A Comprehensive Big-Data-Based Monitoring System for Yield Enhancement in Semiconductor Manufacturing

Kouta Nakata, Ryohei Orihara, Yoshiaki Mizuoka, and Kentaro Takagi

***Abstract*—In this paper, we focus on yield analysis task where engineers identify the cause of failure from wafer failure map patterns and manufacturing histories. We organize yield analysis task into the following three stages, namely, failure map pattern monitoring, failure cause identification, and failure recurrence monitoring, and incorporate machine learning and data mining technologies into each stage to support engineers' work. The important point is that big data analysis enables comprehensive and long-term monitoring automation. We make use of fast and scalable methods of clustering and pattern mining and realize daily comprehensive monitoring with massive manufacturing data. We also apply deep learning, which has been an innovative core technology of machine learning in recent years, to classification of wafer failure map patterns, and explore its performance in detail. Finally, these machine learning and data mining techniques are integrated into an automated monitoring system with interfaces familiar to engineers to attain large yield enhancement.**

***Index Terms*—Data mining, pattern recognition, machine learning, deep learning, semiconductor defects.**

I. Introduction

COMPETITVE strength in semiconductor field depends on lean manufacturing. In a modern semiconductor fabrication plant, owing to rapid advancement of computers and information technology, a large amount of data is collected and used for yield enhancement. But as manufacturing processes today are too complex and data is too big to handle, it is still difficult for engineers to attain rapid yield enhancement by manually finding informative patterns in raw data.

Figure 1 represents a problem with yield analysis. In a large fabrication, there are dozens of product lines, and there are dozens of failure tests for each line, and dozens of failure patterns for each failure test. There are many manufacturing devices, and among them engineers must find a few devices that are responsible for the failure pattern. Obviously, even skilled engineers could not watch all of failures and find responsible process and device of every failure pattern constantly.

Manuscript received June 21, 2017; revised August 17, 2017; accepted September 5, 2017. Date of publication September 18, 2017; date of current version October 27, 2017. *(Corresponding author: Kouta Nakata.)*

The authors are with the Knowledge Media Laboratory, Corporate Research and Development Center, Toshiba Corporation, Kawasaki 212-8582, Japan (e-mail: kouta.nakata@toshiba.co.jp; ryohei.orihara@toshiba.co.jp; yoshiaki.mizuoka@toshiba.co.jp; kentaro1.takagi@toshiba.co.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSM.2017.2753251

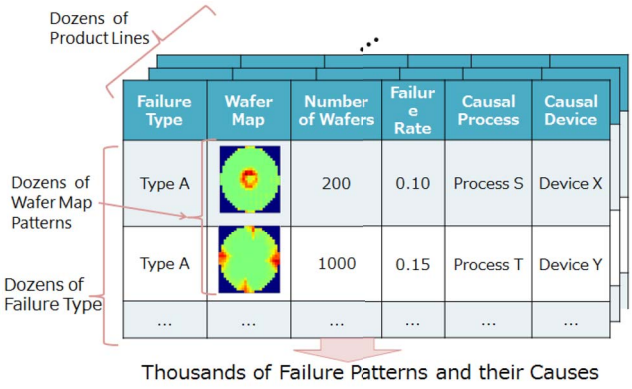

Fig. 1. An Illustration of Difficulty in Comprehensive Monitoring (Red points in wafer map represents failure chips).

In this work, we focus on yield analysis task that engineers identify the cause of failure from wafer failure map patterns and manufacturing histories. We organize yield analysis task into the following three stages, namely, failure map pattern monitoring, failure cause identification and failure recurrence monitoring. We incorporate machine learning and data mining techniques into each stage to support engineers' work. Specially, we make a detailed investigation of deep learning, highly important technique of recent machine learning, for the failure recurrence monitoring stage. We integrate all techniques into a monitoring system to enable comprehensive and long-term monitoring automation.

II. Failure Map Pattern Monitoring by Clustering

Firstly, in order to monitor failure map patterns, we introduce a clustering method. Clustering is one of unsupervised learning approaches which classify objects into groups (clusters) based on their similarities or distances. One of the simplest and most popular clustering methods is K-Means clustering [1]. Clustering approach including K-Means has been used to group wafers with similar failure map patterns (e.g., [2] and [3]). The size of each cluster directly indicates the number of wafers with the failure map pattern, and engineers can understand failure occurrence without checking every wafer by sight.

