4th IEEE Conference on Automation Science and Engineering
Key Bridge Marriott, Washington DC, USA
August 23-26, 2008

Automatic Virtual Metrology System Design and Implementation[1]

Yi-Ting Huang, *Student Member*, *IEEE*, Hsien-Cheng Huang, *Student Member*, *IEEE*,
Fan-Tien Cheng[2], *Fellow*, *IEEE*, Tai-Siang Liao, and Fu-Chien Chang

Abstract— **An automatic virtual metrology (AVM) system that consists of a model-creation server and many AVM servers is proposed. The model-creation server will generate the first set of data quality evaluation models and virtual metrology (VM) conjecture models of a certain equipment type. Under fab-wide VM deployment, the model-creation server can also fan out the first set of models generated to all the AVM servers of the same equipment type. Also, each individual fan-out-accepter's AVM server can perform an automatic model refreshing process to promptly refresh its own data quality evaluation models and VM conjecture models. As such, the VM accuracy can be maintained and the AVM server is then ready to serve. This paper gives an overview to the AVM system and elaborates the automatic data quality evaluation schemes and the automatic model refreshing schemes.**

***Index Terms*— Virtual metrology (VM), automatic virtual metrology (AVM) system, model-creation server, AVM framework, automatic fanning out, automatic model refreshing, automatic data quality evaluation.**

I. Introduction

Virtual Metrology (VM) is a method to conjecture manufacturing quality of a process tool based on data sensed from the process tool and without physical metrology operation [1], [2], [5], [8]. According to VM deployment and automation degree, the VM systems can be classified into four levels as shown in Fig. 1 and depicted as follows.

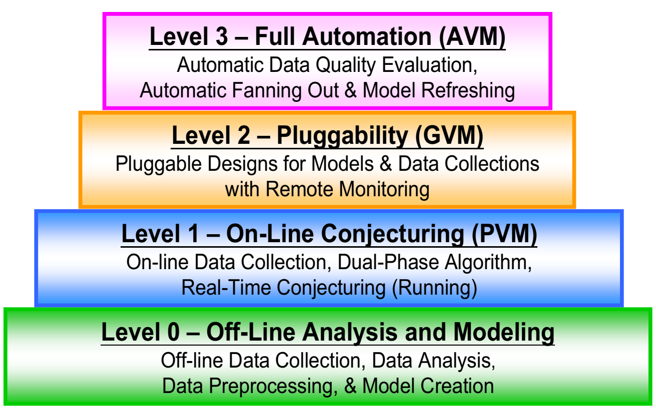

Fig. 1. Automation Levels of Virtual Metrology Systems

Level 0

Level 0 is the most fundamental level in terms of VM automation degree. The Level 0 VM system is capable of executing off-line collection and analysis of the historical process & metrology data. It can also create the first process data quality index (DQI_X) and metrology data quality index (DQI_y) models [11], VM conjecture model [2], [5], reliance index (RI) model [6], and global similarity index (GSI) model [6]. The accuracy of these models can also be evaluated in Level 0.

Level 1

Also referred to as the preliminary VM (PVM) system, the Level 1 VM system focuses on on-line conjecturing. The functions of Level 0 VM system for constructing the first DQI_X/DQI_y and VM & RI/GSI models must be established prior to building PVM. The PVM system should possess capabilities of on-line data collection and dual-phase VM algorithm [8] such that Phase-I VM value (denoted VM_I) and Phase-II VM value (denoted VM_{II}) along with their accompanying RI/GSI can be generated in real time.

However, when the VM application is changed, for example, from chemical vapor deposition (CVD) equipment to etching equipment, the PVM system demands many modifications in the system components, including the data collection driver, the conjecture models, and so on. That is, when a certain component in a PVM system requires adjustment, other related components or modules in the PVM need to be modified as well. Therefore, pluggable designs should be considered for the frequently modified components in PVM so as to enable the VM system becomes more generic.

Level 2

Further improved from Level 1, the Level 2 generic VM (GVM) system aims at module pluggability [10]. The GVM framework not only inherits the on-line conjecturing capability from PVM, but also possesses the pluggability which enables easy exchanges of its data collection driver, VM & RI/GSI models, and communication agent. For example, a GVM system for CVD equipment can also be applied to etching equipment with only a few replacements (i.e., data collection module and VM & RI/GSI models) in the pluggable modules. Furthermore, the GVM system has the capability to empower its remote clients to monitor the system conjecture results at any time and place.

Level 3

Level 3, denoted automatic VM (AVM), is the highest level (full automation) in VM deployment and automation. Besides the functions of PVM and GVM, the AVM system is also

1. The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under contract No: NSC96-2622-E-006-043. This work is Taiwan R.O.C. and U.S.A. Patents Pending with application no.: 97118526.
2. The corresponding author (e-mail: chengft@mail.ncku.edu.tw).

equipped with the capabilities of automatic data quality evaluation, automatic fanning out, and automatic model refreshing.

