A Deep Learning Model for Robust Wafer Fault Monitoring With Sensor Measurement Noise

Hoyeop Lee, Youngju Kim, and Chang Ouk Kim

***Abstract*—Standard fault detection and classification (FDC) models detect wafer faults by extracting features useful for fault detection from time-indexed measurements of the equipment recorded by *in situ* sensors (sensor signals) and feeding the extracted information into a classifier. However, the preprocessing-and-classification approach often results in the loss of information in the sensor signals that is important for detecting wafer faults. Furthermore, the sensor signals usually contain noise induced by mechanical and electrical disturbances. In this paper, we propose the use of a stacked denoising autoencoder (SdA), which is a deep learning algorithm, to establish an FDC model for simultaneous feature extraction and classification. The SdA model can identify global and invariant features in the sensor signals for fault monitoring and is robust against measurement noise. Through experiments using wafer samples collected from a work-site photolithography tool, we confirmed that as the sensor measurement noise severity increased, the SdA's classification accuracy could be as much as 14% higher than those of the twelve models considered for comparison, each of which employed one of three feature extractors and one of four classifiers.**

***Index Terms*—Semiconductor manufacturing, fault detection and classification, sensor measurement noise, deep learning, stacked denoising autoencoder.**

I. Introduction

OVER the past three decades, significant efforts have been made in the semiconductor industry and in academia to develop high-performance fault detection and classification (FDC) models that can detect wafer faults early in the semiconductor fabrication process [1]. In this study, we investigated a stacked denoising autoencoder (SdA) [2], a type of deep learning model, for use as an FDC model. FDC models use, as input, multivariate signal datasets recorded by *in situ* sensors during wafer fabrication and output the predicted class labels (normal or faulty) of the processed wafers.

Standard FDC models perform fault detection by summarizing sensor signals into single values, such as the mean and variance, followed by feeding these summary values into a classifier to determine wafer fault status. However, this approach often results in the loss of sensor information that is important for detecting wafer faults. In addition, the sensor signals usually contain noise (measurement errors) induced by mechanical and electrical disturbances. Park and Lyu [3] revealed that the vacuum sensor in photolithography equipment operates through mechanical contact; thus, its performance deteriorates over time. Sato *et al.* [4] demonstrated that sensor measurements of the voltage parameter during the metal-forming process were different from those predicted by an analytical model. Therefore, sensor noise should be considered when constructing a robust FDC model.

An autoencoder is a simple feed-forward neural network with a single layer of hidden nodes, of which there are usually fewer than the input nodes. It trains these hidden nodes to reconstruct the input at the output by minimizing the reconstruction error. As a result, the hidden nodes learn the features necessary to reproduce the input. When sensor signals contain noise and the shape of the noise distribution is symmetric, minimizing the reconstruction error (*e.g.*, the mean squared error, or MSE) yields an autoencoder that outputs the corresponding sensor signals without noise.

A denoising autoencoder adds more randomness to the learning process by arbitrarily setting some of the input nodes to zero during training to discover features that are robust against the overfitting of the model to the training data [2], [5]. An SdA is a deep neural network consisting of denoising autoencoders that are stacked layer by layer. This multi-layered architecture is capable of learning global features from complex input data, such as multivariate time-series datasets and high-resolution images. The SdA model proposed in this study can effectively learn normal and fault-related features from sensor signals without preprocessing. In addition, it is robust against sensor noise.

Most previous FDC studies have focused only on either feature extraction or classification. Feature extraction is the first stage of FDC and consists of the identification of statistical or geometric features in sensor signals to achieve the best classifier performance. He and Wang [6] used features extracted through principal component analysis (PCA) to reduce the computational complexity of a subsequent k-nearest-neighbors (kNN) analysis. As a variant of PCA that uses the entire monitored data stream, Cherry and Qin [7] suggested a multi-way PCA (MPCA) and applied it to a post-lithography process. Hong *et al.* [8] proposed a modular artificial neural network combined with Dempster-Shafer theory for fault detection in a plasma etching process. Park *et al.* [9] considered an FDC

Manuscript received July 14, 2016; revised September 1, 2016; accepted November 12, 2016. Date of publication November 15, 2016; date of current version February 1, 2017. This work was supported in part by the Technology Innovation Program (Development of Big-Data-Based Analysis and Control Platforms for Semiconductor Manufacturing Plants) by the Ministry of Trade, Industry, and Energy, MOTIE, South Korea, under Grant 10045913, and in part by the National Research Foundation of Korea through the Korean Government (MSIP) under Grant NRF-2016R1A2B4008337.

The authors are with the Department of Information and Industrial Engineering, Yonsei University, Seoul 03722, South Korea (e-mail: kimco@yonsei.ac.kr).

Digital Object Identifier 10.1109/TSM.2016.2628865

model based on sensor signals and proposed a method of extracting features from the signals while minimizing information loss via spline regression. Ko *et al.* [10] proposed an FDC model that employs the structural features of sensor signals, such as their geometric shapes, lengths, and heights, as the input to the classifier.

