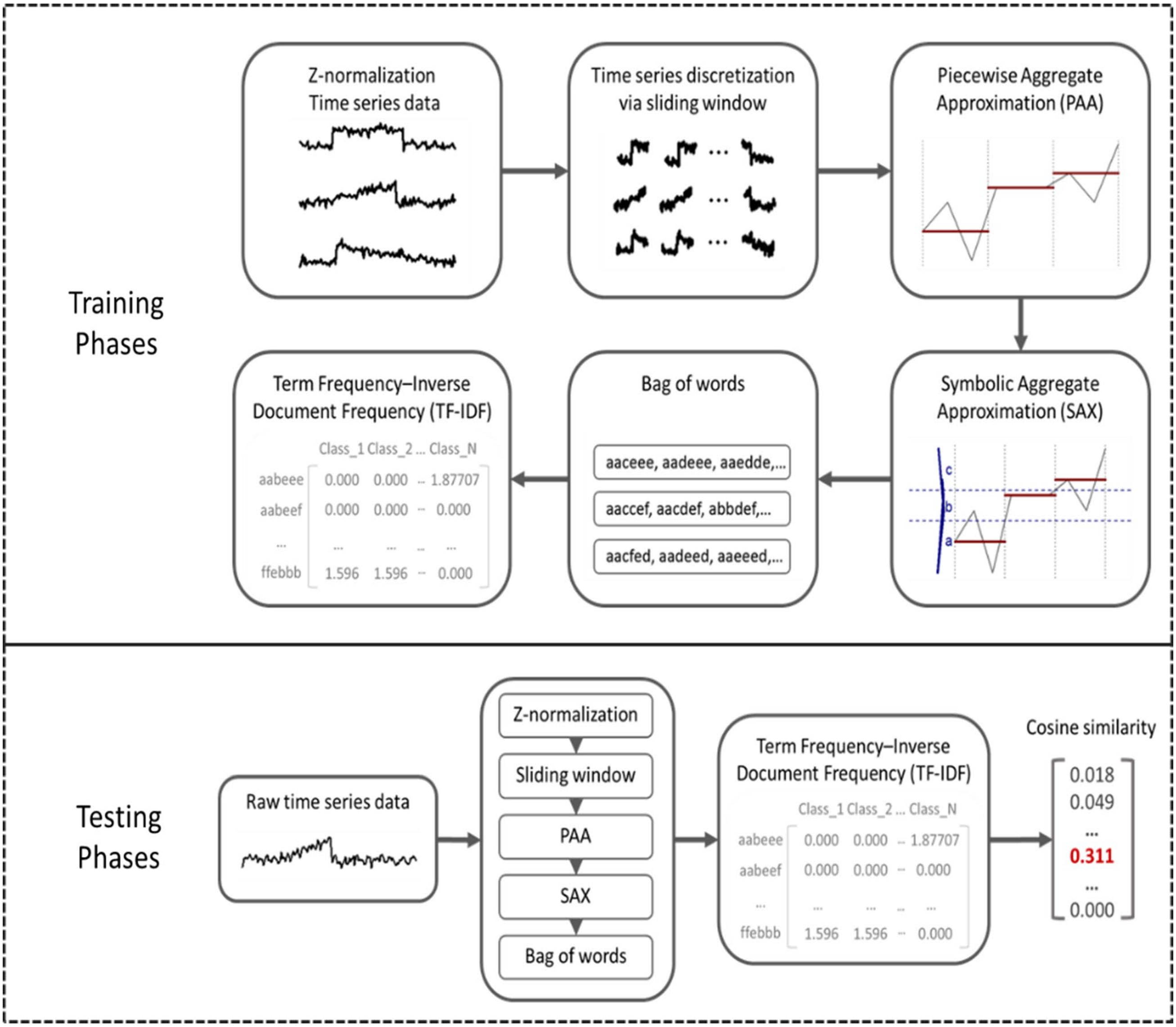

Training Phases
Z-normalization Time series data
Time series discretization via sliding window
Piecewise Aggregate Approximation (PAA)
Symbolic Aggregate Approximation (SAX)
c
b
a
Bag of words
aaceee, aadeee, aaedde,...
aaccef, aacdef, abbdef,...
aacfed, aadeed, aaeeed,...
Term Frequency–Inverse Document Frequency (TF-IDF)
Class_1 Class_2 ... Class_N
aabeee 0.000 0.000 ... 1.87707
aabeef 0.000 0.000 ... 0.000
ffebbb 1.596 1.596 ... 0.000
Testing Phases
Raw time series data
Z-normalization
Sliding window
PAA
SAX
Bag of words
Term Frequency–Inverse Document Frequency (TF-IDF)
Class_1 Class_2 ... Class_N
aabeee 0.000 0.000 ... 1.87707
aabeef 0.000 0.000 ... 0.000
ffebbb 1.596 1.596 ... 0.000
Cosine similarity
0.018
0.049
0.311
0.000

Fig. 3 Structure of SAX–VSM

to find the information for fault diagnosis. The features maps extracted in the deep convolutional layers are the results of highly nonlinear functions of the raw data and thus are difficult to interpret. Therefore, only the feature maps from the first convolutional layer are considered for fault diagnosis (Lee et al. 2017b). However, the information and feature map in the first convolutional layer was simple because of passing one convolution and feature extraction process which is not enough to generate critical feature for fault detection. To bridge the gap, this study proposes an MTS-CNN for fault detection and diagnosis in semiconductor manufacturing. An additional diagnostic layer is added between the last pooling layer and fully connected layer. The diagnostic layer is used to understand the relation among the collected sensor and their importance for the faulty wafer.