The purpose of this paper is to develop an AVM system. Physical characteristics of different equipment are not the same. For maintaining VM conjecture accuracy, DQI_X/DQI_y and VM & RI/GSI models must be created based on the historical process and metrology data acquired from each chamber (a tool is usually composed of 1 to 6 chambers). Therefore, when considering fab-wide VM implementation, we must be aware that the number of DQI_X/DQI_y and VM & RI/GSI models will increase rapidly with the growing number of tools. Under such condition, if we still take the traditional method to create those models one by one with a lot of historical data, huge labor expenses and model-creation time will make fab-wide VM implementation impossible. To solve this problem, the capabilities of automatic fanning out and automatic model refreshing must be implemented in the AVM system to enable those models to be automatically fanned and refreshed to the same type of equipment for maintaining the VM conjecture accuracy and saving tremendous labor expenses and model-creation time. The CVD equipment in a fifth-generation TFT-LCD factory in Taiwan was selected as the illustrative example for automatic VM model refreshing.

The remainder of this paper is organized as follows. Section 2 defines the AVM system. Section 3 then explains the operating procedures for creating the first DQI_X/DQI_y models as well as VM conjecture and RI/GSI models. Next, Section 4 elaborates the advanced dual-phase VM scheme that includes DQI_X, DQI_y, and automatic model refreshing algorithms. Section 5 then presents an illustrative example for automatic VM model refreshing. Finally, a summary and conclusions are made in Section 6.

II. AUTOMATIC VIRTUAL METROLOGY SYSTEM

To achieve the goal of full automation, the abilities of on-line and real-time data preprocessing must be implemented in AVM. As such, the original labor-intensive and time-consuming data pre-processing procedures can be executed automatically and efficiently. However, due to the lack of precedent for reference, expert's experience and manpower are still needed for constructing the first DQI_X/DQI_y and VM & RI/GSI models along with their corresponding DQI_X/DQI_y and RI/GSI thresholds of a certain equipment type. After that, the system will be able to distinguish abnormal process and metrology data by calculating their DQI_X and DQI_y values, respectively. On detecting abnormal (i.e., bigger than the DQI_X threshold) DQI_X values, it will warn process engineers automatically. Similarly, if the system detects abnormal (i.e., bigger than the DQI_y threshold) DQI_y values, besides warning, the metrology data will be deleted automatically to avoid deteriorating the VM conjecture models.

To sum up, for implementing the functions of automatic fanning out and automatic model refreshing in AVM, the system must accomplish the following procedures.

1) Construct the first set of data quality evaluation models and conjecture models. The data quality evaluation models include DQI_X and DQI_y models, and the conjecture models include VM & RI/GSI models [6], [8].

2) Fan out the first DQI_X/DQI_y and VM & RI/GSI models to other equipment of the same type. Namely, refresh the models by adopting the advanced dual-phase VM algorithm until the VM conjecture accuracy is gained.

To fulfill the requirements stated above, this paper proposes an AVM system, as presented in Fig. 2. The AVM system is composed of a model-creation server and several AVM servers. The model-creation server is responsible for executing off-line data analysis and creating the first DQI_X/DQI_y and VM & RI/GSI models of a certain type of equipment and then fanning out those models to all equipment with the same type; while the AVM servers are in charge of automatic model refreshing and on-line VM conjecturing. All the important features of the AVM system are described below.

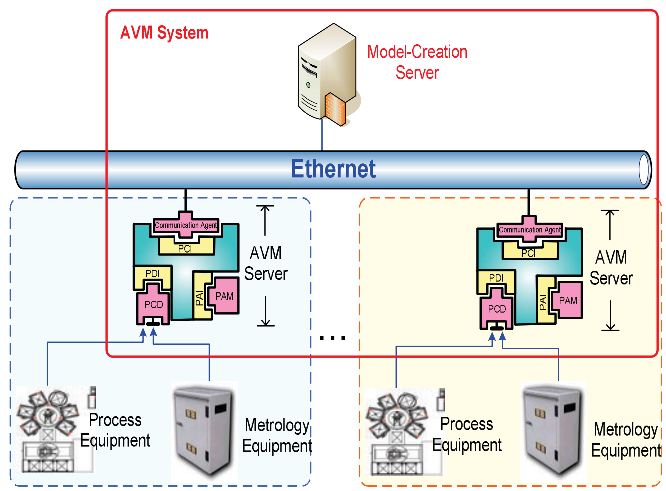

Fig. 2. Automatic Virtual Metrology System

III. OPERATING PROCEDURES FOR CREATING THE FIRST DQI_X/DQI_y AND VM & RI/GSI MODELS

The model-creation server is responsible for constructing the first set of DQI_X/DQI_y and VM & RI/GSI models for a certain type of equipment. As illustrated in Fig. 3, the operating procedure for creating the first DQI_X/DQI_y and VM & RI/GSI models starts with collecting metrology data. Practically, all the actual metrology data have their corresponding equipment process data. Therefore, all the newly collected actual metrology data will be asked to correlate to their corresponding equipment process data via the workpiece identification (ID) number. If the correlation succeeds, the set of data will be stored into the system. On the contrary, if the correlation fails, the set of data will be deleted, and the system will search for a new data set.

