

Cycle-Time Key Factor Identification and Prediction in Semiconductor Manufacturing Using Machine Learning and Data Mining

Yair Meidan, Boaz Lerner, Gad Rabinowitz, and Michael Hassoun

Abstract—Within the complex and competitive semiconductor manufacturing industry, lot cycle time (CT) remains one of the key performance indicators. Its reduction is of strategic importance as it contributes to cost decreasing, time-to-market shortening, faster fault detection, achieving throughput targets, and improving production-resource scheduling. To reduce CT, we suggest and investigate a data-driven approach that identifies key factors and predicts their impact on CT. In our novel approach, we first identify the most influential factors using conditional mutual information maximization, and then apply the selective naïve Bayesian classifier (SNBC) for further selection of a minimal, most discriminative key-factor set for CT prediction. Applied to a data set representing a simulated fab, our SNBC-based approach improves the accuracy of CT prediction in nearly 40% while narrowing the list of factors from 182 to 20. It shows comparable accuracy to those of other machine learning and statistical models, such as a decision tree, a neural network, and multinomial logistic regression. Compared to them, our approach also demonstrates simplicity and interpretability, as well as speedy and efficient model training. This approach could be implemented relatively easily in the fab promoting new insights to the process of wafer fabrication.

Index Terms—Cycle time, data mining, machine learning, production management, semiconductor manufacturing.

I. INTRODUCTION

THE SEMICONDUCTOR manufacturing (SM) industry is characterized by a rough competition, which involves cost, quality, and delivery time [1]–[3], and thus continually forces the need to enhance operational performance through increased equipment utilization and productivity. An improved utilization of the fabrication plant (fab) may also lower overall cost. Some previous approaches for production planning and control relied on rules of thumbs and heuristics [2]. Others utilized simulation, statistical analysis methods, analytical methods, and hybrids of the previous approaches [3]. Since today manufacturing lines are being automated and equipped with numerous sensors [4] and the quantities of data typically collected and stored—measured on the order of terabytes per

quarter [5]—are increasing exponentially [6], SM modeling can be intensified, facilitated, and expedited by using machine learning and data mining (MLDM) [1], [2], [4]–[7]. Being exploratory and data-driven, and not assumption-driven, MLDM find hidden, yet potentially valuable information in terms of specific patterns and thus assist decision makers to use the information buried in massive quantities of raw data more efficiently and more effectively [8]. In SM, MLDM can assist, e.g., in the identification of the factors—among many collected routinely in the manufacturing process—that are the most influential on cycle time (CT) reduction. Using these key factors, and as early as possible during the tool operational cycle, MLDM can predict CT and provide insights, as well as practical ways and rules, to understand, control, and reduce CT.

In this paper, and as part of a broader, joint research project between the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel, and Intel, Kiryat-Gat, Israel, as a data-driven approach is developed to transform synthesized fab data into actionable knowledge for the benefit of manufacturing decision support. An MLDM tool, based on conditional mutual information maximization for feature selection and the selective naïve Bayesian classifier (SNBC), is established and applied to CT key factor identification and prediction. We believe that this tool may become an essential and practical mechanism to promote CT reduction in the fab.

A. Scope and Objectives

Among the key performance indicators of any production line, including a fab, CT—calculated as the cumulative time for completion of a work cycle (a single step or a sequence of steps)—is in the center of attention [9]–[14]. Identifying CT key factors and predicting CT are crucial means for reducing CT. Thereby, these means maintain a competitive edge in the SM industry [2], [3], [6], as they promote shorter time-to-market, earlier detection of yield decrease and its origin, cost reduction (e.g., by decreasing the amount of work in process/progress (WIP) along the process), accomplishment of throughput targets, and cut down on capacity waste. Aside from CT reduction, being able to accurately predict CT has a practical contribution for supporting production-planning decisions (such as scheduling resources and actions for anticipated job completion), planning production-capacity requirements

Manuscript received March 9, 2010; revised October 6, 2010; accepted January 18, 2011. Date of publication February 22, 2011; date of current version May 4, 2011. This work was supported in part by the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

The authors are with the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel (e-mail: yair.meidan@gmail.com; boaz@bgu.ac.il; rgadi@bgu.ac.il; michael.hassoun@gmail.com).

Digital Object Identifier 10.1109/TSM.2011.2118775

and assessing the ability to meet customer due dates. In many cases, the delivery lead time, which is primarily dependent on CT, becomes the most important factor in determining whether a semiconductor supplier wins or loses a particular order [11].

A first step toward reducing CT is to understand the mechanisms of its key factors and how they affect CT. This fundamental knowledge could then assist in estimating CT and in targeting these factors for long-term CT reduction and increased equipment utilization and productivity. Among the CT factors most commonly considered in the literature are WIP levels [6], line-bottlenecks [13], rework rates [14], equipment down time [13], [14], mean time between failures/to repair (MTBF/MTTR) [12], setups [12], [13], capacity of available tools [2], dispatch rules [9]–[13], lot priority and scheduling policies [5], [9]–[14], equipment load [10], [12], [13], and product mix [10], [12]. Many of these factors are taken into account as potential CT factors in the current research as well. Many other candidate explanatory factors—possibly summing up to more than a thousand [15]—are available for further analysis in today’s fabs. The human factor (availability, knowledge, training, and so on) is one of them. However, it is left out of our scope, following the tendency in current (fully) automated fabs to minimize the dependence on human operators. Most of the other factors that are out of the scope of the current research concern process data, equipment data, lot priority, and history that are recorded automatically or semi-automatically and accumulated during the fabrication process [1]. Since the research mainly focuses on introducing and applying an original approach of CT key-factor identification and prediction, we leave the possibility of exhaustively examining fab-related factors to a future research. Previous efforts to reduce CT targeted lot dispatching [9], line balancing, process design and layout, material handling [10], and batch size controlling [11].

The current research proposes a novel and effective data-driven approach to identify the factors that are most influential on CT and least inter-correlated. Based on these key factors, we estimate CT. Once validated on genuine fab data, the proposed approach could easily be deployed at the fab and serve as a practical, efficient, and effective means to analyze, control, and reduce CT.

B. Waiting Time

Cycle time consists of processing, inspection, and transportation times, as well as waiting time for equipment, to complete preventive/breakdown maintenance, or due to engineering hold. However, due to the assumption of rigid processing times, on which the industrial engineer usually has no effect, the focus of the current research is set on the reduction of waiting time (WT) rather than the whole CT. From a business point of view, identifying the top WT factors is a step toward eliminating waste in the manufacturing line and a means to narrow down the range of possible actions for reduction of the whole CT.

The WT key factors to be revealed throughout this research are ought to meet several basic requirements. First, they should be able to profoundly assist in distinguishing between differ-

ent levels of operation waiting times. Second, these factors should be useful and feasible to control by the manufacturing engineers. Another desired quality of the factors is for them to be as limited in number as possible for achieving simplicity and generality of the prediction model without compromising on prediction capability.

In the remainder of this paper, we describe data collection in Section II and methods of preprocessing and discretization of the data in Section III. Section IV is devoted to the introduction of our feature selection approach and Section V to modeling. Section VI presents our experimental methodology and also provides results of applying this methodology to the data. In Section VII, we discuss our findings and conclude this paper and in Section VIII we highlight the contribution of the study and discuss future research.

