DMME-Driven Product Quality Prediction for Semiconductor Manufacturing

Alif Ulfa Afifah, Angga Prastiyan, Fahmi Arif, Fadillah Ramadhan
Industrial Engineering, Institut Teknologi Nasional, Bandung, Indonesia

***Abstract*—Defective products in manufacturing can be reduced by accurately predicting quality outcomes based on process parameters. This study proposes a quality prediction framework for semiconductor manufacturing using the Data Mining Methodology for Engineering Applications (DMME). This study extends DMME with domain-specific preprocessing and demonstrates its superiority on the SECOM dataset compared to other classifiers. Experimental results show that the Random Forest algorithm achieved the highest performance, with 92.99% accuracy and an F-measure of 0.9637, confirming the effectiveness of the proposed approach. These findings highlight the potential of structured, engineering-oriented data mining to improve product quality and support informed decision-making in complex manufacturing environments.**

***Keywords*—Data mining; quality prediction; DMME; semiconductor manufacturing; random forest**

I. INTRODUCTION

Manufacturing refers to the industrial process of converting raw materials into products with higher economic value, whether in the form of finished goods or intermediate components [1]. Semiconductor production is a highly complex stage that involves multiple subprocesses and generates large volumes of process data, which must be effectively controlled and analyzed to improve product quality [2]. Even with continuous advancements in production technology and management practices, product defects still occur in manufacturing processes [3]. Breakdowns in production machinery can further disrupt manufacturing flows and result in substantial losses, emphasizing the importance of preventive maintenance strategies to ensure production stability and consistency in product quality [4]. A practical way to reduce these defects is to analyze production data and pinpoint the process parameters that have the most significant impact on product quality. Understanding these factors enables manufacturers to develop predictive models that help anticipate product outcomes and support informed decision-making for future production runs [5]. When properly built, these models can be more accurate and efficient than traditional inspection-based approaches [6].

Several studies have examined the prediction of quality in various manufacturing contexts. In wafer dicing, for example, models have been shown to predict failures with up to 75% accuracy [7]. In the process of coke production, regression techniques have been employed to map coal properties into performance indicators, such as the coke reactivity index (CRI) and the coke strength after reaction (CSR) [8]. On the other hand, battery production has adopted various methods to model product quality across complex and highly variable production lines [9].

In semiconductor manufacturing, cascade quality prediction techniques that integrate principal component analysis (PCA) with decision tree algorithms, such as ID3, have shown strong results, achieving prediction accuracies of up to 90.02% [10]. While this highlights the promise of predictive modeling in the manufacturing sector, the effectiveness of such models can vary significantly depending on the application area and the characteristics of the data being used.

Reliable quality prediction depends on more than just selecting a good algorithm. The way the problem is defined and the data are prepared play a significant role in the outcome. Quality management, at its core, is about improving process performance and minimizing variation [11], [12]. Predictive models built on process data support this goal by providing a structured, data-driven means of identifying patterns that affect product outcomes [13]. Data mining plays a key part in this process. It combines statistical and machine learning techniques to uncover useful patterns and insights from complex data sources [14], [15], [16]. One common framework for this is the Cross-Industry Standard Process for Data Mining (CRISP-DM), which organizes data mining into six defined phases, from understanding the business context all the way to implementation. Still, CRISP-DM is a general-purpose framework, and it doesn't fully address the needs of engineering-specific problems. To bridge that gap, the Data Mining Methodology for Engineering (DMME) was developed to extend CRISP-DM by incorporating additional steps more suited to technical domains [17].

Despite the availability of various quality prediction methods, many are developed and tested under specific conditions or on a limited set of data. Because of differences in data types, distributions, and feature counts, a model that works well in one context might not deliver the same performance in another. This limitation is especially relevant in semiconductor manufacturing, where the data is often highly complex and challenging to work with. The benchmark that is often used in this space is called the SECOM dataset. It contains many numerical attributes with a nominal label to indicate product quality. While this dataset offers considerable potential for analysis, limited research has explored which machine learning techniques are best suited for handling its specific characteristics, such as high dimensionality and class imbalance. Moreover, there has been little examination of how individual process variables impact prediction outcomes across different modeling approaches.

To fill this research gap, the study focuses on developing a quality prediction model specifically suited to datasets like SECOM. A model that consists of numerical features and categorical labels. The goal is to identify a modeling approach that offers high accuracy while factoring in the distinct difficulties and characteristics of the semiconductor manufacturing process.

The remainder of this study is organized as follows: Section II reviews related works on data-driven quality prediction and outlines the research gap addressed in this study. Section III details the proposed methodology based on DMME. Section IV presents the dataset used in this research, data preparation techniques, describes the modelling approach and selected algorithms. Section V reports the experimental results and Section VI discusses their implications. Finally, Section VII concludes the study with key findings and suggestions for future research.