When the amount of data sets collected is enough for creating new models, process engineers can examine all the collected historical metrology and process data one by one to construct the process data standard temporal patterns and select proper indicators. Then, metrology data abnormal modes are also constructed. By applying the historical process data conforming to the standard temporal patterns and comparing the historical metrology data to the metrology data abnormal modes, data cleaning processes of those historical modeling data sets can then proceed. After data cleaning, the first DQI_X and DQI_y models can be constructed. Next, the key parameters for creating VM conjecture models are selected through execution of the parameter sifting rules [11]. Finally, the selected key parameters are utilized to create the first VM & RI/GSI models.

As presented in Fig. 3, only the procedure with a red frame (for establishing process data standard patterns, selecting proper indicators, and constructing metrology data abnormal modes) requires assistance of manpower. All the other procedures can be executed automatically.

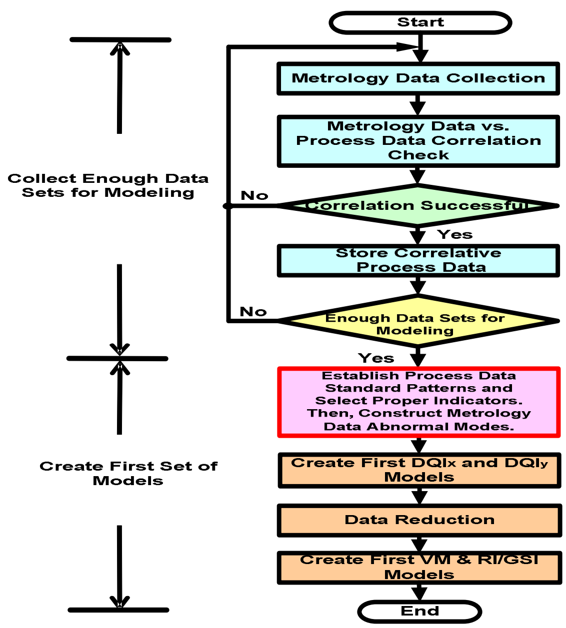

Fig. 3. Operating Procedure for Creating the First DQI_X/DQI_y and VM & RI/GSI Models

IV. ADVANCED DUAL-PHASE VM ALGORITHM

The original dual-phase VM scheme [8] has been successfully implemented in Level 1 PVM and Level 2 GVM systems [10]. To take a step further, an advanced dual-phase VM scheme as diagrammed in Fig. 4 is proposed to be adopted in an AVM server. The improved functions of the advanced scheme are marked by red in Fig. 4 and detailed as follows.

1) The DQI_X and DQI_y models are added into the algorithm to perform automatic data quality evaluation.

2) A model refreshing mechanism is added into the algorithm for automatic model-refreshing control. When VM accuracy is gained, the refreshing procedure will stop automatically and the system will enter the normal operation state.

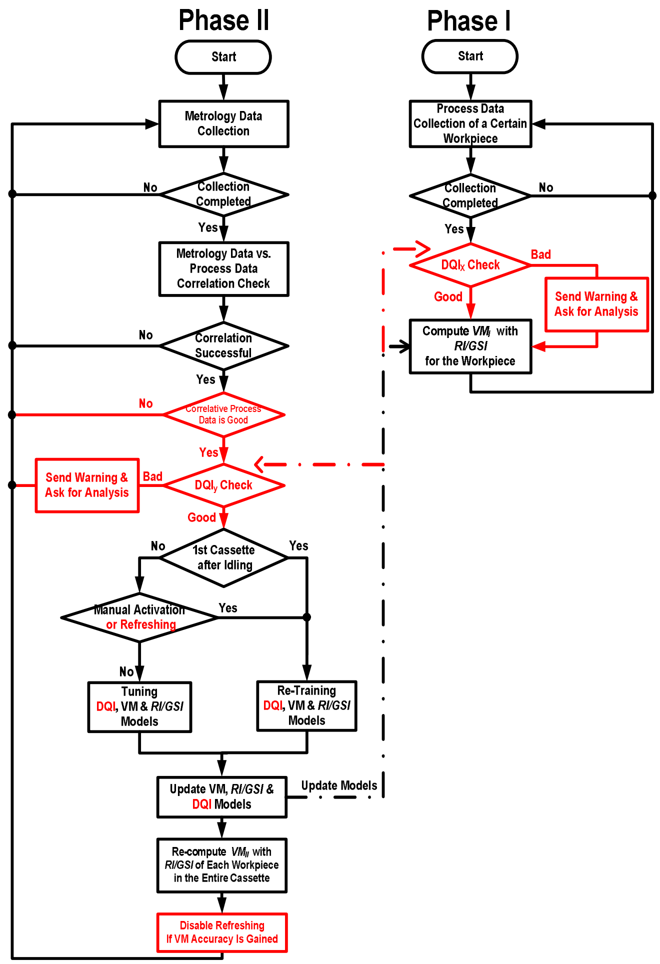

Fig. 4. Advanced Dual-Phase VM Algorithm

The DQI_X, DQI_y, and model-refreshing algorithms are described below.