Classification, which is the second stage of FDC, is the process of determining whether a wafer is normal or faulty using the previously extracted features. He *et al.* [11] presented a decision-tree-based FDC model that can distinguish between different types of wafer faults. In addition, He and Wang [12] proposed a kNN-rule-based FDC model based on the Euclidean distance. Verdier and Ferreira [13] proposed a kNN method based on the Mahalanobis distance. Baly and Hajj [14] proposed the use of a support vector machine (SVM) with the Gaussian radial basis kernel function to detect wafer faults. Kwak *et al.* [15] proposed an incremental clustering-based fault detection algorithm that can accurately find wafer faults even in the case of severely skewed class distributions.

Recently, Lee and Kim [16] conducted statistical experiments to compare the performances of FDC models constructed by combining extant feature extraction methods and classification methods and revealed that the FDC performance varies significantly depending on the combination adopted. Specifically, they discovered that using features that are not suitable for a particular classifier can significantly reduce that classifier's performance. Therefore, it is desirable to consider both the feature extraction and classification stages simultaneously to maximize FDC performance.

To verify the performance of the proposed SdA model, 90 wafer samples were collected from a work-site photolithography tool. Then, four levels of sensor noise were introduced; a total of 70,000 training, validation, and test data sets were generated by imposing noise of each level on every sensor reading. Twelve FDC models, each of which was constructed from one of three feature extractors (none, MPCA, or statistical summary) and one of four classifiers (kNN or an SVM with a linear, polynomial, or radial basis kernel function), were considered for a performance comparison. The results of the experiments confirm that as the sensor noise severity increased, the SdA model could achieve a classification accuracy of up to 14% greater than those of the other models.

The remainder of this paper is organized as follows. Section II briefly introduces deep learning. Section III describes the proposed SdA model. Section IV presents an analysis of the results obtained by comparing the performances of the SdA model and the twelve comparative models. Finally, Section V presents the conclusions and future topics of research.

II. Deep Learning

Artificial neural networks, which began to gain popularity in the early 1980s [17], are machine learning models inspired by the operating principles of the neural network of the human brain and have been widely applied to problems of classification and function approximation. In general, a basic (feed-forward) neural network is composed of an input layer, a hidden layer, an output layer, and full connections between the layers [18]. Neural network learning is the process of determining the connection weights between the layers such that performance is maximized when solving a learning problem.

Deep neural networks contain multiple hidden layers of nodes between the input and output layers. Deep neural networks are known to exhibit excellent performance when they are well trained. However, interest in deep neural networks has greatly decreased since the late 1980s for the following reasons. First, the vanishing gradient problem has arisen [19]. Specifically, if a backpropagation algorithm is used to train an artificial neural network, then only the weights of a few hidden layers close to the output layers are correctly learned, whereas weight learning does not occur properly in the hidden layers far from the output layers. Second, deep neural networks are prone to overfitting. Overfitting occurs when the weights, which are the learning parameters of the model, are determined to function at maximum performance in response to the training data; consequently, the networks performance is lower when it is presented with new data. Finally, problems associated with computer performance have arisen when training deep neural networks. As the size of a neural network increases to solve more complex problems, the number of weights that must be learned also increases exponentially, as does the amount of time required to train the model.

Recently, various solutions to these problems have been presented. Srivastava *et al.* [20] proposed the dropout method to prevent overfitting, and Oh and Jung [21] suggested a method of increasing the learning rates of artificial neural networks by employing graphical processing units. Hinton and Salakhutdinov [22] developed deep neural networks called deep belief networks by stacking multiple restricted Boltzmann machines and trained these networks one layer at a time to resolve the vanishing gradient and overfitting problems associated with the classical backpropagation algorithm. The learning objective for a restricted Boltzmann machine is to ensure that the reproduced data are as close as possible to the input data by employing a weight update formula. Therefore, the weights of the trained hidden nodes are effective for feature extraction.

As a result of the significant efforts directed toward solving the problems previously associated with artificial neural networks, deep learning has attracted attention in both academia and a variety of businesses. Consequently, various other deep learning models, in addition to deep belief networks [22], have been developed. One recently developed deep learning model is SdA [2], in which denoising autoencoders are stacked layer by layer. An autoencoder generates an output that is identical to the input. Therefore, autoencoders, similarly to restricted Boltzmann machines, are well-known feature extractors [23], although their network structures are different from those of restricted Boltzmann machines. Other deep learning models include convolutional neural networks [24], which are primarily used for image classification, and deep recurrent neural networks [25], which are used for signal processing. Thus far,

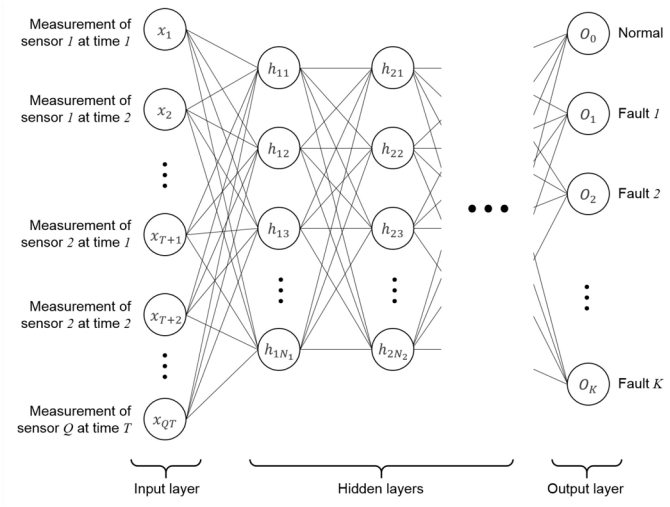


Fig. 1. Deep neural network structure.

however, deep learning studies have mostly focused on speech recognition or image classification problems.