II. Related Work

Recent advancements have demonstrated the increasing role of machine learning (ML) and data-driven approaches in enhancing semiconductor manufacturing processes. A comprehensive review of ML applications for semiconductor process optimization was presented by [18], in which predictive models, virtual metrology, and advanced process control were highlighted as methods that have improved yield and quality across various stages of production. While these approaches have demonstrated significant improvements, most focus on specific techniques or stages rather than employing a holistic methodology to systematically address data challenges in manufacturing. This study addresses this gap by leveraging the Data Mining Methodology for Engineering Applications (DMME) to develop a comprehensive quality prediction framework for semiconductor processes. In [19], the authors systematically reviewed automated defect inspection using convolutional neural networks (CNNs) on scanning electron microscope (SEM) images, emphasizing the potential of ML to replace traditional, labor-intensive inspection methods. In specific process applications, [20] developed a machine-learning-based prediction model to optimize plasma etching parameters, demonstrating that domain-specific ML techniques can enhance process stability and output quality. Beyond improving accuracy, [21] integrated explainability methods into manufacturing quality prediction models, enabling better transparency and trust in model-driven decisions. Furthermore, [22] addressed a critical challenge of ML adoption in manufacturing by analyzing the effects of data quality and class imbalance on predictive performance, recommending robust preprocessing pipelines for reliable outcomes.

In addition to algorithmic improvements, recent works have explored leveraging modern computing infrastructures to enhance quality management systems. In [23], the authors demonstrated the feasibility of integrating data mining techniques such as Decision Tree, k-Nearest Neighbor, Naïve Bayes, and Random Tree into a Software-as-a-Service (SaaS) platform, enabling low-cost and accessible quality prediction for manufacturing environments. Similarly, [24] proposed a Cloud-based Quality Analyzer (CQA) that incorporates real-time analysis, automated feedback loops, and reduced reliance on human quality engineers, aligning with Industry 4.0 principles of cyber-physical integration and data-driven decision-making. These studies highlight the growing trend toward combining advanced predictive models with cloud-based and intelligent platforms to achieve scalable, efficient, and adaptive quality management solutions.

Despite these contributions, many existing approaches either focus on specific manufacturing tasks or rely heavily on singular algorithmic solutions without a structured methodology to handle data quality issues systematically. To bridge this gap, the present study employs the Data Mining Methodology for Engineering Applications (DMME), which integrates domain-specific preprocessing, feature selection, and algorithm comparison to develop a robust product quality prediction framework tailored for high-dimensional, imbalanced semiconductor manufacturing data.

III. Methodology

This research uses the Data Mining Methodology for Engineering Applications (DMME), which builds on the widely recognized CRISP-DM framework and adapts it specifically for engineering challenges [25]. DMME introduces important technical extensions to the process, including feature selection, dimensionality reduction, and techniques to address class imbalance, which are critical steps due to the high-dimensional and unbalanced nature of the SECOM dataset analyzed here. For example, principal component analysis (PCA) was considered to simplify the dataset. On the other hand, methods like SMOTE were explored to address the uneven distribution between defective and acceptable products. For the classification task, the study focused on five algorithms, i.e., Decision Tree, Random Forest, Support Vector Machine, k-Nearest Neighbor, and Naive Bayes. These were selected because they offer a good mix of transparency, speed, and effectiveness for structured data typically found in manufacturing. More complex options, such as XG Boost or neural networks, were set aside for this work in favor of models that are easier to interpret and commonly used in process engineering. All analyses were conducted using Rapid Miner Studio and Microsoft Excel, ensuring a systematic evaluation of how various process parameters affect product quality in semiconductor manufacturing.

IV. Data Understanding and Methodology

This stage describes the process of collecting and selecting data to be processed for research. The process of predicting product quality requires historical data owned by the company to be used as a reference in the prediction process. The data used must have elements that directly affect the quality of the product or can be referred to as attributes. The data studied in this study were defect data in semiconductor products [25]. The semiconductor dataset can be seen in Table I.

The data above displays several columns of identity information, including the time of manufacture of the product, the product number in the second column, the third column, which contains parameters with both numeric and real characteristics, and the last column, which shows the quality results of the product. There are 590 types of parameters, each with a unique value in the column. The last column in the table indicates the quality of the product, with an accept/reject

classification status characterized by binomial characteristics. The dataset used has multivariate characteristics, with the value of each attribute having numerical characteristics. The dataset is in the manufacturing area. The dataset used contains attributes with numeric characteristics, comprising 591 columns and 1567 rows of production data. This semiconductor dataset accounts for missing values and bias resulting from disturbances in the data collection process, with varying intensities depending on the features of each product. This null value must be considered when preprocessing data or when the algorithm is applied. The data is represented in a raw text file, with each line corresponding to an individual and a feature in the process.