Proposed Method

The MTS-CNN model for fault detection and diagnosis is consisted of feature extraction and fault detection and diagnosis as shown in Fig. 4. A multi-channel CNN is used whereby each channel is determined by one SVID and the important features from the data collected by one sensor are extracted through two stacked structures with convolution, activation, and pooling layers. Additionally, the relationships between each SVD and faulty wafer is identified by

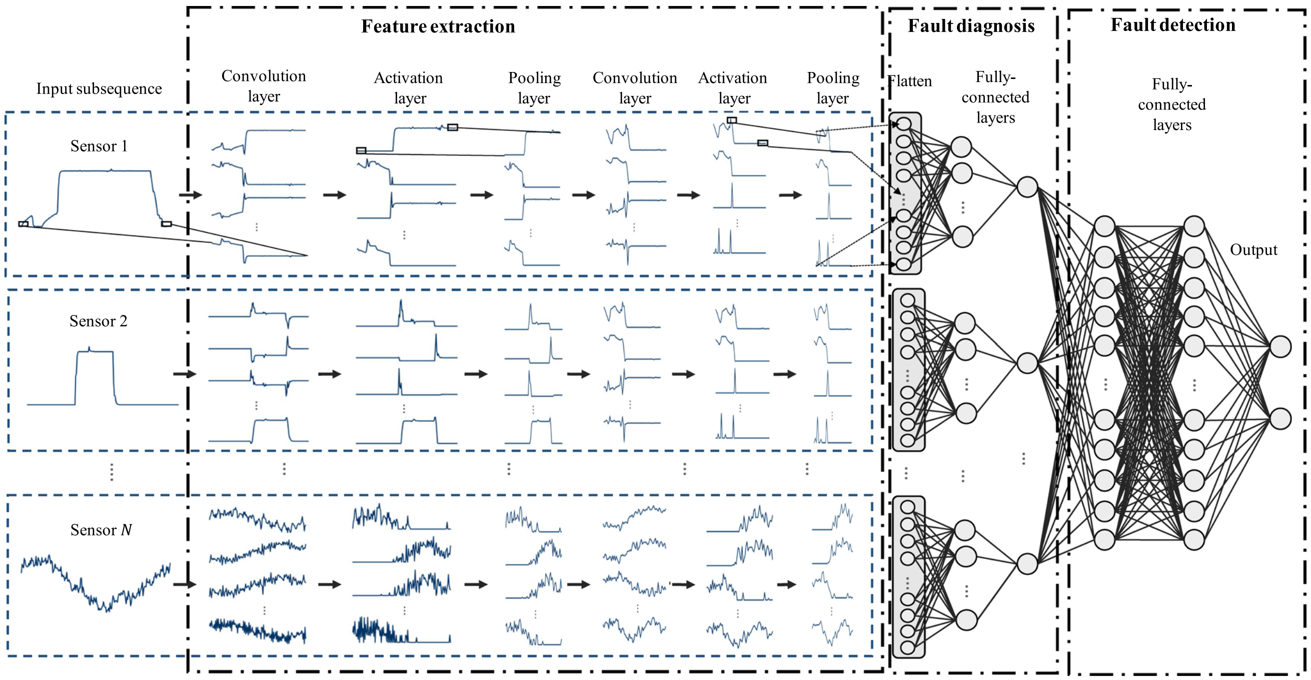


Fig. 4 The proposed MTS-CNN model

a diagnostic layer after the process of feature extraction by each channel. By observing the diagnostic layer, the status of each sensor can be capable of distinguishing faulty wafers from normal ones. In the final layer of MTS-CNN, fault detection is performed based on the extracted features in conjunction with a fully connected neural network.

Data preprocessing

The measurements of different equipment variables (e.g., temperature, pressure and humidity) during the manufacturing process are on different numerical scales. To avoid scale errors, the original equipment sensor data are standardized. The standardization process transforms the equipment sensor data to a distribution with an average value of 0 and a standard deviation of 1 (Goldin and Kanellakis 1995).

The subsequences of the entire time series are used as the input data. Two methods are employed to perform data augmentation to avoid overfitting during training the model. First, the dropout technique is used when the convolutional network extracts features (Bouthillier et al. 2015). Second, the sensor data are decomposed into various subsequences using a sliding window, in which are then used as the input data whose features are subsequently extracted by the CNN (Le Guennec et al. 2016).

Assuming the data $\mathbf{X}$ is a three-dimensional data matrix including M wafers, K sensors, and recorded time n_p. The $\mathbf{X}_q$ is the qth sensor data as follows:

$$\mathbf{X}_q = \begin{bmatrix} x_{1,q,1} & x_{1,q,2} & \cdots & x_{1,q,n_p} \\ x_{2,q,1} & x_{2,q,2} & \cdots & x_{2,q,n_p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M,q,1} & x_{M,q,2} & \cdots & x_{M,q,n_p} \end{bmatrix} \tag{2}$$

where n_p is the sensor data length. When a sliding window w is used to acquire the subsequences of the qth sensor data $\mathbf{X}_q$ as shown in Fig. 5, a $n_p - w + 1$ number of subsequences $s_i \; (i = 1, \ldots, n_p - w + 1)$ are selected from sensor data $\mathbf{X}_q$ of length n_p using a sliding window of length w.

$$sliding\ (\mathbf{X}_q, w) = S, \quad S \in \{s_1, s_2, \ldots, s_{n_p - w+1}\} \tag{3}$$

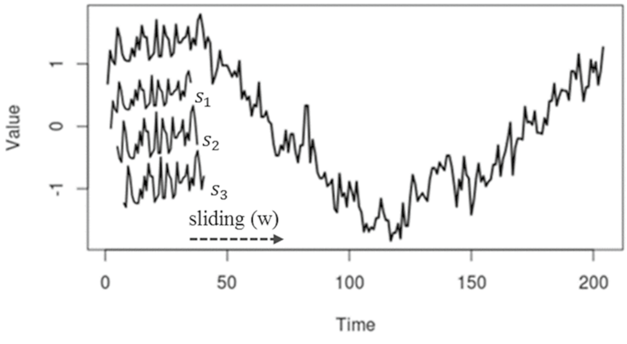


Fig. 5 Sliding window-based subsequence extraction

Feature extraction

A convolution layer and an activation layer are used to extract features from the input time-series data. Subsequently, a pooling layer is used to reduce the dimension to find the most important information recorded by the sensor, which will be used as the basis for fault detection and diagnosis. First, in the convolution layer, a number of trained convolution kernels k_{ij} make convolution with the original input data and then obtain the feature maps x_i^{l-1} in the previous layer and add a bias b_j^l to obtain the convoluted feature maps z_j^l. The convolution operation slides the convolution kernels on the data and calculates the new feature obtained by the inner product. If sensor data of length n_p are input into the model and the convolution kernels have a length of w, sensor data of length $n_p - w + 1$ will be obtained after convolution.

$$z_j^l = \sum_i x_i^{l-1} * k_{ij}^l + b_j^l \quad (4)$$

where the notation of $*$ is a convolution operator. To compare with the existing CNN for fault detection (e.g., FDC-CNN), we extract the critical feature of each SVID along time horizon instead of performing convolution among the all SVIDs together. The shortcoming of extract feature from all multivariate time-series is difficult to trace the importance of key SVID after deep convolutions. The nonlinear correlation of each SVID to detect abnormality needs more stacked convolutions as well.