III. Stacked Denoising Autoencoder for Wafer Fault Monitoring

Let us assume that the number of sensors in a wafer manufacturing apparatus is Q and that the sensor data measured over the time duration T that is required to manufacture a wafer are $x_{11}, \ldots, x_{1T}, \ldots, x_{Q1}, \ldots, x_{QT}$, where x_{qt} is the measurement recorded by sensor q at time t. In this study, x_{qt} was normalized to take values between 0 and 1. The deep neural network had the structure shown in Fig. 1. Without loss of generality, the sensor signals can be expressed as the input layer data $\boldsymbol{X} = [x_1, x_2, \ldots, x_{QT-1}, x_{QT}]^\top$. The nodes of the jth hidden layer and the output layer are denoted by $\boldsymbol{H}_j = [h_{j1}, \ldots, h_{jN_j}]^\top$ and $\boldsymbol{O} = [o_0, \ldots, o_K]^\top$, respectively. Here, the lth node of the output layer, o_l, corresponds to fault type l and produces the probability of that fault type. Note that we regard fault type 0, corresponding to the first node o_0, as representative of a normal wafer. The number of hidden layers, L, and the number of hidden nodes in the jth hidden layer, N_j, are defined by the user.

The weight matrices for the connections from the input layer to the first hidden layer, from the jth hidden layer to the $(j+1)$th hidden layer, and from the last hidden layer to the output layer are denoted by $\boldsymbol{W}_0$, $\boldsymbol{W}_j$, and $\boldsymbol{W}_L$, respectively. In addition, bias terms are denoted by $\boldsymbol{B}_0$, $\boldsymbol{B}_j$, and $\boldsymbol{B}_L$, and $f(.)$ is defined as the activation function. Then, the relationships between $\boldsymbol{H}_1$, $\boldsymbol{H}_j$, and $\boldsymbol{O}$ are as shown in (1), (2), and (3), respectively:

$$\boldsymbol{H}_1 = f(\boldsymbol{W}_0\boldsymbol{X} + \boldsymbol{B}_0) \tag{1}$$

$$\boldsymbol{H}_{j+1} = f\big(\boldsymbol{W}_j\boldsymbol{H}_j + \boldsymbol{B}_j\big) \quad (1 \le j \le L-1) \tag{2}$$

$$\boldsymbol{O} = \mathrm{softmax}(\boldsymbol{W}_L\boldsymbol{H}_L + \boldsymbol{B}_L). \tag{3}$$

In these equations, f is a component-wise function that satisfies $f([x_1, \ldots, x_n]^\top) = \big[f(x_1), \ldots, f(x_n)\big]^\top$ and generally uses the hyperbolic tangent function to resolve the gradient vanishing problem [26]. The softmax function in (3) is defined as

$$\mathrm{softmax}\Big([x_1, \ldots, x_n]^\top\Big) = \left[\frac{\exp(x_1)}{\sum_{i=1}^n \exp(x_i)}, \ldots, \frac{\exp(x_n)}{\sum_{i=1}^n \exp(x_i)}\right]^\top. \tag{4}$$

The neural network learns to minimize the difference between the actual fault vector $\boldsymbol{D} = [d_0, \ldots, d_K]^T$, where $d_m = 1$ and $d_n = 0 (m \neq n)$ if a fault of type m has occurred, and the fault probability vector $\boldsymbol{O}$ that is predicted through the network by computing the MSE.

$$MSE = \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{K}\big(d_{pk} - o_{pk}\big)^2. \tag{5}$$

Here, P is the number of training samples.

An SdA is constructed by stacking denoising autoencoders in layers. Pre-training is performed in a layer-wise fashion, followed by a fine-tuning process throughout the entire network. A detailed description of this learning process is as follows.

Step 1: As shown in Fig. 2, train the weight matrices $\boldsymbol{W}_0$ and $\boldsymbol{W}_0'$ of the first denoising autoencoder using the backpropagation algorithm with all training data.

Step 2: Remove the output layer of the trained first denoising autoencoder (also removing $\boldsymbol{W}_0'$, but retaining the learned $\boldsymbol{W}_0$), causing the hidden layer of this first autoencoder to become the input layer of the second. Then, update the weight matrices $\boldsymbol{W}_1$ and $\boldsymbol{W}_1'$ of the second autoencoder via backpropagation with the training data.

Step 3: Repeat steps 1 and 2 for all hidden layers (*i.e.*, remove the output layer of the previous denoising autoencoder, replace it with the current denoising autoencoder, and train via backpropagation).