A. On-Line and Real-Time DQI_X Algorithm

By applying principal component analysis (PCA) [13], Euclidean distance (ED) [14], and cross validation's leave-one-out (LOO) method [12], the DQI_X algorithm, as depicted in Fig. 5, is capable of real-time and on-line detecting abnormalities and updating the DQI_X model. Suppose that there are n historical process data sets for constructing the first DQI_X model and each data set consists of p parameters. These n historical process data sets are applied to generate p eigenvectors with p corresponding eigenvalues in descending order ($\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$) by PCA [13]. Then, a set of k significant eigenvectors (with $\lambda \geq 1$) is selected for constructing the feature-extraction matrix, **M**, which is expressed as:

$$\boldsymbol{M} = \begin{bmatrix} [\text{eigenvector 1}] \\ [\text{eigenvector 2}] \\ \vdots \\ [\text{eigenvector k}] \end{bmatrix}_{k*p} \quad (1)$$

The process data quality evaluation procedure begins with transforming the i-th input process data $\mathbf{X}_i$ with p parameters into the k data feature variables ($\boldsymbol{A}_i = [a_1, a_2, ..., a_k]$) by:

$$\boldsymbol{A}_i = \boldsymbol{M} \bullet \boldsymbol{X}_i \tag{2}$$

The first feature-extraction matrix, $\mathbf{M}$, was obtained by Eq. (1). Subsequently, the k data feature variables are transformed into a standardized Z-score data set, $\boldsymbol{Z}_{\boldsymbol{A}} = [z_{a_1}, z_{a_2}, ..., z_{a_k}]$, which is then converted by means of ED into a consolidated index, i.e., DQI_X :

$$DQI_{X_i} = \sqrt{\sum_{j=1}^{k} (z_{a_{i,j}} - \bar{z}_{a_j}) * (z_{a_{i,j}} - \bar{z}_{a_j})^T} \tag{3}$$

where i represents the number of the process-data set and $\bar{z}_{a_j}$ is the mean of the j-th standardized variable of the training samples. Theoretically, the value of $\bar{z}_{a_j}$ is zero, therefore Eq. (3) can be simplified as:

$$DQI_{X_i} = \sqrt{\sum_{j=1}^{k} (z_{a_{i,j}}) * (z_{a_{i,j}})^T} \tag{4}$$

Meanwhile, a reasonable DQI_X threshold, DQI_{X_T} , should be defined by applying LOO in the first DQI_X model-creating process:

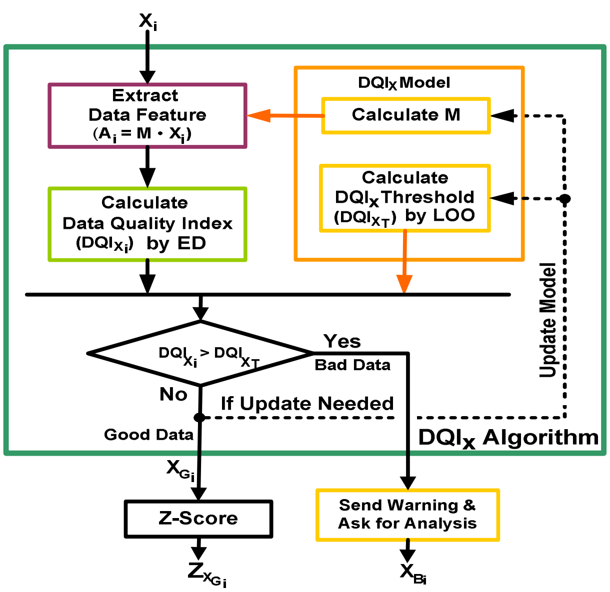

Fig. 5. On-Line and Real-Time DQI_X Algorithm

$$DQI_{X_T} = a * \overline{DQI}_{X_{LOO}} \tag{5}$$

where $\overline{DQI}_{X_{LOO}}$ stands for the 90% trimmed mean of all the DQI_X values calculated from all the modeling samples via LOO. The constant "a" in Eq. (5) is about 2 to 3 and the default value is set to be 3.

A set of input process data is considered to be abnormal (denoted $\mathbf{X}_{B_i}$) when its DQI_{X_i} is larger than DQI_{X_T} , otherwise it is considered normal (denoted $\mathbf{X}_{G_i}$). When a set of abnormal process data, $\mathbf{X}_{B_i}$, is detected, a warning message will be sent to process engineers for subsequent verification and analysis. As for those sets of normal process data, $\mathbf{X}_{G_i}$, they will be converted to $\mathbf{Z}_{X_{G_i}}$ by Z-score and sent to the VM conjecture models. The $\mathbf{X}_{G_i}$ will also be used for updating the DQI_X model, including $\mathbf{M}$ and DQI_{X_T} , if necessary.