TABLE I SEMICONDUCTOR DATASET

Time	Prod	Attribute				Accept/Reject
		0	*1*	*…*	*589*	
19/07/2008 11:55	1	3030,9	2564,0	…		Accept
19/07/2008 12:32	2	3095,8	2465,1	…	208,2	Accept
19/07/2008 13:17	3	2932,6	2559,9	…	82,9	Reject
19/07/2008 14:43	4	2988,7	2479,9	…	73,8	Accept
19/07/2008 15:22	5	3032,2	2502,9	…	73,8	Accept
…	…	…	…	…	…	…
17/10/2008 6:07	1567	2944,9	2450,8	…	137,8	Accept

A. Data Preparation

The data preparation process involves adjusting the type of data to be used in conjunction with the character data from the dataset. The data prepared for the data type adjustment process is part of the attribute column, where each parameter value has special characteristics that can be classified as a particular data type. An explanation of the data preparation stage is provided below.

1) Data cleansing: The raw dataset, before the cleansing process is carried out to eliminate missing values, can be seen in Table II.

TABLE II SEMICONDUCTOR RAW DATASETS

Time	Prod	Attribute				Accept/Reject
		0	*1*	*…*	*589*	
19/07/2008 11:55	1	3030,9	2564,0	…		Accept
19/07/2008 12:32	2	3095,8	2465,1	…	208,2	Accept
19/07/2008 13:17	3	2932,6	2559,9	…	82,9	Reject
19/07/2008 14:43	4	2988,7	2479,9	…	73,8	Accept
19/07/2008 15:22	5	3032,2	2502,9	…	73,8	Accept
…	…	…	…	…	…	…
17/10/2008 6:07	1567	2944,9	2450,8	…	137,8	Accept

The data cleansing process, which involved eliminating missing values in this study, was performed using Microsoft Excel on the rows and columns of the dataset. After the process of removing missing values is completed in Microsoft Excel, the data can be viewed in Table III.

TABLE III SEMICONDUCTOR DATASETS AFTER CLEANSING DATA FOR MISSING VALUE

Attribute				Accept/Reject
0	*1*	*…*	*589*	
2932,6	2559,9	…	82,9	Reject
2988,7	2479,9	…	73,8	Accept
2946,3	2432,8	…	44,0	Accept
3030,3	2430,1	…	44,0	Accept
3058,9	2690,2	…	95,0	Accept
…	…	…	…	…
3246,3	2499,8	…	23,6	Accept

Then, the data cleansing process is carried out to handle outlier values in the semiconductor dataset using the auto-cleansing feature in RapidMiner Studio software. An overview of auto cleansing can be seen in Fig. 1.

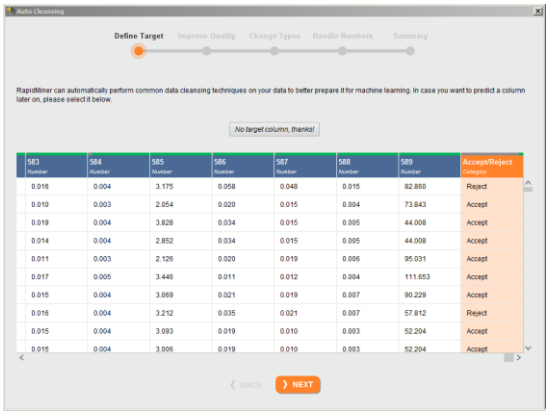


Fig. 1. Cleansing data for outlier handling.

After the auto cleansing process is done, the data after the process can be seen in Table IV.

TABLE IV SEMICONDUCTOR DATASETS AFTER CLEANSING DATA FOR OUTLIERS

Attribute				Accept/Reject
0	*1*	*…*	*589*	
2932,6	2559,9	…	82,9	Reject
2988,7	2479,9	…	73,8	Accept
2946,3	2432,8	…	44,0	Accept
3030,3	2430,1	…	44,0	Accept
3058,9	2690,2	…	95,0	Accept
…	…	…	…	…
3246,3	2499,8	…	23,6	Accept

2) Data transformation: Before the data transformation process is carried out, the dataset to be studied is prepared in advance. Data transformation is performed using a read operator in Rapid Miner, which can be used for various data extensions, including CSV, Excel, URL, SAS files, and several other file extensions that resemble databases. Researchers in this study used an Excel file type (xlsx), so that the operator used was Read Excel. Four steps must be taken when importing data to this operator. First, select files; second, determine the format to be used; third, determine the rows and columns; and finally, select labels to predict. The determination of the label is based on the classification results expected in the study. The prediction process carried out refers to the classification results to be achieved. The processed elements consisted of 392 attribute variables and the outcome of the decision variable, which was labeled. The type of data used depends on the value in each attribute column. In this research dataset, all values in the attribute columns are numeric data types, and the label variable, resulting from the accept/reject decision, is nominal.