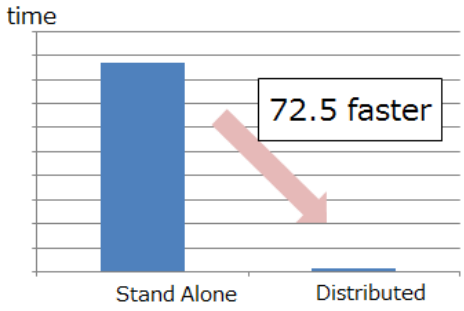

Fig. 2. Runtime of Parallel Distributed Clustering. Standalone represents calculation time on single computational resource, and Distributed represents calculation time on distributed computational resources by K-Means++.

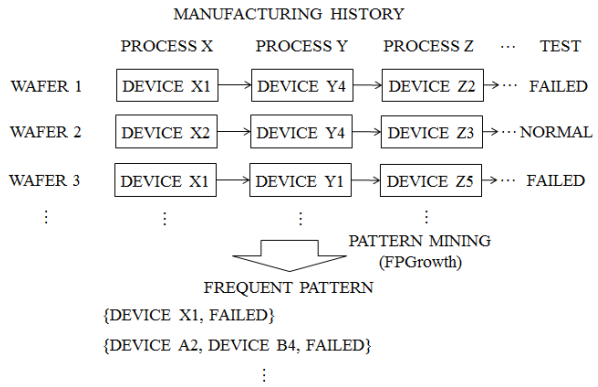

Fig. 3. An Example of Utilization of Pattern Mining. Manufacturing history of each wafer is viewed as a transaction, and pattern mining algorithms extract frequent patterns that appear in histories.

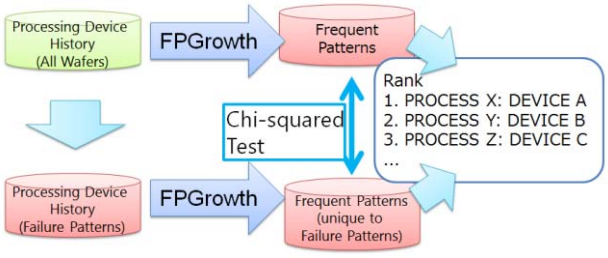

Fig. 4. An Illustration of Cause Identification Utilizing Pattern Mining Techniques. Frequent device patterns are extracted both from all and failed wafer histories, and are ranked in order of p-value of chi-squared test.

In this work, we adapt a distributed clustering method scalable K-Means++ [4]. As stated in Section I, the problem is that there are many wafers, production lines, failure tests and patterns. Since in real-world system clustering should be done by the start of the work, computation time of clustering algorithm is also important. One solution to a large data problem is to parallelize computation on distributed computers. Scalable K-Means++ algorithm distributes dataset to calculation units and provides faster clustering solutions.

By introducing scalable K-Means++, we reduce runtime of clustering for wafers 72.5 times faster with 32 processing units (Fig. 2). Decrease in clustering time enables timely monitoring of all failure tests of all product lines.

III. Cause Identification

Secondly, we utilize a pattern mining method to identify responsible devices of the failure patterns. Usually, engineers investigate huge amount of manufacturing history data, that is, which device have wafers gone through at each process. Intuitively, the more failed wafers are processed in the device, the more it is likely to be the cause of the failure. Engineers focus attention on the processes that are logically or empirically related to the failure type, and investigate devices of the processes through manual works.

Figure 3 represents an example of utilization of pattern mining. Pattern mining algorithms such as Apriori are designed to discover frequent patterns in transactions [19]. One of its most successful applications is to extract frequent patterns of purchase from transactions made by customers in stores. When it is applied to the semiconductor manufacturing, manufacturing history can be viewed as a transaction, and it extracts frequent patterns of devices. In Fig. 3, frequent patterns of devices that are related to the results of tests are extracted.

