Pattern Anal Applic (2014) 17:863–881
DOI 10.1007/s10044-014-0386-6

INDUSTRIAL AND COMMERCIAL APPLICATION

Evaluating the reliability level of virtual metrology results for flexible process control: a novelty detection-based approach

Pilsung Kang · Dongil Kim · Sungzoon Cho

Received: 12 April 2012 / Accepted: 19 May 2014 / Published online: 26 June 2014
© Springer-Verlag London 2014

Abstract The purpose of virtual metrology (VM) in semiconductor manufacturing is to support process monitoring and quality control by predicting the metrological values of every wafer without an actual metrology process, based on process sensor data collected during the operation. Most VM-based quality control schemes assume that the VM predictions are always accurate, which in fact may not be true due to some unexpected variations that can occur during the process. In this paper, therefore, we propose a means of evaluating the reliability level of VM prediction results based on novelty detection techniques, which would allow flexible utilization of the VM results. Our models generate a high-reliability score for a wafer's VM prediction only when its process sensor values are found to be consistent with those of the majority of wafers that are used in model building; otherwise, a low-reliability score is returned. Thus, process engineers can selectively utilize VM results based on their reliability level. Experimental results show that our reliability generation models are effective; the VM results for wafers with a high level of reliability were found to be much more accurate than those with a low level.

Keywords Virtual metrology · Reliability level · Novelty detection · Semiconductor · Process monitoring

P. Kang (✉)
School of Industrial Management Engineering, Korea University, Seoul, South Korea
e-mail: pilsung.kang@gmail.com

D. Kim · S. Cho
Department of Industrial Engineering, Seoul National University, 599 Gwanak-ro, Gwanak-gu, Seoul 151-744, South Korea

1 Introduction

Semiconductor manufacturing consists of hundreds of individual steps that a wafer must pass through in order to become a final product. Recently, these individual operations have become more complex and the process dimensions have become smaller. This has increased the importance of a precise process monitoring and quality control. In typical semiconductor manufacturing, process monitoring and quality control involves an actual metrology process and statistical process control (SPC) techniques [24, 51, 54].

Although metrology-based SPC is the most widely used quality control scheme, it has some limitations. First, there is a trade-off between the effectiveness (high yield) and the efficiency (cycle time) of the manufacturing process. Metrology is not a value-added operational process, and it is only used for process monitoring. If process engineers implement more frequent metrology processes between operational processes, the total number of processes increases. In addition, the remaining wafers must be held until the investigation of the sampled wafers is completed. As a consequence, the total production cycle time increases at the expense of a higher yield rate [13, 55]. Second, wafer-to-wafer quality control is practically impossible as long as sampling techniques are utilized in the metrology process [14, 41]. Sampling-based metrology assumes that the metrology measurements of the other wafers are consistent with those of one or two wafers sampled from the same lot. In a real process, however, variations in quality arise due to many unexpected deviations in the process. Therefore, there is always a higher risk of both missed wafers (faulty wafers that are not picked out by the process) and false alarms (in which normal wafers are picked out for being faulty), as compared to a scenario in which wafer-to-wafer quality control is possible.

In order to overcome such limitations of metrology-based SPC, virtual metrology has been highlighted as a new scheme of advanced process control (APC) that makes possible wafer-to-wafer quality control in semiconductor manufacturing [17, 18, 56]. The purpose of virtual metrology (VM) is to support process monitoring and quality control by predicting the metrological values of every wafer without implementing an actual metrology process, based on process sensor data that is collected during the operation. The development of an accurate virtual metrology model offers many benefits. First, process engineers can take more appropriate actions to improve the final yield, such as adjusting operation recipes, based on information that is richer than produced by an actual metrology process [11]. Second, once a prediction model is built, the number of wafers measured by the actual metrology equipment can be significantly decreased, because only a few wafers are required in order to maintain and update the model. Thus, the total production cycle time and resources required by the actual metrology process is decreased, resulting in higher production efficiency [13]. Third, real-time process drift detection [41] as well as wafer-to-wafer (run-to-run; R2R) process control [35] becomes possible, since virtual metrology provides continuous process monitoring on a wafer level.

Due to many benefits of virtual metrology, it has been widely studied since the latest 1990s, and the research in this area has developed in two main directions. The first direction is to improve the prediction accuracy of virtual metrology models by developing new prediction algorithms or by selecting (extracting) relevant input variables (predictors) [4, 15, 41, 42, 46]. The second direction of research involves building a real-time process control system by integrating virtual metrology and an R2R control scheme [6, 12, 35, 47, 48, 64]. Although previous studies have achieved noticeable progresses in both directions, most of them rely on a common but not realistic assumption; VM prediction results are quite accurate and reliable. However, despite its many benefits, in practice VM is subject to two types of intrinsic risks. The first risk is model risk, which is related to inaccurate prediction results. When R2R control is running, process recipe manipulation or equivalent proper actions are selectively activated based upon each wafer's VM prediction results. If the VM prediction result is accurate, the follow-up actions are appropriate. However, if the VM result is inaccurate, the follow-up actions will cause several additional problems because the operation is based upon wrong information. The second type of risk is data risk, which is related to the difference between the data used to build a VM model (training data) and the data used for predicting metrological values (test data). When VM prediction is highly accurate, this means that the functional relation between the input variables (process sensor parameters) and target variables (metrological values) of the training data was well-captured by the model. However, no model can make a very accurate prediction when highly heterogeneous data, which was not seen when the model was built, is provided, even though a certain degree of generalization is possible. As a consequence, this heterogeneous data may increase the uncertainty of the R2R control. As mentioned earlier, a great deal of research focused upon lowering the model risk. Only a few works, however, have been devoted to lowering the data risk [16].

In this paper, in order to reduce the data risk, we propose a means of evaluating the reliability level of VM prediction results based upon novelty detection techniques. To do so, VM prediction models and novelty detection models are built based upon the same training data. When a new wafer arrives, its process sensor data is provided simultaneously to both the VM and novelty detection models. If the sensor data are similar enough to those of the training wafers, a high-reliability score is assigned to the wafer's VM prediction result; if not, a low-reliability score is assigned. Process engineers can then increase the flexibility of process control and enhance overall productivity by selectively utilizing a wafer's VM prediction results, based on its reliability level. The main contributions of this paper can be summarized as follows:

- Reliance level for each VM prediction result is evaluated.
- Novelty detection algorithms and their combinations are employed to evaluate the homogeneity of the input sensor values.
- Practical applicability of the proposed framework is verified.

The rest of this paper is structured as follows. In Sect. 2, we briefly review the research articles related to VM and novelty detection. In Sect. 3, we present the structure of our reliability evaluation system and its individual components. In Sect. 4, we explain the experimental settings such as data description, variable selection methods, VM prediction models, algorithm parameters, and performance measures. In Sect. 5, we analyze the effect of the reliability evaluation models in terms of two prediction accuracy measures. In Sect. 6, along with some concluding remarks, we discuss areas of future work.

2 Related work

2.1 Virtual metrology

In semiconductor manufacturing, a general process of metrology-based SPC is as follows. First, 25 wafers in a

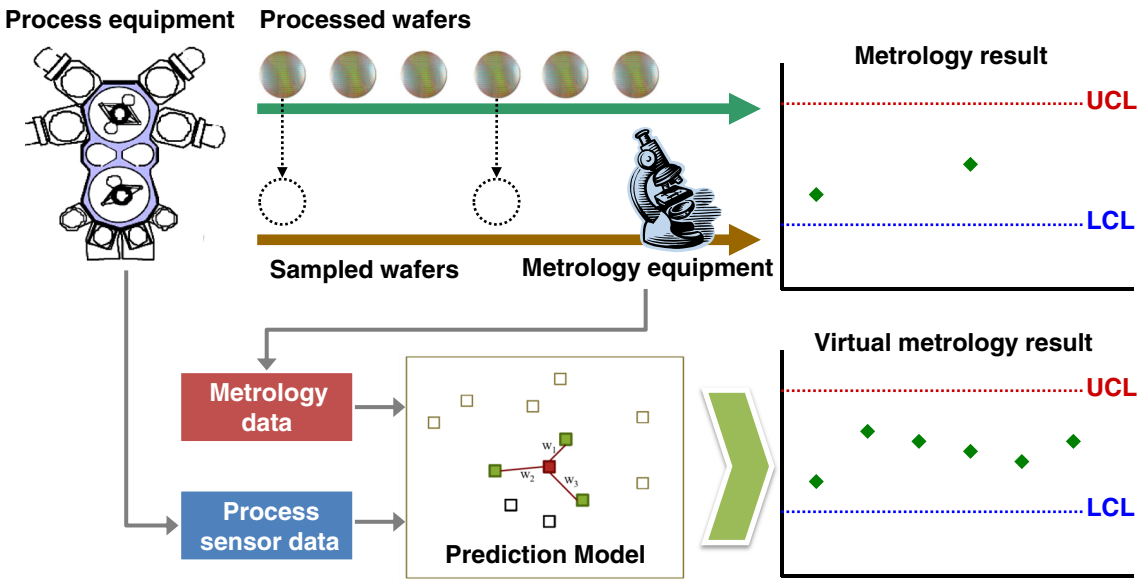


Fig. 1 The conceptual difference between actual metrology (*top*) and virtual metrology (*bottom*)

single unit, called a lot, are processed in an individual piece of operational equipment that is guided by a predefined operational manual called a recipe. Second, in order to check whether the wafers in the lot were processed properly, only one or two wafers are provided as samples to the metrology equipment. This equipment then measures the pre-determined parameters that are considered critical to the yield rate of the final product, such as translation, rotation, and magnification. If all these measurements of the sampled wafer meet the process control criteria, then all the wafers in the same lot are transferred to the next process; if not, they either undergo an additional calibration process or are discarded.

The conceptual difference between actual metrology and virtual metrology is illustrated in Fig. 1. In actual metrology, only a few wafers are sampled when an operating process is completed, and they are provided to the metrology equipment in order to measure quality-related indicators. If the measurements of these indicators are within the control limit, all the wafers in the same lot pass the examination and are transferred to the next operational process. If the measurements are not within the control limit, then, either an additional operation is conducted or the wafers are discarded, depending upon the degree of error. In virtual metrology, on the other hand, a prediction model is built based upon equipment sensor data that is collected during the operation (inputs, predictors, independent variables) as well as actual metrological values (outputs, targets, dependent variables). Because sampled wafers provide both input and output data, the model is trained with these wafers. Once the model is built, the sensor data from the process equipment for every wafer are provided to the model, and its metrological values are predicted in real time without an actual metrology process. If the model can determine the functional relationship between the process sensor data and the metrological values, it becomes possible to obtain metrological values for every wafer in the lot without an actual metrology process.