An activation function f is used in the activation layer, and x_j^l is obtained by calculating the activation function f with the input data z_j^l of the previous layer. To avoid the vanishing gradient problem that can easily occur when performing backpropagation in a deep neural network model, a rectified linear unit (*ReLU*) activation function is used in the hidden layer. The distinct features of a *ReLU* are can cause a neuron's partial output to be zero, and can further reduce overfitting.

$$x_j^l = f(z_j^l) \quad (5)$$

$$\mathrm{Re}LU(x) = \max(0, x) \quad (6)$$

Pooling is also referred to as subsampling. There are two primary types of pooling as shown in Fig. 6, i.e., max and average pooling. Max pooling returns the maximum mapping value for each feature. Average pooling returns the average mapping value for each feature. Each feature that passes through the convolution and activation layers is pooled to become a new feature. A max pooling layer is used in the proposed framework, where x_j^l is the input data; x_j^{l+1} is the pooling output value; and $down(\cdot)$ represents the subsampling value generated by max pooling). By using a number of non-overlapped $1 \times n$ kernels and calculating the maximum or average values with the kernels, the dimension of the output sensor data is reduced by n number of folds. For example, there are a time series with 8 data (4, 8, 16, 10, 2, 0, 1, 9). Assuming that a 1×2 kernal with stride size is 2 as shown in Fig. 6, then the original time series is used to perform max pooling and result in (8, 16, 2, 9). If we select average pooling and the time series is transformed into (6, 13, 1, 5).

$$x_j^{l+1} = down(x_j^l) \quad (7)$$

Fault detection and diagnosis

Fault detection and diagnosis layers are constructed after two stacked convolution-pooling layers. There is a one

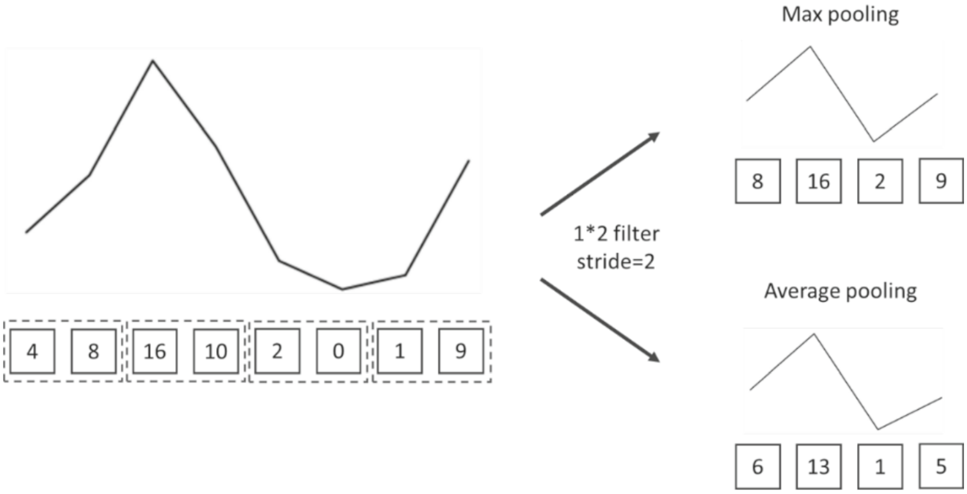


Fig. 6 Max pooling and average pooling

fully connected neural network, in which one flatten layer is assigned after the second pooling layer, one hidden layer, and one neuron node. The diagnostic layer can determine the difference between the sensors and the faulty wafers. The ReLU activation function can also be used to set a threshold value between faulty and normal products for each sensor to represent the important information of the entire data set.

In the diagnostic layer, a set of optimal weights will be obtained after the training of the MTS-CNN model is completed. The weights represent the relations between the sensors. These weights are used in conjunction with a ReLU activation layer. Based on the features of a ReLU, the sensors with a negative weight are found to have a zero output. These sensors with a zero output are relatively unimportant. The remaining sensors with a non-zero output will affect the final prediction. By further observing the output values of normal and faulty wafers, a threshold value can be set to determine which sensor detects the highest number of faulty products.

After the diagnostic layer, two hidden layers with fully connected networks are used to predict output status (i.e., faults or normal). In the output layer, the softmax function is used to perform classification prediction as follows:

$$f(z)_r = \frac{e^{z_r}}{\sum_{r=1}^{C} e^{z_r}}, \quad r = 1, \ldots, C \tag{8}$$

where C is the number of considered class in the output layer.

Training a neural network requires the use of forward propagation and backpropagation. An output value will be generated by forward propagation. By calculating fault detection is a classification problem and using a square error function as a loss function to adjust the weight in each neuron may be overly smooth, and result in the poor convergence of the network, a cross-entropy function (*CE*) is used as the loss function as follow (Bishop 1995; Golik et al. 2013):

$$CE = -\sum_{p=1}^{M} CE_p \tag{9}$$

$$CE_p = -\sum_{r=1}^{C} L_r \log_2(f(z)_r) \tag{10}$$

where CE_p is the cross-entropy value of pth wafer and L_r is the binary value. Indeed, only the softmax value of true label will be calculate and others are 0. Therefore, the cross-entropy of total sample is

The weight of the entire MTS-CNN model can be adjusted based on the loss function *CE*, using a backpropagation algorithm and a stochastic gradient descent (SGD) algorithm as an optimizer to adjust the weight of the entire MTS-CNN model, until the error reaches the minimum value and is convergent. Several model training techniques can be applied when using backpropagation and SGD algorithms to train the neural network to increase the convergence speed of the neural network and find the optimal parameters, e.g., standardizing the input data and randomizing the sequence of the training data (LeCun et al. 2012). To consider one convolution-pooling networks are constructed to extract features from the data collected by each sensor. The full-batch learning method would be difficult to find a good combination of weights in the proposed MTS-CNN model because of the large amount of equipment sensor data, resulting in overly long training times and heavy computer memory requirements. Therefore, the mini-batch method is used to train the weights of the MTS-CNN model. The mini-batch method uses only one batch of data for each training epoch and makes weights update by finding the average after each epoch (Krizhevsky et al. 2012). This method is advantageous because it leads to a more efficient weight update of the network within the same training time.