B. On-Line and Real-Time DQI_y Algorithm

The purpose of the DQI_y algorithm is to detect the abnormal metrology data caused by: (1) measuring errors and (2) external factors, *e.g.* particle pollution. Since a VM value is conjectured based on its process data, the relationship between the process data and the corresponding metrology data is influential to the VM conjecture model. In general, a group of similar process data sets in a conjecture model will generate similar metrology values. Therefore, when a dissimilar metrology datum occurs within a similar metrology-data group, the outlier might be resulted from a measuring error or external factor.

Considering the above description, the DQI_y algorithm is designed based on the similarity of the process data. Through comparison among the corresponding metrology data that are correlated to a similar process-data group, a metrology outlier can be detected and further deleted from the model. Nevertheless, before applying the DQI_y algorithm, the corresponding DQI_X value is recalled to assure the process data's normality. In other words, only those metrology data with normal corresponding process data are examined by the DQI_y algorithm.

Prior to entering the on-line and real-time DQI_y algorithm as displayed in Fig. 6, the m similar patterns, $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m\}$, should be ready. The first set of similar patterns, $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m\}$, was generated by adaptive resonance theory 2 (ART2) [15] with ρ= 0.98 during the first DQI_y model-creating process.

Observing Fig. 6, when a new metrology datum, y_j, is collected, its corresponding standardized Z-score process data ($\mathbf{Z}_{X_{G_j}}$) is applied to search for the most similar pattern, denoted $\mathbf{P}_q$, by ART2 from the similar-pattern model $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_m\}$. In general, the degree of similarity of the pattern searched should surpass the threshold of ρ (0.98). However, in reality, the condition for ρ> 0.98 may not be satisfied. As such, the pattern with the maximal similarity will serve as the reference pattern.

The v samples ($v \geq 2$) inside the $\mathbf{P}_q=[\mathbf{X}_{q,1}, \mathbf{X}_{q,2}, ..., \mathbf{X}_{q,v}]$ with their corresponding metrology values, $\mathbf{Y}_q=[y_{q,1}, y_{q,2}, ..., y_{q,v}]$, and this new metrology datum, y_j, are utilized to calculate the DQI_{y_j} and DQI_{y_T} of the new metrology datum.

The DQI_{y_j} of y_j is obtained as (normalized variability, NV):

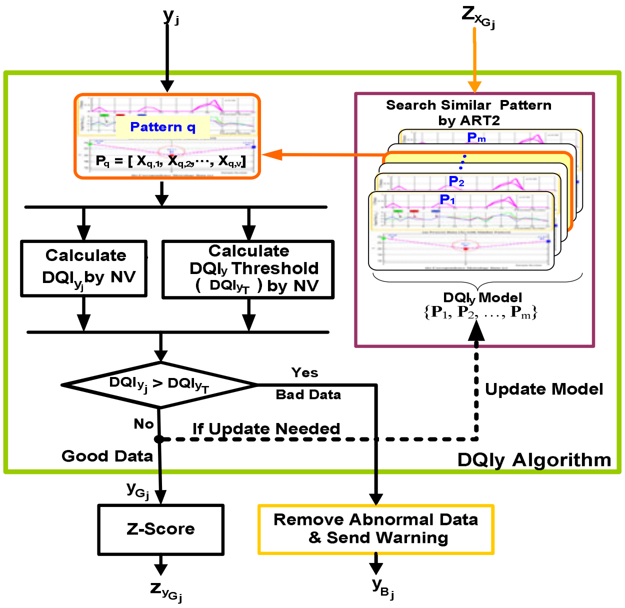

Fig. 6. On-Line and Real-Time DQI_y Algorithm

$$DQI_{y_j} = \frac{| y_j - \bar{y}_q |}{\bar{y}_q} \tag{6}$$

with $$\bar{y}_q = \frac{1}{v}\sum_{l=1}^{v} y_{q,l} \tag{7}$$

where $\bar{y}_q$ is the mean of $\boldsymbol{Y}_q$ or the mean of all the metrology values in $\boldsymbol{P}_q$, and v is the number of samples inside the pattern ($\boldsymbol{P}_q$).

The DQI_{y_T} of a certain pattern, $\boldsymbol{P}_q$, is defined to be the maximal-tolerable variance of that $\boldsymbol{P}_q$. Suppose y_t is the maximal-tolerable metrology value that possesses the maximal-tolerable variance in $\boldsymbol{P}_q$, then y_t can be presented as:

$$y_t = \bar{y}_q + R_{max} \text{ or } y_t = \bar{y}_q - R_{max} \tag{8}$$

where R_{max} is the maximal-tolerable variance:

$$R_{\max} = \max(R_{P_1}, R_{P_2}, \ldots, R_{P_m}) \tag{9}$$

where R_{P_i}, $i=1, 2, \ldots, m$, is the range in pattern $\boldsymbol{P}_i$, and m is the total number of all the similar-pattern groups.

By adding y_t into the similar pattern, $\boldsymbol{P}_q$, the DQI_{y_T} can be acquired as:

$$DQI_{y_T} = \frac{| y_t - \bar{y}_q |}{\bar{y}_q} \tag{10}$$

After obtaining the DQI_{y_j} and DQI_{y_T}, if $DQI_{y_j} > DQI_{y_T}$ is true, then it means the new metrology data (y_{B_j}) may be abnormal, which should be discarded and needs further analysis. Otherwise, the new metrology data (y_{G_j}) is normal, which can be included into the similar-pattern model for model re-training.