II. DATA COLLECTION

For the simulation of a fab, we generate data using SEMATECH Dataset 1—a common standard for SM data modeling [16]. This simple mechanism of data generation allows controlled analysis of fab factors and their impact on each other in different fab settings and product manufacturing scenarios and for various purposes, such as the understanding of the relationships and trends in a full-scale SM plant [17], identification of the relationship between the X-factor metric and the effective capacity of the line [18], and comparison of scheduling policies [19]. Our model represents a reduced-scale fab of Flash non-volatile memory, in which the tools, routing, products, and flow characteristics are taken from actual fabs, providing numerous fab scenarios for the analysis. This model, implemented using Autosched AP simulation software, is comprised of two products that are different in their routes and characteristics. The numbers of operations (production steps) needed to complete routes one and two are 208 and 243, respectively. The model includes 68 types of 211 tools as well as features of inspection, maintenance, rework, and in-line scrap. The release rate of the 48-wafer lots is constant and stands at about one lot every 3 h for route one and one lot every 6 h for route two, leading to a total of 4000 wafers per week. Using three tuning variables—lot starts (time intervals between releasing new lots into the process), MTBF, and MTTR—we simulate different product mixes and tool availability.

Following 123 simulation scenarios, each containing 451 operations of both routes ($123 \times 451 = 55473$) instances are collected. Each instance, representing an operation, is a 182-dimensional feature (factor, variable) vector. Features are divided into *identifiers* (such as operation or tool ID), *parameters* (infrastructural variables, such as operation position in the process, tool type running the operation, and batch size), *performance measures* (such as means and standard deviations of WT and CT, inter-arrival time, and WIP levels) and *calculated variables* (such as the ratio of tool utilization to availability or expected work load on tools).

While forming the final data set for analysis, we decided to further utilize the simulated data set, and test the assumption that an operation WT depends also on characteristics of previous operations. For example, an operation with a short CT

but a high throughput rate may extend the queue of the next operation, and hence may lengthen its mean WT. Therefore, besides features of the current operation, every instance in the data set also holds values of the same features measured in the two preceding operations. The features of the three operations are regarded as candidate CT factors of the current operation.

Representing two products and more than a hundred scenarios, each having more than four hundred combinations of product-operation, the derived data set—we believe—is large enough to represent a real fab.

III. PREPROCESSING AND DISCRETIZATION OF DATA

Basic preprocessing of the collected raw data set to make it ready for the analysis includes omitting unnecessary experiment-specific identifiers, such as operation ID and tool ID, and WT-related performance measures, such as CT mean and standard deviation. It also consists of reclassifying alphanumeric values (e.g., yes/no) in variables indicating batch operations as nominal (e.g., 1/0) and relabeling of values of categorical features with numeric values (e.g., the values none \start\middle\end of “positioning within rework loop” are changed to 0\1\2\3, respectively). By the end of this stage, the number of features in the data set is reduced from 182 to 155.

In this paper, we predict WT using a version of the naïve Bayesian classifier (NBC) [20], [21]. This is an accurate and efficient model that has many advantages over other models, especially when considering implementation and deployment in the fab. Since NBC is usually applied to discrete rather than continuous data [22], we consider WT classification (as opposed to general prediction) and discretize all continuous variables in the data set. Discretization is the transformation of a continuous-valued variable into a discrete-valued variable, usually by setting certain thresholds on the variable values to separate them to different ranges (“bins”), where each is represented by a discrete value. There is substantial evidence [23] that discretization expedites classifier design, mainly because a discrete variable—having a finite, usually a small, number of values—is easier to model than a continuous variable that theoretically has an infinite number of values. When considering the NBC accuracy, there are studies that demonstrate inferiority of classification based on continuous variables over that based on these variables following discretization [23], [24] and other studies that show the opposite [25]. Nevertheless, our experience shows that any apparent gap, if exists, between accuracies achieved by the two approaches could easily be bridged if we carefully choose the method of modeling the continuous variables or the parameters of discretization. Therefore, we see no risk of losing prediction accuracy by applying NBC to the discretized variables.

Since our generated data set has never been evaluated before, we have no prior knowledge regarding which discretization method or discretization thresholds would be most appropriate for WT prediction and hence evaluate several methods empirically. In addition, we make a distinction between discretization of WT factors (i.e., the independent variables) and discretization of the WT variable itself (i.e., the dependent variable). This distinction is primarily due to

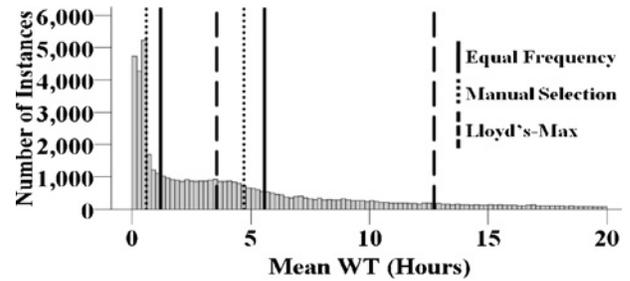


Fig. 1. WT histogram and discretization thresholds derived using equal-frequency binning, manual selection, and Lloyd-Max quantization.

the different roles and distributions of WT and its factors. Following a preliminary study, we determined three levels—low, medium, and high—that reasonably represent WT, could demonstrate our methodology, and may easily be adopted by the fab. Nevertheless, we further recommend optimizing this number of levels for each particular data set and before the deployment in the fab according to the fab needs. In contrary to discretization of the WT variable, there is no reason to limit the discretization of WT factors by the number of possible levels. Therefore, our methodology is flexible and does not depend on this number or on the values of thresholds. Clearly, the fab can easily adapt them to its own needs to meet its common practices and unique constraints.

Numerous approaches [23], [24], [26], [27] have previously been utilized for variable discretization, two of which are selected here for WT factors and three for the WT variable. The first, and one of the simplest discretization methods, is known as *equal-frequency binning*. Given m instances and a user-defined value k , the method divides the range of each feature value into k bins where each bin contains m/k instances of adjacent values. Thus, in our case, this discretization method maintains equal prior probabilities for the three WT levels and, thereby, may also eliminate possible bias of WT classification due to imbalanced data. That is, in cases of acute differences in prior probabilities of WT levels (e.g., if most instances have a single WT value), a WT classifier that is trained using a discretization method other than equal-frequency binning would hardly ever opt for a minority WT class and would, therefore, make it harder to predict this class. Fig. 1 shows a histogram of WT values of our simulated fab data set with indications (continuous lines) of two thresholds created following a three-level equal-frequency binning.

The second method of discretization relies on *manual selection of thresholds*. Thresholds where WT seems to undergo major changes are manually selected to separate between bulks of instances that represent different WT values (dotted lines in Fig. 1). Prior probabilities are no longer equal; however they are not extremely different from one another. The third method we evaluate, known as *L-Level Lloyd-Max quantization* [26], is an algorithm that originated from the field of pulse code modulation. The algorithm finds the thresholds needed to quantize the continuous variable, such that the mean square quantization error between the continuous variable values and their quantized (discrete) values is minimized. The result of applying this algorithm to the simulated fab data is shown in Fig. 1 (dashed lines). When data of a real fab is acquired,

manual modification of the WT thresholds derived by either method may be applied—if necessary—according to actual fab objectives, demands or constraints, e.g., due to durations of operator working shifts.

