Comparison of Classification Machine Learning Models for Production Flow Analysis in a Semiconductor Fab

Ivan Kristianto Singgih[1,2,3,*], Stefanus Soegiharto[1], Arida Ferti Syafiandini[4,5]

[1] Department of Industrial Engineering, University of Surabaya, Surabaya, Indonesia
[2] The Indonesian Researcher Association in South Korea (APIK), Seoul, 07342, South Korea
[3] Kolaborasi Riset dan Inovasi Industri Kecerdasan Artifisial (KORIKA), Jakarta, Indonesia
[4] Department of Library and Information Science, Yonsei University, Seoul, South Korea
[5] Research Center for Computing, National Research and Innovation Agency, Indonesia (BRIN), Cibinong, Indonesia

ivanksinggih@staff.ubaya.ac.id;
s.soegiharto@staff.ubaya.ac.id;
afsyafiandini@yonsei.ac.kr

**Corresponding author*

Abstract. A semiconductor fab has complex wafer lot movements between machines and workstations. To ensure a smooth flow of the wafer lots, the system must be observed appropriately. Observation of such a complicated system is possible using machine learning. In this study, various machine learning techniques are applied to predict the semiconductor fab's throughput when considering wafer lot processing and queuing status at the machines and the machine utilization. The accuracies of the models are compared. It is shown that the random forest model obtained the best accuracy of more than 97%. Compared with the previous study, this study considers more models to allow a more comprehensive evaluation. The findings are important for providing suggestions on machine learning model selection for predicting the output of a semiconductor fab.

Keywords: Semiconductor Fab, Classification, Prediction, Machine Learning, Model Evaluation.

1 Introduction

Semiconductor fab has a complex environment due to the re-entrants of wafers to workstations and the usage of parallel machines [1]. The movement of wafers in a semiconductor fab (Intel minifab) is illustrated in Figure 1. Before being processed at workstation 1, the wafer lots must be grouped into batches first. Please refer to Singgih [1] to obtain more details on the simulation used for the data collection, the required processing times on the machines, and the arrival schedule of the wafer lots.

© The Author(s) 2023

M. Hartono et al. (eds.), *Proceedings of the 4th International Conference on Informatics, Technology and Engineering 2023 (InCITE 2023)*, Atlantis Highlights in Engineering 21,
https://doi.org/10.2991/978-94-6463-288-0_24

Singgih [1] has shown that some classification machine learning models could be used to predict the system's weekly throughput when considering various product and machine-related information, e.g., the number of processed wafer lots on each machine, number of wafer lots on the machine queues, and machine utilization.

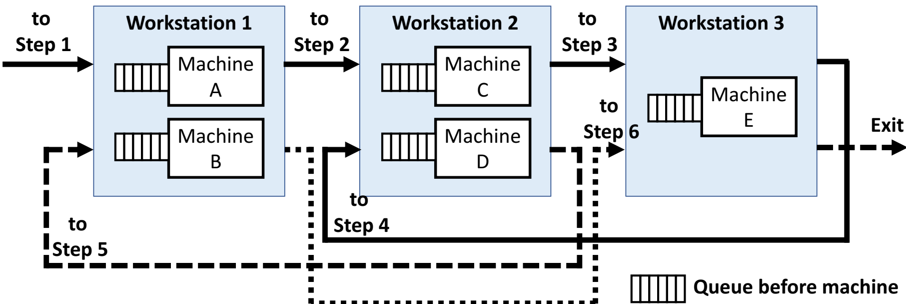

Fig. 1. Wafer flow in Intel semiconductor minifab.