C. Automatic Model Refreshing Algorithm

The physical properties of two different chambers are unlikely to be the same. When conducting automatic fanning out, from chamber A to chamber B for example, the conjecture accuracy of the fanned out model will not be as good as that of the model created with the original data of chamber B. Therefore, automatic model refreshing is essential for recovering the conjecture accuracy of the fanned out model.

During the refreshing stage, the AVM server will apply the advanced dual-phase VM algorithm to automatically refresh the old models in chamber B. Through this way, the chamber B models, which were originally copied from chamber A, can gradually retain its freshness and conjecture accuracy.

The model refreshing mechanism works according to the following description. Observing the Phase-II portion of Fig. 4, whenever the system detects a new training data set (i.e., the new actual metrology data (y) of a workpiece and its correlative equipment process data (X)), if the data quality of this new training data set is good (checked by DQI_X/DQI_y), the oldest training set in the VM model will be deleted, and the new training set will be added into the model for re-training.

As mentioned in [6], the RI is defined as the overlapping-area value between the standardized statistical distribution of the neural-network (NN) conjecture value and the multiple-regression (MR) predictive value. Because the AVM server possesses the RI scheme, the AVM server should have both the NN and MR VM results; while NN conjecture value is assigned as the default VM output and MR predictive value is considered as the reference output [6], [8].

Thus, the completeness of the refreshing procedure can be evaluated by monitoring the NN and MR VM_I MAPE (mean absolute percentage error) [8] values and its accompanying GSI values. To enable the advanced dual-phase VM algorithm with the function of disabling the refreshing procedure when VM accuracy is gained (as shown in Fig. 4), the refreshing thresholds for NN and MR VM_I MAPE values (denoted NN_{RT} & MR_{RT}) and the refreshing threshold for the accompanying GSI values (denoted GSI_{RT}) must be determined. With NN_{RT}, MR_{RT}, and GSI_{RT}, a successful and complete refreshing procedure must satisfy the following conditions simultaneously.

Condition (1) The NN VM_I MAPE value is smaller than NN_{RT} for three consecutive refreshing samples.

Condition (2) The MR VM_I MAPE value is smaller than MR_{RT} for three consecutive refreshing samples.

Condition (3) The GSI value is smaller than GSI_{RT} for three consecutive refreshing samples.

The procedures for defining NN_{RT}, MR_{RT} and GSI_{RT} during creating the first set of NN, MR and GSI models are detailed as follows.

First, according to cross validation's leave-one-out (LOO) method [12], select one sample data set (among all the historical data sets for establishing the first set of models) as the test sample set, and utilize the remaining historical data sets to construct the NN conjecture model and MR predictive model [6]. Next, calculate the NN conjecture error (E_N), MR predictive error (E_M), and GSI_{LOO} value of the test sample set by applying the LOO models constructed in the previous step. Repeat the above steps until all the E_N, E_M, and GSI_{LOO} values of each individual test sample sets are acquired. Consequently, the maximal and minimal E_N/E_M, which represent the worst and best NN conjecture / MR predictive accuracy of the VM models, and the 90% trimmed mean of all the GSI_{LOO} values calculated from all the modeling samples via LOO are obtained. Accordingly, NN_{RT} and MR_{RT} are defined as follows.

$$NN_{RT} = \frac{Max(E_N) + Min(E_N)}{2} \tag{11}$$

$$MR_{RT} = \frac{Max(E_M) + Min(E_M)}{2} \tag{12}$$

where $Max(E_N)$, $Min(E_N)$, $Max(E_M)$, $Min(E_M)$ represent the maximal (E_{Ni}, $i=1, 2, \ldots, n$), minimal (E_{Ni}, $i=1, 2, \ldots, n$), maximal (E_{Mi}, $i=1, 2, \ldots, n$), minimal (E_{Mi}, $i=1, 2, \ldots, n$), respectively. And, *n* stands for the number of historical data sets for modeling.

On the other hand, GSI_{RT} is determined as:

$$GSI_{RT} = a * \overline{GSI_{LOO}} \tag{13}$$

Where $\overline{GSI_{LOO}}$ stands for the 90% trimmed mean of all the GSI_{LOO} values calculated from all the modeling samples via LOO. The constant "a" in Eq. (13) is about 2 to 3 and the default value is set to be 3. By the same token, the refreshing thresholds of DQI_X/DQI_y can also be defined and applied for determining whether the DQI_X/DQI_y refreshing process is successful and complete.

The functions of automatic fanning out and automatic model refreshing are to copy the DQI_X/DQI_y and VM & RI/GSI models of a certain chamber and apply it to other chambers of the same equipment type. As such, at the initial stage, the first DQI_X/DQI_y and VM & RI/GSI models of a new type of equipment can be established according its physical properties and with enough number of training data sets (about two to three times of the number of key parameters). Then, other pieces of equipment of the same type can adopt automatic fanning out and automatic model refreshing to create their own models, with an average requirement of less than 15 sets of training samples only. As a result, VM can be automatically deployed fab-wide without waste in massive manpower and time for establishing all the DQI_X/DQI_y and VM & RI/GSI models.