Concerning WT factors, two methods are considered. One is the *equal-frequency binning* previously described, this time with $k = 10$ bins for each continuous variable. Such a value of k is common in variable discretization using equal-frequency binning. Since this discretization method does not utilize class labels (i.e., the three WT levels) in setting partition boundaries, it is likely that some information necessary for WT classification will be lost as a result of combining values that might be associated with different class labels in the same bin.

As opposed to this unsupervised discretization method [23], the *entropy-based method* is supervised since it utilizes WT labels. According to this method [27], a threshold T_A for a continuous-valued feature A in a data set S is selected among all candidates T_{AS} that partition A into two intervals. The selected T_A is the one that minimizes the weighted average of entropies of the subsets S_1 and S_2 of instances created after the partition

$$E(A, T_A, S) = \frac{|S_1|}{|S|} \cdot H(S_1) + \frac{|S_2|}{|S|} \cdot H(S_2). \quad (1)$$

The weights are ratios of numbers of instances in each subset to the number of instances in the data set before partitioning A according to T_A , and $H(S_j)$ is the class entropy of a subset S_j

$$H(S_j) = - \sum_{i=1}^{N_c} P(c_i, S_j) \cdot \log P(c_i, S_j) \quad (2)$$

where $P(c_i, S_j)$ is the proportion of instances in S_j that belong to class c_i of N_c classes of class variable C .

After employing T_A to partition a continuous feature into two intervals, the algorithm continues and selects the best threshold to each such interval and then recursively creates a multi-interval discretization of the feature. Stopping the recursion to avoid unnecessary splits is performed according to the minimum description length (MDL) principle [28], defined as the minimal number of bits required to uniquely specify discretization based on a particular number of intervals given the data. By penalizing a too large number of intervals, MDL guarantees that the discretization result is the simplest possible according to the data, and thus also provides the lowest probability of erroneous discretization [27]. The result of such discretization is that some areas in the continuous spaces are partitioned very finely whereas others, which have relatively low entropy, are partitioned coarsely. Due to this quality, several studies [23], [27] showed that NBC significantly improves accuracy when continuous-feature discretization is based on entropy minimization.

IV. FEATURE SELECTION

Preliminary experimentation with data sets differing by the number of features raised a couple of challenges. The first was the computation time of SNBC induction, which is exponentially increasing with the number of features. The

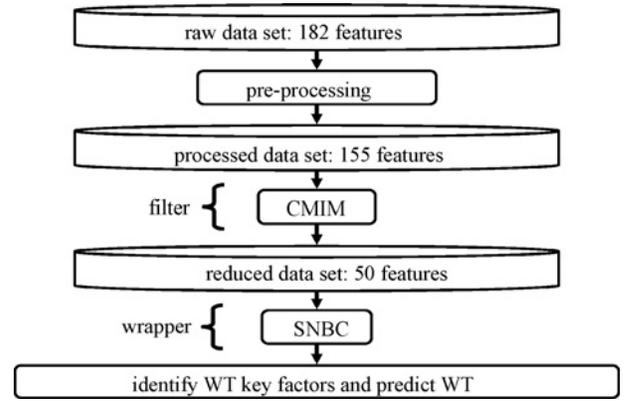


Fig. 2. Methodology of feature selection and classification developed in the research.

other challenge was redundancy and irrelevance of features. Meeting these challenges should also comply with identifying features that are possible to be controlled by the fab. Another study [15] for diagnosis of yield-loss factors in SM reported similar challenges. Moreover, it was clearly seen in preliminary experiments that after selecting an optimal set of features to classify WT, not only that the addition of more features did not contribute to classification accuracy, but it actually offended it. This phenomenon might be an indication that SNBC over-fitted the data, and its generalization ability was negatively affected when too many features were selected as WT factors. The solution to both challenges and to over-fitting is feature selection [8] by which a subset of the features that contribute to correct classification and are independent of each other as much as possible is selected. Not only that this feature subset is smaller and less redundant than the original set but it also usually improves classification performance.

Feature selection methods can be grouped into two types—filters and wrappers [29]. Filters are classifier agnostic, as they are not specific to any type of classification method. Standard filters rank features according to their individual predictive power, which can be estimated by various means [15], [30], [31]. Wrappers evaluate the discriminability of a set of features using classifiers and as part of classification [25], [29]. The main drawback of wrapper methods is their computation-time cost incurred when repeatedly train and evaluate the classifier on each examined feature subset. On the other hand, the main drawback of filters is that they do not necessarily meet both essential aspects of feature selection, i.e., providing accurate classification by a small number of independent features.

To get the best of both worlds, the methodology of the current research (Fig. 2) employs a time-saving filter to extract the top-50 most informative and independent features, followed by a wrapper that combines further feature selection and classification. We describe the filter below and the wrapper, which is basically a classifier, in Section V.

The filter relies on the criterion of conditional mutual information maximization (CMIM) [32]. Using CMIM, we can incrementally select features that carry information about the class that has not been caught by any of the already-selected features. For this purpose, we first consider the *class entropy* quantifying the uncertainty in each of the labels

$(c_1, c_2, \dots, c_{N_c})$ of the class variable C using the class prior probabilities $(P(c_1), P(c_2), \dots, P(c_{N_c}))$

$$H(C) = - \sum_{i=1}^{N_c} P(c_i) \cdot \log P(c_i). \quad (3)$$

After observing a feature U , the uncertainty in the class label becomes conditional on each of the N_U values $(u_1, u_2, \dots, u_{N_U})$ of this feature. The *conditional class entropy* computed using the probabilities of the conditionals reflects the remaining uncertainty of C conditional on U

$$H(C|U) = - \sum_{j=1}^{N_U} P(u_j) \left(\sum_{i=1}^{N_c} P(c_i|u_j) \cdot \log P(c_i|u_j) \right). \quad (4)$$

The conditional class entropy $H(C|U)$ is always less than, or equal to, the class entropy $H(C)$, as U can only reduce the uncertainty in the label of C . Equality exists if and only if the feature and the class are probabilistically independent. The amount by which the uncertainty about the class is decreased by considering the feature is defined as the mutual information (MI) between these two variables

$$\begin{aligned} MI(C; U) &= H(C) - H(C|U) \\ &= \sum_{c_i, u_j} P(c_i, u_j) \cdot \log \frac{P(c_i, u_j)}{P(c_i)P(u_j)}. \end{aligned} \quad (5)$$

In the context of WT prediction, C and U represent, respectively, WT and any of its potential key factors. A factor having a high degree of MI with WT substantially reduces the uncertainty about WT, and contributes to WT prediction [33]–[35]. However, when two such factors are selected and there is a high degree of dependency between them, one of them is redundant to classification. This is since its addition to a feature set that already includes the other factor cannot increase the classification accuracy dramatically while at the same time it may even undermine the accuracy due to increasing the “curse-of-dimensionality” [36]. Therefore, CMIM [32]—the feature selection algorithm we apply—incrementally selects factors that contribute to WT prediction conditioned that they are not dependent on already-selected factors. To measure this conditional contribution, CMIM has to replace MI with conditional MI (CMI) and compute MI between WT and a candidate WT factor U conditioned on any already-selected factor V represented using N_V values $(v_1, v_2, \dots, v_{N_V})$

$$\begin{aligned} CMI(C; U|V) &= H(C|V) - H(C|U, V) \\ &= \sum_{c_i, u_j, v_k} P(c_i, u_j|v_k) \cdot \log \frac{P(c_i, u_j|v_k)}{P(c_i|v_k)P(u_j|v_k)}. \end{aligned} \quad (6)$$

That is, the CMI criterion, reflecting an estimate of the amount of information shared between C and U when V is known, guarantees a good tradeoff between WT discrimination and factor independence.