There are two mainstreams in VM-related research: (1) to develop new prediction algorithms or by selecting (extracting) relevant input variables (predictors) to improve prediction accuracy of virtual metrology systems; and (2) to integrate virtual metrology and R2R control scheme to build a real-time process control system.

With regard to the first VM research direction, Cheng and Cheng [15] employed a 4-layer feed-forward neural network to build a VM prediction model. A total of 2,356 input variables are utilized to predict three metrological values (thickness mean, range, and uniformity) of an advanced 300 mm FAB environment in Taiwan. Despite the complicated network structure, their VM model achieved 1.7 % of maximum error rate and 0.39 % of maximum average projection error (MAPE). Besnard and Toprac [4] built a regression tree based on various types of data such as raw FDC data, preceding metrology measurements, and context information. Before training the regression tree, irrelevant input variables, such as not normally distributed or highly correlated each other, are removed. Then, their VM model achieved an 85 % correlation between actual and predicted metrological values. Lin et al. [41] extracted relevant variables using principal component analysis (PCA), then built a prediction model based upon radial basis function (RBF) networks. Their virtual metrology model achieved a <1 % mean absolute

percentage error (MAPE) in the CVD process environment. Pang et al. [46] showed that a very low MAPE could be achieved by taking into account the effects of different tools in different steps, based upon a combination of clustering techniques and multivariate analysis of variance (MANCOVA). Lynn et al. [42] improved the prediction accuracy of VM models by employing a weighted partial least squares regression to reflect the relative importance of process sensor parameters.

With regard to the second VM research direction. Qin et al. [48] presented a fab-wide R2R control framework by combining fault detection control (FDC) and VM, and highlighted critical issues for the success of the framework, such as updating prediction models and embedding the FDC inside the VM models. Khan et al. [35] tried to improve the VM prediction accuracy as well as R2R control flexibility by designing an R2R framework in which VM models are embedded inside an operational process, and adjacent VM models are connected and exchange information on the processes. In order to integratethe VM models into an R2R control system, statistical or machine learning algorithms are employed. The (multivariate) linear regression is the simplest R2R controller and it was adopted in early studies such as photolithograph overlay control [6] and lithography process [12]. As a non-linear R2R controller, neural networks are most commonly used. It was adopted in various semiconductor processes such as reactive ion etching [40], chemical vapor decomposition (CVD) [61], chemical–mechanical planarization [10, 64], and photolithographic steppers [47].

2.2 Novelty detection

Novel instances or outliers are defined as "observations that deviate so much from other observations as to arouse suspicions that they were generated by a different mechanism" [29]. The purpose of novelty detection is to identify those novel observations that occur rarely among abundant normal instances [33]. For a novelty detection task, two different learning frameworks are available: binary classification and one-class classification. The formal learns both normal and novel classes during the training, whereas the latter generalizes only normal class during the training. The class boundary difference generated by binary classification and one-class classification is illustrated in Fig. 2. Because a small number of crosses are located in the right side, binary classification algorithms divide the data space as shown in Fig. 2a. Assuming that the points A and B are newly given, they are classified as circles. On the other hand, since only circles are used to describe the normal class in one-class classification, the decision boundary becomes a rectangle that envelops the given observations Fig. 2b. In this example, the points A and B are determined as novel. One-class classification is more effective than binary classification under certain circumstances, such as when the class imbalance is severe, or when it is practically impossible to gather data for a certain class. Tax and Duin [59] pointed out that the sample size and class overlap are two main features of one-class datasets so when developing a new classifier, it should be designed to cover these features as wide as possible. Due to its practical importance, a number of one-class classification algorithms have been introduced and they can be grouped into four major categories: (1) distribution-based, (2) clustering-based, (3) distance-based, and (4) support vector-based methods. Distribution-based methods have an assumption that normal observations are drawn from a specific distribution so the main task of algorithms is to estimate its parameters. Gaussian density estimation [3], mixture of Gaussian density estimator [44], and Parzen window density estimator [21] belong to this category. Clustering-based methods relieve the assumption of the shape of distribution in distribution-based methods. In clustering-based methods, normal class is defined as a union of some number of distinctive arbitrary shape of clusters. They can be grouped into three sub-categories: (1) partitional clustering, (2) hierarchical clustering, and (3) density-based clustering. *K*-Means clustering [9] and *K*-medoids clustering [65] are representative partitional clustering algorithms, whereas BIRCH [66], CURE [25], ROCK [26], Chameleon [34], and Z-windows [7] are representative hierarchical clustering algorithms. As density-based clustering methods, DBSCAN [22], OPTICS [2], and LOF [8] are commonly used. Distance-based methods employ nearest neighbor learning for novelty detection. The novelty score of a new observation is proportional to the aggregated distance to its nearest neighbors. Based on the distance measure and the aggregation method, various algorithms can be possible [1, 27, 33, 37, 49]. Support vector-based methods generate an arbitrary shape of closed class boundary that can describe the normal class well in the input space by mapping the data into a higher dimensional feature space to achieve a better generalization ability. The one-class support vector machine (1-SVM) [52] and support vector data description (SVDD) [57] are two well-known support vector-based algorithms. The former finds the farthest hyperplane from the origin, above which as many normal observations are placed as possible, whereas the latter finds the most compact hypersphere that envelops as many normal observations as possible. It has been proved that 1-SVM and SVDD produce the same class boundary when a Gaussian kernel

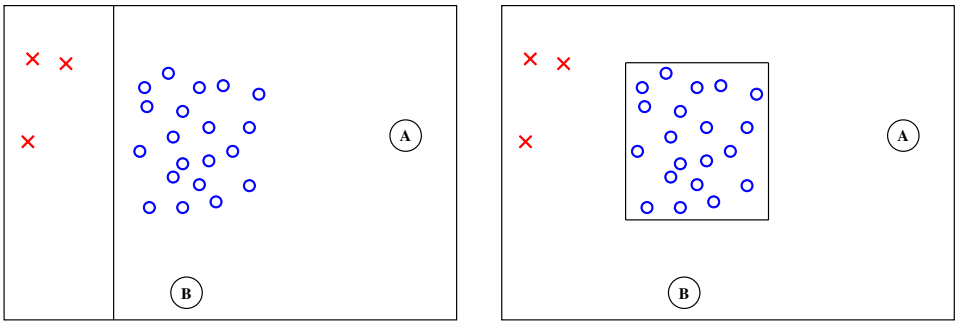


Fig. 2 The classification boundary of binary classification (**a**) and one-class classification (**b**)

(a) Binary classification **(b)** Novelty detection

function is used [57]. Due to their high generalization ability, support vector-based novelty detection algorithms have been successfully applied to various practical domains such as image classification [20, 38] and chemical process monitoring [31].

Rather than single novelty detection algorithms, an ensemble of one-class classification algorithms has been highlighted as a means of improving the detection performance. Krawczyk and Wozniak [39] proposed five diversity measures for selecting effective committee members. Based on the empirical study with a large number of datasets, the entropy-based measure returned the best performances, followed by the sphere intersection measure and the energy measure. Krawczyk and Filipczuk [38] proposed an efficient medical decision support framework for breast cancer diagnosis. In their work, the entire dataset is decomposed to one of the three classes and novelty detection algorithm is applied to each class. In order to improve the detection performance, an ensemble of one-class classification algorithms is constructed for each class. Cyganek [20] and Yeh et al. [63] attempted to construct one-class support vector ensembles; the former divided the training data into some number of homogeneous clusters in the feature space and applied a 1-SVM in each cluster, whereas the latter adopted the AdaBoost framework [23]. Wilk and Wozniak [62] extend the binary classification into multi-class classification by employing a fuzzy inference system with a set of one-class classifiers. Their experimental results show that the fuzzy combiner yields consistently lower error rates than other combination methods.

In this study, since we assume that all training wafers are homogeneous, we only have examples of a normal class, so the one-class classification-based novelty detection scheme is more suitable than binary classification for the assignment of a reliability level of VM prediction results. In addition, we combine a set of one-class classifiers to improve the stability of reliability level produced by individual novelty detectors.

3 Reliability evaluation of virtual metrology prediction results

The conceptual structure of our reliability evaluation system for VM, which is illustrated in Fig. 3, differs from a traditional VM system in the following ways. In a traditional VM system, a prediction model is trained based on the process sensor data and the actual metrological values of wafers that are inspected by actual metrology equipment. When an operation on a wafer is completed, its process sensor data are provided to the VM model for prediction of its metrological values. In our reliability evaluation system, however, a novelty detection model is also built, in addition to the VM model, and it is based only on the process sensor data of the training wafers. When an operation on a new wafer is completed, process sensor data are provided to the VM model and the novelty detection model at the same time, in order to predict the metrological values, and the similarity between the sensor data of the new wafer and those of the training wafers, respectively. If the novelty detector determines that the process sensor data of a new wafer is similar enough to those of the training wafers, the new wafer is considered to be drawn from the same underlying distribution as the training wafers, and a high-reliability score is therefore assigned to its VM prediction results. If the degree of similarity is insufficient, the new wafer is considered to be drawn from an underlying distribution that is different from that of the training wafers, and a low-reliability score is then assigned.

In order to build the reliability evaluation system for VM, two types of prediction models are necessary: a regression model for VM and a novelty detection model for reliability evaluation. Regression models are used in the generation of continuous outcomes by configuring the functional relationships between predictors, either discrete or continuous, and targets. Novelty detection models are associated with the generation of binary outcomes (0 or 1) produced by generalizing given data that consist of only predictors. In order to explore the effects and consequence

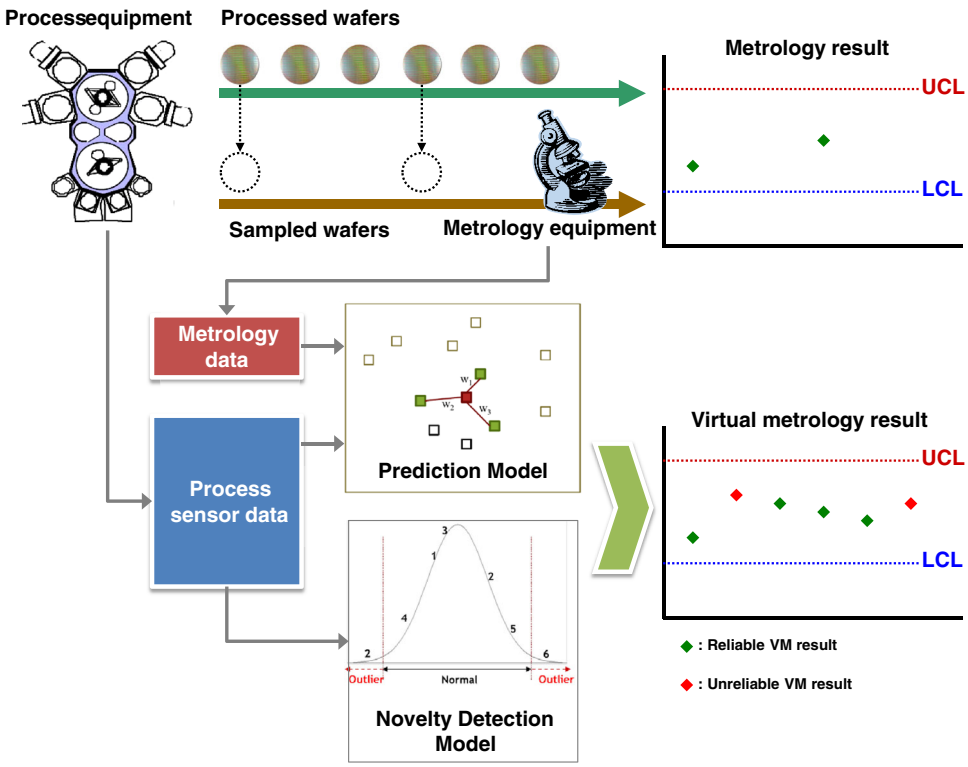


Fig. 3 The conceptual structure of our reliability evaluation system for VM

of reliability evaluation, we employed three regression algorithms for VM prediction and five novelty detection algorithms for reliability evaluation. In the next subsections, we briefly introduce the regression and novelty detection algorithms adopted in our experiments.