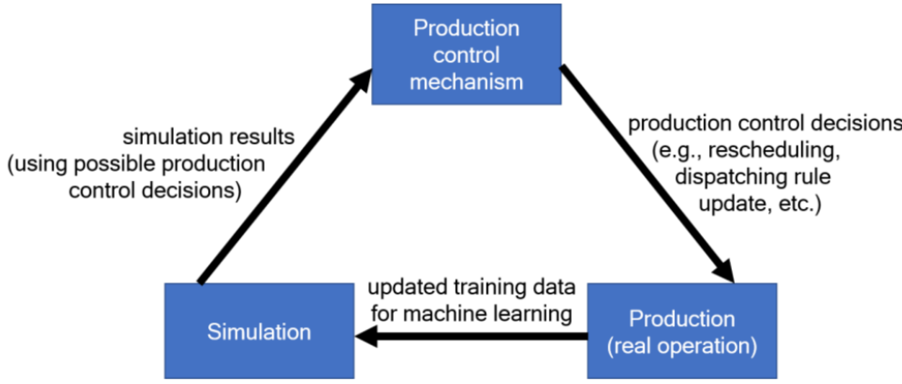

Fig. 2. The digital twin framework for the semiconductor process analysis.

The system proposed by Singgih [1] was a digital twin (Figure 2). In this digital twin system, real-time data related to the wafer lot processing and machine status can be collected using IoT sensors placed on the machines and their queues. This wafer lot processing and machine status show how well the operation was optimized, e.g., using batching, allocation, and scheduling decisions. In other words, when good optimization decisions are made, the wafer lot flow would be smooth, e.g., all machines are highly utilized, less queuing times, etc., and the weekly target throughput would be satisfied. By observing the production status information as input data, and the throughput satisfaction level (e.g., low and high) as the target (output) each week, Singgih [1] stated that the relationships between those input and output data could be identified. Using the same framework for finding the relationships between the input

and output data, the importance of each input data could also be observed, e.g., by iteratively considering a different set of input data, as conducted by Singgih [1].

This study continues the study of Singgih [1] by applying more classification machine learning models to allow a more comprehensive evaluation of various models. The contributions of this study are:

1. This study proposes a multi-model performance-based general feature selection scheme. Based on models with high accuracy in Singgih [1], this study identifies important input factors that tend to remain longer in each of the best models when the number of factors was reduced iteratively in Singgih [1]. Then, the factors are sorted based on their importance and a threshold is set to obtain generally important input factors.
2. This study tests more classification machine learning models using the selected factors to allow a more comprehensive evaluation of the machine learning models. This evaluation allows researchers to understand the performance of more machine learning models when dealing with the semiconductor fab optimization and provides insights for selecting methods when solving similar problems in the semiconductor fab.

The structure of this study is as follows. Solution methodology section presents the proposed feature selection scheme and the considered classification machine learning models. Numerical experiments and analysis section shows the numerical experiment results. The last section concludes the study.

2 Solution Methodology

The classification machine learning models considered by Singgih [1] are considered with the addition of models listed in Scikit-learn [2] that are classified into (1) ensemble methods, (2) Gaussian processes, and (3) Naive Bayes. In this study, models that were not included yet by Singgih [1] are selected, especially those in categories with no representative yet or models from categories with good performing methods in Singgih [1]. The complete list of considered classification machine learning models in this study is shown in Table 1.

The research methodology is shown in Figure 3. This study lists the features related to wafer processing and machine information based on results of Singgih [1]. Singgih [1] reduced the number of considered input data in the four best prediction models (AB, GB, RF, CART) by removing one input data from the complete model (of 42 input data) iteratively until the accuracy of each model is no longer improved (Figure 4, left side). The iteration number when each input data (out of 42 data) was removed is shown in Figure 4 (right side). Given the iteration numbers on the right side, each input data is sorted from the most important one (the ones removed at the larger number of iterations and tend to remain at final models). When selecting the final important input data, this study chooses them starting from the ones removed at the largest number of iterations and continues selecting the less important ones until input data with a certain number of iterations, as long as the input data is still included in the final version of any of the four selected models. After selecting the final im-

portant input data, they are used to test all classification machine learning models listed in Table 1. The details on the classification results for the model with the best accuracy are presented.

Table 1. List of considered classification machine learning models in this study.