Within a search for the top- N features, the CMIM algorithm makes certain that a feature similar to those already picked is not selected, even if it is individually predictive. The motivation is that this feature does not carry extra information about the class to predict, other than the information already captured by features previously chosen. Previous experiments

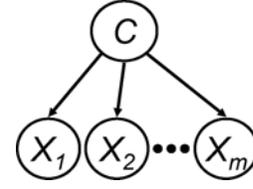


Fig. 3. NBC represented as a Bayesian network.

[32] showed how using CMIM and NBC outperformed other combinations of feature selection and classification methods, while proving robustness when challenged by noisy data sets and achieving error rates similar to, or lower than, those of state-of-the-art classifiers. Finally, as [32] concerned only binary problems (having binary features and two classes), we extend the CMIM criterion to multivariate variables and multiclass problems to make it applicable to WT prediction.

V. MODELING

A classifier assigns an instance—represented by values of a set of features—to one of a finite number of class labels. In this paper, we search for the classifier maximizing the probability of correct classification of a fab operation to a WT label, and also identify those operation features that contribute to the classifier highest accuracy and therefore can be considered as WT key factors.

Previous studies of CT prediction using MLDM in SM relied on decision trees [2], [5], [6], neural networks [2], [5], [6], [10], the k -nearest neighbor algorithm [5], [6], and clustering [2], [6]. We suggest here (Fig. 2) a different approach based on preliminary feature selection using CMIM (the filter) followed by further feature selection and classification using a version of NBC called SNBC (the wrapper). SNBC [21] runs NBC [20] while conducting a greedy forward selection search over the space of features. Starting from an empty feature subset, SNBC incrementally adds to the current subset the feature, among all unselected features, that maximizes the NBC classification accuracy of the subset. NBC is regarded as an accurate MLDM classifier [22], [25], [37], and although not considered as accurate as a neural network or as intuitive as a decision tree, it has advantages in terms of simplicity, training and classification speeds, and problem scalability [21], [22], [37].

NBC can be presented (Fig. 3) as a Bayesian network in which observed variables X_1, X_2, \dots, X_m are independent of each other given the class variable C [20]. That is, the network observed variables are represented by nodes onto which all arcs from the class variable (node) are directed. For WT classification, the class and the observed variables represent, respectively, WT and its factors.

During training and using the data, the NBC parameters are estimated by the class-conditional probability distributions $P(x_j|c_i)$ for each value x_j of variable X_j and each label c_i of the class variable and by the label priori probabilities $P(c_i)$. A test instance I , represented as a conjunction (x_1, x_2, \dots, x_m) of m values of the observed variables in the form $(\wedge x_j)$ is classified using Bayes’ theorem

$$P(c_i|I) = \frac{P(c_i) \cdot P(I|c_i)}{P(I)} = \frac{P(c_i) \cdot P(\wedge x_j|c_i)}{\sum_k P(\wedge x_j|c_k) \cdot P(c_k)}. \quad (7)$$

Due to the (naïve) independence assumption, the computation of the m -dimensional likelihood, $(\wedge x_j | c_j)$, can be decomposed by a product of m 1-D class-conditional probabilities $(x_j | c_j)$. Hence, the class-posterior probability (7) can be estimated based on

$$P(c_i | I) = \frac{P(c_i) \cdot \prod_j P(x_j | c_i)}{\sum_k \prod_j P(x_j | c_k) \cdot P(c_k)} \quad (8)$$

and the predicted WT is identified by the class c^* achieving the highest *a posteriori* probability, $P(c^* | I)$.

By decomposing the likelihood, NBC gains simplicity, scalability, and the possibility of speedy training, which may contribute to fast and efficient deployment of the model in the fab. The robustness of the model—demonstrated in almost no need in setting parameters or applying heuristics while training the classifier—makes this deployment even easier. Another advantage is that NBC results are intuitive and easy for human interpretation. In addition, the above advantages may prevail even when confronting the rapidly changing environment of the fab if the model is allowed to re-train. Finally, since the naïve assumption of feature conditional independence rarely holds for natural data sets and may therefore undermine the NBC classification accuracy, our methodology feeds SNBC only with features that CMIM has previously considered as both discriminative and independent to each other.

VI. METHODOLOGY AND RESULTS

In the first stage, we implemented in MATLAB the CMIM, NBC, and SNBC algorithms as well as a code for executing the experimental methodology (Fig. 2). In the second stage, we performed preliminary feature selection using the CMIM algorithm to reduce the problem 155-dimensionality, and thereby facilitate the SNBC operation. Results of initial runs revealed that the minimal number of features that were needed to reach the maximal accuracy over a validation set— independent of the training set—was approximately 47. Hence, it was decided to perform preliminary feature selection using CMIM to extract the top-50 of the available features, and consider only them in the stage of modeling.

Fig. 4 shows CMI scores (6) for the first 50 iterations of the CMIM algorithm in which the first 50 individual features achieving the highest CMI values using the validation set are selected. CMI scores are averaged over ten runs, each using 10 000 random instances for training and another 10 000 random instances for validation of the CMIM algorithm. Having a maximal value of MI with WT class, the “number of operations carried out by the tool” is the first feature selected for WT classification. The second selected feature is the “mean time in processing state” that has the maximal value of MI with WT class conditioned on the “number of operations carried out by the tool” that has already been selected. Further, the most discriminative and least redundant features—relative to already-selected features—are (in this order) “tool availability,” “number of previous tool loops,” “standard deviation of the percent of tool’s time in down state,” and “load.” As expected from CMIM, Fig. 4 keeps descending monotonically, reflecting the diminishing effect of

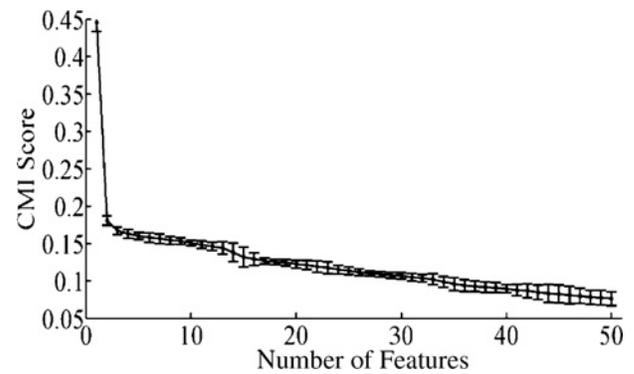


Fig. 4. Means and 95% confidence intervals of CMI scores for features selected during the first 50 iterations of the CMIM algorithm.