In order to deal with a large amount of history data, we adopt a pattern mining method FPGrowth ([8], [9]). FPGrowth is an improvement of Apriori and much more efficient for finding complete sets of frequent patterns in a large dataset. FPGrowth is fast and scalable, and its runtime increases linearly with the number of transactions and items while runtime of Apriori increases exponentially. FPGrowth is expected to extract frequent patterns of devices from large datasets of all products and tests in a short time, especially when it is used on distributed computational resources.

Figure 4 is an illustration of cause identification utilizing the pattern mining techniques. In Fig. 4, frequent device patterns are extracted both from all and failed wafer histories. By applying chi-squared test on frequent patterns of failed wafers and arranging the patterns in order of p-value, the method can present major candidates of the causes in ranking form. We have confirmed that our method ranks true responsible devices in the top three for real failure examples.

IV. Failure Recurrence Monitoring

Thirdly, we introduce Deep Learning approach to monitor recurrences of failures. Deep Learning actually means the modeling with neural networks with many hidden layers. Machine learning methods such as deep neural networks or Support Vector Machine have been introduced to classification of failure map patterns in recent works ([5]–[7]). In this work, we follow a standard supervised learning approach. By learning classification models with engineers' supervision on patterns, the models can automatically classify new wafers and indicate long-term trend of failure occurrence.

Although deep neural networks are known as the most successful machine learning method in many research and industrial fields, networks has a large amount of important hyper-parameters to be determined, including network structure itself, and it is not easy to make networks to exhibit

a certain classification performance. We make detailed experiments on neural networks for classification of failure map patterns with real-world dataset.

A. Dataset

In order to adopt deep learning approach to classification of failure map pattern classification, we use complete 2 month's wafer map data collected in real fabrications. In real data, various map patterns could happen. In this paper, we introduce a concrete example of radial pattern in Fig. 1, where failed chip appears in 4 radial directions. In this case, deep learning algorithm learns two-class classification models to distinguish radial map patterns from the others. To simulate the real situation, we divide whole dataset into two halves. We train classification models with the first month data, and evaluate the accuracy of classification with the second month data. Since failed wafers are much fewer than normal wafers in real production line, learning classification models of high performance is harder than in experimental settings where all labels are included equally, as known as class imbalance problems in machine learning.

B. Network Structure

The first thing to do for utilizing neural networks is to determine network structure. Many complex networks have been reported in recent researches (e.g., [14] and many other reviews on deep learning). Recently, owing to improvement of performance of computers or Graphical Processing Units (GPUs), a complex network such as GoogLeNet [16] can be learned with feasible computational resources even in industrial fields.

In this paper, however, we make use of relatively simple Convolutional Neural Network (CNN) structures. CNNs are designed to treat 2D array data and show high performances in image recognition fields (e.g., [15]). Since wafer map patterns can be regarded as simple 2D images, CNNs are promising for their classification.

Table I shows an example of network structure of 5-layer CNN. A 5-layer CNN consists of Input, Convolutional layer, Pooling layer, Full Connection Layer, and Output Layer. In addition to 5-layer CNN, 7-layer CNN is made by adding a set of Convolutional and Pooling layer to 5-layer CNN, and 9-layer is made by adding two sets. As the activation function, we use Rectangle Linear Unit (ReLU) which is known to avoid gradient vanishing problem and make training of deep networks efficient [18].

Table II shows the performance of networks of different size. The performance is evaluated with metrics of F-Measure, Precision, and Recall. Here, Precision is calculated by TP/(TP+FP), Recall is TP/(TP+FN), and F-Measure is the harmonic mean of Precision and Recall, that is,

$$fmeasure = 2 \times \frac{precision \times recall}{precision + recall}$$

TP represents True Positive, the number of radial pattern samples rightly classified, FP represents False Positive, the number of non-radial pattern samples misclassified as radial pattern, and FN represents False Negative, the number of radial pattern samples misclassified as non-radial patterns.