The entire CNN framework contains a large number of parameters. To avoid overfitting, the dropout technique is used in the diagnostic and FDC layers when training the network. The dropout technique randomly allows neurons in the hidden layer to disappear at a certain probability p, set by the user when correcting the weight during each training epoch. As a result, the weight may not be updated for all the neurons during the weight update process, thereby preventing the occurrence of overfitting. The dropout technique is not used on the test data. All the neurons with their weights multiplied by $(1 - p)$ are used in prediction (Srivastava et al. 2014).

Empirical study

To validate the proposed MTS-CNN model, an empirical data from the chemical vapor deposition (CVD) process in semiconductor fabrication was used. The CVD process is used to develop thin films on ICs. Total 189 wafers including 148 normal wafers and 41 abnormal wafers were used for empirical analysis and model evaluation. In particular, the equipment abnormality results in the peeling issue on wafer and yield loss. Total 21 sensors were originally installed on the CVD equipment. We remove 4 SVIDs because of the constant value and the remaining 17 SVIDs were used further analysis. The recorded length of each SVID are from 199 to 205.

Before MTS-CNN model construction, all the measured data from each sensor were performed standardization.

Then, the original time-sequence data were selected by a moving window. For example, the length of a SVID was 204 and window size was 149, then total 56 subsequence with 149 points were selected.

Fault detection

The processes of feature extraction were consisted of two convolution-activation-pooling layers. The kernel size and number of feature maps are two main hyperparameters in convolution layers. In order to extract the slight change in each subsequence, the kernel size is 5. The number of feature maps in first convolution layer is 16 and the number of feature maps in second convolution layer is 64. The activation function in activation layer is used ReLU. In order to represent the trend of each subsequence, the pooling method is used average-pooling. The kernel size in pooling layer is 2 and the parameter of stride is 2.

In order to capture the contribution of each sensor, a fully connected layer with 256 input nodes and 1 output node are used. The benefit of proposed diagnosis structure can represent the importance of each sensor while the prediction result is abnormal. The engineer can trace the potential correlated sensors. In the fault detection layer, two fully connected layers with 732 nodes were used. To reduce the effect of overfitting, a 0.5 dropout rate was used.

To determine the hyperparameter setting in the training of deep learning model, a preliminary experiment was used. The learning rate representing the rate of updating the connected weights between neurons is used as 0.01. The optimizer is used stochastic gradient descent and the value of momentum is 0.9. The batch size is 128, which refers the amount of training sample used for updating weights at one time. Each deep learning model was trained for each epoch, which represents the all training samples have been input into the training model. The maximum epoch for each deep learning model is 100.

To demonstrate the effectiveness of the proposed framework, the performance of the proposed MTS-CNN was compared with the performance of the five time-series analysis approaches, including 1NN-DTW (Xi et al. 2006), SAX–VSM (Senin and Malinchik 2013), Shapelet Forests (Patri et al. 2014), MC-DCNN (Zheng et al. 2014), and FDC-CNN (Lee et al. 2017b). 1NN-DTW calculates the similarity between two series. SAX–VSM algorithm uses SAX to extract the feature into bag-of-words and then build the vector space among these bag-of-word. Based on the cosine similarity, this algorithm can determine the similar class. Shapelet Forests is to select the time interval as shaplet which can separate these time series and then decision tree-based approach was used as classifier. Both MC-DCNN and FDC-CNN were deep learning-based approaches.

We perform fivefolds cross-validation to evaluate the performance of proposed model. The 20% of original data was used for testing data (39) and remaining data were used for model training (150). The data for model training can also separate into 80% training data and 20% validation data. For example, the data in first fold include 39 testing data (20%), 120 training data (150 × 80%) and 30 validation data (150 × 20%).

To consider the fault detection in CVD process is a binary classification problem, which is to predict the wafer could be abnormal or not. Therefore, precision, recall, and F_1 score were used to measure the performance of proposed MTS-CNN model. Precision denotes the proportion of classified abnormal wafer that are actual abnormal wafers. Recall denotes the proportion of actual abnormal wafers that are correctly detected. The high value of precision and recall indicates that the model is able to detect the abnormal wafers well. The related metrics were defined as follows:

$$\text{Precision } (p) = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall } (r) = \frac{TP}{TP + FN} \tag{12}$$

$$\mathrm{F}_1 = \frac{2}{1/p + 1/r} \tag{13}$$

where TP is the number of wafer with the predicted label and true label being 1. FP is the number of wafer whose predicted label is 1, but the true label is 0. FN is the number of wafer whose predicted label is 0, but the true label is 1.

After five-folds cross-validation, Table 1 summarized the precision, recall, F_1, and accuracy in 1NN-DTW,

Table 1 Performance comparisons in precision, recall, F_1, and accuracy

Fivefolds cross-validation	Precision	Recall	F_1	Accuracy (%)
1NN-DTW (Xi et al. 2006)	0.3076	0.2694	0.2872	70.88
SAX–VSM (Senin and Malinchik 2013)	0.9320	0.3944	0.5542	85.66
Shapelet forests (Patri et al. 2014)	0.9714	0.7861	0.8690	94.76
MC-DCNN (Zheng et al. 2014)	0.9278	0.9500	0.9388	96.86
FDC-CNN (Lee et al. 2017b)	0.9750	0.9500	0.9617	98.36
MTS-CNN	1.0000	0.9750	0.9873	99.48

SAX–VSM, Shapelet Forests, MC-DCNN, FDC-CNN, and MTS-CNN. In terms of accuracy, Shapelet Forests, MC-DCNN, FDC-CNN, and MTS-CNN are large than 0.9. 1NN-DTW is worst among the other approaches. Although the precision (0.9320) of SAX–VSM is better than 1NN-DTW, but the recall (0.3944) of SAX–VSM is still low. The precision (0.9714) and recall (0.7861) of Shapelet Forests, which is better than that of SAX–VSM. In terms of precision and recall, MC-DCNN, FDC-CNN, and the proposed MTS-CNN have high better solution than 1NN-DTW, SAX–VSM, Shapelet Forests. In particular, the proposed MTS-CNN outperform than other five approaches with high precision (1.0), recall (0.9750), and F_1 (0.9617).