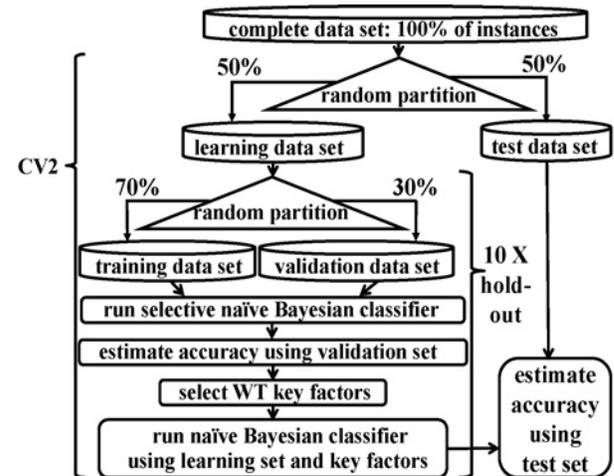


Fig. 5. Methodology of experiments.

marginal information to WT class added by additional features. The confidence intervals measured for CMI values are narrow, indicating the stability of the algorithm.

At the end of this stage, narrowing the list of candidate WT factors down from 155 to 50 features by using the CMIM filter, the data set is more suitable to be processed by the wrapper that further selects features, as part of WT classification.

In the third stage, which is modeling, we establish a methodology (Fig. 5) by which we perform two loops of experiments. In the outer loop, we perform 2-fold cross-validation (CV2). That is, we randomly partition the data set into two equal subsets. We use the first subset for learning a model and the second subset for testing the trained model and computing its classification accuracy. This accuracy is the percentage of instances (operations) for which the WT class predicted by the model is equal to the actual WT class. We then change the roles of the two subsets and repeat the procedure, before averaging the two test accuracies and reporting this average accuracy.

In the inner loop of the CV2 experiment, and using the learning subset, we select the WT key factors and the model that uses these factors to predict WT class most accurately. For this purpose, we first randomly partition the learning set into a training set having 70% of the learning set and a validation set having the remaining 30% of the data using

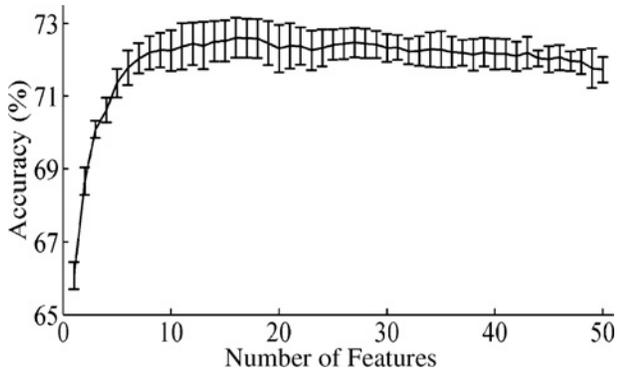


Fig. 6. SNBC classification accuracy (mean \pm std.) for increasing numbers of features, when applying equal-frequency binning of WT and entropy-based discretization of WT factors for a fab manufacturing both products.

a hold-out procedure [36]. The training set is used to train the SNBC algorithm whereas the validation set is used to stop the algorithm when classifier accuracy ceases to increase. The stopping point determines a WT key factor set. Using these factors and the entire learning set, an NBC is trained and then tested on the test set. This training-validation-test procedure repeats for 10 random 70/30 partitions of the learning set, and then the mean test classification accuracy (and its standard deviation) are recorded and a majority vote over the 10 sets of WT key factors is performed to determine the WT key factors reported for this fold of the CV2. Finally, the experiment repeats with the learning and test sets exchanged and the final results are averaged over the two folds of the CV2 methodology and reported.

Fig. 6 demonstrates the results of applying the above methodology to find the optimal number of WT key factors and estimate their accuracy in WT class prediction in a fab manufacturing both products (Section II). WT is discretized using equal-frequency binning and the WT factors are discretized using the entropy-based method (Section III). The figure shows that the SNBC classification accuracy measured using a validation set for increasing numbers of selected features peaks for 20 features.

We repeat the same procedure for a fab manufacturing both products using any of the combinations of WT discretizations (i.e., equal frequency, manual selection, and Lloyd's) and WT factor discretizations (i.e., equal frequency, and entropy-based). Fig. 7 shows similar graphs to that in Fig. 6 for all six combinations of discretization methods. Accuracies in all cases increase steeply when adding the first key factors and then, after including the first 10–30 factors (depending on the discretization methods), are stabilized before slightly decline. The peak performance in each graph for a certain number of features, clearly demonstrates how a model having too few features under fits the data and a model having too many features over fits the data. In both cases, performance is inferior to performance using the optimal number of key factors. In addition, Fig. 7 demonstrates superiority of equal-frequency binning of WT (upmost lines) with respect to WT classification accuracy.

Test results for equal-frequency binning applied to both WT and its factors and each of the products, as well as

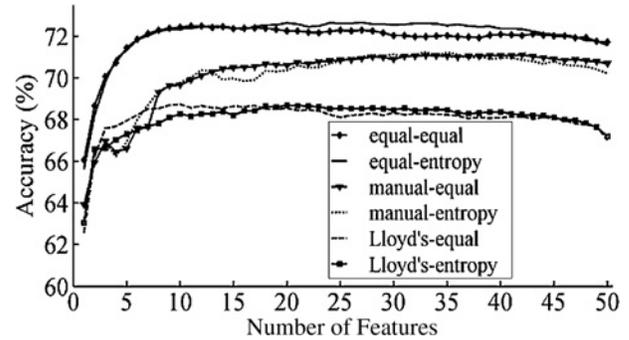


Fig. 7. SNBC mean classification accuracy for increasing numbers of features and a fab manufacturing both products. The graphs exhibit different combinations of discretization methods.

TABLE I
CLASSIFICATION PERFORMANCE

Products	WT Discretization	Factor Discretization	# of Selected Key Factors	Accuracy Mean (std) (%)
First	Equal	Equal	25.3 (13.8)	75.3 (0.4)
Second	Equal	Equal	36.7 (7.1)	73.3 (0.2)
Both	Equal	Equal	18.5 (9.1)	72.2 (0.4)
Both	Equal	Entropy	20.3 (7.8)	72.6 (0.5)
Both	Manual	Equal	29.7 (9.2)	70.7 (0.4)
Both	Manual	Entropy	29.0 (7.0)	70.6 (0.4)
Both	Lloyd's	Equal	21.8 (6.0)	64.7 (0.6)
Both	Lloyd's	Entropy	15.5 (7.7)	64.6 (0.8)

for all six combinations of discretizations for manufacturing both products, are presented in Table I. Results show that although WT models for a single product are based on more features than those for both products, they are also more accurate. This matches expectation that modeling the behavior of a single product would be more accurate than that of two non-identical products. Table I also manifests that WT discretization using equal-frequency binning is advantageous to other WT discretization methods, and that the discretization method applied to WT factors has no significant effect on classification accuracy.

The last result matches results of empirical studies [23], [24] of a variety of discretization methods that show that entropy-based methods perform slightly better in some data sets, but on average, all discretization methods perform roughly the same. We note that these conclusions are limited to discretization of the independent variables but not of the dependent variable. Indeed, Table I and Fig. 7 demonstrate that WT (our dependent variable) is affected by the method of discretization. In addition, Fig. 7 shows that for each case there is a range of sizes of possible feature subsets that provide near optimal performance. For example, though a model comprising 20 features is selected for both products and equal-entropy discretization, a model comprising, say ten features is not inferior significantly. This manifests the robustness of our feature-selection methodology.