3.1 Virtual metrology models

Three regression algorithms were employed for VM prediction in our experiments: multiple linear regression (MLR), *k*-nearest neighbor (*k*-NN) regression, and artificial neural networks (ANN). MLR [50] estimates the functional relationship between multiple input variables and single or multiple target variables of given data in the form of linear equation. Compared to other complex algorithms, MLR offers a number of advantages such as a closed analytic form, computational efficiency, and less user-specific parameters. However, its performance is degraded when there is a non-linear relationship between the predictors and targets.

Let y_{ki} denote the ith metrological value of the kth wafer, while x_{kj} denotes the jth process sensor data of the kth wafer. Then, the MLR equation with p predictors, d targets, and n training wafers can be written as:

$$y_{ki} = \beta_{k0} + \beta_{k1}x_{i1} + \beta_{k2}x_{i2} + \cdots + \beta_{kp}x_{ip}, \quad \text{for } k = 1, 2, \ldots, d, \quad i = 1, 2, \ldots, n. \tag{1}$$

This can be rewritten in a matrix form as:

$$\mathbf{Y} = \mathbf{X}\beta,$$
$$\mathbf{Y} = \begin{pmatrix} y_{11} & \cdots & y_{1d} \\ \vdots & \ddots & \vdots \\ y_{n1} & \cdots & y_{nd} \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_{10} & \cdots & \beta_{d0} \\ \vdots & \ddots & \vdots \\ \beta_{1p} & \cdots & \beta_{dp} \end{pmatrix}. \tag{2}$$

The intercept β of the above equation can be obtained by minimizing the squared error (residual) between the targets ($\mathbf{Y}$) and the predictions ($\hat{\mathbf{Y}}$), as shown in Eq. (3) using the ordinary least square (OLS) method as follows:

$$E = \frac{1}{2}\sum_{i=1}^{n} e_i^2 = \frac{1}{2}\det\left|(\mathbf{Y} - \hat{\mathbf{Y}})^{\mathrm{T}}(\mathbf{Y} - \hat{\mathbf{Y}})\right| = \frac{1}{2}\det\left|(\mathbf{Y} - \mathbf{X}\beta)^{\mathrm{T}}(\mathbf{Y} - \mathbf{X}\beta)\right|. \tag{3}$$

$$\frac{\partial E}{\partial \boldsymbol{\beta}} = \mathbf{X}^{\mathrm{T}}\mathbf{Y} - \mathbf{X}^{\mathrm{T}}\mathbf{X}\beta = 0, \quad \boldsymbol{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}. \tag{4}$$

ANN [5] is one of the most widely used non-parametric regression algorithms in many fields, including that of virtual metrology, due to its ability to capture non-linear relationships between predictors and targets. A 3-layer feed-forward neural network was employed in our experiments. In ANN, the targets are expressed as a combination of input values and weights as follows:

$$y_k = \sum_{q=1}^{h} w_{kq}^{(2)} g\left(\sum_{r=1}^{d} w_{qr}^{(1)} x_r\right), \quad k = 1, 2, \ldots, d, \tag{5}$$

where $w_{kq}^{(2)}$, $w_{qr}^{(1)}$, and g(•) denote the weight connected between the kth output node and the qth hidden node, the weight connection between the qth hidden node and the rth input node, and the activation function, respectively. Training ANN is equivalent to optimizing the weights in Eq. (5), which are obtained by minimizing the objective loss function, which is generally done using the least squared residual in Eq. (3).

k-NN [28] is the most popular memory-based learning algorithm. Since it does not require a training procedure, it is employed in a number of tasks that require rapid model update. k-NN predicts the target values of a new instance based on the similarity information between the new instance and its neighbor instances. Once a new instance is provided, k-NN first searches the k most similar instances in the reference data set. Next, the weight for each selected neighbor instance is assigned; the greater the similarity, the greater the weight. The target values of the selected neighbors are then aggregated using a predefined combining rule to produce the target value of the new instance:

$$\hat{y} = \sum_{j \in \mathrm{NN}(x)} w_j y_j, \tag{6}$$

where NN(x) and w_j denote the index set of k-nearest neighbors of the new instance x, and the weight assigned to the jth nearest neighbor, respectively. In k-NN learning, two user-specific parameters must be declared: the number of nearest neighbors (k), and the weight allocation method. Here, we adopted the locally linear reconstruction (LLR) method [32], due to its ability to determine the two parameters in a structured way, unlike other heuristic-based approaches. LLR finds the optimal weights for the nearest neighbors by minimizing the reconstruction error $E(\mathbf{w})$ between the target instance and the projection made by its neighbors, which is defined as follows,

$$E(\mathbf{w}) = \frac{1}{2}\left|\mathbf{x}_t - \sum_{j=1}^{k} \mathbf{w}_j \tilde{\mathbf{x}}_j\right|^2, \tag{7}$$

where $\mathbf{x}_t$, $\tilde{\mathbf{x}}_j$, and $\mathbf{w}_j$ are the target instance, jth nearest neighbor of $\mathbf{x}_t$, and the weight assigned to $\tilde{\mathbf{x}}_j$. By solving this quadratic programming, LLR can find the optimal set of weights systematically.

3.2 Novelty detection (one-class classification) algorithms

In order to assign a level of reliability to a wafer's VM prediction results, we performed an evaluation to compare the homogeneity of the process sensor data for a new wafer with that of the training wafers based on novelty detection techniques. Once a set of instances is provided, novelty detection algorithms characterize and generalize the data, assuming that they are drawn from the same underlying distribution. When a new instance is provided, its novelty score is computed. It is determined as being novel if the novelty score is greater than the given threshold; if not, it is considered normal.

A total of five novelty detection algorithms were employed: a Gaussian density estimator (Gauss), a mixture of Gaussians (MoG), KMC, k-nearest neighbor (k-NN), and SVDD. Gauss [3] is the simplest parametric novelty detection method. It assumes that normal data is generated from a Gaussian distribution, as shown in Eq. (8).

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{2/d}|\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]. \tag{8}$$

When a set of training instances are given, Gauss estimates its two model parameters, $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, which are the mean vector and the covariance matrix of the normal training data, respectively. Then, whenever a new instance is provided, its probability is computed using Eq. (8) with the estimated parameters ($\boldsymbol{\mu}$, $\mathbf{\Sigma}$). If the probability is high enough, the new instance is considered to be from the same distribution as the training data, so it is given a high-reliability score. If the probability is low, the new data are not considered to be from the same distribution, and is given a low-reliability score.

Gauss requires a very strict assumption of unimodality, which is often violated in practice. To obtain a more flexible density estimate, MoG [44] allows more than one modal, and the probability is estimated by a linear combination of K individual distribution components as follows:

$$p(\mathbf{x}) = \sum_{k=1}^{K} P(k) p_k(\mathbf{x}). \tag{9}$$

where K, $P(k)$, and $p_k(\mathbf{x})$ are the number of components in a mixture model, the prior probability of the kth component, and the conditional probability of $\mathbf{x}$ for the kth component, respectively. When a new instance is provided, the probability is computed, and it is determined as being normal only if the probability is high enough. In MoG, each component is assumed to be a Gaussian distribution, and the parameters of each Gaussian are optimized by an expectation–maximization algorithm [5].

KMC [57] is similar to MoG in that it groups the normal data into K clusters, where instances within the same cluster are homogeneous, while those in different clusters are heterogeneous. However, KMC does not require a Gaussian assumption for each cluster. With a given normal

data set, KMC finds K centroids that minimize the within-cluster sum of the squared error,

$$\arg \min_{C} \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathrm{c}_i||^2, \tag{10}$$

where c_i is the centroid of C_i, and C is the union of all clusters ($C = C_1 \cup \ldots \cup C_K$). When a new instance $\mathbf{x}_\mathrm{n}$ is provided, its novelty score is determined based on the distance to the nearest cluster, as follows:

$$\begin{aligned} &\text{Novelty score } (\mathbf{x}_n) = ||\mathbf{x}_n - \mathbf{c}_i||, \text{ where} \\ &||\mathbf{x}_n - \mathbf{c}_i|| \le ||\mathbf{x}_n - \mathbf{c}_j||, \text{ for all } k,\ i \neq j. \end{aligned} \tag{11}$$

In k-nearest neighbor learning, when a new instance is provided, its k most similar instances are selected based on a certain similarity metric, such as the Euclidean distance. Then, the novelty score is computed by aggregating this similarity information. Among various similarity combination methods, we adopted a hybrid novelty score [33], due to its ability to consider distance and local topology simultaneously, which are computed as follows:

$$d_{\text{hybrid}}(\mathbf{x}) = d_{\text{avg}}(\mathbf{x}) \times \left(\frac{2}{1 + \exp(-d_{\text{c-hull}}(\mathbf{x}))} \right), \tag{12}$$

where d_{avg} is the average distance to the k-nearest neighbors, and $d_{\text{c-hull}}$ is the distance to the convex hull made by the neighbors as shown in Eq. (13).

$$d_{\text{avg}}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{k} ||\mathbf{x} - \mathbf{x}^i||, \quad d_{c-\text{hull}(\mathbf{x})} = ||\mathbf{x} - \sum_{i=1}^{k} w_i \mathbf{x}^i||, \tag{13}$$

where $\mathbf{x}_\mathrm{i}$ is the ith nearest neighbor, and w_i is its corresponding weight obtained by solving LLR.