Fault diagnosis

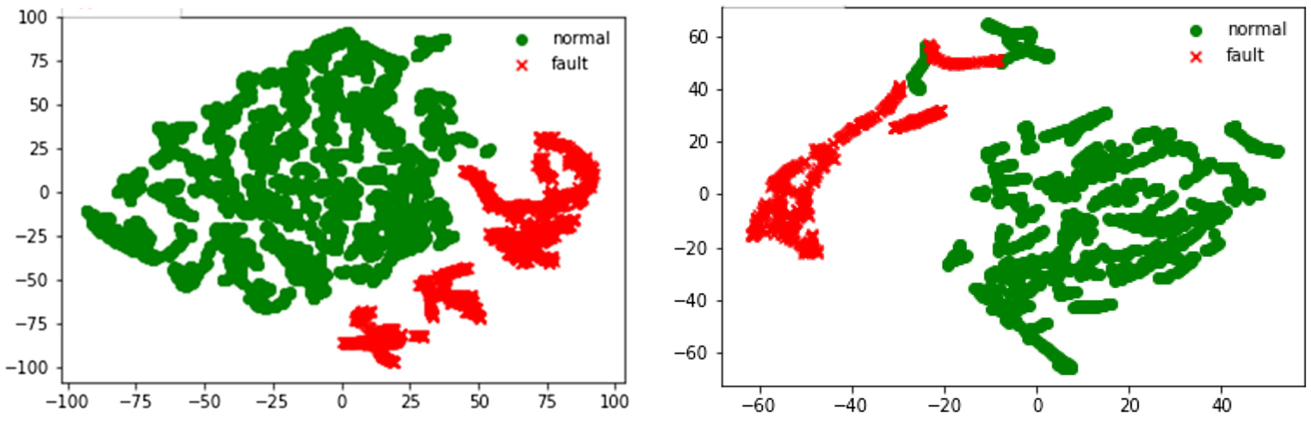
To examine the effect of the extracted feature of each subtime series by the proposed MTS-CNN, t-SNE (van der. Maaten and Hinton 2008) is used to perform a 2-D data visualization between normal and fault wafers. The input data of t-SNE were the output in diagnosis layer for 17 sensors in different time periods. The perplexity used in the t-SNE algorithm is set to 30, the exaggeration is set to 12, the learning rate is set to 200, and the number of iteration is set to 1000. Figure 7 illustrates the visualization of normal and faulty wafers generated by using t-SNE for training dataset and testing dataset in first cross-validation. The green points represent the normal wafers and the red points denotes faulty wafer. The great merit of data visualization is more straightforward for engineers to perform fault detection and identify different types of faults. Looking at the Fig. 7a, the normal wafers are clustered closely and the faulty wafers are clustered into several groups. Figure 7b shows that most of normal wafer and faulty wafers in terms of the 2-D t-SNE can be separated. Therefore, the 2-D t-SNE map via the output value of MTS-CNN could be potential used for process monitoring and identifying the abnormality real-time.

After detecting the abnormality, it should be identified the cause of the abnormal wafer among these monitoring sensors. This information are captured in the weight of diagnosis layer in the proposed MTS-CNN. The large output weight of each weight represent the large effect for the final classification. Figure 8 shows the testing result of the weight in 17 sensors of all normal and abnormal wafers. It finds that the sensor #7, #13, #15, and #16 have large deviation. This information can help engineer identifying critical sensor and remove assignable cause quickly.

Figure 9 shows weights of 17 sensors from normal and faulty wafers in fivefolds cross validation. We can find that the output value in sensors #7 and #15 have large difference between normal wafers and faulty wafers. To further check the original sensor value in sensors #7 and #15 as shown in Figs. 10 and 11, the faulty wafers have abnormal behaviors in radio frequency (RF) voltage during the CVD process and slightly high temperature in the end of CVD process. Therefore, the related causes of faulty wafers can be identified according to the MTS-CNN effectively.

Conclusion

This paper proposes a MTS-CNN to detect the abnormality of observed wafers and identify the correlated SVID with useful information for further root cause diagnosis. Data argumentation is used for variable-length time-series data, in which each sensor were selected by moving window to generate various subsequences with more diversity of sequential pattern. Through the short varied intervals which have difference between normal and abnormal can be extracted through convolution, pooling and learning processes, the