Steps 1, 2, and 3 are pre-training steps that efficiently initialize the weights throughout the network. This pre-training resolves the vanishing gradient problem. In SdA learning, the features (*i.e.*, hidden nodes) in each hidden layer are fed into the next hidden layer, thereby creating more highly abstracted features [27]. Hence, the stacking of hidden layers makes it possible to automatically identify global features in the layers close to the output layer. However, the SdA learning process is unsupervised. The model accepts sensor signals as input and generates an output that is similar to the input. There is no mapping between the input and the class labels (*e.g.*, normal or faulty). For example, even if the network is trained to detect wafer faults, it is still not possible to map the nodes from the last feature extractor (*i.e.*, the hidden layer of the last denoising autoencoder) to the wafer fault types. A practical solution is to add one fully connected layer to the last layer using the softmax function, whose weight matrix is denoted by $\boldsymbol{W}_L$ in Fig. 2. As a final step, the weight matrices $\boldsymbol{W}_0, \ldots, \boldsymbol{W}_L$ of the entire network are trained via backpropagation (this step is called fine-tuning). Together, pre-training

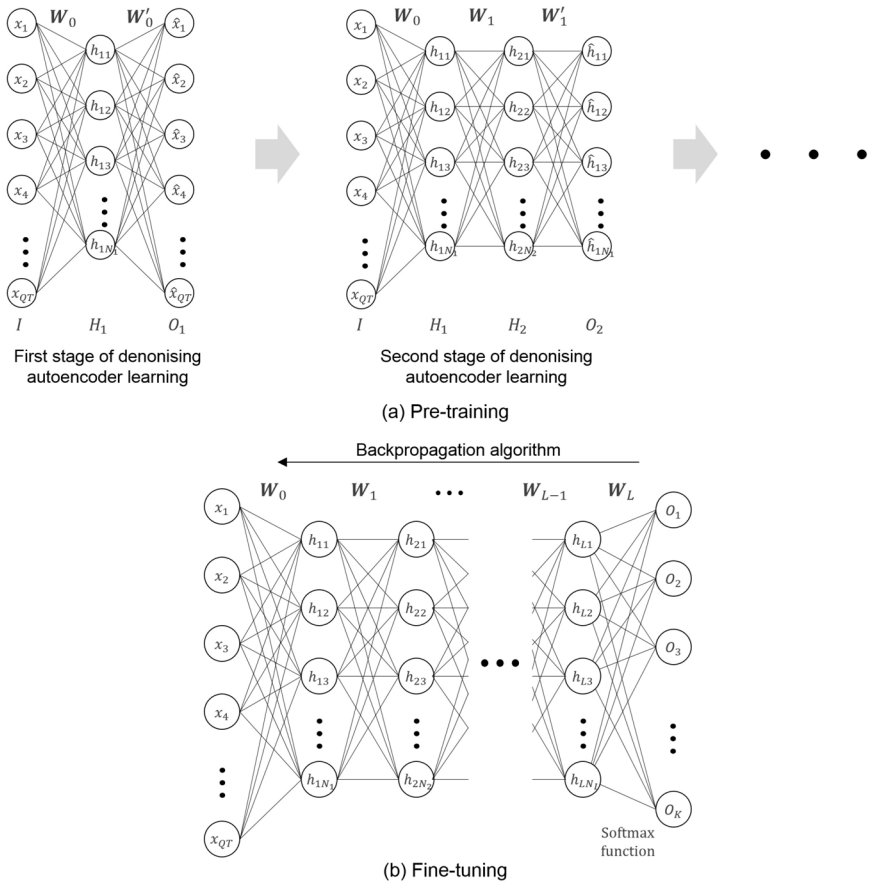


Fig. 2. (a) Pre-training and (b) fine-tuning of stacked denoising autoencoder.

and fine-tuning produce an effective SdA for wafer fault monitoring.

IV. Experiments

A. Experimental Design

For the experiments, data from 39 normal wafer samples and 51 faulty wafer samples were collected from three sensors mounted on a work-site photolithography tool. In general, the photolithography process consists of six steps: cleaning, preparation, photoresist application, exposure, developing, and resist stripping [28]. In this paper, because a lack of photoresist adhesion results in a distorted circuit image, we focused on the sub-step in which adhesion promoter is applied to ensure the firm adhesion of the photoresist. In a closed chamber, hexamethyldisilazane (HMDS) vapor in a stream of dry nitrogen is applied to the wafer, and the HMDS is allowed to spin dry at a high temperature. If the HMDS does not dry properly, the adhesive strength will be lost. In this experiment, three in situ sensors read out the volume of adhesion promoter, the volume of nitrogen, and the temperature 116 times during the manufacturing of each wafer. We identified that the faulty wafer samples were caused by deviations in these three process parameters during the adhesion promotion process.

We observed that the sensor readings at each measurement time for the normal wafers were not identical but rather randomly fluctuated up and down with respect to the set points of the process recipe. Based on this observation, we considered four levels of sensor measurement noise: low, moderate, high, and extreme. At each sensor measurement time, the maximum absolute value of the sensor readings at that time point was obtained from the normal wafer samples, and the four levels of noise at that point were generated using uniform distributions with identical means of zero and ranges of 0.01%, 0.1%, 1% and 10%, respectively, of the maximum absolute value. The noise was generated following the jittering method, which is a widely used data augmentation method [24]. This method creates artificial data that are similar to, but slightly different from, the real data and helps to avoid overfitting in the analysis results [29]. For each noise level, 50,000 training data, 10,000 verification data, and 10,000 test data were generated by applying the jittering method to the sensor signals from the normal and faulty wafer samples. The data sizes were determined based on those of the MNIST dataset [26], which is a popular dataset for use in deep learning studies.