SVDD [57, 58] is a novelty detection algorithm that is based on structural risk minimization [60], and it solves a problem in feature space using a kernel trick [45, 53]. SVDD finds a hypersphere with a minimum volume that encloses as many normal instances as possible in the feature space. Let R and $\mathbf{a}$ denote the radius and the center of the hypersphere, respectively, in an optimization problem to be solved that is stated as:

$$\begin{aligned} &\min R^2 + C \sum_{i=1}^{n} \xi_i, \\ &s.t. \quad ||\Phi(\mathbf{x}_i) - \mathbf{a}||^2 \le R^2 + \xi_i, \quad \xi_i \ge 0, \ \forall \mathbf{x}_i, \end{aligned} \tag{14}$$

where $\Phi(\mathbf{x}_i)$ and $\mathbf{a}$ are a transformed input data and the center of the normal class instances in the feature space, respectively. The solution can be found by formulating it as a Wolfe's dual problem and utilizing a kernel trick. When a new instance $\mathbf{x}_n$ is provided, its novelty score can be measured as follows:

$$\text{Novelty score } (\mathbf{x}_n) = R^2 - ||\Phi(\mathbf{x}_i) - \mathbf{a}||^2. \tag{15}$$

As an attempt to improve the stability of reliance level obtained by individual novelty detection models, we construct a fusion model of novelty detectors. Since the main purpose of this study is to verify the practical applicability of novelty detection algorithms as a reliability indicator for VM prediction results, we adopted a simple majority voting scheme for aggregating the novelty detection algorithms rather than sophisticated methods discussed in Sect. 2.2.

$$\mathrm{NI}_{\text{Fusion}}(\mathbf{x}_n) = \delta \left(\sum_{j=1}^{p} \mathrm{NI}_j(\mathbf{x}_n) > \frac{p}{2} \right), \tag{16}$$

where p is the number of individual novelty detectors ($p = 5$ in our experiment). $\mathrm{NI}_{\text{Fusion}}$ and NI_j denote the novelty indicator of the fusion and jth individual novelty detector that returns 1 if $\mathbf{x}_n$ is determined as novel or returns 0 if it is determined as normal. δ is an indicator function that return 1 if the condition in the parenthesis is met, otherwise return 0.

4 Experimental settings

4.1 Data

In order to analyze the effect of the proposed reliability evaluation models, at an actual semiconductor manufacturing company in South Korea, we collected the data from 117 process sensors in two pieces of photo-lithography equipments as inputs, and eight metrological values as outputs. Since preventive maintenance (PM) was performed seven times during the data collection, we divided the entire data into eight segmented periods, using the occasions of PM as the points of separation. The number of wafers collected in each period for each piece of equipment is summarized in Table 1. The first 100 wafers in each period were used for training the VM prediction models and novelty detection algorithms (including cross-validation for selecting algorithm parameters and variable selection), and the remaining wafers were used for performance evaluation.

Table 1 The number of wafers collected in each period for each equipment

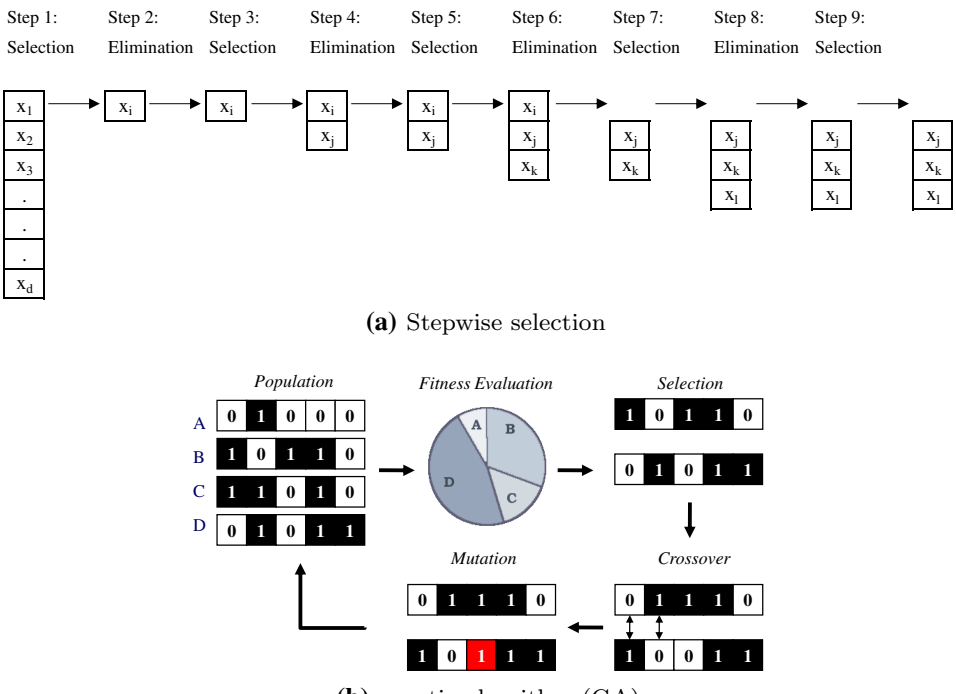
Equipment no.	Prd. 1	Prd. 2	Prd. 3	Prd. 4	Prd. 5	Prd. 6	Prd. 7	Prd. 8
EQ1	230	172	137	167	452	818	138	195
EQ2	226	180	136	170	450	816	138	195

4.2 Variable selection

In our experiments, a total of 117 input variables (process sensor parameters) were collected. Not only was this too many compared to the number of training wafers, but a number of irrelevant variables were also included in the raw data set. Therefore, we reduced the dimensions of the input in order to improve the prediction performance and the model training efficiency. We adopted stepwise variable selection and a genetic algorithm (GA) in order to select the most relevant variables. Stepwise variable selection process begins with the single most relevant input variable, and the following two procedures are conducted alternately until every significant variable is included: (1) among the candidates, one that most improves the prediction accuracy is added (selection); (2) and among the selected variables, one that is most irrelevant to improve the prediction accuracy is removed (elimination). Note that it is not necessary to remove a variable in the elimination step. A selected variable is removed again if and only if the prediction performance can be maintained without it. Figure 4a illustrates an example of stepwise variable selection. In steps 2 and 4, no variable is eliminated because there is no prediction performance improvement. However, variable x_i is removed in step 6 since the prediction performance is enhanced when it is excluded from the selected variable set. In step 9, when there are no variables to add, the stepwise variable selection is finalized. Although the stepwise variable selection can rapidly converge to a subset of significant variables, it is usually not an optimal subset when a large number of input variables are considered. In this circumstance, GA can be a better alternative. GA finds the optimal set of input variables based upon an evolutionary procedures such as selection, crossover, and mutation [19, 30]. Figure 4b illustrates the process of GA variable selection. Initially, a sufficient number of chromosomes, called a population, are created. Each chromosome has the form of binary vector where each element, called a gene, designates the usage of the corresponding input variable: 1 for used, 0 for not used. Next, VM models and novelty detection algorithms are trained with the candidate variables in each chromosome and its fitness value is evaluated. Since the purpose of our study is to discriminate the normal and novel wafers well in a VM process for flexible process control, it is more desirable when the difference of VM prediction errors between highly reliable wafers and unreliable wafers is maximized. Thus, we define the fitness function of the GA as follows,

$$\text{Fitness functions} = \text{MAE}(W_L) - \text{MAE}(W_H), \tag{17}$$

where MAE is the mean absolute error (MAE) that is defined as Eq. (18), while W_L and W_H denote a set of wafers that are classified as novel (low reliability) and normal (high reliability), respectively. Chromosomes with high fitness values survive and generate a new population by imitating biological reproductive processes such as crossover and mutation. Crossover is associated with exchanging some genes between two chromosomes, whereas mutation is associated with reversing the value of certain genes (ex: from 0 to 1) with a low probability. In



(a) Stepwise selection

(b) genetic algorithm (GA)

Fig. 4 Variable selection based on the stepwise selection and genetic algorithm (GA)

Table 2 The number of selected input variables

Prd.	EQ1						EQ2					
	Stepwise			GA			Stepwise			GA		
	MLR	*k*-NN	ANN	MLR	*k*-NN	ANN	MLR	*k*-NN	ANN	MLR	*k*-NN	ANN
1	10	6	9	38	23	15	13	6	6	17	13	16
2	12	6	7	31	23	19	14	4	7	24	18	20
3	7	2	6	38	21	8	8	3	6	26	18	17
4	11	5	6	27	18	22	13	6	6	25	19	15
5	8	6	6	28	11	15	12	4	7	34	14	10
6	9	3	6	18	13	11	4	5	7	12	8	5
7	8	5	6	22	18	17	4	4	7	15	20	21
8	8	5	6	16	18	15	14	4	6	37	21	18

doing so, input variables with high prediction performance are kept throughout the generation process, while those with low performance naturally die out. Once this cycle (selection, crossover, and mutation) is repeated a sufficient number of times, we can identify a pseudo-optimal set of variables.

The number of selected input variables in each period for each VM prediction model is summarized in Table 2. It was observed that there was significant redundancy among the process sensor parameters. At most, 38 input variables were selected for EQ1's first and third periods by GA with MLR, which still represented a 67 % reduction of the original variables. In an extreme case, only two input variables were selected for EQ1's third period by stepwise selection with *k*-NN. We would note that, regardless of the prediction algorithm used, fewer input variables were selected with the stepwise selection than by GA for both pieces of equipment; the number of input variables selected by GA for the same equipment/period/prediction algorithm pair was more than twice the number obtained by stepwise selection. The reason for this is that GA has a larger coverage of the search space than stepwise selection, so an individual variable has a greater chance of being considered for selection.

4.3 Algorithm parameters and performance measures

In our experiments, three regression algorithms (MLR, *k*-NN, ANN) were employed for VM prediction, and five novelty detection algorithms (Gauss, MoG, KMC, *k*-NN, SVDD) were employed for reliability evaluation. Besides MLR and Gauss, each of the adopted algorithms required that algorithm-specific parameters be determined. The parameters for each algorithm and their candidate values are summarized in Table 3. *k*-NN regression and *k*-NN novelty detection requires the number of nearest neighbors (k), while MoG and KMC require the number of clusters K. H is the number of hidden nodes for ANN. With the Gaussian kernel, two parameters must be optimized for SVDD: the width of the Gaussian kernel (σ) and the cost of the errors (C). Note that although SVDD can take other form of kernels such as linear kernel or polynomial kernel, we adopted the Gaussian kernel since it has been most commonly adopted and reported better performance for practical use [36, 43, 67]. These algorithm parameters are optimized by fivefold cross-validation process using the training dataset. Initially, a set of parameters for regression algorithms and novelty detectors are fixed. Then, the variable selection is conducted with these fixed parameters. As a result, the best variable sets are obtained for each set of algorithm parameters. Finally, the best parameter–variable set combination is determined using the same fitness function used in the GA.