(a) in the training dataset

(b) in the testing dataset

Fig. 7 t-SNE of normal and faulty wafers

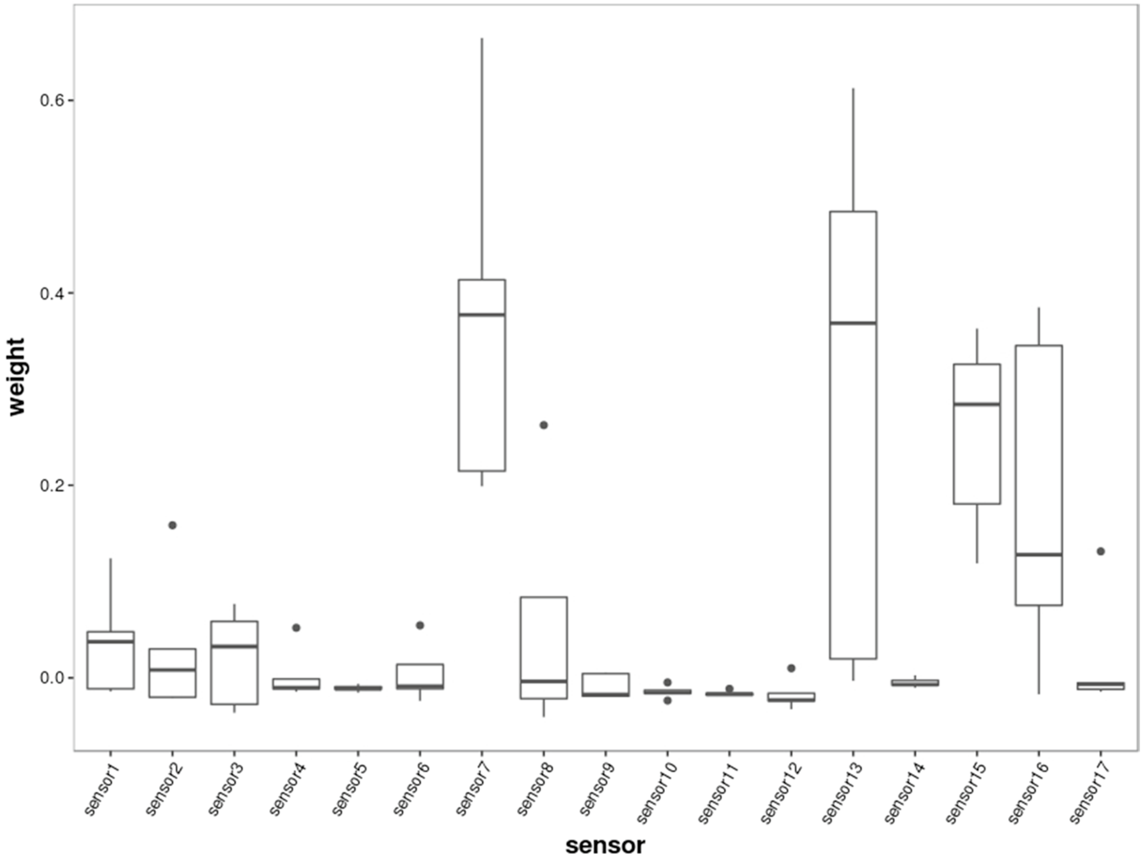


Fig. 8 Weights of 17 sensors from normal and faulty wafers in fivefolds cross validation

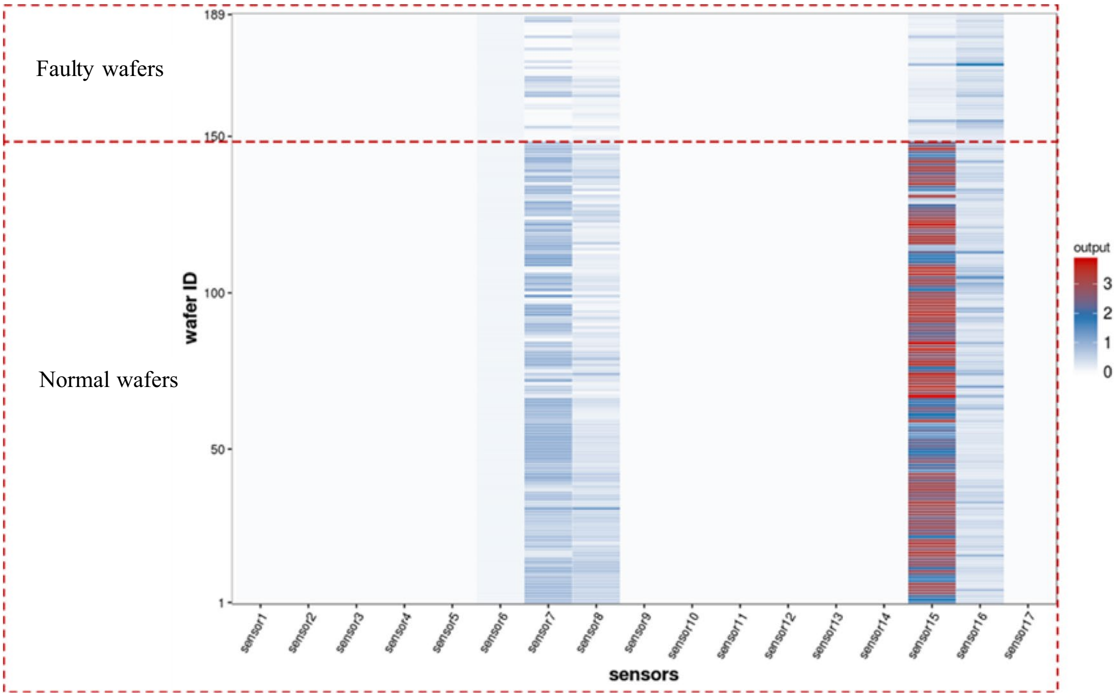


Fig. 9 Weights of 17 sensors from normal and faulty wafers in fivefolds cross validation