B. Modelling

The modeling process is carried out by extracting the data that has been prepared in advance. The extraction process is carried out with the selected algorithms, namely, random forest, decision tree, k-NN, support vector machine, and naïve Bayes. The extraction processes up to the selection of Rules can be seen below.

1) Data extraction: The data extraction stage on the prepared dataset utilizes several operator processes to facilitate the data extraction process. Extraction begins by reading the data from the dataset using the Read Excel operator and then proceeds through the preprocessing process (data preparation) until it reaches the cross-validation process.

2) Cross-validation: Cross-validation is a statistical method for evaluating and comparing algorithms by dividing the dataset into two classifications, namely the training set and the testing set. Validation in this study employs a 10-fold cross-validation approach, performing iterations with 10 data waves that alternate as test data. Cross-validation is used because the validation process emphasizes the use of test data interchangeably, ensuring that the data used to create a model is similar to the original characteristics. The training set is part of the dataset that is used to train the model, while the testing set is a subset of the dataset that is used to evaluate its performance. The principle of the cross-validation process is to conduct repeated experiments (iterations) on the test and train sets. The number of these iterations can be determined according to the intended results. The extraction stage involves several supporting operators that optimize the data to be extracted, ensuring that at the time of cross-validation, the data is relevant for validation.

V. Results

A. Selected Algorithm

The accuracy value of each algorithm used can be seen in Table V.

TABLE V ALGORITHM ACCURACY

Algorithm	Precision	Recall	Accuracy	F *Measure*	G-*Mean*
Random Forest	92.99%	100.00%	92.99%	0.9637	0.4649
k-NN	99.72%	93.21%	92.99%	0.9636	0.4648
SVM	99.91%	92.98%	92.90%	0.9632	0.4645
Decision Tree	98.70%	93.64%	92.55%	0.9610	0.4621
Naive Bayes	93.63%	21.88%	25.97%	0.3547	0.1024

The Rapid Miner results recapitulation for algorithms that have been used in graphical form can be seen in Fig. 2.

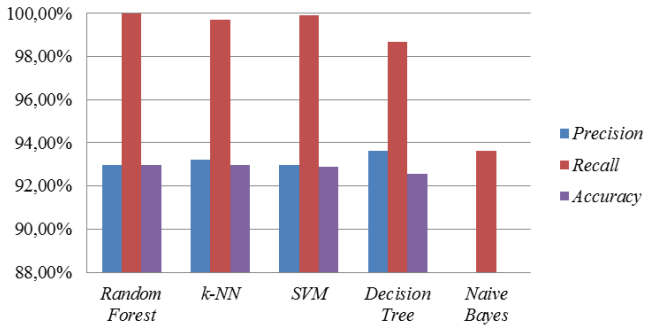


Fig. 2. Comparison of algorithms.

Table V presents the performance metrics for each algorithm. Random Forest achieved the highest accuracy (92.99%) and F-measure (0.9637), outperforming all other classifiers. While Random Forest provided the best performance, its computational cost was higher than single-tree models, which may impact real-time applications.

The five models used have their characteristics based on their work specifications. Based on this research, the random forest model has the highest accuracy and F-measure value, at 92.99% for accuracy and 0.9637 for the F-measure. Therefore, it was selected for use on the dataset. However, this model has weaknesses in terms of computation time. The random forest algorithm uses random attributes and classifies the trees that are formed. The tree building is done recursively based on the data associated with the same label. Tree splitting is performed to divide the data based on the type of attribute used. The Naïve Bayes algorithm classifies labels using probability and statistical methods. The SVM algorithm pursues the optimal hyperplane by maximizing the distance between classes. A hyperplane serves as a separator between classes. Another classification algorithm is the k-Nearest Neighbor algorithm, which classifies new nodes by assigning a value based on the category with the highest number of nearest k neighbors. Finally, the decision tree algorithm serves as the basis for the random forest algorithm, which converts data into decision rules to form a tree, enabling

decision makers to better interpret solutions to the problems that arise.

There is a visualization of the random forest algorithm, which consists of several decision tree diagrams. An illustration of one of the first decision tree visualizations from the Random Forest algorithm, is demonstrated in Fig. 3, following the knowledge mining process already conducted.

The decision tree model in the random forest algorithm can be interpreted as a set of rules, making it easier to understand. Rules are a concise summary of descriptive algorithms that enable the deduction of solutions to a problem, making them easy to understand. One of the rules for the decision tree shown in Fig. 4 is evident.