No	[Abbreviation] Model Name	Explanation
1	[AB] Adaptive Boosting	Please refer to Singgih [1] for the definition of AB until SVM
2	[SGD] Linear classifiers with stochastic gradient descent training	
3	[NNMLP] Neural Network (Multilayer Perceptron)	
4	[GB] Gradient Boosting	
5	[RF] Random Forest	
6	[KNN] K-Nearest Neighbors	
7	[CART] Classification and Regression Tree	
8	[NB] Gaussian Naive Bayes	
9	[SVM] Support Vector Machine (C-support Vector)	
10	[Ensemble: BC] Bagging Meta-estimator	Bagging meta-estimator is an ensemble meta-estimator that utilizes a number of weaker prediction models. The performances of individual base classifiers are aggregated to make a final prediction [3].
11	[Ensemble: HBC] Histogram-Based Gradient Boosting	Histogram-based gradient boosting is a boosting ensemble that selects the best splits based on the feature histograms [4].
12	[Naive Bayes: MNB] Multinomial Naive Bayes	Multinomial Naive Bayes is a Naive Bayes implemented for a multinomially distributed data [5]. Naive Bayes methods consider that the conditional probabilities of independent variables are statistically independent [6].
13	[Naive Bayes: CoNB] Complement Naive Bayes	Complement Naive Bayes is a complement variant of MNB [7]. CoNB deals with the “severe assumptions” in the MNB [8].
14	[Naive Bayes: BNB] Bernoulli Naive Bayes	Bernoulli Naive Bayes is a Naive Bayes method that classifies data, which are distributed in multivariate Bernoulli distributions [9].

3 Numerical Experiments and Analysis

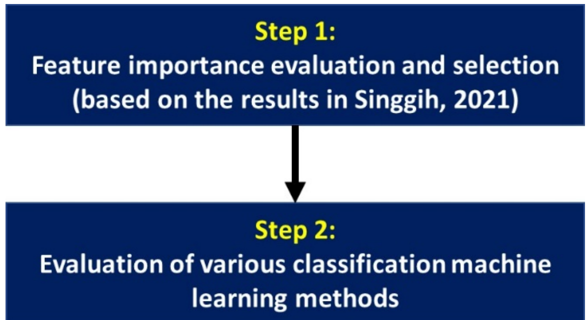

Fig. 3. Research methodology.

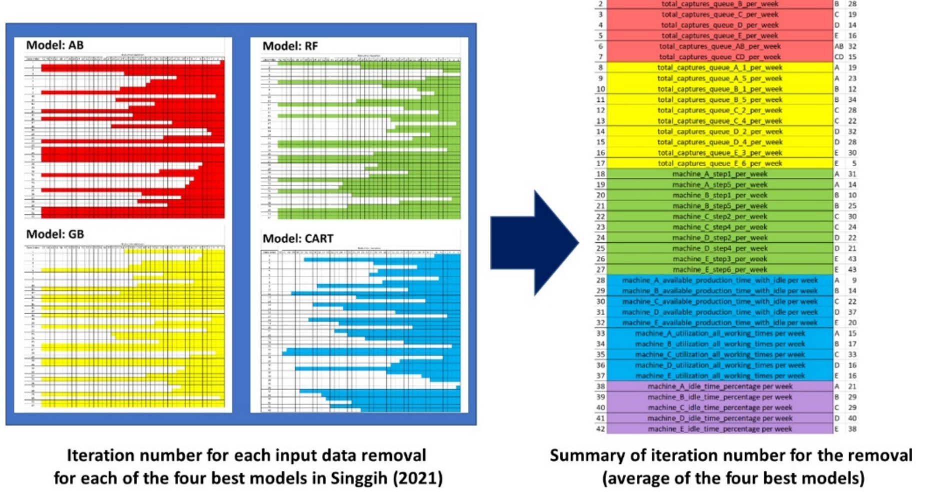

Fig. 4. A complete list of 42 input data in Singgih [1] and the iteration number when each input data was removed from the models (average of four best models).

List of the 42 initial input data was considered in Singgih [1]. The output data is the high throughput and low throughput classes. The Intel minifab was designed to satisfy 84 wafer lot-throughput target every week. Therefore, the high throughput class is defined with 84-95 wafer lots per week, meanwhile, the low throughput class is defined with 73-83 wafer lots per week. The generated throughput during the simulation in Singgih [1] was ranged between 73-95 wafer lots, based on the dynamics caused by the stochasticity in the emergency maintenance, the limited capacity in the machines' queues, and the capacities of the machines. The input data are sorted based on the iteration number for their removal (presented in Figure 4, right side) in descending

order, and the results are shown in Table 2. The most important input data are listed at the top. Input data that remained until iteration 19 during the removal procedure in Singgih [1] are selected. The other data are considered to be less important.