In the last stage of the study, the SNBC classification accuracy is compared, using the same experimental methodology, with those of two of the most common MLDM classifiers—C5.0 decision tree and artificial neural network (ANN)—

TABLE II
CLASSIFICATION ACCURACIES ACHIEVED BY COMPETITIVE MODELS FOR
A FAB MANUFACTURING TWO PRODUCTS

Classifier	Average (%)	StDev (%)
ANN	73.2	0.7
MLR	73.2	0.3
SNBC	72.6	0.4
C5.0	72.3	0.4

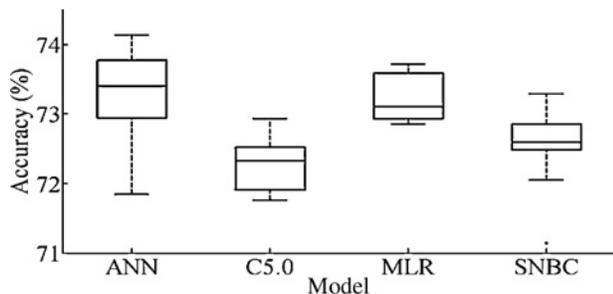


Fig. 8. Box plot for classification accuracies of the four models.

TABLE III
ANOVA TABLE FOR CLASSIFICATION ACCURACIES ACHIEVED BY THE
ANN, MLR, SNBC, AND C5.0 CLASSIFIERS

Source of Variation	Sum of Squares	df	Mean Square	F Value	P Value	F Critical
Between groups	14.55	3	4.85	19.35	0.000	2.725
Within groups	19.04	76	0.25			
Total	33.59	79				

and with that of multinomial logistic regression (MLR)—generalizing logistic regression to more than two outcomes. Note that ANN and decision trees are also the most popular MLDM models used for CT prediction in SM in Section V. All models are trained using the same learning sets and top-50 features selected using CMIM, and their classification accuracies are evaluated using the same test sets. Table II shows the means and standard deviations of the test classification accuracies of the four models when the fab manufactures the two products.

Fig. 8 provides a richer representation for the comparison using a box plot manifesting graphically for each model the sample minimum, lower quartile, median, upper quartile, and maximum.

All classifiers seem to perform comparably well, where ANN and MLR are superior to SNBC (due to its naive assumption of variable conditional independence) and C5.0, and SNBC is superior to C5.0. Results of ANOVA (Table III) prove that not all models perform the same (P -value < 0.0001). Fisher's least significant difference procedure—a two-step test for pairwise comparisons made following ANOVA—reveals that ANN and MLR accuracies are significantly higher than that of SNBC, and that SNBC accuracy is significantly higher than that of C5.0.

TABLE IV
DIFFERENCES BETWEEN MODELS EVALUATED, AS REVEALED IN THE
STUDY

Classifier Property	ANN	MLR	SNBC	C5.0
Accuracy	Supreme	Supreme	Good	Least good
Model complexity	High	Low	Low	Moderate
Speed of training	Long	Fast	Very fast	Average
Human interpretation	Difficult	Moderate	Easy	Moderate
Inherent feature selection	Not trivial	Not existed in factorial	Simple and Efficient	Moderately efficient
Scalability	Not good	Good	Supreme	Poor
Attractiveness to deployment	Moderate	Moderate	High	High

VII. DISCUSSION AND CONCLUSION

Besides achieving a classification accuracy which is comparable to those of state-of-the-art classifiers, SNBC has several advantages over the other classifiers. First is SNBC overwhelming scalability—that is the classifier ability to keep its accuracy for increasing numbers of variables—more than any other classifier. This is a result of likelihood decomposition by NBC [8]. Second is SNBC computational and time complexities, which are lower than those of the other classifiers. Third is the inherent feature selection the classifier performs. It allows the classifier to reduce the size of the feature set to 20 WT key factors for optimal accuracy and even to around 10 factors with almost no performance degradation. ANN, for example, needs all 50 features, two hidden layers, and an intensive training period. Reducing the list of 50 features to several key factors is not immediate in ANN and needs exploration of the network parameters. The list of key factors derived using C5.0 is shorter than that of ANN but the tree depth is of around 9 levels undermining quick and easy interpretation of the mechanisms contributing to WT. Also MLR finds all 50 factors important for prediction (stepwise mode was disabled by the full factorial mode) and yields a too-complex model. Table IV summarizes the differences between the four models as found in this paper (and not necessarily in general), and demonstrates SNBC advantages. These advantages of SNBC make the model a recommended WT predictor for SM.

Table I shows that the most accurate SNBC model comprises about 20 features on average (though a model comprising ten features is not too inferior). Out of which, the ones selected first by the model and hence are considered most influential on WT (a.k.a. “WT key factors”) are the “frequency of tool idleness” (waiting for operator or material), “tool availability” (found influential also in [12] and [14], indicating that WT decreases with tool availability), “number of previous tool loops” (re-entrant behavior), “number of operations carried out by the tool,” and “standard deviation of the queue length” (discussed in the context of WIP in [2], [5], [11]–[14], and [38]).

Fig. 9 presents a distribution of WT levels [low (1), medium (2), and high (3)] for increasing levels of “tool availability” from 1 to 10. The figure shows that, as expected, high levels of availability mainly lead to low WT, whereas low levels of availability relate primarily to medium to high

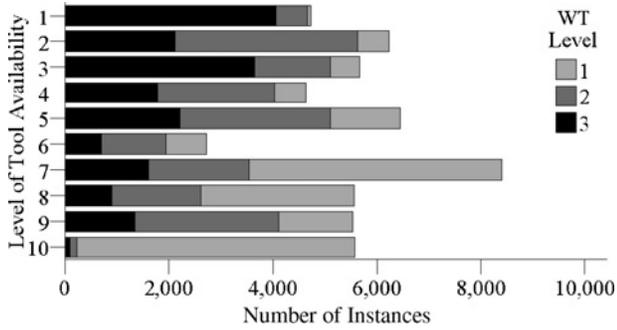


Fig. 9. WT distribution for increasing tool availability levels.

WT levels. The very strong correspondence between level 1 of tool availability (and for lesser correspondence levels 2–4) and high WT and between level 10 (and other high levels) of tool availability and low WT, provides some explanation to why “tool availability” is ranked second on the list of top WT key factors. Also “tool availability” of previous operation is ranked high (seventh place).

The “number of such tools in the fab” [38] is also found influential on WT. Interestingly however, the two corresponding factors, ranked ninth and eighth on the list of the top 20 WT key factors, are of the previous operation and the one before, respectively, rather than of the current operation. Moreover, concerning these two previous operations, it is found that of the top-20 factors, 12 describe the current operation, 6 describe its predecessor (4 of which were among the top-10) and 2 factors belong to the former operation (both in the top-10). Apparently, previous operations do affect the average WT of the current operation more significantly than we expected.

Regarding the nature of the 20 extracted WT key factors, the trained model has a 50%–50% mixture of infrastructural parameters (which are available before the simulation run) and performance measures (available only after the run). Among the top-10 factors, 80% are infrastructural parameters and only 20% are performance measures, either of the current operation or its predecessors. For a manufacturing engineer, it would certainly be preferable to obtain a model for decision support that consists of as many factors as possible that their values are known in advance and can be further controlled even before the operation has started.