Table 3 The algorithm-specific parameters for each algorithm and the candidate values

Purpose	Algorithm	Parameter	Candidates
Regression	*k*-NN	k	[1,2, …, 10,15,20]
	ANN	H	[1,2,…,10,15,20]
Novelty detection	MoG	K	[1, 3, 5, 7,10]
	KMC	K	[1, 3, 5, 7,10]
	k-NN	k	[1,2,…,10]
	SVDD	σ	$[2^{-5},2^{-4},\ldots,2^{4},2^{5}]$
		C	$[2^{-5},2^{-4},\ldots,2^{4},2^{5}]$

When a new wafer is provided, a binary reliability outcome (low or high) for VM prediction is determined by the novelty detection model as follows. Once a novelty detection model is trained with the same wafers used for building a VM prediction model, the novelty scores of the training wafers are computed and sorted in descending order. Next, the 5 percentile value is set at a cut-off value (threshold). If the novelty score of the new wafer is higher than the threshold, its VM prediction results are labeled as low; if not, its prediction results are labeled as high. This

means that if the process sensor data of a new wafer are similar to more than 95 % of training wafers, the training wafers and the new wafer are considered homogeneous, and the VM prediction results of the new wafer are considered highly reliable because the new wafer's sensor data were sufficiently learned by the VM model. In the opposite case, the training wafers and the new wafer are considered heterogeneous, so the VM prediction results of the new wafer are considered unreliable because the VM model did not have enough learning opportunities.

Based on the evaluated reliability level, the performance of VM is analyzed in terms of two accuracy measures: the MAE and the percentage of absolute range error (PARE). MAE is based on computing the absolute difference between the actual and the predicted metrology values as.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{18}$$

where n is the total number of test wafers, and y_i and $\hat{y}_i$ are the actual and predicted metrological values, respectively, of the ith wafer. Since the scale of actual metrological values is very small, i.e., $<10^{-2}$, we used an adjusted MAE by multiplying Eq. (18) by 100. PARE is defined as the proportion of wafers whose prediction error is within the level of tolerance, and is computed as:

$$\text{PARE} = \frac{1}{n}\sum_{i=1}^{n} I(|y_i - \hat{y}_i| < \theta), \tag{19}$$

where $I(\bullet)$ is an indication function that returns 1 if the condition in the parenthesis is satisfied, and returns 0 if it is not. θ is the tolerance level determined by the process recipe. In our experiments, θ was set to 0.003 because the same value was used in the actual manufacturing process.

5 Experimental results

Note again that we use the term W_H to refer to wafers whose reliability level is labeled high and W_L for wafers whose reliability level is labeled low. The summary statistics of the proportions of W_L, as determined by individual novelty detection algorithms and their fusion model, are summarized in Table 4, and their distributions are shown in Fig. 5. We would note that there were a total of 48 cases for each novelty detection algorithm for each piece of equipment: 8 periods × 3 VM prediction models × 2 variable selection methods. Although we used the same cut-off threshold for both the high and low reliability levels, i.e., the top 5 % novelty score of the training wafers, the distributions of the proportions of WL were quite different, depending upon the novelty detection algorithm. It

Table 4 The summary statistics of the proportion of W_L

Algorithm	EQ1				EQ2			
	Mean	Median	SD	Q3–Q1	Mean	Median	SD	Q3–Q1
Gauss	0.0589	0.0270	0.0745	0.077	0.0431	0.0264	0.0524	0.043
MoG	0.0711	0.0597	0.0523	0.0751	0.0621	0.0526	0.0511	0.0548
KMC	0.0282	0.0243	0.0259	0.0395	0.0212	0.0148	0.0209	0.0337
k-NN	0.0546	0.0443	0.0486	0.0619	0.0480	0.0296	0.042	0.0450
SVDD	0.1482	0.1187	0.0933	0.1447	0.1247	0.0947	0.0875	0.1172
Fusion	0.0434	0.0327	0.0380	0.5260	0.0309	0.0250	0.0275	0.0361

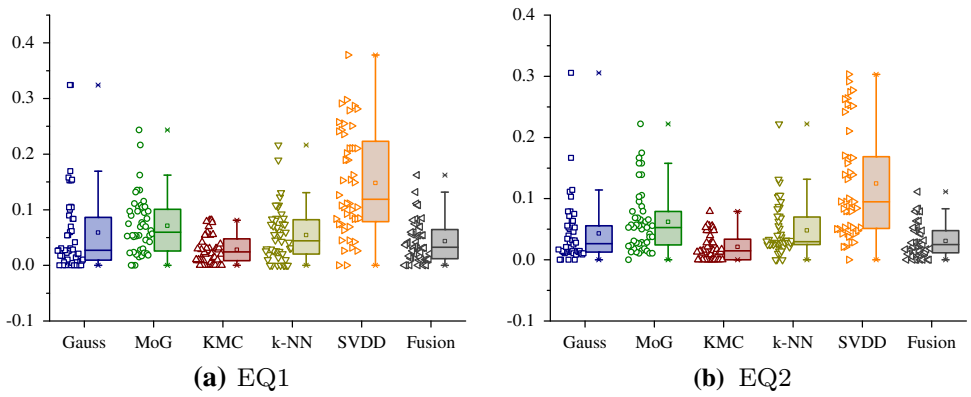


Fig. 5 The proportion of W_L determined by each novelty detection algorithm

was observed that KMC assigned a low level of reliability to the wafers most strictly, as only 2:82 % (EQ1) and 2:12 % (EQ2) of the test wafers were labeled as low, on average. In addition, the variation of the proportions obtained using KMC was the smallest among the novelty detection algorithms. Gauss and k-NN assigned low reliability to 4–6 % of the test wafers on average, and these proportions were consistent with the threshold setting, i.e., 5 %. In general, MoG and SVDD were found to over-fit the training data, because they assigned low reliability to more test wafers than was expected, and the variations were also large. MoG assigned low reliability to 7:11 % (EQ1) and 6:21 % (EQ2) of the test wafers, on average. The discrepancy between the proportions of W_L in the training data versus the test data was the greatest when SVDD was used. More than 10 % of the test wafers were determined as W_L, and with the highest degree of variation, as shown in Fig. 5. When it comes to the Fusion model, it assigned low reliability to the wafers slightly lower than the threshold; 4:34 % and 3:09 % of the test wafers are determined as unreliable for EQ1 and EQ2, respectively. Based on these results, we are able to make an immediate suggestion regarding the adoption of a novelty detection algorithm. If process engineers require strict process control, it would be better to use a novelty detection algorithm that sounds an alarm frequently, such as SVDD. If they require a lesser degree of control, then a novelty detection algorithm that seldom assigns low reliability to wafers, such as KMC, should be employed.

The VM prediction performance according to the reliability level, in terms of the adjusted MAE, is summarized in Table 5. Theoretically, 64 MAEs for both W_H and W_L can be obtained for each VM model-novelty detector algorithm pair, because there were two pieces of equipment, eight periods, and four targets. However, for a certain equipment–period–target variable cases, all wafers resulted in high reliability level so that the MAE(W_L) cannot be obtained. We discarded those cases when computing the statistics for W_L. In addition, Table 6 shows the proportion of the equipment-period-target variable pairs that resulted in lower adjusted MAEs for WH than those for WL. If this proportion is large, we can conclude that the novelty detection algorithm is effective, because the wafers with high reliability resulted in smaller errors than those with low reliability. In other words, the greater the proportion, the more effective the novelty detection algorithm.

Based on Tables 5 and 6, the following observations can be made. First, W_L resulted in a much higher MAE than W_L, regardless of the VM models, novelty detection algorithms, or variable selection methods. On average, the MAE of W_L was more than 50 % higher than that of W_H for the same VM model/variable selection/novelty detection pair, with just a few exceptions. We would note that there were a few extremely high adjusted MAEs for W_L, which would make the average adjusted MAEs biased. In terms of the median value of the adjusted MAEs, however, W_L still resulted in more than 20 % higher MAEs than W_H for most pairs. This finding supports our hypothesis that a VM prediction result would not be reliable when a test wafer's input data and those of the training wafers are heterogeneous.

Second, among the variable selection methods, the average adjusted MAEs for W_H and W_L is generally lower with the stepwise selection when MLR and ANN are employed as VM models, whereas the average adjusted MAEs for W_H are not significantly different, but those for W_L is much lower with GA selection than stepwise selection when k-NN is employed. Therefore, the proportions in Table 6 are greater with GA than with stepwise selection for MLR and ANN for most cases, whereas stepwise selection and GA selection resulted in higher proportions for three cases, respectively. We would note that in general, the input variables selected by GA selection outnumbered those obtained by stepwise selection. Looking at the adjusted MAEs of the training data, we see that the error rates with GA selection were lower than those with stepwise selection for all VM prediction models. However, their levels of performance with the test data were reversed with an only exception of k-NN for WL. A possible explanation for this is that because GA selection takes a broader search space into account than stepwise selection, GA selection brings a higher risk of over-fitting, in practice.