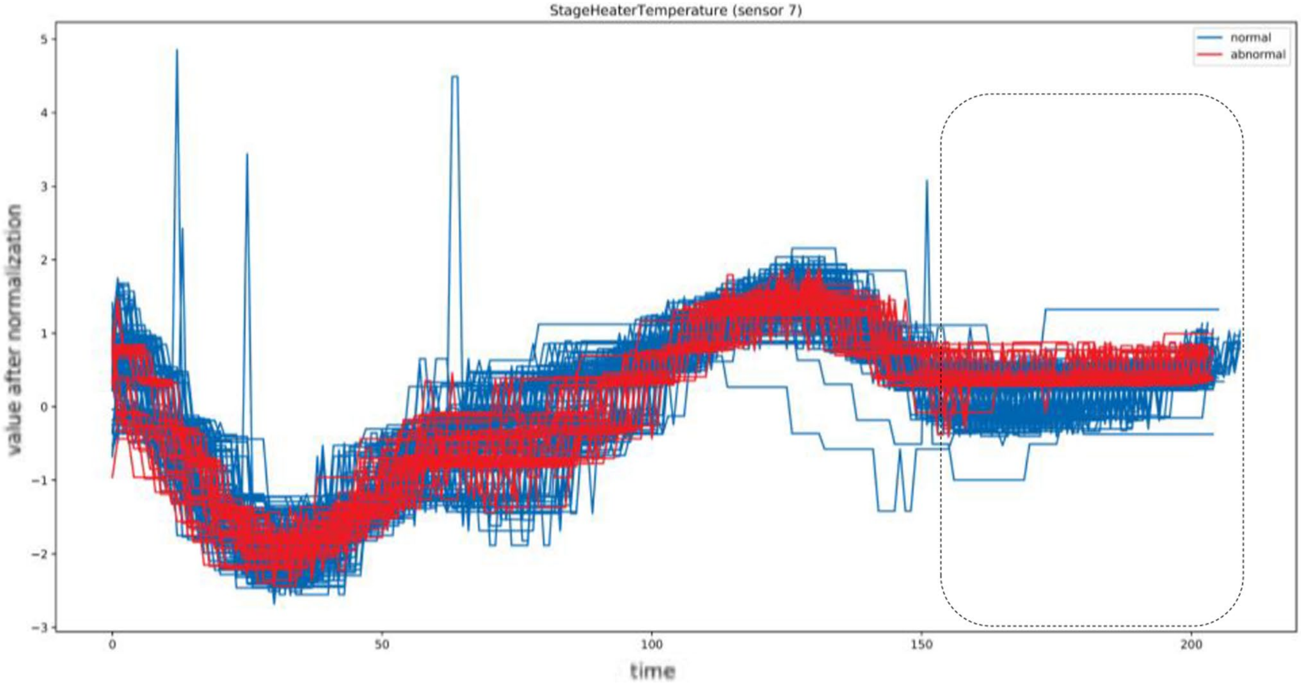


Fig. 10 Value of sensor #7 (StageHeaterTemperature)

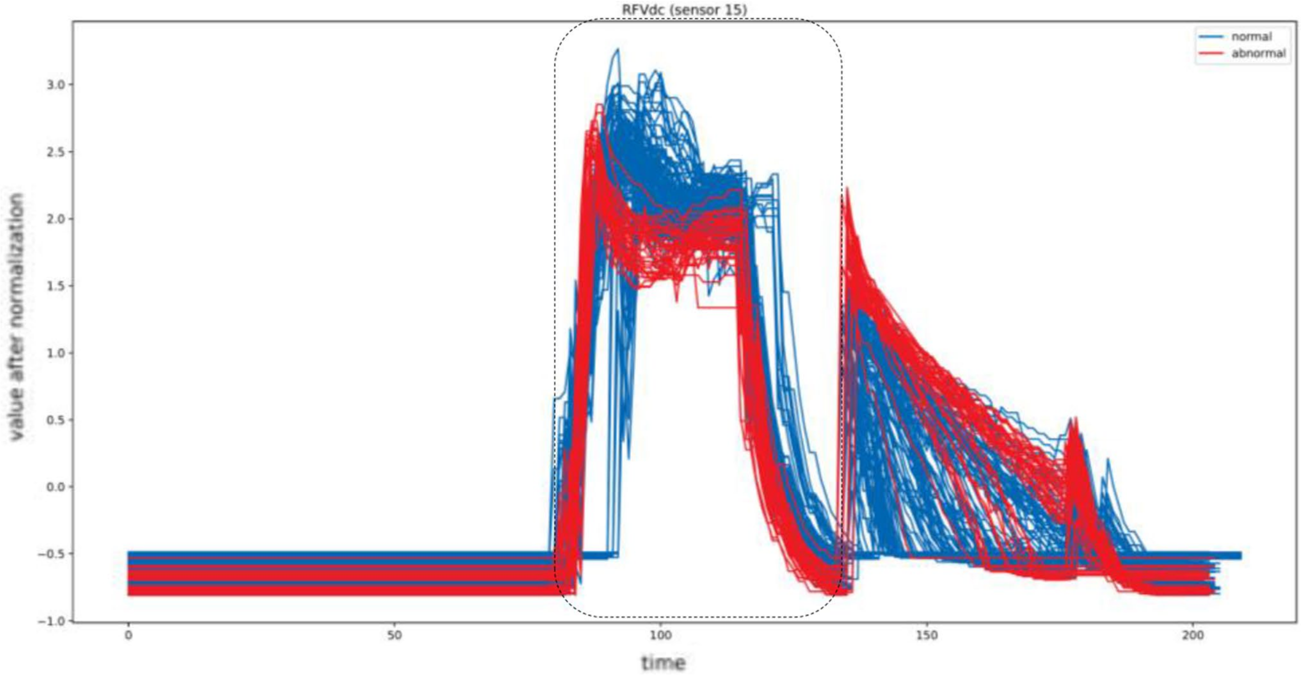


Fig. 11 Value of sensor #15 (RFVdc)

fault can be detected quickly without the full time-series of each sensor. In order to identify the contribution of each sensor, one diagnosis layer with ReLU function is added before fully connected layer. It can not only capture the useful feature to form the nonlinear function but also maintain the interpretation of the caused SVID while a faulty wafer is detected. Comparing with the existing FDC-CNN, the proposed MTS-CNN can use more stacked CNN layers to learn the highly nonlinear function and also provide traceable information for fault diagnosis.

The empirical results demonstrate that the proposed MTS-CNN not only has better prediction performance than 1NN-DTW, SAX–VSM, Shapelet Forests, MC-DCNN, FDC-CNN, but also provides the source of deviation for each abnormal wafer and assist engineers for fault diagnosis. The proposed MTS-CNN has good interpretation ability in differing normal and abnormal wafer among these collected sensors.

This study mainly considers the fault detection and diagnosis in semiconductor manufacturing. However, in high fully automation manufacturing, unexpected equipment breakdown results in throughput loss. In order to capture the failure or deviation as early as possible, the time-to-failure or remaining useful life of each equipment should be predicted. The further research can be addressed how to apply deep learning for the predictive maintenance and anomaly detection in smart manufacturing. Additionally, amounts of accurately labeled data are usually difficult to obtain in real industries. Data argumentation with synthetic noise could be considered to enhance the data distribution for model training (Li et al. 2020; Luo et al. 2020).

Acknowledgements This research was supported by the Ministry of Science and Technology, Taiwan (MOST 107-2221-E-027-127-MY2; MOST 108-2745-8-027-003).

References

Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.

Bouthillier, X., Konda, K., Vincent, P., & Memisevic, R. (2015). Dropout as data augmentation. arXiv preprint arXiv:1506.08700.