TABLE I
AN EXAMPLE OF NETWORK STRUCTURE (5-LAYERS)

Layers	Patch	Hidden Unit	Activation Function
Input			
Convolutional	5 x 5 x 20		ReLU [18]
Pooling	2 x 2		
Full Connection		500	ReLU [18]
Output			

TABLE II
PERFORMANCE OF NETWORKS OF DIFFERENT SIZE

Layers	F-Measure	Precision	Recall
SVM(Baseline)	0.884	0.889	0.880
5-Layer	**0.920**	**0.934**	**0.905**
7-Layer	0.906	0.930	0.882
9-Layer	0.915	0.931	0.900

In Table II, CNN outperforms Support Vector Machine (SVM) which is one of the most competitive methods other than deep learning. Generally, deeper neural networks are expected to exhibit better performance. But in Table II, shallower 5-layer CNN provides better performance than 7, 9-layer CNNs. This is because failure map patterns are simpler than general image of object recognition tasks, and it can be considered that deeper CNNs are too complex and have too many parameters for wafer map patterns. From this experiment, a deep but simple neural network structure is considered as one of the most powerful options for wafer map pattern classification.

C. Learning Rate Settings

Neural networks has a large amount of hyper parameters to affect performances of resultant models, and learning rate is considered as one of the most influential [17]. Learning rate is a value to determine how quickly parameters of a model are adjusted at each stage of learning and is known to have a huge effect on the performance and convergence speed of learning.

In standard algorithms for neural networks, learning can be considered as an optimization of network parameters to training dataset. Gradient descent is one of the most popular algorithms to perform optimization. Especially, Stochastic Gradient Descent (SGD) which performs a parameter update for every small group (called mini-batch) of training samples is widely used because it is efficient and can avoid trap to local optimum by randomly choosing samples.

SGD needs a constant learning rate which represents how largely an update change the parameters. Learning rate is often determined arbitrarily and difficult to find appropriate settings. In this experiment, we set learning rate to 0.1, 0.01, 0.001, and call them SGD-0.1, SGD-0.01, SGD-0.001.

In addition to using constant learning rate of SGD-0.1, SGD-0.01, SGD-0.001, we evaluate multiple methods to of

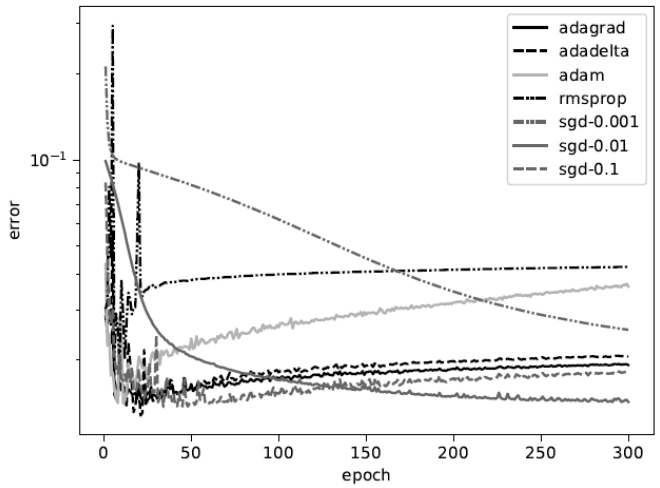

Fig. 5. The Relation between epoch (the number of learning) and test error. Each line represents the method to determine learning rate.

TABLE III
PERFORMANCE OF DIFFERENT LEARNING RATE

Learning Rate	F-Measure	Precision	Recall
SGD-0.1	0.911	0.928	0.895
SGD-0.01	0.897	0.912	0.882
SGD-0.001	0.889	0.935	0.847
AdaGrad	0.906	0.914	0.898
RMSProp	0.873	0.960	0.801
AdaDelta	0.906	0.903	0.908
Adam	0.905	0.907	0.903

TABLE IV
EFFECTS OF DROPOUT

Networks	F-Measure	Precision	Recall
5-Layer-Dropout	0.891	0.919	0.865
7-Layer-Dropout	0.901	0.927	0.877
9-Layer-Dropout	0.877	0.881	0.872

AdaGrad [10], RMSProp [11], AdaDelta [12], and Adam [13] which algorithmically adjust learning rate in the course of learning.

AdaGrad [10] and RMSProp [11] are methods that automatically decrease initial learning rate as learning advances. AdaGrad modifies learning rate based on the past gradient, and performs smaller update for parameters that have been largely updated. RMSProp attempts to fix shortcomings of AdaGrad that learning rate sometime diminish rapidly to 0 in the course of learning. RMSProp has a term to take priority to last gradient update, and prevent learning rate from diminishing.