Third, among the VM prediction models, MLR and k-NN resulted in a similar level of performance in terms of the adjusted MAE for W_H and the adjusted MAE difference between W_H and W_L, while ANN was not as accurate as the others. However, when we look at the adjusted increase in MAE shown in Table 6, MLR and ANN were more effective than k-NN. We would note that in an ideal situation, the value in each cell in Table 6 should be 1, because a good VM model with a proper novelty detection algorithm always makes more accurate predictions for W_H than W_L. However, for some equipment/period/target pairs, only a few wafers, such as a number of less than five, were identified as W_L, and some of them were false alarms in practice. In such cases, the MAE of W_L could be smaller than W_H. Although k-NN made fairly accurate predictions for W_H, it failed to distinguish W_L from W_H. ANN, on the other hand, succeeded in distinguishing W_L from W_H, but its MAE for W_H is lower than that of MLR. Overall, when considering both the MAE for W_H and the adjusted increase in MAE, MLR was found to be the best model using our experimental settings. However, we should recall that only 100 training wafers were used in our experiment, due to the difficulty of acquiring actual data. If a sufficient number of

Table 5 The summary statistics of the adjusted MAE with respect to W_H and W_L

Variable selection			Stepwise				GA			
VM	ND	Reliability	Mean	Median	SD	Q3–Q1	Mean	Median	SD	Q3–Q1
MLR	Gauss	W_H	0.1270*	0.1220	0.0271	0.0300	0.1337*	0.1284	0.0294	0.0271
		W_L	0.3229	0.1866	0.4691	0.1644	0.3826	0.2409	0.4706	0.2459
	MoG	W_H	0.1259*	0.1231	0.0262	0.0319	0.1348*	0.1284	0.0339	0.0241
		W_L	0.2205	0.1704	0.1973	0.1081	0.4069	0.2075	0.6270	0.2110
	KMC	W_H	0.1281*	0.1238	0.0275	0.0318	0.1406*	0.1330	0.0460	0.0266
		W_L	0.2660	0.1867	0.2213	0.1518	0.5535	0.2658	0.8426	0.3510
	k-NN	W_H	0.1264*	0.1240	0.0267	0.0311	0.1376*	0.1309	0.0452	0.0282
		W_L	0.2376	0.1796	0.1728	0.1414	0.5078	0.2426	0.8511	0.2857
	SVDD	W_H	0.1255*	0.1218	0.0264	0.0298	0.1371*	0.1308	0.0418	0.0271
		W_L	0.1884	0.1581	0.1033	0.0841	0.2937	0.1814	0.4550	0.1129
	Fusion	W_H	0.1277*	0.1235	0.0275	0.0319	0.1376*	0.1314	0.0418	0.0266
		W_L	0.2709	0.2025	0.2250	0.1633	0.4140	0.2169	0.6219	0.2944
k-NN	Gauss	W_H	0.1254*	0.1244	0.0258	0.0323	0.1242*	0.1283	0.0261	0.0359
		W_L	0.2311	0.1578	0.3548	0.0857	0.1544	0.1450	0.0715	0.0883
	MoG	W_H	0.1254*	0.1240	0.0270	0.0315	0.1241*	0.1263	0.0261	0.0347
		W_L	0.1993	0.1500	0.2933	0.0638	0.1653	0.1556	0.0792	0.0915
	KMC	W_H	0.1265*	0.1241	0.0269	0.0306	0.1261*	0.1294	0.0268	0.0337
		W_L	0.1825	0.1490	0.1615	0.0806	0.1547	0.1348	0.0919	0.0748
	k-NN	W_H	0.1246*	0.1216	0.0268	0.0313	0.1248*	0.1277	0.0265	0.0337
		W_L	0.1950	0.1627	0.1756	0.0755	0.1740	0.1587	0.0814	0.0761
	SVDD	W_H	0.1268*	0.1245	0.0291	0.0347	0.1258*	0.1266	0.0276	0.038
		W_L	0.1435	0.1302	0.0817	0.0514	0.1351	0.1313	0.0418	0.0424
	Fusion	W_H	0.1261*	0.1245	0.0284	0.0332	0.1251*	0.1275	0.0265	0.0331
		W_L	0.1919	0.1588	0.1861	0.0813	0.1612	0.1443	0.0822	0.0835
ANN	Gauss	W_H	0.1311*	0.1283	0.0263	0.0308	0.1343*	0.1293	0.0303	0.0317
		W_L	0.3028	0.1876	0.3178	0.1776	0.2954	0.1879	0.5418	0.1495
	MoG	W_H	0.1302*	0.1277	0.0267	0.032	0.1339*	0.1297	0.0324	0.0344
		W_L	0.2213	0.1771	0.1989	0.1209	0.2354	0.1730	0.3808	0.076
	KMC	W_H	0.1336*	0.1280	0.0346	0.0303	0.1355*	0.1298	0.0316	0.0339
		W_L	0.2645	0.1737	0.2626	0.1881	0.3452	0.1912	0.8132	0.1835
	k-NN	W_H	0.1326*	0.1270	0.0343	0.0315	0.1341*	0.1293	0.0312	0.0346
		W_L	0.2356	0.1735	0.1707	0.143	0.3530	0.1831	1.0109	0.0687
	SVDD	W_H	0.1317*	0.1270	0.0283	0.0336	0.1329*	0.1269	0.0326	0.0380
		W_L	0.1888	0.1470	0.1942	0.0646	0.2540	0.1650	0.5103	0.0768
	Fusion	W_H	0.1319*	0.1280	0.0274	0.0314	0.1346*	0.1295	0.0306	0.0327
		W_L	0.2944	0.1795	0.4195	0.1706	0.3292	0.1908	0.8145	0.1200

An asterisk (*) denotes that the average adjusted MAE of W_L is greater than that of W_H at the significant level of 0.05

Table 6 The proportion of equipment–period–target pairs where the adjusted MAE increase is greater than 0

VM model	Variable selection	Gauss	MoG	KMC	*k*-NN	SVDD	Fusion
MLR	Stepwise	0.8500	0.9063	0.7500	0.9063	0.8000	0.9063
	GA	0.9375	0.8750	0.8750	0.9583	0.8167	0.9843
k-NN	Stepwise	0.6786	0.7292	0.7000	0.7250	0.5000	0.7500
	GA	0.6250	0.7000	0.5000	0.7857	0.5385	0.8281
ANN	Stepwise	0.7500	0.7000	0.9000	0.8750	0.6000	0.9063
	GA	0.9063	0.8958	0.8333	1.0000	0.7500	0.9841

training wafers were provided, it is possible that more complex regression algorithms such as k-NN or ANN would perform better than a simple linear model.

Fourth, the best novelty detection algorithm depended upon the VM prediction model. Gauss was found to be the best for MLR in terms of the median adjusted MAE, the difference between the MAE in WH and W_L, find the proportion of adjusted MAE increase. For the other two VM models, k-NN was found to be the best when using the same criteria. It is interesting that the simplest novelty detection algorithm, i.e., Gauss, was best suited for the simplest (linear) VM model, while the more complicated novelty detection algorithm, i.e., k-NN, went well with non-linear VM models. It is worth noting that although the fusion of individual novelty detectors did not result in the lowest adjusted MAEs for W_H, it gave a remarkable performance in terms of the proportion of adjusted MAE increase. For the six VM model-variable selection pairs, the fusion novelty detector resulted in the highest proportions of the adjusted MAE increase (Table 6), with an exception of ANN–GA pair. Even in the ANN–GA pair, its adjusted MAE increase proportion is 0.9841, which is very close to the best result (1, MLR) and much higher than the others. This implies that the fusion novelty detectors can reduce the variation of individual novelty detectors so a more stabilized performance can be achieved. We would also note that among the novelty detection algorithms, SVDD displayed behavior that was different from that of the other algorithms. Its mean and median adjusted MAE of W_H was as low as that of the other algorithms, but the gap between W_H and W_L was significantly narrower. As explained earlier, SVDD rejected many wafers as it gave a high reliability level. Some of these rejected wafers were actually not similar to the training wafers, but the others were labeled low even though they were actually drawn from the same underlying distribution. The MAE of those wafers was not as large as that of an actual novel wafer; thus, this diluted the MAE of W_L. As a consequence, we would not recommend the use of a conservative novelty detection algorithm such as SVDD unless one wishes a very strict process monitoring and is willing to accept a number of alarms.

The VM prediction performance according to the reliability level, in terms of PARE, are summarized in Table 7. First of all, similar to the results obtained in terms of the adjusted MAE, W_H resulted in higher average PAREs (better performance) than W_L, regardless of the VM model, variable selection algorithm, or novelty detection algorithm. The average PARE of W_H for a certain VM model/variable selection/novelty detection algorithm pair is at least 10 % higher than that of W_L. With stepwise selection, the performance of MLR and ANN seemed indistinguishable from one another, since the PAREs of W_H were always >0.9, but those of W_L were smaller than 0.8, except for SVDD. Although k-NN resulted in similar PAREs for WH, the PAREs for W_L were greater than those obtained with the other VM models. Therefore, the difference in the PARE for W_H versus W_L became narrower. This is not desirable unless a VM model can predict both W_H and W_L very well, so the reliability level for the prediction results becomes of no use. When we looked back at Table 5, it seemed unfortunate that, k-NN did not have as a good prediction power for W_L as for W_H. Therefore, we can conclude that the high PARE for W_L obtained with k-NN was not so much due to the fact that many of its predictions were accurate, but rather, that they were marginally within the threshold θ in Eq. (18).

Second, it is worth pointing out that the variable selection with GA was comparable to the stepwise variable selection only when MLR was adopted as a VM model. With the other two regression models, the average PARE of W_H was not significantly greater than that of WL, and was even lower in some cases. We suspect that because GA covers a broader search space than stepwise selection, it was likely to over-fit the training data. Since MLR is a linear model, it has a relatively lower degree of complexity than k-NN and ANN, and this low level of complexity compensates for the over-fitted variable selection results. k-NN and ANN, on the other hand, are models with a higher level of complexities, which are able to generate the arbitrary shape of a curve for regression fitting. Thus, the over-fitted variable selection results were not controlled by the VM model, which resulted in prediction performance degradation.

Third, among the VM model-novelty detection pairs, MLR with Gauss was found to be the best combination. Although the average PARE was not the highest with the MLR–Gauss pair, the difference between the best PARE and that of the MLR–Gauss pair was negligible. However, the difference between the PARE of W_H and W_L was maximized with the MLR–Gauss combination. However, similar to the results in terms of adjusted MAE, the fusion of novelty detectors results in the best when looking at the performance stability. Table 8 shows the proportion of equipment–period–target pairs in which the average PARE of W_H is greater than that of W_L among a total of 64 pairs. It is confirmed that the fusion model was outstanding for all VM model-variable selection combinations. At least 70 % of pairs resulted in a higher average PARE of WH than WL (k-NN-GA), while more than 92 % of them resulted in a higher average PARE of WH than WL when MLR-GA combination is employed.

In summary, in terms of adjusted MAE and PARE, we can make the following observations. First, every novelty detection algorithm was useful in detecting wafers that would produce less reliable VM predictions.