The input layer of the SdA used in this study consisted of 348 ($= 3 \times 116$) nodes, and the output layer consisted of two nodes: normal and faulty. To observe how the SdA's fault detection performance was affected by the number of hidden layers, up to four hidden layers were tested, where the first, second, third, and fourth hidden layers consisted of 1,000, 500, 300, and 100 hidden nodes, respectively. The weight learning parameters for pre-training and fine-tuning were set to 0.005 and 0.0005, respectively, and the probability of masking noise

TABLE I
COMPARISON RESULTS FOR THE CASES OF LOW AND MODERATE SENSOR MEASUREMENT NOISE

Sensor measurement noise level	Algorihtm		Type I error	Type II error	Accuracy
	Feature extractor	Classifier			
Low	None	kNN	0.000	0.000	1.000
		SVM-L	0.157	0.125	0.857
		SVM-P	0.021	0.000	0.988
		SVM-R	0.000	0.000	1.000
	MPCA	kNN	0.000	0.000	1.000
		SVM-L	0.271	0.798	0.501
		SVM-P	0.000	0.000	1.000
		SVM-R	0.000	0.000	1.000
	Statistical summary values (mean and variance)	kNN	0.005	0.005	0.995
		SVM-L	0.414	0.503	0.547
		SVM-P	0.455	0.540	0.508
		SVM-R	0.324	0.840	0.452
	SdA-1		0.154	0.113	0.870
	SdA-2		0.008	0.014	0.989
	SdA-3		0.000	0.000	1.000
	SdA-4		0.001	0.006	0.996
Moderate	None	kNN	0.151	0.129	0.859
		SVM-L	0.237	0.266	0.750
		SVM-P	0.213	0.227	0.781
		SVM-R	0.006	0.007	0.994
	MPCA	kNN	0.141	0.224	0.822
		SVM-L	0.600	0.354	0.506
		SVM-P	0.099	0.625	0.674
		SVM-R	0.484	0.300	0.596
	Statistical summary values (mean and variance)	kNN	0.081	0.116	0.904
		SVM-L	0.221	0.596	0.617
		SVM-P	0.223	0.577	0.624
		SVM-R	0.072	0.875	0.581
	SdA-1		0.234	0.136	0.822
	SdA-2		0.097	0.065	0.922
	SdA-3		0.003	0.009	0.994
	SdA-4		0.005	0.004	0.996

was set to 0.1. Learning was terminated when the MSE of the validation data was minimized. In this study, the SdA was implemented using pylearn2 [30].

The performances of the twelve other FDC models, each of which consisted of one of three feature extractors and one of four classifiers, were measured for comparison with that of the SdA. The settings for the feature extractors (none, MPCA, and statistical summary) were as follows. "None" indicates that all of the sensor signals were used directly as inputs without performing feature extraction. MPCA is an extension of the standard PCA and utilizes every measured sensor signal to extract principal components. That is, this feature extraction model regards every sensor reading as an input feature for PCA. For example, a wafer sample with ten process parameters, each with 100 sensor measurements, yields 1,000 features. In our experiments, we reduced the 348 sensor data features generated by the three process parameters into 20 principal components via PCA. In the statistical summary method, the mean and standard deviation of the signal of each sensor were used. The four classification methods investigated in this study were kNN analysis and SVMs with linear, polynomial, and radial basis kernel functions. The parameter values that yielded the highest performances, as identified through cross-validation on the training data, were set as the parameter values for the classifiers.

The performance indicators used in this study were the type I error, the type II error, and the accuracy, as shown in (6), (7), and (8), respectively:

$$\text{Type I error} = \frac{\text{number of normal wafers misclassified as fault}}{\text{total number of normal wafers}} \tag{6}$$

$$\text{Type II error} = \frac{\text{number of faulty wafers misclassified as fault}}{\text{total number of faulty wafers}} \tag{7}$$

$$\text{Accuracy} = \frac{\text{number of misclassified wafers}}{\text{total number of wafers}}. \tag{8}$$

B. Results and Analysis

Tables I and II present the performance results for the test data after the introduction of the four levels of sensor measurement noise. In the tables, the number of hidden layers n in each SdA model is indicated by SdA-n; the SVMs with linear, polynomial, and radial kernel functions are denoted by SVM-L, SVM-P, and SVM-R, respectively; and the values indicating the models with the best performance are underlined. When the sensor measurement noise level was low or moderate (Table I), the best accuracies of the SdA models were similar to the best accuracies of the other twelve models (0% for low noise and 0.2% for moderate noise). However, when the noise level was increased to high or extreme (Table II), the average performance gap increased to approximately 14% (18.4% for high noise and 10.3% for extreme noise).