AdaDelta [12] and Adam [13] are methods that not only decrease but also increase learning rate according to the past update. AdaDelta decreases learning rate as AdaGrad, and in addition modifies learning rate with respect to approximated second order differential of the gradients. With Adadelta, we also do not need to set an initial learning rate, as it has been eliminated from the update rule. Adam updates parameters based on estimation of first and second moment of historical gradients, and takes an idea that learning rate become large in initial steps of learning.

Figure 5 shows the relation between epoch (the number of learning) and error of classification. In machine learning, the performance of model is expected to be improved as optimization as learning advances. But often a model learns detail or noise too much from training data, and the performance to new testing data declines seriously. This phenomenon is known as overfitting. In Fig. 5, it is obvious that learning rate largely affect learning speed and how the model suffers overfitting. All of AdaGrad, RMSProp, AdaDelta, and Adam make learning fast, and learning converges within 100 epochs in all methods, and become largely overfit from then on.

Table III shows the performance of each learning rate and method. We use 5-layer CNN for this experiment. A constant value of learning rate SGD-0.1 is the highest performance, and AdaGrad, AdaDelta, and Adam also works well. When we take both performance and learning speed, AdaGrad, AdaDelta, and Adam is competitive candidate for a practical use. AdaDelta is also advantageous because it does not need initial constant learning rate which is hard to determine.

D. Dropout Technique

Dropout is a technique to improve generalization performance [20]. Dropout randomly inactivate some parts of network and forcibly avoid overfitting of models to detailed or noise of dataset. Generally, dropout works better for complex networks because they tend to learn too much from training data.

Table IV shows effects of dropout. Compared with the results in Table II, the performance is significantly lower even for the most complex 9-layer CNN, and dropout does not work. It is considered that failure map patterns that occur in real production lines are more or less similar each other during a few months and generalization by dropout might not be always necessary for wafer map pattern classification.

E. Model Averaging

Model averaging is a technique that learns multiple models and combines them to classify samples. It is known that an ensemble of models performs better than individual model. In process of investigating hyper-parameters, we have made several competitive classification models. By combining models, we attempt to improve final classification performance.

Table V shows the performance of ensemble models. We ensemble 2 models of 5/9-layer, 4 models of 5/7/9-layer and 7-layer-dropout, and 6 models of 5/7/9-layer and all their dropout models. Ensemble of 2 competitive models performs best. It is interesting that ensemble of 6 models including less competitive 5-layer-dropout and 9-layer-dropout outperforms ensemble of 4 better models. This result implies that ensemble of models of various performance is effective to improve final classification performance.

TABLE V
EFFECTS OF MODEL AVERAGING

Number of models	F-Measure	Precision	Recall
2	0.926	0.939	0.913
4	0.919	0.944	0.895
6	0.921	0.944	0.900

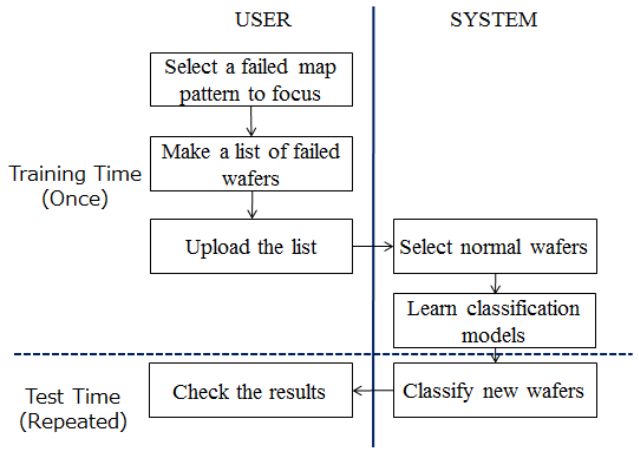

Fig. 6. A Procedure of Learning Failure Patterns User only needs to make a list of failed wafers, which is much easier than labeling all data. The system automatically selects the negative samples (normal wafers) and learns classification models.