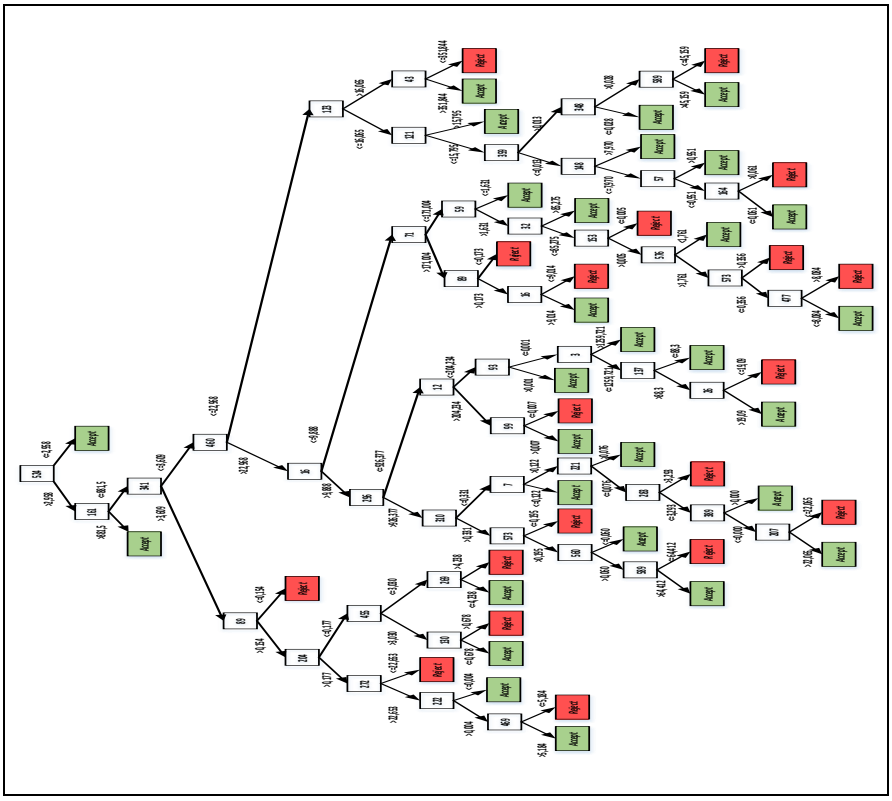


Fig. 3. One of the random forest model diagram visualization.

524 > 2.958
| 161 > 8811.500: Accept {Reject=0, Accept=86}
| 161 ≤ 8811.500
| | 341 > 3.609
| | | 89 > 0.154
| | | | 204 > 0.177
| | | | | 272 > 22.653
| | | | | | 222 > 0.004
| | | | | | | 469 > 5.184: Accept {Reject=0, Accept=10}
| | | | | | | 469 ≤ 5.184: Reject {Reject=2, Accept=0}
| | | | | | 222 ≤ 0.004: Accept {Reject=0, Accept=57}
| | | | | 272 ≤ 22.653: Reject {Reject=2, Accept=0}
| | | | 204 ≤ 0.177
| | | | | 455 > 3.030
| | | | | | 130 > 0.678: Reject {Reject=8, Accept=0}
| | | | | | 130 ≤ 0.678: Accept {Reject=0, Accept=4}
| | | | | 455 ≤ 3.030
| | | | | | 269 > 4.238: Reject {Reject=2, Accept=0}
| | | | | | 269 ≤ 4.238: Accept {Reject=0, Accept=10}
| | | 89 ≤ 0.154: Reject {Reject=5, Accept=0}
| | 341 ≤ 3.609
| | | 460 > 22.968
| | | | 16 > 9.888
| | | | | 296 > 926.377
| | | | | | 310 > 0.331
| | | | | | | 573 > 0.195
| | | | | | | 93 ≤ -0.001
| | | | | | | | 3 > 1259.721: Accept {Reject=0, Accept=22}
| | | | | | | | 3 ≤ 1259.721
| | | | | | | | | 137 > 88.300
| | | | | | | | | | 26 > 1.909: Reject {Reject=5, Accept=1}
| | | | | | | | | | 26 ≤ 1.909: Accept {Reject=0, Accept=3}
| | | | | | | | | 137 ≤ 88.300: Accept {Reject=0, Accept=8}
| | | | 16 ≤ 9.888
| | | | | 71 > 171.004
| | | | | | 89 > 0.173
| | | | | | | 16 > 9.014: Accept {Reject=0, Accept=12}
| | | | | | | 16 ≤ 9.014: Reject {Reject=2, Accept=0}
| | | | | | 89 ≤ 0.173: Reject {Reject=4, Accept=0}
| | | | | 71 ≤ 171.004
| | | | | | 59 > 1.631
| | | | | | | 32 > 85.275: Accept {Reject=0, Accept=52}
| | | | | | | 32 ≤ 85.275
| | | | | | | | 153 > 0.005
| | | | | | | | | 576 > 1.761
| | | | | | | | | | 573 > 0.356: Reject {Reject=5, Accept=1}
| | | | | | | | | | 573 ≤ 0.356
| | | | | | | | | | | 477 > 8.084: Reject {Reject=1, Accept=1}