VIII. CONTRIBUTION AND FUTURE RESEARCH

This paper aims at introducing a novel and complete MLDM-based methodology for CT (WT) key factor identification and prediction in SM. It shows how uncertainty in WT estimation could decrease by using MLDM methods while employing only fraction of the available factors, which interestingly also include factors of past operations. From a base line of 33.3% prior probability for any WT level (low, medium, or high), the learned model has reached an average accuracy of 72.6% on a test set of unseen-before operations, reflecting an improvement of nearly 40% in WT estimation accuracy. In addition, the research demonstrates the

TABLE V
FEATURES REMOVED DURING PRELIMINARY FILTERING

Feature Description	Associated Operation(s)	Filtering Reason
Index of experiments' set	Any	Identifier
Layer index	Any	Identifier
Machine/tool type name	Any	Identifier
Operation index	Any	Identifier
Operation name	Any	Identifier
Product type (1 or 2)	Any	Identifier
Simulation run index	Any	Identifier
Tool-set index	Any	Identifier
X Factor: CT/process time	Current, segment	Performance measure
Mean CT	Current, segment	Performance measure
Mean WT	Segment	Performance measure
St.dev. CT	Segment	Performance measure
St.dev. operation's CT	Current	Performance measure
St.dev. WT	Current, segment	Performance measure

contribution of CMIM and SNBC to accurate classification of WT.

The research also lays practical and methodological grounds for a future research, which can be extended to other target functions (such as X-factor or yield-oriented performance indicators) and more complex fab environments having several products with various versions. As the research assumes a truthful representation of a fab by the simulated data, its resultant classification models and conclusions are yet to be validated on genuine data, collected through routine manufacturing. Once validated, in addition to assisting semiconductor manufacturers in better focusing their resources on future CT reduction efforts, the models could easily be integrated into manufacturing lines, automatically collecting fab-wide data and processing them into WT estimations. In order to maintain validity over time, it is recommended to periodically refresh the models using current data by re-running CMIM and updating the conditional probabilities table of the selected key-factors. As NBC is computationally inexpensive, it can even serve as a basis for real-time decision support system in manufacturing, identifying potential delays in production and their causes, as well as better assessing due dates and fundamentally enhancing production planning. Finally, semiconductor manufacturers can effortlessly adopt the modular structure of our methodology and have it customized to their own special needs, restrictions, and objectives, e.g. by means of adjusting the number of levels and the values of thresholds for discretization, as well as the selection of the analyzed features to begin with.

APPENDIX FEATURES IN THE DATA SET

See Tables V–VII.

TABLE VI
FEATURES REMOVED BY CMIM

Feature Description	Associated Operation(s)
Number of times that the lot has been previously processed by a tool of that type	Any
Distance from last bottleneck	Any
Distance to next bottleneck	Any
Distance from last processing by such tool	Any
Distance to next processing by such tool	Any
Distance from line start	Any
Distance to line end	Any
Existence of batch size restrictions	Any
Lot scrapping probability	Any
Wafer scrapping probability	Any
Mean inter-arrival time to operation	Any
Mean percent of tool's time in down state	Any
Mean percent of tool's time in setup state	Any
Mean time in setup state	Any
Minimum batch size (diffusion)	Any
Maximum batch size (diffusion)	Any
Number of products using that tool	Any
Positioning within rework loop (lithography)	Any
St.dev. inter-arrival time to operation	Any
St.dev. percent of tool's time in setup state	Any
St.dev. time in setup state	Any
St.dev. tool type availability	Any
X factor: CT/process time	Prev., before prev.
Load (manufacturing rate)/(maximum possible rate)	Before prev.
Availability's coefficient of variation	Prev., before prev.
Frequency of tool's down state	Prev., before prev.
Mean availability of tool type	Prev., before prev.
Mean CT	Before prev.
Mean percent of time in material loading state	Prev., before prev.
Mean percent of time in unloading state	Prev., before prev.
St.dev. operation's CT	Prev., before prev.
St.dev. percent of time in idleness state	Before prev.
St.dev. percent of time in material loading state	Prev., before prev.
St.dev. percent of time in processing state	Prev., before prev.
St.dev. percent of time in unloading state	Prev., before prev.
St.dev. percent of tool's time in down state	Before prev.
St.dev. queue length	Before prev.
St.dev. time in processing state	Before prev.
St.dev. tool's down state frequency	Current, before prev.
St.dev. WT	Prev., before prev.
Tool availability	Before prev.
Utilization/availability	Before prev.

TABLE VII
WT KEY FACTORS, ORDERED BY CMIM AND SNBC SELECTION

Feature Description	Associated Operation	CMIM	SNBC
Number of operations carried out by the tool	Current	1	4
Mean time in processing state	Current	2	11
Tool availability	Current	3	2
Number of previous tool loops	Previous	4	3
St.dev. percent of tool's time in down state	Current	5	18
Load (manufacturing rate)/(maximum possible rate)	Current	6	14
Mean percent of tool's time in processing state	Current	7	-
Number of previous tool loops	Before prev.	8	6
Mean time in processing state	Previous	9	13
Mean percent of time in idle state	Current	10	16
St.dev. queue length	Current	11	5
Mean percent of tool's time in processing state	Previous	12	-
Mean queue length	Current	13	-
Mean time in processing state	Before prev.	14	-
St.dev. time in processing state	Previous	15	-
St.dev. percent of time in processing state	Current	16	-
Mean percent of tool's time in processing state	Before prev.	17	-
St.dev. percent of idle time state	Current	18	17
Utilization/availability	Current	19	-
Number of such tools in the fab	Before prev.	20	8
Frequency of tool idleness	Before prev.	21	-
Mean percent of time in material loading state	Current	22	19
Frequency of tool idleness	Current	23	1
St.dev. time in processing state	Current	24	-
Mean queue length	Before prev.	25	-
Mean percent of time in unloading state	Current	26	-
St.dev. percent of time in unloading state	Current	27	15
St.dev. percent of time in material loading state	Current	28	-
Number of such tools in the fab	Current	29	-
Mean weekly number of station setups	Current	30	-
St.dev. tool's idleness frequency	Current	31	-
Mean queue length	Previous	32	-
St.dev. queue length	Previous	33	-
St.dev. tool's idleness frequency	Previous	34	-
Number of such tools in the fab	Previous	35	9
Load (manufacturing rate)/(maximum possible rate)	Previous	36	-
St.dev. tool's idleness frequency	Before prev.	37	-
Mean CT	Previous	38	12
Mean availability of tool type	Current	39	-
Utilization/availability	Previous	40	-
St.dev. percent of tool's time in down state	Previous	41	-
Mean percent of time in idle state	Previous	42	-
Frequency of tool's down state	current	43	-
Tool availability	Previous	44	7
Number of operations carried out by the tool	Previous	45	10
Mean percent of time in idle state	Before prev.	46	-
Availability's coefficient of variation	Current	47	20
St.dev. percent of time in idle state	Previous	48	-
St.dev. tool's down state frequency	Previous	49	-
Frequency of tool idleness	Previous	50	-