TABLE II
COMPARISON RESULTS FOR THE CASES OF HIGH AND EXTREME SENSOR MEASUREMENT NOISE

Sensor measurement noise level	Algorihtm		Type I error	Type II error	Accuracy
	Feature extractor	Classifier			
High	None	kNN	0.340	0.473	0.602
		SVM-L	0.339	0.365	0.650
		SVM-P	0.159	0.599	0.650
		SVM-R	0.322	0.364	0.659
	MPCA	kNN	0.335	0.512	0.588
		SVM-L	0.140	0.913	0.524
		SVM-P	0.072	0.901	0.568
		SVM-R	0.043	0.949	0.564
	Statistical summary values (mean and variance)	kNN	0.157	0.254	0.801
		SVM-L	0.210	0.634	0.606
		SVM-P	0.200	0.671	0.596
		SVM-R	0.202	0.663	0.598
	SdA-1		0.277	0.140	0.800
	SdA-2		0.105	0.085	0.906
	SdA-3		0.066	0.063	0.936
	SdA-4		0.006	0.022	0.985
Extreme	None	kNN	0.327	0.525	0.586
		SVM-L	0.468	0.340	0.588
		SVM-P	0.053	0.845	0.598
		SVM-R	0.281	0.544	0.603
	MPCA	kNN	0.338	0.549	0.569
		SVM-L	0.174	0.741	0.576
		SVM-P	0.046	0.940	0.560
		SVM-R	0.175	0.784	0.557
	Statistical summary values (mean and variance)	kNN	0.254	0.388	0.687
		SVM-L	0.204	0.706	0.575
		SVM-P	0.203	0.718	0.570
		SVM-R	0.178	0.777	0.558
	SdA-1		0.317	0.177	0.762
	SdA-2		0.274	0.171	0.784
	SdA-3		0.260	0.178	0.786
	SdA-4		0.224	0.200	0.790

The SdA models exhibited superior performance compared with the other FDC models.

As shown in Table I, when the level of sensor measurement noise was low, the feature extractor based on statistical summary values failed to identify sensor signal features that could effectively discriminate faulty wafers from normal ones. Sensor signals contain information regarding the time-indexed signal shape that is useful for fault detection. This feature extractor lost this pattern information during data summarization. When the noise level was low, the "none" and MPCA feature extraction methods, which do not require the summarization of the sensor signals, exhibited good results. However, when the sensor measurement noise was increased to the moderate level, the performances of these two methods were not excellent. These two feature extractors were not capable of filtering the measurement noise, which caused high type I and type II errors. The features of normal and faulty wafers that were generated by all three feature extractors were located close to each other in the feature space. As a result, the SVM classifiers failed to construct single classification boundaries with good performance in the feature space. By contrast, kNN is a local classifier and assigns a class label to each new wafer based on only a few instances located near that wafer. This instance-based learning property results in good performance for hard classification problems. As shown in Table II, the performances of the three feature extractors were further decreased for the test data with high and extreme levels of sensor measurement noise. In particular, all of the FDC models except the SdA models tended to exhibit higher type II errors than type I errors; thus, these models would be undesirable for use in actual applications because they could lead to decreased manufacturing yields.

Table I shows that SdA-3 and SdA-4, *i.e.*, the SdA models with three and four hidden layers, exhibited the best performance for the low and moderate noise levels, respectively, with small type I and type II errors. In addition, in Table II, the SdA models show a tendency for the type II error to be smaller than the type I error. Tables I and II illustrate the effect of the autoencoders – namely, the sensor measurement noise was filtered out during the training of the model. In the autoencoder models, the neural network was trained using the backpropagation algorithm to reconstruct the input noise-affected sensor signals at the output with minimal reconstruction error. As a result, the backpropagation algorithm forced the neural network to generate filtered sensor signals at the output. In addition, the results in Tables I and II suggest that as the difficulty of fault detection increases, a greater number of hidden layers, which act as feature extractors, is required. Each autoencoder learns the abstracted features of the input sensor signals with fewer hidden nodes than input nodes to reproduce the input signals at the outputs. Thus, the first hidden layer (autoencoder) of an SdA model learns the classification features of the source sensor measurements, and the second layer (autoencoder) combines the features found in the first layer to create more highly abstracted and synthesized features. As the number of hidden layers increases, more global and more