Table 7 The summary statistics of the PARE with respect to W_H and W_L

Variable selection			Stepwise				GA			
VM	ND	Reliability	Mean	Median	SD	Q3–Q1	Mean	Median	SD	Q3–Q1
MLR	Gauss	W_H	0.9332*	0.9490	0.0587	0.0485	0.9211*	0.9343	0.0602	0.0527
		W_L	0.7104	0.8000	0.3535	0.4143	0.6685	0.7500	0.2430	0.3417
	MoG	W_H	0.9356*	0.9474	0.0563	0.0461	0.9186*	0.9374	0.0647	0.0475
		W_L	0.7854	0.8947	0.2871	0.2971	0.7037	0.7778	0.2633	0.4231
	KMC	W_H	0.9313*	0.9452	0.0581	0.0527	0.9130*	0.9252	0.0660	0.0557
		W_L	0.7582	0.8333	0.2888	0.3667	0.6054	0.6333	0.3429	0.3990
	k-NN	W_H	0.9350*	0.9464	0.0561	0.0504	0.9164*	0.9322	0.0652	0.0506
		W_L	0.7455	0.9161	0.3598	0.3056	0.6467	0.7500	0.3172	0.4183
	SVDD	W_H	0.9360*	0.9524	0.0552	0.0474	0.9178*	0.9313	0.0651	0.0568
		W_L	0.8387	0.8949	0.1696	0.2582	0.7980	0.8333	0.1761	0.2246
	Fusion	W_H	0.9319*	0.9453	0.0586	0.0495	0.9163*	0.9268	0.0626	0.0536
		W_L	0.7270	0.8571	0.3400	0.3750	0.7044	0.7813	0.3001	0.5000
k-NN	Gauss	W_H	0.9279*	0.9353	0.0558	0.0734	0.9266	0.9352	0.0573	0.0750
		W_L	0.8029	1.0000	0.3065	0.2500	0.8822	1.0000	0.1618	0.2000
	MoG	W_H	0.9282	0.9423	0.0600	0.0700	0.9280	0.9358	0.0537	0.0760
		W_L	0.8338	0.8571	0.2325	0.2153	0.8527	0.9393	0.2154	0.1875
	KMC	W_H	0.9253	0.9367	0.0585	0.0794	0.9239	0.9281	0.0556	0.0855
		W_L	0.8394	0.9375	0.2207	0.2917	0.8917	1.0000	0.2387	0.1214
	k-NN	W_H	0.9298	0.9394	0.0586	0.0750	0.9257	0.9343	0.0550	0.0821
		W_L	0.8130	0.8411	0.2053	0.2788	0.8573	0.9129	0.2036	0.1951
	SVDD	W_H	0.9262	0.9393	0.0617	0.0791	0.9239	0.9310	0.0598	0.0870
		W_L	0.8732	0.9166	0.1896	0.1484	0.9147	0.9179	0.0935	0.1002
	Fusion	W_H	0.9278*	0.9362	0.0573	0.0710	0.9255	0.9366	0.0561	0.0736
		W_L	0.7981	0.8889	0.2726	0.2667	0.8723	1.0000	0.1995	0.2000
ANN	Gauss	W_H	0.9226*	0.9380	0.0561	0.0706	0.9166*	0.9253	0.0627	0.0774
		W_L	0.6817	0.7750	0.3510	0.5000	0.7437	0.8333	0.3200	0.3333
	MoG	W_H	0.9256*	0.9369	0.0563	0.0755	0.9181*	0.9291	0.0648	0.0731
		W_L	0.7822	0.8333	0.2189	0.3205	0.8149	0.8297	0.1669	0.2609
	KMC	W_H	0.9210*	0.9358	0.0594	0.0825	0.9151*	0.9250	0.0638	0.0727
		W_L	0.7088	0.9000	0.3442	0.5000	0.7281	0.8333	0.3133	0.5000
	k-NN	W_H	0.9235*	0.9390	0.0600	0.0815	0.9179*	0.9302	0.0617	0.0687
		W_L	0.7270	0.8063	0.2998	0.3979	0.7571	0.8571	0.2949	0.3333
	SVDD	W_H	0.9222	0.9400	0.0594	0.0832	0.9190*	0.9301	0.0633	0.0675
		W_L	0.8513	0.9191	0.1740	0.2321	0.8049	0.8633	0.2166	0.2230
	Fusion	W_H	0.9214*	0.9376	0.0582	0.0820	0.9166*	0.9246	0.0628	0.0675
		W_L	0.7163	0.8452	0.3215	0.4167	0.7413	0.8462	0.2804	0.5000

An asterisk (*) denotes that the average adjusted PARE of W_H is greater than that of W_L at the significant level of 0.05

Table 8 The proportion of equipment–period–target pairs where the PARE of W_H is greater than that of W_L

VM model	Variable selection	Gauss	MoG	KMC	*k*-NN	SVDD	Fusion
MLR	Stepwise	0.8500	0.8438	0.7500	0.6875	0.7250	0.8750
	GA	0.9375	0.8750	0.8750	0.8333	0.7667	0.9219
k-NN	Stepwise	0.7143	0.7083	0.6500	0.7000	0.4500	0.7302
	GA	0.5750	0.6750	0.4375	0.6429	0.5000	0.7031
ANN	Stepwise	0.6429	0.7250	0.7000	0.8750	0.6750	0.8438
	GA	0.6563	0.8125	0.7083	0.7500	0.6923	0.8281

Second, stepwise variable selection resulted in better reliability estimation performance than GA selection in general, because it prevented over-fitting of the training data. Third, among the candidate regression models and novelty detection algorithms, the MLR–Gauss pair produced effective reliability evaluation as well as accurate VM prediction. Fourth, constructing a fusion model can improve the stability of the proposed framework since the proportion of the equipment–period–target variable pairs that result in better performance for WH than WL in terms of both adjusted MAE and PARE is higher with the fusion model than the other individual novelty detectors.

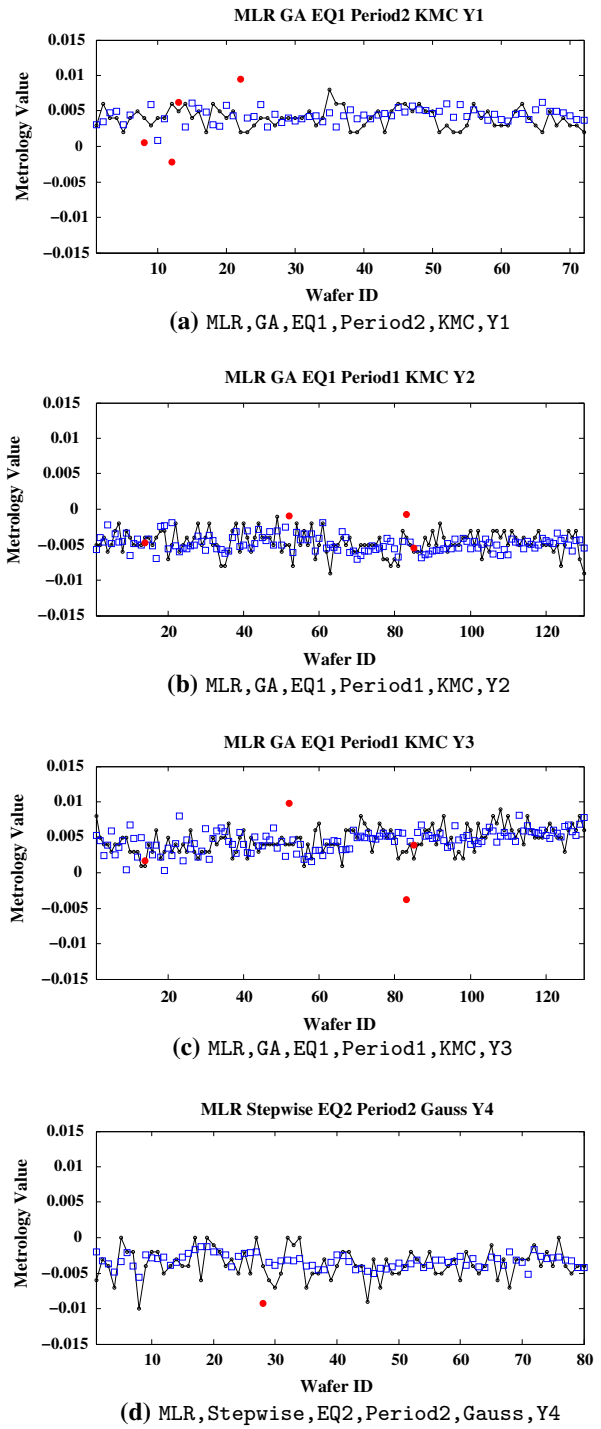


Fig. 6 The actual VM results (*small circle*) and the VM prediction of W_H (*empty square*) and W_L (*large circle*)

Figure 6 shows a number of VM prediction results and their corresponding reliability levels for certain equipment/period/target pairs with certain VM model/variable selection/novelty detection pairs. We would note that the small circles represent actual metrological values, while the empty squares and large circles represent the predicted metrological values of W_H and W_L, respectively. We would also note that, in general, the variation of the predicted metrological values is smaller than that of the actual values, because none of the regression models was designed to learn the natural noise. In Fig. 6a there are four wafers (wafer ID 7, 11, 12, and 21) with low reliability, and their actual and predicted metrology values are notably different from those of the wafers with high reliability, except for one wafer (wafer 12). In Fig. 6b, c, two wafers with low reliability have VM values that are very different from the actual ones, while the other two wafers have VM predictions that are similar to the actual ones, despite the low reliability. In Fig. 6d, only one wafer turned out to be unreliable, and its prediction value is very different from its actual metrology value. With these VM prediction results and the evaluated reliability, a process engineer could take appropriate action as follows. Let us assume that a wafer's reliability level is high. If its predicted metrology value is within the control limit, we can conclude that the process is operating properly, and no action need to be taken. If, on the other hand, its predicted metrology value is outside the control limit, one can conclude that something has gone wrong during the operation. In this case, proper follow-up action such as tool adjustment or recipe modification should be performed. If, however, a wafer's reliability level turns out to be low, then no further action should be taken, based on its predicted VM values alone, until its actual metrology value is measured, because the prediction is not trustworthy. If the predicted metrology value is outside the control limit but it turns out that its actual metrology value is within the control limit, we are able to avoid the performance of additional unnecessary operations. The problem with the reliability evaluation occurs when a wafer's reliability level is low, but its predicted VM value is within the control limit and its actual metrology value is also within the limit. The cost of this is the resource needed to provide additional actual metrology. However, by updating the novelty detection

model with the inclusion of that wafer, we can improve the reliability evaluation model in the long run.

6 Conclusion

In this paper, we have proposed a framework for evaluating the reliability of VM predictions to support the selective usage of VM results, in order to facilitate flexible process control. In order to determine the reliability level, we propose the use of novelty detection algorithms that determine the homogeneity between a test wafer and training wafers. If the test wafer is determined to be similar to the training wafers, its VM prediction is considered highly reliable; if not, it is considered unreliable. In order to analyze the effect of the proposed reliability evaluation methods, we conducted extensive experiments using two variable selection methods, three VM prediction models, and five novelty detection algorithms as well as their fusion model, based on actual process and metrology data. The experimental results showed that every novelty detection algorithm could satisfy our purpose, but specifically, the MLR–Gauss or MLR–fusion pair with stepwise variable selection was outstanding. We also demonstrated that, based on the evaluated reliability level and predicted metrological values, an appropriate follow-up action can be taken that will facilitate accurate and flexible process control.