REFERENCES

- [1] 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.
- [2] C. F. Chien, C. W. Hsiao, C. Meng, K. T. Hong, and S. T. Wang, "Cycle time prediction and control based on production line status and manufacturing machine learning," in *Proc. IEEE Int. Symp. Semicond. Manuf.*, Sep. 2005, pp. 327–330.
- [3] S. H. Chung and H. W. Huang, "Cycle time estimation for wafer fab with engineering lots," *IIE Trans.*, vol. 34, no. 2, pp. 105–118, Feb. 2002.
- [4] R. Ganesan, A. N. V. Rao, and T. K. Das, "A multiscale Bayesian SPRT approach for online process monitoring," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 3, pp. 399–412, Aug. 2008.
- [5] R. Goodwin, R. Miller, E. Tuv, A. Borisov, M. Janakiram, and S. Louchheim, "Advancements and applications of statistical learning/machine learning in semiconductor manufacturing," *Intel Technol. J.*, vol. 8, no. 4, pp. 324–337, Nov. 2004.
- [6] P. Backus, M. Janakiram, S. Mowzoon, G. C. Runger, and A. Bhargava, "Factory cycle-time prediction with a data-mining approach," *IEEE Trans. Semicond. Manuf.*, vol. 19, no. 2, pp. 252–258, May 2006.
- [7] S. C. Hsu and C. F. Chien, "Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing," *Int. J. Prod. Econ.*, vol. 107, no. 1, pp. 88–103, May 2007.
- [8] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.
- [9] Y. Narahari and L. M. Khan, "Modeling the effect of hot lots in semiconductor manufacturing systems," *IEEE Trans. Semicond. Manuf.*, vol. 10, no. 1, pp. 185–188, Feb. 1997.
- [10] D. Y. Liao and C. N. Wang, "Neural-network-based delivery time estimates for prioritized 300 mm automatic material handling operations," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 3, pp. 324–332, Aug. 2004.
- [11] S. Johal, "Application report: Simulation reduces product cycle time," *Semicond. Int.*, vol. 21, no. 4, pp. 101–102, Jan. 1998 [Online]. Available: http://www.semiconductor.net/article/206898-Application_Report_Simulation_Reduces_Product_Cycle_Time.php
- [12] D. Fronckowiak, A. Peikert, and K. Nishinohara, "Using discrete event simulation to analyze the impact of job priorities on cycle time in semiconductor manufacturing," in *Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf. Workshop*, vol. 12, no. 14, Nov. 1996, pp. 151–155.
- [13] V. Ramamurthi, M. E. Kuhl, and M. E. Hirschman, "Analysis of production control methods for semiconductor research and development fabs using simulation," in *Proc. 37th Conf. Winter Simul.*, 2005, pp. 2177–2185.
- [14] E. Akcali, K. Nemoto, and R. Uzsoy, "Cycle-time improvements for photolithography process in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 14, no. 1, pp. 48–56, Feb. 2001.
- [15] C. M. Fan and Y. P. Lu, "A Bayesian framework to integrate knowledge-based and data-driven inference tools for reliable yield diagnoses," in *Proc. 40th Conf. Winter Simul.*, 2008, pp. 2323–2329.
- [16] SEMATECH, "Modeling data standards, version 1.0," Sematech, Inc., Austin, TX, Tech. Rep., 1997.
- [17] J. Hunter, D. Delp, D. Collins, and J. Si, "Understanding a semiconductor process using a full-scale model," *IEEE Trans. Semicond. Manuf.*, vol. 15, no. 2, pp. 285–289, May 2002.
- [18] D. Delp, J. Si, Y. Hwang, and B. Pei, "A systematic study of the extended X-factor in relation to effective system capacity," *J. Manuf. Syst.*, vol. 24, no. 4, pp. 289–301, Dec. 2005.
- [19] J. G. Dai and S. Neuroth, "DPPS scheduling policies in semiconductor wafer fabs," in *Proc. Int. Conf. Modeling Anal. Semicond. Manuf.*, 2002, pp. 194–199.
- [20] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. 10th Natl. Conf. Artif. Intell.*, 1992, pp. 223–228.
- [21] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proc. 10th Conf. Uncertainty Artif. Intell.*, 1994, pp. 399–406.
- [22] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 131–163, Nov.–Dec. 1997.
- [23] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. 12th Int. Conf. Mach. Learning*, 1995, pp. 194–202.
- [24] C. N. Hsu, H. J. Huang, and T. T. Wong, "Implications of the Dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers," *Mach. Learning*, vol. 53, no. 3, pp. 235–263, Dec. 2003.
- [25] R. Malka and B. Lerner, "Classification of fluorescence in situ hybridization images using belief networks," *Pattern Recognit. Lett.*, vol. 25, no. 16, pp. 1777–1785, Dec. 2004.
- [26] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
- [27] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, vol. 13, Sep. 1993, pp. 1022–1027.
- [28] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, Sep. 1978.
- [29] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
- [30] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, Oct. 2000.
- [31] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. 6th Pacific Rim Int. Conf. Artif. Intell.*, 2000, pp. 679–689.
- [32] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learning Res.*, vol. 5, pp. 1531–1555, Nov. 2004.
- [33] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.
- [34] B. V. Bonnländer and A. S. Weigend, "Selecting input variables using mutual information and nonparametric density estimation," in *Proc. Int. Symp. Artif. Neural Netw.*, 1994, pp. 42–50.
- [35] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *J. Mach. Learning Res.*, vol. 3, pp. 1415–1438, Mar. 2003.
- [36] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [37] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, nos. 2–3, pp. 103–130, Nov. 1997.
- [38] A. Raddon and B. Grigsby, "Throughput time forecasting model," in *Proc. 8th IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Sep. 1997, pp. 430–433.



Yair Meidan received the B.S. and M.S. degrees, both from the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel, in 2006 and 2008, respectively.

During his studies he was a Research and Teaching Assistant in data mining and machine learning. After graduation, he was an Adjunct Instructor and a Quantitative Analyst, mostly for marketing, behavioral targeting, and predictive analytics. He is currently a Data Mining and Predictive Analytics

Consultant with IBM-SPSS, Petah Tikva, Israel.



Boaz Lerner received the B.A. degree in physics and mathematics from Hebrew University, Jerusalem, Israel, in 1982, and the Ph.D. degree in computer engineering from the Ben-Gurion University of the Negev, Beer-Sheva, Israel, in 1996.

He researched with the Neural Computing Research Group, Aston University, Birmingham, U.K., and the Computer Laboratory, Cambridge University, Cambridge, U.K. He is currently with the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev. His current

research interests include machine learning and data mining approaches to data analysis and their application to real-world problems.



Gad Rabinowitz received the B.S. and M.S. degrees from the Ben-Gurion University of the Negev, Beer-Sheva, Israel, and the Ph.D. degree in operations research from Case Western Reserve University, Cleveland, OH.

He is currently the Head of the Department of Industrial Engineering and Management, Ben-Gurion University of the Negev. His current research interests include the theory and practice of operation and scheduling of production and logistics systems, and modeling of quality engineering and management

issues.



Michael Hassoun received the Ph.D. and M.Sc. degrees from the Industrial Engineering and Management Department at Ben-Gurion University of the Negev, Israel, and the B.Sc. degree in mechanical engineering from the Technion, Israel.

He is currently a Lecturer at the Ariel University Center. His current research interests include performance analysis and improvement of semiconductor fabs.