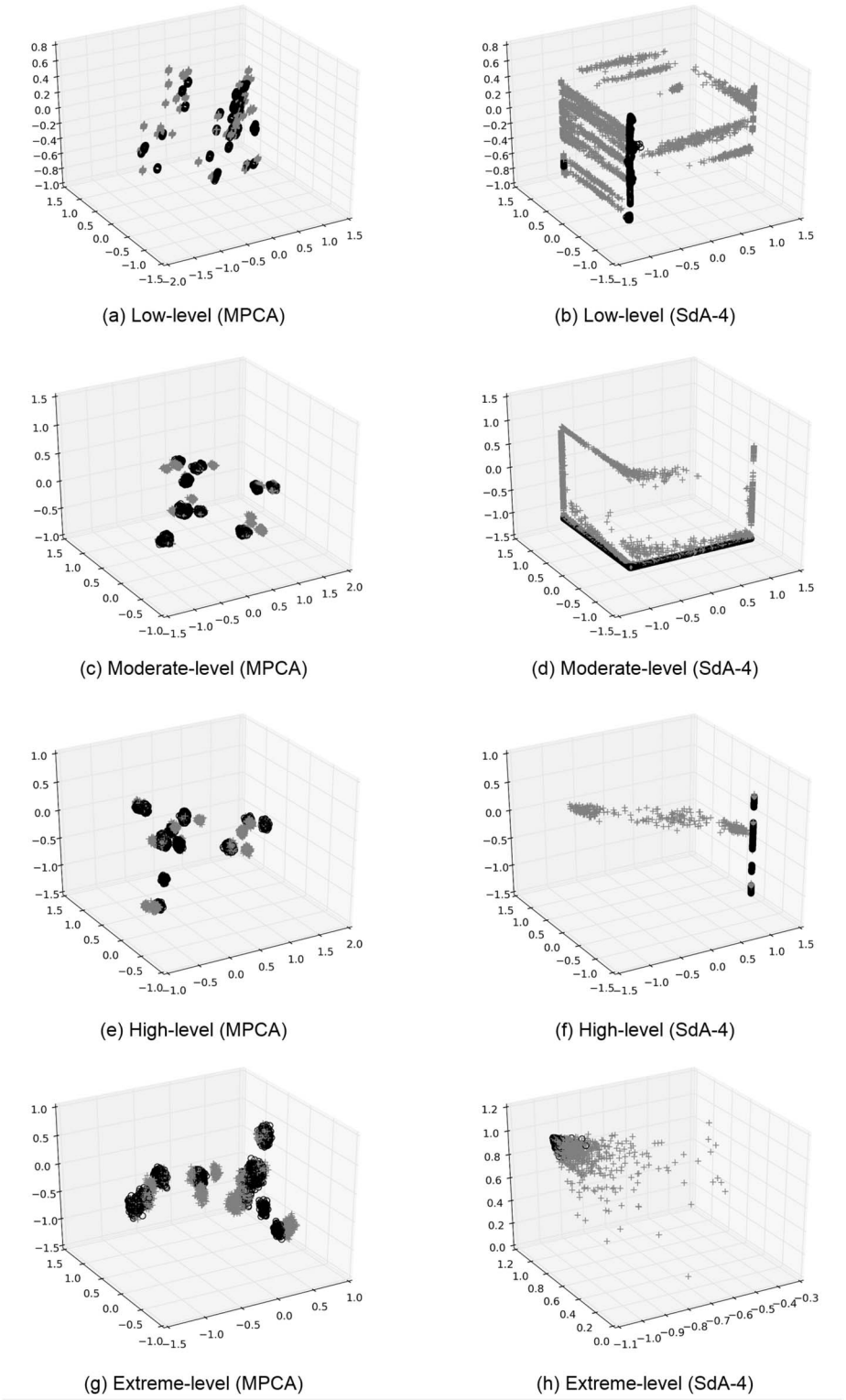


Fig. 3. Visualizations of the normal wafers (black circles) and faulty wafers (gray + signs) in three-dimensional space. The axes in (a), (c), (e), and (g) are the principal components, and those in (b), (d), (f), and (h) are three hidden nodes of the fourth hidden layer in SdA-4.

highly abstracted features are obtained, which are more effective for wafer classification. From these results, we conclude that the SdA architecture can learn global and abstracted features that are useful not only for image classification [2] but also in FDC applications for semiconductor manufacturing.

Fig. 3 presents three-dimensional scatter plots with axes defined by three hidden nodes of the trained SdA-4 models, *i.e.*, the nodes of the fourth hidden layer that most significantly influenced the output layer, and by the first three principal components extracted via MPCA. Normal and faulty wafers

are represented by black circles and gray + signs, respectively. Figs. 3(a), 3(c), 3(e), and 3(g) show the input data, i.e., the sensor signals, in terms of the three principal component features, and it is evident that the features corresponding to the normal and faulty wafers are mixed together. These results illustrate that the principal component features were not effective for solving the fault detection problem in the presence of sensor noise.

By contrast, Figs. 3(b), 3(d), 3(f), and 3(h) show the results in terms of the three SdA hidden-node features. For low and moderate levels of sensor measurement noise, the normal wafers in Figs. 3(b) and 3(d) are concentrated in specific areas that are distinct from those in which the faulty wafers are concentrated. Specifically, these features have a tendency to cluster near the walls of the feature space because the hyperbolic tangent activation functions of the hidden nodes produce extreme values of -1 or +1. These results imply that the hidden nodes of the trained SdA-4 models effectively distinguished the faulty wafers from the normal ones, resulting in high fault detection performance. However, as the noise level was increased to high and then extreme, the performance of the SdA models decreased, as seen from Figs. 3(f) and 3(h). The normal wafers remain concentrated in specific areas, but significant numbers of the faulty wafers are mixed in with the normal wafers because of the severe sensor measurement noise.

V. Conclusion

In this paper, we proposed an SdA as an FDC model that is robust against measurement noise in sensors mounted on semiconductor manufacturing equipment. An SdA is a deep learning algorithm that can extract decisive features for wafer fault detection from sensor signals and perform wafer classification using the extracted features. We experimentally confirmed that with increasing noise severity, the SdA's classification accuracy could be up to 14% greater than those of twelve other data-mining models considered for comparison, each of which employed one of three feature extractors and one of four classifiers. Furthermore, because the type II error was smaller than the type I error when the SdA approach was used, the proposed SdA model is expected to be useful in real applications.