Apart from a number of experimental results which we noted, there are a few limitations of the present work that suggest further directions for research. First, there is no clear definition of the outlier in an actual manufacturing control system; we cannot evaluate the performance of novelty detectors using more diversified measures, such as the rejection ratio of normal class (false alarm) and the acceptance ratio of novel class (miss). Thus, what we have done is to evaluate the latent effect of outliers indirectly by comparing the performances for the normal and novel classes determined by the novelty detectors. Therefore, it should be worth applying our framework to a process control system which has a clear definition of outliers. Second, because we had difficulty collecting actual data from a semiconductor manufacturing process, we could not investigate the long-term effect of the reliability evaluation models. Therefore, long-term-based VM prediction and reliability evaluation models should be developed and analyzed. Third, although we provided a general guideline for the selective usage of the reliability evaluation results, its practical impact should be studied by implementing a reliability evaluation methodology in a wafer-to-wafer control scheme.

Acknowledgments The first author was supported by the research program funded by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2011-0021893) and by the Ministry of Science, ICT, and Future Planning (NRF-2014R1A1A1004648). The dataset used in this paper can be available upon request.

References

1. Angiulli F, Pizzuti C (2005) Outlier mining in large high-dimensional data sets. IEEE Trans Knowl Data Eng 17(2):203–215
2. Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: Ordering points to identify the clustering structure. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp 49–60
3. Barnett V, Lewis T (1994) Outliers in statistical data. Wiley and Sons, New York
4. Besnard J, Toprac A (2006) Wafer-to-wafer virtual metrology applied to run-to-run control. In: Proceedings of the Third ISMI Symposium on Manufacturing Effectiveness, Austin, TX, USA
5. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, New York
6. Bode C, Ko B, Edgar T (2004) Run-to-run control and performance monitoring of overlay in semiconductor manufacturing. Control Eng Pract 12(7):893–900
7. Boutsinas B, Tasoulis DK, Vrahatis MN (2006) Estimating the number of clusters using a windowing technique. Pattern Recogn Image Anal 16(2):143–154
8. Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, pp 93–104
9. Burge P, Shawe-Taylor J (1997) Detecting cellular fraud using adaptive prototypes. In: Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, Rhode Island, USA, pp 9–13
10. Castillo DE, Yeh JY (1998) An adaptive run-to-run optimizing controller for linear and nonlinear semiconductor processes. IEEE Trans Semicond Manuf 11(2):285–295
11. Chang YJ, Kang Y, Hsu CL, Chang CT, Chan T (2006) Virtual metrology techniques for semiconductor manufacturing. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006), Vancouver, BC, Canada, pp 5289–5293
12. Chemali CE, Freudenberg J, Hankinson M, Bendik JJ (2004) Run-to-run critical dimension and sidewall angle lithography control using the PROLITH simulator. IEEE Trans Semicond Manuf 17(3):388–401
13. Chen P, Wu S, Lin J, Ko F, Lo H, Wang J (2005) Virtual metrology: a solution for wafer to wafer advanced process control. In: Proceedings of International Symposium on Semiconductor Manufacturing (ISSM 2005), San Jose, CA, USA, pp 155–157
14. Chen YT, Yang HC, Cheng FT (2006) Multivariate simulation assessment for virtual metrology. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, pp 1048–1053
15. Cheng CY, Cheng FT (2005) Application development of virtual metrology in semiconductor industry. In: Proceedings of The 31st Annual Conference of IEEE Industrial Electronics Society (IECON 2005), Raleigh, NC, USA, pp 124–129

16. Cheng FT, Chen YT, Su YC, Zeng DL (2008) Evaluating reliance level of a virtual metrology system. IEEE Trans Semicond Manuf 21(1):92–103
17. Cheng FT, Chang JC, Huang HC, Kao CA, Chen YL, Peng JL (2011) Benefit model of virtual metrology and integrating AVM into MES. IEEE Trans Semicond Manuf 24(2):261–272
18. Cheng FT, Huang HC, Kao CA (2012) Developing an automatic virtual metrology system. IEEE Trans Autom Sci Eng 9(1):181–188
19. Cho SJ, Hermsmeier MA (2002) Genetic algorithm guided selection: variable selection and subset selection. J Chem Inf Model 42(4):927–936
20. Cyganek B (2012) One-class support vector ensembles for image segmentation and classification. J Math Imaging Vis 42(2–3):103–117
21. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York
22. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Second international conference on knowledge discovery and data mining. AAAI Press, Menlo Park, pp 226–231
23. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
24. Gergeret F, Gall G (2003) Yield improvement using statistical analysis of process data. IEEE Trans Semicond Manuf 16(3):535–542
25. Guha S, Rastogi R, Shim K (1998) CURE: An efficient clustering algorithm for large databases. In: Proceedings of ACM SIGMOD Conference on Management of Data, Seattle, WA, USA, pp 73-84
26. Guha S, Rastogi R, Shim K (2000) ROCK: a robust clustering algorithm for categorical attributes. Inf Syst 25(5):345–366
27. Harmeling S, Dornhege G, Tax D, Meinecke F, Müller KR (2006) From outliers to prototypes: ordering data. Neurocomputing 69(13–15):1608–1618
28. Hastie T, Tibshirani R, Friedman J (2002) The element of statistical learning: data mining, inference, and prediction. Springer-Verlag, New York
29. Hawkins DM (1980) Identification of outliers. Chapman & Hall, London
30. Jarvis RM, Goodacre R (2004) Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data. Bioinformatics 21(7):860–868
31. Jiang H, Liu G, Xiao X, Mei C, Ding Y, Yu S (2012) Monitoring of solid-state fermentation of wheat straw in a pilot scale using FT-NIR spectroscopy and support vector data description. Microchem J 102:68–74
32. Kang P, Cho S (2008) Locally linear reconstruction for instance-based learning. Pattern Recogn 41(11):3507–3518
33. Kang P, Cho S (2009) A hybrid novelty score and its use in keystroke dynamics-based user authentication. Pattern Recogn 42(11):3115–3127
34. Karypis G, Han EH, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. IEEE Comput Mag 32(8):68–75
35. Khan A, Moyne J, Tilbury D (2007) An approach for factory-wide control utilizing virtual metrology. IEEE Trans Semicond Manuf 20(4):364–375
36. Khazai S, Safari A, Mojaradi B, Homayouni S (2012) Improving the SVDD approach to hyperspectral image classification. IEEE Geosci Remote Sens Lett 9(4):594–598
37. Knorr EM, Ng RT, Tucakov V (2000) Distance-based outliers: algorithms and applications. VLDB J 8(3–4):237–253
38. Krawczyk B, Filipczuk P (2013) Cytological image analysis with firely nuclei detection and hybrid one-class classification decomposition. Eng Appl Artif Intell. doi:10.1016/j.engappai.2013.09.017
39. Krawczyk B, Woźniak M (2014) Diversity measures for one-class classifier ensembles. Neurocomputing 126:36–44
40. Limanond S, Si J, Tsakalis K (1998) Monitoring and control of semiconductor manufacturing processes. IEEE Control Syst Mag 18(6):46–58
41. Lin TH, Hung MT, Lin RC, Cheng FT (2006) A virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, pp 1054–1059
42. Lynn S, Ringwood J, MacGrearailt N (2010) Weighted windowed PLS models for virtual metrology of an industrial plasma etch process. In: Proceedings of IEEE International Conference on Industrial Technology (ICIT 2010), Vina del Mar, Chile, pp 309–314
43. Mũnoz-Marí J, Bovolo F, Gomez-Chova L, Bruzzone L, Camp-Valls G (2010) Semisupervised one-class support vector machines for classification of remote sensing data. IEEE Trans Geosci Remote Sens 48(8):3188–3197
44. McLachlan GJ, Peel D (2000) Finite mixture models. Wiley and Sons, New York
45. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12(2):181–201
46. Pang TH, Sheng BQ, Wong DSH, Jang SS (2011) A virtual metrology system for predicting end-of-line electrical properties using MANCOVA model with tools clustering. IEEE Trans Ind Inf 7(2):187–195
47. Park SJ, Lee MS, Shin SY, Cho KH, Lim JT, Cho BS, Jei YH, Kim MK, Park CH (2005) Run-to-run overlay control of steppers in semiconductor manufacturing systems. IEEE Trans Semicond Manuf 18(4):605–613
48. Qin S, Cherry G, Good R, Wang J, Harrison C (2006) Semiconductor manufacturing process control and monitoring: a fab-wide framework. J Process Control 16(3):179–191
49. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of International Conference on Management of Data (SIGMOD 2000), Dallas, TX, USA
50. Ross SM (2004) Introduction to probability and statistic for engineers and scientists. Academic Press, San Diego
51. Sachs E, Hu A, Ingolfsson A (1995) Run by run process control: combining SPC and feedback control. IEEE Trans Semicond Manuf 8(1):26–43
52. Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
53. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
54. Spanos C, Guo HF, Miller A, Levine-Parril J (1992) Real-time statistical process control using tool data. IEEE Trans Semicond Manuf 5(4):308–318
55. Su AJ, Jeng JC, Huang HP, Yu CC, Hung SY, Chao CK (2007) Control relevant issues in semiconductor manufacturing: overview with some new results. Control Eng Pract 15(10):1268–1279
56. Su YC, Lin TH, Cheng FT, Wu WM (2008) Accuracy and real-time considerations for implementing various virtual metrology algorithms. IEEE Trans Semicond Manuf 21(3):426–434
57. Tax D (2001) One-class classification: Concept-learning in the absence of counterexamples. PhD thesis, Delft University of Technology, URL http://wwwict.ewi.tudelft.nl/davidt
58. Tax D, Duin R (1999) Support vector domain description. Pattern Recogn Lett 20(11–13):1191–1199
59. Tax D, Duin R (2005) Characterizing one-class datasets. In: Proceedings of the Sixteenth Annual Symposium of the Pattern

Recognition Association of South Africa, Langebaan, South Africa, pp 21–26
60. Vapnik V (1998) Statistical learning theory. Wiley and Sons, New York
61. Wang XA, Mahajan R (1996) Artificial neural network model-based run-to-run process controller. IEEE Trans Compon Packag Manuf Technol part C 19(1):19–26
62. Wilk T, Wozniak M (2012) Soft computing methods applied to combination of one-class classifiers. Neurocomputing 75(1):185–193
63. Yeh CY, Lee ZY, Lee SJ (2009) Boosting one-class support vector machines for multi-class classification. Appl Artif Intell 23(4):297–315
64. Yi J, Sheng Y, Xu C (2003) Neural network based uniformity profile control of linear chemical-mechanical planarization. IEEE Trans Semicond Manuf 16(4):609–620
65. Ypma A, Duin RPW (1998) Support objects for domain approximation. In: Proceedings of International Conference on Artificial Neural Networks, Skovde, Sweden
66. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In: Proceedings of ACM SIGMOD Conference on Management of Data, Montreal, Canada, pp 103–114
67. Zhuang L, Dai H (2006) Parameter optimization of kernel-based on-class classifier on imbalance learning. J Comput 1(7):32–40