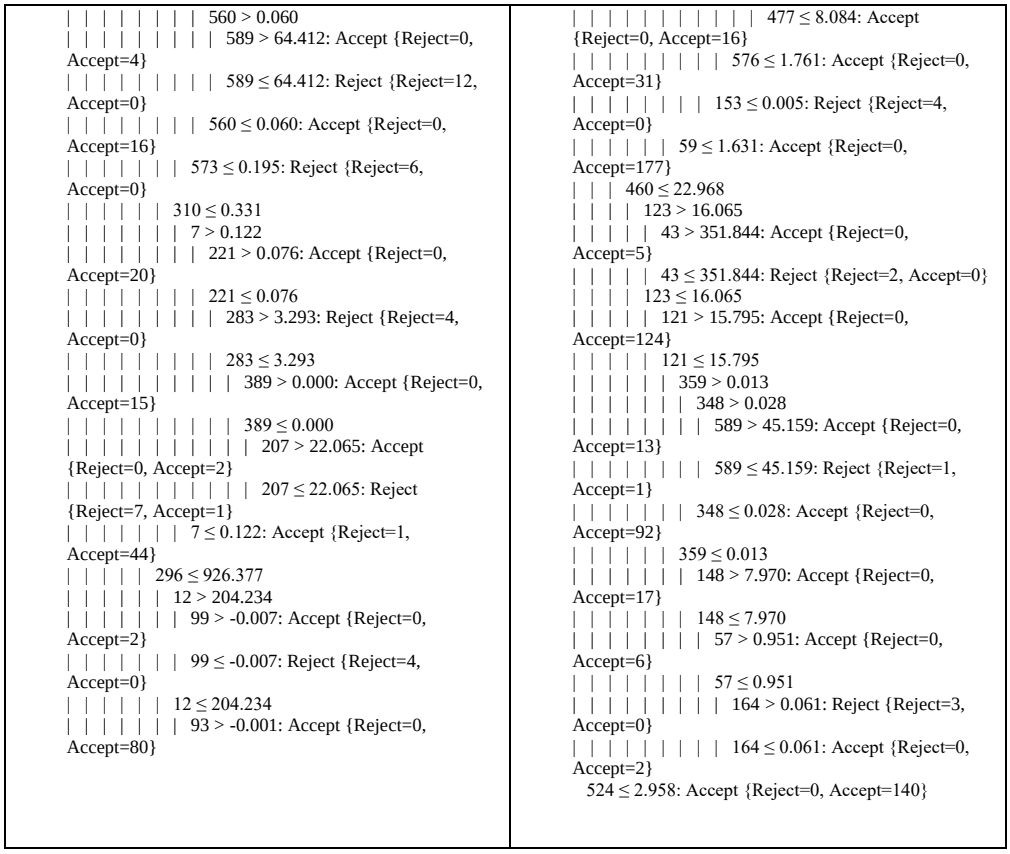


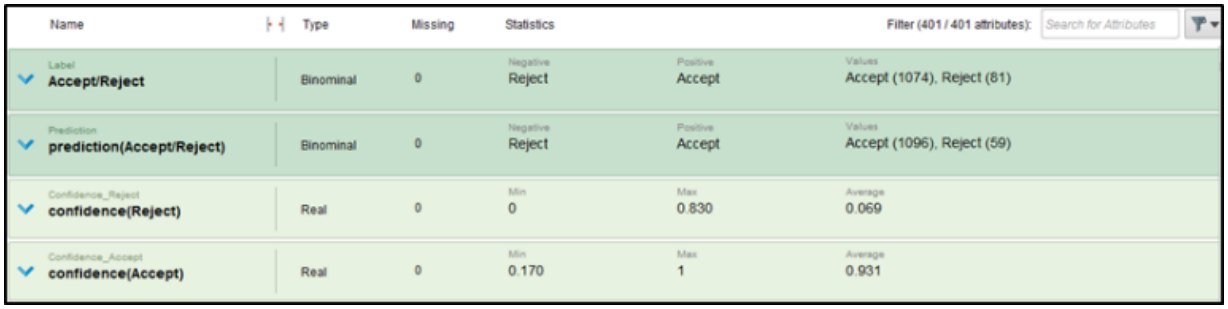
Fig. 4. Rules in the random forest model.

The formulation of the quality prediction model, performed on semiconductor data sets, produces rules with the highest accuracy level, specifically the random forest algorithm, which achieves an accuracy value of 92.99% and an F-measure of 96.37%. This is supported by some knowledge from the results of data extraction regarding the relationship between process parameters and the final quality of the product, as represented by red and green, as shown in Fig. 5.

The results of the quality prediction model are close to the accepted results from the previous production, with a total of 1096 units of products and a failure rate of 59 units. The predicted results are shown in Fig. 6.