Two promising topics for further research can be identified. First, only the photolithography process was investigated in our study to test the performance of the SdA. Additional analyses should be conducted on data from different semiconductor manufacturing processes to verify the suitability of the SdA for wafer fault detection. Furthermore, it will be necessary to investigate a trained SdA to identify the process parameters that most significantly impact the classification results to develop a procedure for analyzing the main causes of wafer faults.

References

[1] S. P. Cunningham, C. J. Spanos, and K. Voros, "Semiconductor yield improvement: Results and best practices," *IEEE Trans. Semicond. Manuf.*, vol. 8, no. 2, pp. 103–109, May 1995.

[2] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[3] S.-J. Park and K.-M. Lyu, "Vacuum-adsorbing apparatus of semiconductor device fabrication facility," U.S. Patent 6,166,646, 2000.

[4] T. Sato, D. Sylvester, Y. Cao, and C. Hu, "Accurate in situ measurement of peak noise and delay change induced by interconnect coupling," *IEEE J. Solid-State Circuits*, vol. 36, no. 10, pp. 1587–1591, Oct. 2001.

[5] J. Li and A. Cichocki, "Deep learning of multifractal attributes from motor imagery induced EEG," in *Proc. Int. Conf. Neural Inf. Process.*, Kuching, Malaysia, 2014, pp. 503–510.

[6] Q. P. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *Proc. Amer. Control Conf.*, Seattle, WA, USA, 2008, pp. 1606–1611.

[7] G. A. Cherry and S. J. Qin, "Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 159–172, May 2006.

[8] S. J. Hong, W. Y. Lim, T. Cheong, and G. S. May, "Fault detection and classification in plasma etch equipment for semiconductor manufacturing *e*-diagnostics," *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 1, pp. 83–93, Feb. 2012.

[9] J. Park, I.-H. Kwon, S.-S. Kim, and J.-G. Baek, "Spline regression based feature extraction for semiconductor process fault detection using support vector machine," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5711–5718, 2011.

[10] J. M. Ko, C. O. Kim, S. J. Lee, and J. P. Hong, "Structural feature-based fault-detection approach for the recipes of similar products," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 2, pp. 273–283, May 2010.

[11] S. He, G. A. Wang, M. Zhang, and D. F. Cook, "Multivariate process monitoring and fault identification using multiple decision tree classifiers," *Int. J. Prod. Res.*, vol. 51, no. 11, pp. 3355–3371, 2013.

[12] Q. P. He and J. Wang, "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345–354, Nov. 2007.

[13] G. Verdier and A. Ferreira, "Adaptive mahalanobis distance and *k*-nearest neighbor rule for fault detection in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 1, pp. 59–68, Feb. 2011.

[14] R. Baly and H. Hajj, "Wafer classification using support vector machines," *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 3, pp. 373–383, Aug. 2012.

[15] J. Kwak, T. Lee, and C. O. Kim, "An incremental clustering-based fault detection algorithm for class-imbalanced process data," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 3, pp. 318–328, Aug. 2015.

[16] T. Lee and C. O. Kim, "Statistical comparison of fault detection models for semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 80–91, Feb. 2015.

[17] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[19] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[21] K.-S. Oh and K. Jung, "GPU implementation of neural networks," *Pattern Recognit.*, vol. 37, no. 6, pp. 1311–1314, 2004.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

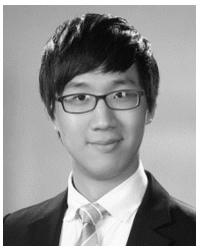
[23] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, vol. 19. Cambridge, MA, USA: MIT Press, 2007, pp. 153–160.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural. Inf. Process. Syst.*, 2012, pp. 1106–1114.

[25] A. Graves, A.-R. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, 2013, pp. 6645–6649.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[27] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Dept. IRO, Université de Montréal, Montreal, QC, Canada, Tech. Rep. 1341, vol. 4323, 2009.
[28] G. S. May and C. J. Spanos, *Fundamentals of Semiconductor Manufacturing and Process Control*. Hoboken, NJ, USA: Wiley, 2006.
[29] R. M. Zur, Y. Jiang, and C. E. Metz, "Comparison of two methods of adding jitter to artificial neural network training," *Int. Congr. Series*, vol. 1268, pp. 886–889, Jun. 2004.
[30] I. J. Goodfellow *et al.*, "Pylearn2: A machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013.



Hoyeop Lee received the B.S. degree in information and industrial engineering from Yonsei University, Seoul, South Korea, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include data mining, deep learning, and fault detection and classification model for semiconductor manufacturing.

Youngju Kim received the B.S. degree in information and industrial engineering from Yonsei University, Seoul, South Korea, in 2016, where she is currently pursuing the Ph.D. degree. Her research interests include machine leaning, deep learning, and feature learning, and their applications.

Chang Ouk Kim received the B.S. and M.S. degrees from Korea University, Seoul, South Korea, in 1988 and 1990, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1996, all in industrial engineering. He is a Professor with the Department of Information and Industrial Engineering, Yonsei University, South Korea. He has published over 100 papers in journals and conference proceedings. His current research interest is data science for manufacturing.