V. ILLUSTRATIVE EXAMPLE OF AUTOMATIC VM MODEL REFRESHING

A piece of CVD equipment in a fifth-generation TFT-LCD factory in Taiwan was selected as the illustrative example for automatic VM model refreshing.

The CVD equipment is composed of six chambers marked from chambers A to F. In this case, chamber A is chosen to generate the first set of VM & RI/GSI models, whereas chamber F is assigned to perform automatic model refreshing. In this example, 60 (n=60) historical data sets of chamber A were collected for creating the first set of models. Twenty-four (24) data sets of chamber F were applied to evaluate the refreshing results.

The experimental results are shown in Fig. 7. Simple recurrent neural networks (SRNN) [6], [8] are adopted as the NN tool in this example. Observing Fig. 7, all the values shown at sample 0 are obtained by applying the first set of models created from chamber A and without refreshing. Apparently, the NN VM_I and MR VM_I MAPE values at sample 0 are rather high. That is because the physical property of chamber F is somewhat different from that of chamber A. This point can be verified by inspecting the corresponding GSI value (in Fig. 7), which is extremely high. However, observing sample 1, both NN VM_I and MR VM_I MAPE values are becoming moderate after the first data set of chamber F has been considered for refreshing. Meanwhile, the corresponding GSI value at sample 1 becomes moderate, too. It appears that the refreshing function works.

The corresponding NN_{RT}, MR_{RT} and GSI_{RT} threshold values are 1.58, 1.27, and 3.48, which are calculated from Eqs. (11), (12), and (13), respectively with a = 3. From Fig. 7 we can see that conditions (1) and (2) are satisfied by samples no. 1~3, while condition (3) is not. Therefore, the refreshing procedure continues until all the three conditions are satisfied by samples no. 3~5. In other words, the conjecture accuracy of chamber F is gained after 5 refreshing samples. Now the refreshing process can be terminated and the system is ready to provide all the VM related services.

To verify that the conjecture accuracy of chamber F is gained after completing the refreshing process, chamber F VM conjecturing results with the conjecturing models built by chamber F itself is also performed and displayed in Fig. 8. Observing sample 0 of Fig. 8, because the conjecturing models were built by chamber F itself, the GSI values should be low and the NN VM_I and MR VM_I MAPE values are quite smaller than those of the refreshing case shown in Fig. 7. Further, the NN VM_I MAPE values at samples 5 & 24 of Figs. 7 and 8 are 0.87 & 1.04 and 0.89 & 0.91, respectively; while the MR VM_I MAPE values at samples 5 & 24 of Figs. 7 and 8 are 0.95 & 0.96 and 1.17 & 0.95, respectively. They are almost the same. Also, the GSI curves at Figs. 7 and 8 are similar after sample 5. According to the above observations, it is evident that a complete and successful refreshing process can indeed maintain the VM conjecturing accuracy automatically and efficiently.

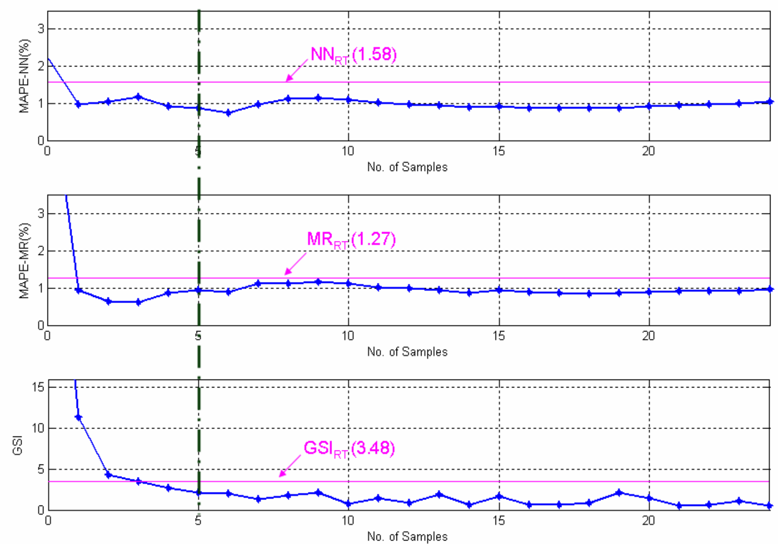

Fig. 7. Automatic VM Model Refreshing with Chamber A's First Set of Models and Performing the Refreshing Process on Chamber F