Accept/Reject	prediction(Accept/Reject)	confidence(Reject)	confidence(Accept)	0	1	2	3	4	5	6
Reject	Reject	0.600	0.400	2932.610	2559.940	2186.411	1698.017	1.510	100	95.488
Accept	Accept	0.067	0.933	2988.720	2479.900	2199.033	909.793	1.320	100	104.237
Accept	Accept	0.008	0.992	2946.250	2432.840	2233.367	1326.520	1.533	100	100.397
Accept	Accept	0.002	0.998	3030.270	2430.120	2230.422	1463.661	0.829	100	102.343
Accept	Accept	0	1	3058.880	2690.150	2248.900	1004.469	0.788	100	106.240
Accept	Accept	0.100	0.900	2967.680	2600.470	2248.900	1004.469	0.788	100	106.240
Accept	Accept	0.150	0.850	3016.110	2428.370	2248.900	1004.469	0.788	100	106.240
Reject	Reject	0.614	0.386	2994.050	2548.210	2195.122	1046.147	1.320	100	103.340
Accept	Accept	0.007	0.993	2920.070	2507.400	2195.122	1046.147	1.320	100	103.340
Accept	Accept	0.101	0.899	3051.440	2529.270	2184.433	877.627	1.467	100	107.871
Reject	Reject	0.612	0.388	2963.970	2629.480	2224.622	947.774	1.292	100	104.849
Accept	Accept	0.083	0.917	2988.310	2546.260	2224.622	947.774	1.292	100	104.849
Accept	Accept	0.083	0.917	3028.020	2560.870	2270.256	1258.456	1.395	100	104.808
Accept	Accept	0	1	3032.730	2517.790	2270.256	1258.456	1.395	100	104.808
Accept	Accept	0	1	3040.340	2501.160	2207.389	962.532	1.204	100	104.031

Fig. 5. Important factor.



Name	Type	Missing	Statistics		
Label **Accept/Reject**	Binominal	0	Negative Reject	Positive Accept	Values Accept (1074), Reject (81)
Prediction **prediction(Accept/Reject)**	Binominal	0	Negative Reject	Positive Accept	Values Accept (1096), Reject (59)
Confidence_Reject **confidence(Reject)**	Real	0	Min 0	Max 0.830	Average 0.069
Confidence_Accept **confidence(Accept)**	Real	0	Min 0.170	Max 1	Average 0.931

Fig. 6. Results of the quality prediction model.

VI. Discussion

The Random Forest algorithm achieved the highest performance among all evaluated classifiers, with an accuracy of 92.99% and an F-measure of 0.9637 on the SECOM dataset. This superior performance can be attributed to Random Forest's ensemble learning approach, which mitigates overfitting and effectively handles high-dimensional data with complex inter-feature dependencies.

The variation in performance across datasets indicates that the proposed Random Forest-based model is particularly effective for high-dimensional, structured process data like SECOM. For datasets with fewer features or lower variance, simpler models such as SVM or k-NN may suffice. This suggests that the algorithm's suitability is strongly linked to data characteristics, reinforcing the importance of feature engineering in DMME. These insights confirm that structured methodologies like DMME can deliver practical, high-performance solutions in real-world manufacturing contexts.

These findings highlight the value of adopting a structured methodology that integrates domain-specific preprocessing, feature selection, and model comparison, as facilitated by DMME. By systematically addressing data quality issues, the framework ensures that the most suitable algorithm is selected for the given manufacturing context. Future research could extend this work by incorporating cost-sensitive learning, real-time adaptation, and integration with cloud-based quality management systems to further improve predictive performance in dynamic production environments.

VII. Conclusion

This study applied the Data Mining Methodology for Engineering Applications (DMME) to predict product quality in semiconductor manufacturing. By integrating advanced preprocessing techniques and comparing multiple classification algorithms, the framework effectively addressed the challenges of high dimensionality and data imbalance in the SECOM dataset. Experimental results identified Random Forest as the most suitable classifier, achieving 92.99% accuracy and an F-measure of 0.9637.

The findings confirm that enhancing traditional CRISP-DM with domain-specific preprocessing and algorithm evaluation improves predictive performance in complex manufacturing environments. This work contributes to the development of structured, data-driven methodologies for quality prediction and provides a practical approach for industries seeking reliable, interpretable models. Future research will explore broader manufacturing applications, incorporate cost-sensitive learning, and integrate this framework into real-time, multi-stage production systems to further strengthen defect prevention and decision-making.

References

[1] Hajo Wiemer, Lucas Drowatzky, Steffen Ihlenfeldt, 'Data Mining Methodology for Engineering Applications (DMME) – A Holistic Extension to the CRISP-DM Model, Chair of Machine Tools Development and Adaptive Controls, Institute of Mechatronic Engineering, Technische Universität Dresden, 01069 Dresden, Germany. 2019.

[2] Pedro Espadinha-Cruz, Radu Godina, Eduardo M. G. Rodrigues, 'A Review of Data Mining Applications in Semiconductor Manufacturing', A Review of Data Mining Applications in Semiconductor Manuacturing', Processes, 9(2), 305, 2021.