F. Learning Procedures

In addition to making high performance classification models with deep learning, we attempt to reduce engineers' labor in a real-world system. Figure 6 shows the procedure of learning, where engineers are assumed to only make a list of failed wafers. Given a list, our system automatically selects normal wafers and starts a learning algorithm. In general supervised learning, users have to give data labels for machine learning algorithms to have them learn from data. But labeling is a painful task and it is not realistic for yield enhancement engineers to label all data as their daily operations. The procedure in Fig. 6 is expected to reduce engineers' labor of labeling for supervised learning since they only need to pick up failed wafers.

In practice, we also have to run many learning processes for many settings of parameters and many kinds of map patterns to make high performance classification models. For this reason, it is necessary for deep learning algorithms to finish calculation of learning models within a practically short time.

It is known that Graphical Processing Units (GPUs) largely accelerate speed of learning in deep learning applications, because deep learning involves a large amount of matrix and vector multiplication that can be parallelized on GPUs. In this work, we reasonably make use of GPUs for learning. Table VI shows the calculation time of CNN calculated with a high performance Central Processing Unit (CPU), and with a GPU. We have confirmed that learning with a large amount of wafers finishes in 4 minutes by utilizing a GPU. This calculation time is short enough for practical use of deep learning.

TABLE VI
CALCULATION TIME OF LEARNING

Number of models	F-Measure	Calculation Time
SVM	0.889	5.0 min
5-layer (CPU)	0.920	3.2 h
5-layer(GPU)	0.920	4.0 min

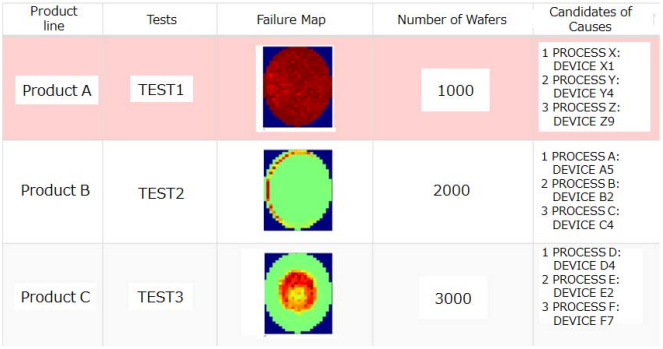

Fig. 7. An Example of Failure Map Pattern Monitoring / Cause Identification Interface. Engineers can see a summary of failures and its responsible candidates in one view.

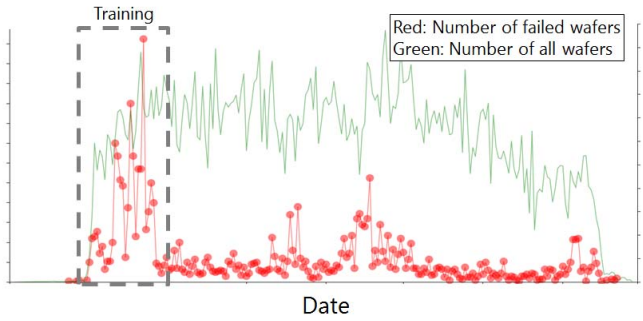

Fig. 8. An Example of Recurrence Monitoring Interface. Models automatically classify failed wafer. Engineers can easily watch long-term trend and identify recurrence of failures as soon as it happens.

V. INTEGRATED SYSTEM

Finally, we integrate all techniques above-mentioned into an automated monitoring system with interfaces familiar to engineers. Figure 7 shows an example of integration of failure map pattern monitoring and cause identification. In Fig. 7, engineers can see a summary of failures and its responsible candidates in one view. If they discover new failure map patterns in the screen, they can start an investigation from the candidates of causes, and if first candidate is actually a cause, they can start taking measures to the device in a very short time. The fast machine learning and data mining algorithms enables daily comprehensive monitoring over all product lines and tests.

Figure 8 shows an example of failure recurrence monitoring. In Fig. 8, engineers trained a classification model with wafers in the period enclosed by dotted line based on the procedure in Fig. 6. Red points represent the number of wafers classified as the failure by the model. Engineers can easily watch long-term trend and identify recurrence of failures as soon as it happens.