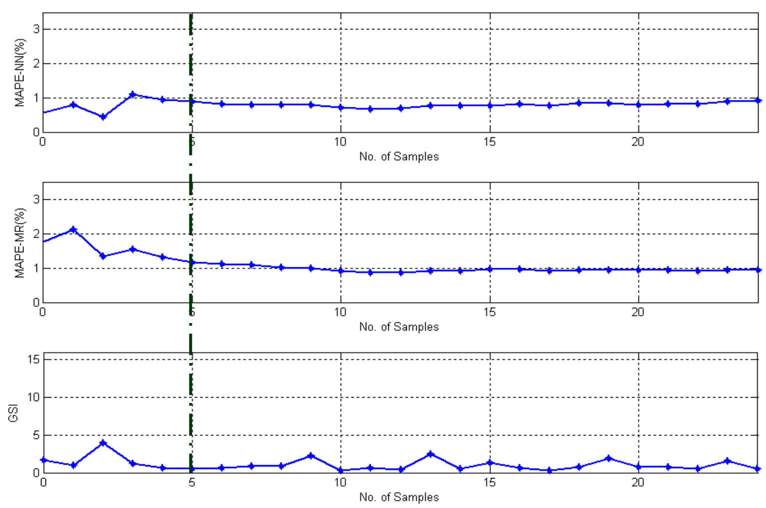

Fig. 8. Chamber F VM Conjecturing Results with the Conjecturing Models Built by Chamber F Itself

VI. SUMMARY AND CONCLUSTIONS

This paper proposes an AVM system for fab-wide VM deployment. The AVM system is composed of a model-creation server and many AVM servers. The AVM system is capable of automatic model fanning out and automatic model refreshing. By applying the AVM system, tremendous model-creation time and labor expenses for fab-wide VM deployment can be greatly reduced. As such, fab-wide VM deployment can become feasible and affordable.

ACKNOWLEDGMENT

The authors would like to thank Chi Mei Optoelectronics Corporation, Ltd. (CMO) at Taiwan for providing the raw data of CVD equipment used in the illustrative example.

REFERENCES

[1] A. Weber, "Virtual Metrology and Your Technology Watch List: Ten Things You Should Know about This Emerging Technology," *Future Fab International*, issue 22, section 4, pp. 52-54, Jan. 2007.

[2] Y.-C. Su, M.-H. Hung, F.-T. Cheng, and Y.-T. Chen, "A Processing Quality Prognostics Scheme for Plasma Sputtering in TFT-LCD Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 183-194, May 2006.

[3] Y.-C. Su, F.-T. Cheng, M.-H. Hung, and H.-C. Huang, "Intelligent Prognostics System Design and Implementation," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 195-207, May 2006.

[4] Y.-J. Chang, Y. Kang, C.-L. Hsu, C.-T. Chang, and T. Y. Chan, "Virtual Metrology Technique for Semiconductor Manufacturing," in *Proc. 2006 International Joint Conference on Neural Networks (IJCNN'06)*, pp. 5289-5293, July 2006.

[5] M.-H. Hung, T.-H. Lin, F.-T. Cheng, and R.-C. Lin, "A Novel Virtual Metrology Scheme for Predicting CVD Thickness in Semiconductor Manufacturing," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 308-316, June 2007.

[6] F.-T. Cheng, Y.-T. Chen, Y.-C. Su, and D.-L. Zeng, "Evaluating Reliance Level of a Virtual Metrology System," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 92-103, February 2008.

[7] Y.-T. Huang, F.-T. Cheng, and Y.-T. Chen, "Importance of Data Quality in Virtual Metrology," in *Proc. of The 32th Annual Conference of the IEEE Industrial Electronics Society (IECON 2006)*, Paris, France, pp. 3727-3732, November 2006.

[8] F.-T. Cheng, H.-C. Huang, and C.-A. Kao, "Dual-Phase Virtual Metrology Scheme," *IEEE Transactions on Semiconductor Manufacturing*, vol. 20, no. 4, pp. 566-571, November 2007.

[9] Y.-C. Su, T.-H. Lin, F.-T. Cheng, and W.-M. Wu, "Implementation Considerations of Various Virtual Metrology Algorithms," in *Proc. 2007 IEEE International Conference on Automation Science and Engineering*, Scottsdale, Arizona, U.S.A., pp. 276-281, September 2007.

[10] H.-C. Huang, Y.-C. Su, F.-T. Cheng, and J.-M. Jian, "Development of a Generic Virtual Metrology Framework," in *Proc. of 2007 IEEE International Conference on Automation Science and Engineering*, Scottsdale, Arizona, U.S.A., pp. 282-287, September 2007.

[11] T.-H. Lin, F.-T. Cheng, A.-J. Ye, W.-M. Wu, and M.-H. Hung, "A Novel Key-variable Sifting Algorithm for Virtual Metrology," in *Proc. 2008 IEEE International Conference on Robotics and Automation*, Pasadena, California, U.S.A., pp. 3636-3641, May 2008.

[12] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, San Francisco, CA: Morgan Kaufman, 2005.

[13] T. W. Anderson, "Asymptotic Theory for Principal Component Analysis," *Ann. Math. Statist.*, vol. 34, pp. 122–148, 1963.

[14] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*, Meboo, 2005.

[15] G. A. Carpenter & S. Grossberg, "ART 2: Self-Organization of Sable Category Recognition Codes for Analog Input Patterns," *Applied Optics*, vol.26, no.12, pp.4919-4930, Dec 1987.