Cai, W., Zhang, W., Hu, X., & Liu, Y. (2020). A hybrid information model based on long short-term memory network for tool condition monitoring. *Journal of Intelligent Manufacturing*, 1–14. https://doi.org/10.1007/s10845-019-01526-4

Cherry, G. A., & Qin, S. J. (2006). Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *IEEE Transactions on Semiconductor Manufacturing, 19*(2), 159–172.

Chien, C.-F., Hsu, C.-Y., & Chen, P. (2013). Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence. *Flexible Services and Manufacturing Journal, 25,* 367–388.

Dalpiaz, G., & Rivola, A. (1997). Condition monitoring and diagnostics in automatic machines: Comparison of vibration analysis techniques. *Mechanical Systems and Signal Processing, 11*(1), 53–73.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 419–429).

Fan, S. K. S., Lin, S. C., & Tsai, P. F. (2016). Wafer fault detection and key step identification for semiconductor manufacturing using principal component analysis, AdaBoost and decision tree. *Journal of Industrial and Production Engineering, 33*(3), 151–168.

Gertler, J. (1998). *Fault detection and diagnosis in engineering systems*. Boca Raton: CRC Press.

Goldin, D. Q., & Kanellakis, P. C. (1995). On similarity queries for time-series data: Constraint specification and implementation. In *Proceedings of international conference on principles and practice of constraint programming* (pp. 137–153).

Golik, P., Doetsch, P., & Ney, H. (2013). Cross-entropy vs. squared error training: A theoretical and experimental comparison. *Interspeech, 13,* 1756–1760.

Han, Y., & Song, Y. H. (2003). Condition monitoring techniques for electrical equipment—A literature survey. *IEEE Transactions on Power Delivery, 18*(1), 4–13.

He, Q. P., & Wang, J. (2007). Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 20*(4), 345–354.

Hsu, C. Y., Chen, W. J., & Chien, J. C. (2020). Similarity matching of wafer bin maps for manufacturing intelligence to empower Industry 3.5 for semiconductor manufacturing. *Computers & Industrial Engineering, 142,* 106358.

Huang, Z., Zhu, J., Lei, J., Li, X., & Tian, F. (2020). Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. *Journal of Intelligent Manufacturing, 31,* 953–966.

Kampouraki, A., Manis, G., & Nikou, C. (2009). Heartbeat time series classification with support vector machines. *IEEE Transactions on Information Technology in Biomedicine, 13*(4), 512–518.

Kim, E., Cho, S., Lee, B., & Cho, M. (2019). Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(3), 302–309.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). *Efficient backprop* (2nd ed., pp. 9–48). Berlin: Springer.

Le Guennec, A., Malinowski, S., & Tavenard, R. (2016). Data augmentation for time series classification using convolutional neural networks. In *Proceedings of ECML/PKDD workshop on advanced analytics and learning on temporal data* (pp. 1–8).

Lee, H., Kim, Y., & Kim, C. O. (2017a). A deep learning model for robust wafer fault monitoring with sensor measurement noise. *IEEE Transactions on Semiconductor Manufacturing, 30*(1), 23–31.

Lee, K. B., Cheon, S., & Kim, C. O. (2017b). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 30*(2), 135–142.

Lee, W. J., Xia, K., Denton, N. L., Ribeiro, B., & Sutherland, J. W. (2020). Development of a speed invariant deep learning model with application to condition monitoring of rotating machinery. *Journal of Intelligent Manufacturing*, 1–14. https://doi.org/10.1007/s10845-020-01578-x

Li, X., Zhang, W., Ding, Q., & Sun, J. Q. (2020). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing, 31*(2), 433–452.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 2–11).

Luo, J., Huang, J., & Li, H. (2020). A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis. *Journal of Intelligent Manufacturing*, 1–19. https://doi.org/10.1007/s10845-020-01579-w

Mahadevan, S., & Shah, S. L. (2009). Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control, 19*(10), 1627–1639.

Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001). Feature-based classification of time-series data. *International Journal of Computer Research, 10*(3), 49–61.

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing, 31*(1), 127–182.

Park, E. L., Park, J., Yang, J., Cho, S., Lee, Y. H., & Park, H. S. (2014). Data based segmentation and summarization for sensor data in semiconductor manufacturing. *Expert Systems with Applications, 41*(6), 2619–2629.

Patri, O. P., Sharma, A. B., Chen, H., Jiang, G., Panangadan, A. V., & Prasanna, V. K. (2014). Extracting discriminative shapelets from heterogeneous sensor data. In *Proceedings of 2014 IEEE international conference on big data* (pp. 1095–1104).

Rakthanmanon, T., & Keogh, E. (2013). Fast shapelets: A scalable algorithm for discovering time series shapelets. In *Proceedings of the 13th SIAM international conference on data mining* (pp. 668–676).

Rato, T. J., Blue, J., Pinaton, J., & Reis, M. S. (2017). Translation invariant multiscale energy-based PCA for monitoring batch processes in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering, 14*(2), 894–904.

Rostami, H., Blue, J., & Yugma, C. (2018). Automatic equipment fault fingerprint extraction for the fault diagnostic on the batch process data. *Applied Soft Computing, 68,* 972–989.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(1), 43–49.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Senin, P., & Malinchik, S. (2013). SAX–VSM: Interpretable time series classification using sax and vector space model. In *Proceedings of the IEEE 13th international conference on data mining* (pp. 1175–1180).

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9,* 2579–2605.

Wang, T., Qiao, M., Zhang, M., Yang, Y., & Snoussi, H. (2018). Data-driven prognostic method based on self-supervised learning approaches for fault detection. *Journal of Intelligent Manufacturing*, 1–9. https://doi.org/10.1007/s10845-018-1431-x

Xi, X., Keogh, E., Shelton, C., Wei, L., & Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on machine learning* (pp. 1033–1040).

Yang, J., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of twenty-fourth international joint conference on artificial intelligence* (pp. 3995–4001).

Ye, L., & Keogh, E. (2009). Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 947–956).

Yu, J. (2011). Fault detection using principal components-based Gaussian mixture model for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 24*(3), 432–444.

Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2014). Time series classification using multi-channels deep convolutional neural networks. In *Proceedings of international conference on web-age information management* (pp. 298–310).

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