Once classification model is learned, new wafers are classified and recurrence is notified automatically for years. Supervised learning approach enables long-term monitoring and reduces risk of missing recurrence of failures.

VI. Conclusion

In this work, we focused on yield analysis task. We incorporated machine learning and data mining technologies into failure map pattern monitoring, failure cause identification and failure recurrences monitoring to support engineers' work. We applied clustering and pattern mining methods of K-Means++ and FPGrowth, and confirmed that these fast and scalable methods enable failure map monitoring and cause identification for real-world massive manufacturing data. We also applied Deep Learning to the classification of wafer failure map patterns. We carefully explored structures and hyperparameters of Deep Learning and showed it also works well in a semiconductor manufacturing application. In addition to applying machine learning and data mining technologies, we considered operational factors such as learning procedures, calculation time, and interfaces to support engineers' work. Our integrated comprehensive "Big Data Based" monitoring system is expected to lead to reduction of engineers' labor and large yield enhancement.

References

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, vol. 1. 1967, pp. 281–297.

[2] G. DeNicolao, E. Pasquinetti, G. Miraglia, and F. Piccinini, "Unsupervised spatial pattern classification of electrical failures in semiconductor manufacturing," in *Proc. Artif. Neural Netw. Pattern Recognit. Workshop*, 2003, pp. 125–131.

[3] C.-F. Chien, W.-C. Wang, and J.-C. Cheng, "Data mining for yield enhancement in semiconductor manufacturing and an empirical study," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 192–198, 2007.

[4] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proc. VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012.

[5] C.-S. Liao, T.-J. Hsieh, Y.-S. Huang, and C.-F. Chien, "Similarity searching for defective wafer bin maps in semiconductor manufacturing," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 953–960, Jul. 2014.

[6] F. Adly *et al.*, "Randomized general regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 2, pp. 145–152, May 2015.

[7] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015.

[8] C. Borgelt, "An implementation of the FP-growth algorithm," in *Proc. 1st Int. Workshop Open Source Data Min. Frequent Pattern Min. Implement.*, Chicago, IL, USA, 2005, pp. 1–5.

[9] L. Zhou *et al.*, "Balanced parallel FP-growth with MapReduce," in *Proc. Inf. Comput. Telecommun.*, 2010, pp. 243–246.

[10] J. Duchi, H. Elad, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[11] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning," Univ. at Toronto, Toronto, ON, Canada, Tech. Rep., 2012.

[12] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 1–9.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[18] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Heidelberg, Germany: Springer, 2012, pp. 437–478.

[19] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, vol. 1215. 1994, pp. 487–499.

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

Kouta Nakata received the Ph.D. degree in science from the University of Tokyo in 2006. He is a Senior Research Scientist with the Knowledge Media Laboratory, Toshiba Corporate Research and Development Center, Japan. He has worked on the developement of data mining, content recommendation, and speech recognition technologies. His current research interests include machine learning, data mining, and their application for semiconductor manufacturing.

Ryohei Orihara received the Ph.D. degree in engineering from the University of Tsukuba in 1999. He has been with Toshiba Corporation, Kawasaki, Japan, since 1988, where he is currently a Chief Research Scientist with the Corporate Research and Development Center. He is currently a Visiting Professor with the University of Electro-Communications, Tokyo, Japan. His current research interests include artificial intelligence, machine learning, data mining, and text mining. He is currently the Vice President of JSAI. He is an IPSJ Fellow.

Yoshiaki Mizuoka received the M.S. degree in computer science from the Tokyo Institute of Technology, Japan, in 2009. He is a Researcher with the Knowledge Media Laboratory, Toshiba Corporate Research and Development Center. He has worked on the developement of machine learning, natural language processing, content recommendation and solutions of speech synthesis. His current research interests include developing engineering solutions for advanced technology yield enhancement.

Kentaro Takagi received the Ph.D. degree in science from Kyoto University, Japan, in 2015. He is a Researcher with the Knowledge Media Laboratory, Toshiba Corporate Research and Development Center. His research interest includes machine learning with special focus on the semi-supervised learning in deep neural network and its applications for semiconductor manufacturing.