[3] Xinmin Zhang, Manabu Kano, Masahiro Tani, Junichi Mori, Junji Ise, Kohhei Harada, "Prediction and causal analysis of defects in steel products: Handling nonnegative and highly overdispersed count data", Control Engineering Practice, Volume 95, 2020.

[4] Alif Ulfa Afifah, Fifi Herni Mustofa, Ainun Zahra Mustika, 'Determining Preventive Maintenance Schedule on Press Machine Components using Age Replacement Methods', E3S Web of Conferences 484, 01024, 2023.

[5] R. W. House and T. Rado, "An approach to artificial intelligence," in *IEEE Transactions on Communication and Electronics*, vol. 83, no. 70, pp. 111-116, Jan. 1964.

[6] E. Demirbilek and J. -C. Grégoire, "Machine learning based reduced reference bitstream audiovisual quality prediction models for realtime communications," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Hong Kong, China, pp. 571-576, 2017.

[7] Te-Jen Su, Yi-Feng Chen, Jui-Chuan Cheng, Chien-Liang Chiu, "An artificial neural network approach for wafer dicing saw quality prediction", Microelectronics Reliability, Volume 91, Part 2, 2018.

[8] Lauren North, Karen Blackmore, Keith Nesbitt, Merrick R. Mahoney, Models of coke quality prediction and the relationships to input variables: A review, Fuel, Volume 219, Pages 446-466, 2018.

[9] Sebastian Thiede, Artem Turetskyy, Arno Kwade, Sami Kara, Christoph Herrmann, Data mining in battery production chains towards multi-criterial quality prediction, CIRP Annals, Volume 68, Issue 1, Pages 463-466, 2019.

[10] Fahmi Arif, Nanna Suryana, Burairah Hussin, Cascade Quality Prediction Method Using Multiple PCA+ID3 for Multi-Stage Manufacturing System, IERI Procedia, Volume 4, Pages 201-207, 2013.

[11] Joseph m. Juran, Joseph A. De Feo, Juran's Quality Handbook: The Complete Guide to Performance Excellence, McGraw Hill, 2010.

[12] Peter D. Mauch, Quality Management Theory and Application, CRC Press, Boca Raton, 2009.

[13] Roger Sauter, Introduction to Engineering Statistics and Six Sigma, Technometrics, Vol 49, 2007.

[14] Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques. 2012.

[15] Efraim Turban, Jay E. Aronson, Ting Peng Liang, Decision Support Systems and Intelligent Systems, Prentice Hall. 2003.

[16] M. Kantardzic, Data Mining: Concepts, Models, Methods & Algorithms. 2011

[17] Steffen Huber, Hajo Wiemer, Dorothea Schneider, Steffen Ihlenfeldt, DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model, Procedia CIRP, Volume 79, Pages 403-408, 2019.

[18] Ying-Lin Chen, Sara Sacchi, Bappaditya Dey, et al. Exploring Machine Learning for Semiconductor Process Optimization: A Systematic Review. TechRxiv. July 16, 2024.

[19] Thibault Lechien, Enrique Dehaerne, Bappaditya Dey, Victor Blanco, Sandip Halder, Stefan De Gendt, Wannes Meert, 'Automated Semiconductor Defect Inspection in Scanning Electron Microscope Images: a Systematic Review', arxiv, Cornell University, 2023.

[20] Changmin Kima, Seunghwan Leeb, Muyoung Kima, Min Sup Choic, Taesung Kimb, and Hyeong-U Kima , 'Machine Learning-Based Prediction of Atomic Layer Control for MoS2 via Reactive Ion Etcher', Applied Science and Convergence Technology 2023; 32(5): 106-109, 2023.

[21] Dennis Gross, Helge Spieker, Arnaud Gotlieb, Ricardo Knoblauch, 'Enhancing Manufacturing Quality Prediction Models through the Integration of Explainability Methods', arxiv. Cornell University, 2024.

[22] Jiarui Xie, Lijun Sun, Yaoyao Fiona Zhao, 'On the Data Quality and Imbalance in Machine Learning-based Design and Manufacturing—A Systematic Review', Engineering, 2024.

[23] Fahmi Arif, Fadillah Ramadhan, Wildan Sayf, 'Data Mining for Quality Prediction in Software-as-A-Service Concept: A Case Study in Offset Printing Company', Quality Innovation Prosperity/Kvalita Inovacia Prosperita 27/3, 2023.

[24] Fahmi Arif and Isa Setiasyah Toha, 'Enhancing manufacturing quality system using cloud-based quality analyzer', AIP Conference Proceedings 2772, 080012, 2023.

[25] McCann, M. and Johnston, A, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/SECOM]. Irvine, CA: University of California, School of Information and Computer Science, 2018.