Table 2. Selected input data sorted based on their average number of iterations when they were removed from the four best models in Singgih [1].

No	Data Name (Unit in One Week)	Removal Iteration Index	Inclusion in Which Model (Among the Best Four Ones)
26	Amount of lots that completed step 3 at machine E	43	AB, GB, RF
27	Amount of lots that completed step 6 at machine E	43	AB, GB, RF, CART
41	Idle time portion of machine D (in percentage)	40	AB, GB, RF
42	Idle time portion of machine E (in percentage)	38	AB, GB, RF
31	Available processing time on machine D (in percentage, after removing the preventive and emergency maintenance times)	37	AB, GB
11	Amount of lots for step 5 at machine B's buffer	34	GB
35	Utilization of machine C (in percentage, after removing the preventive, emergency maintenance, and idle times)	33	AB, GB, RF
6	Amount of lots for any step at machine A and B's buffers	32	GB, RF
14	Amount of lots for step 2 at machine D's buffer	32	AB, GB
18	Amount of lots for step 1 at machine A's buffer	31	AB, GB
16	Amount of lots for step 3 at machine E's buffer	30	AB, GB
22	Amount of lots that completed step 2 at machine C	30	AB, RF
39	Idle time portion of machine B (in percentage)	29	GB
40	Idle time portion of machine C (in percentage)	29	AB
2	Amount of lots for any step at machine B's buffer	28	AB
12	Amount of lots for step 2 at machine C's buffer	28	RF
15	Amount of lots for step 4 at machine D's buffer	28	RF
21	Amount of lots that completed step 5 at machine B	25	GB
23	Amount of lots that completed step 4 at machine C	24	RF
9	Amount of lots for step 5 at machine A's buffer	23	AB
13	Amount of lots for step 4 at machine C's buffer	22	GB
24	Amount of lots that completed step 2 at machine D	22	AB
30	Available processing time on machine C (in percentage, after removing the preventive and emergency maintenance times)	22	GB
25	Amount of lots processed at machine D for processing step 4	21	AB
38	Idle time portion of machine A (in percentage)	21	-
32	Available processing time on machine E (in percentage, after removing the preventive and emer-	20	RF

No	Data Name (Unit in One Week)	Removal Iteration Index	Inclusion in Which Model (Among the Best Four Ones)
	gency maintenance times)		
1	Amount of lots for any step at machine A's buffer	19	RF
3	Amount of lots for any step at machine C's buffer	19	AB
8	Amount of lots for step 1 at machine A's buffer	19	-

Classification machine learning models in Table 1 are evaluated. Among 10,086 data records, 80% of them are used as training data, while 20% of them are used as testing data. Using the training data and 10-fold cross-validation, the accuracy of the models is obtained, as shown in Figure 5. The average and standard deviation of the accuracy for each model are presented in Table 3.

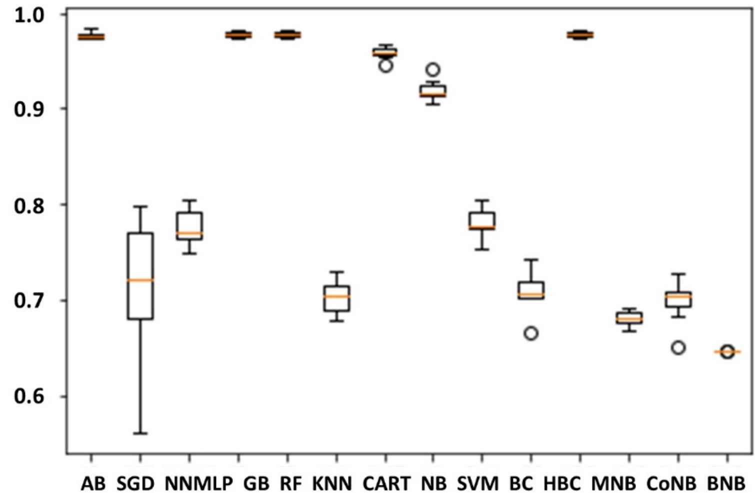

Fig. 5. Accuracy of the classification machine learning models represented in box plots.

Table 3. Accuracy of each classification machine learning model.

Model	Average of the accuracy (%)	Standard deviation of the accuracy (%)
AB	97.695	0.389
SGD	71.255	7.101
NNMLP	77.578	1.773
GB	97.794	0.308
RF	97.831	0.279
KNN	70.339	1.689
CART	95.910	0.593
NB	91.968	0.970
SVM	78.173	1.457

Model	Average of the accuracy (%)	Standard deviation of the accuracy (%)
BC	70.823	1.865
HBC	97.744	0.247
MNB	68.047	0.716
CoNB	69.930	1.994
BNB	64.564	0.029

The best model is the random forest which reaches an average accuracy of 97.831%. Other models that produce an accuracy of more than 95% and are worth further investigation are Adaptive Boosting, Gradient Boosting, Classification and Regression Tree, and Histogram-Based Gradient Boosting. The random forest was then tested using the testing data, and the obtained accuracy was 97.671%. This study concluded that the best model is the random forest, which is the same with the conclusions made by Singgih [1]. However, differently from Singgih [1], this study found that Histogram-Based Gradient Boosting has a high accuracy as well.

4 Conclusions

In this study, a framework to identify important input data for predicting throughput in a semiconductor fab is proposed. More classification machine learning models are also tested and it was shown that random forest obtained the best average accuracy of more than 97%. Other models worth further investigation are Adaptive Boosting, Gradient Boosting, Classification and Regression Tree, and Histogram-Based Gradient Boosting.

The following topics are suggested for further studies: (1) studying how specific operations research-related decisions could be made to obtain better values for the important input data, e.g., more number of processed wafer lots, less machine idle times, etc., (2) observing the same production problem using regression models, instead of the classification ones to allow observing more detailed behaviors of the semiconductor fab.

References

1. Singgih, I.K.: Production Flow Analysis in a Semiconductor Fab Using Machine Learning Techniques. Processes. 9, 407 (2021). https://doi.org/10.3390/pr9030407
2. Scikit-learn: 1. Supervised learning, https://scikit-learn/stable/supervised_learning.html
3. Munteanu, C.R., Gutiérrez-Asorey, P., Blanes-Rodríguez, M., Hidalgo-Delgado, I., Blanco Liverio, M. de J., Castiñeiras Galdo, B., Porto-Pazos, A.B., Gestal, M., Arrasate, S., González-Díaz, H.: Prediction of Anti-Glioblastoma Drug-Decorated Nanoparticle Delivery Systems Using Molecular Descriptors and Machine Learning. International Journal of Molecular Sciences. 22, 11519 (2021). https://doi.org/10.3390/ijms222111519

4. Alazba, A., Aljamaan, H.: Software Defect Prediction Using Stacking Generalization of Optimized Tree-Based Ensembles. Applied Sciences. 12, 4577 (2022). https://doi.org/10.3390/app12094577
5. Macrohon, J.J.E., Villavicencio, C.N., Inbaraj, X.A., Jeng, J.-H.: A Semi-Supervised Approach to Sentiment Analysis of Tweets during the 2022 Philippine Presidential Election. Information. 13, 484 (2022). https://doi.org/10.3390/info13100484
6. Pan, Y., Gao, H., Lin, H., Liu, Z., Tang, L., Li, S.: Identification of Bacteriophage Virion Proteins Using Multinomial Naïve Bayes with g-Gap Feature Tree. International Journal of Molecular Sciences. 19, 1779 (2018). https://doi.org/10.3390/ijms19061779
7. Gan, S., Shao, S., Chen, L., Yu, L., Jiang, L.: Adapting Hidden Naive Bayes for Text Classification. Mathematics. 9, 2378 (2021). https://doi.org/10.3390/math9192378
8. Scikit-learn: sklearn.naive_bayes.ComplementNB, https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html
9. Shukla, R., Sinha, A., Chaudhary, A.: Electronics | Free Full-Text | TweezBot: An AI-Driven Online Media Bot Identification Algorithm for Twitter Social Networks. Electronics. 11, 743 (2022). https://doi.org/10.3390/electronics11050743

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.