Enhancing the Accuracy of Manufacturing Process Error Detection through SMOTE-based Oversampling Using Machine Learning and Deep Learning Techniques

Sai Venkat Boyapati*, Guduru Bhanu Sri Rakshitha†, Maryada Rishitha Reddy‡, Dr. Sumalatha Saleti§
Department of Computer Science Engineering, SRM University AP
India
Email: *saivenkat_boyapati@srmap.edu.in, †bhanusrirakshitha_guduru@srmap.edu.in, ‡rishithareddy_maryada@srmap.edu.in, §sumalatha.s@srmap.edu.in

***Abstract*—A production competency study leads to a rise in the manufacturing sectors' strategic emphasis. Developing Semiconductor materials is a highly complex approach that necessitates numerous evaluations. It is impossible to emphasize the significance of the quality of the product. We put up a number of methods for automatically creating a prognostic model that is effective at identifying equipment flaws throughout the semiconductor materials' wafer fabrication process. The SECOM dataset is representative of semiconductor production procedures that go through numerous tests performed. There are imbalanced statistics in the dataset, so our proposed methodology incorporates SMOTE functionality that is introduced to mitigate the imbalance of the training dataset by leveling off any unbalanced attributes. Detecting faults in the manufacturing process improves semiconductor quality and testing efficiency, and is used to validate both approaches to Machine Learning and Deep Learning algorithms. This is accomplished by collecting performance metrics during the development process. Another aspect of our effort to cut down on the training time for testing is highlighted in our research report.**

***Index Terms*—Semiconductor, Manufacturing process errors, SECOM dataset, Wafer production, Training and Testing data, Imbalance data, Balance Data, Machine Learning, Deep Learning, SMOTE technique.**

I. Introduction

The corporate setting of modern times is constantly evolving. The development of semiconductors has significantly altered our world [1]. The primary objective of semiconductor manufacturers is to raise their quality on an annual basis. Because semiconductors form the foundation of every hardware system, the demand for them has increased tremendously along with both the personal and business use of all technologies.

Machine learning, data mining, and deep learning offer a multitude of possibilities for the efficient management of industrial processes. By means of distributed data collection, data cleansing, extracting useful data from noisy data, and updating optimization ideas from flowing data in real-time, these technologies can provide a wealth of opportunities for industrial applications [2]. These changes produced enormous volumes of data that need to be evaluated. This article suggests machine learning and deep learning methods for evaluating manufacturing process failures.

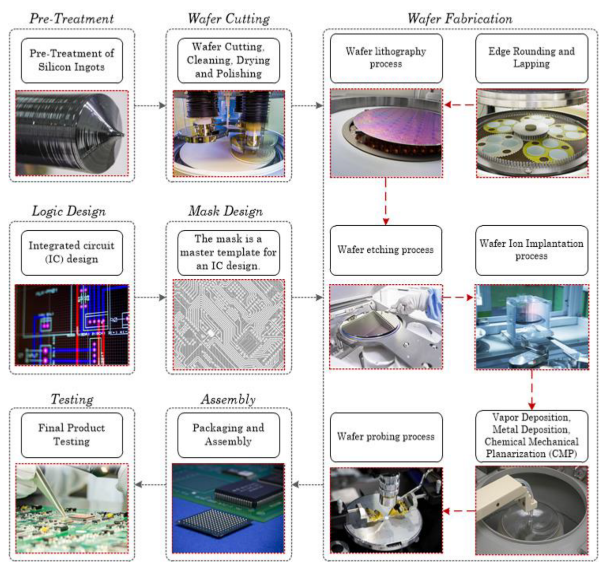

Fig. 1. A succinct illustration of the semiconductor manufacturing process [3].

The production of semiconductors is the industry that is taken into account for the validation of these processes. Predicting process faults is crucial for reducing failure rates. The SECOM dataset serves as a support for comparing our suggested methodologies and is indicative of semiconductor manufacturing operations. This dataset acts as the norm for assessing if the deliverables of a series of manufacturing activities are incorrect or not. It conducts numerous tests on the semiconductor and assesses its functional capability to see if it is functioning properly. In our dataset, we have a significant quantity of data from several semiconductors that were undertaking these checks. The results of the tests indicate whether the semiconductors were successful or not.

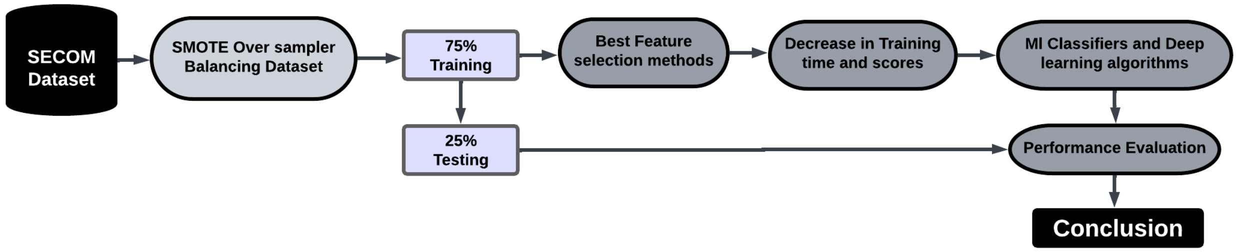

Fig. 2. Proposed Methodology for SECOM Dataset workflow.

Below is a brief outline of this article's workflow and key contributions [4]:

1) Determining an approach that utilizes the following processes for spotting faults in the manufacturing process which are data preprocessing which involves data cleaning and it corrects the noisy data, features selection, and dataset will be splitting into train and test data.
2) Before doing these data preprocessing techniques SMOTE operation is applied to correct the class imbalance data.
3) The modification, applying as well as assessment of two deep learning and machine learning algorithms.

II. Review of Literature

The initial literature papers about our topic actually assist us in first comprehending the importance, manufacturing techniques, and testing procedures for semiconductors. Using earlier articles as a guide, Let's now talk about the methodologies that numerous other published studies use.

Balancing the Dataset as the dataset is faulty data balancing is the main part. Kim et al [5] and Salem et al [6] authors employed the SMOTE technique to balance the dataset. Their results indicated that SMOTE-based oversampling could help improve fault detection in semiconductor manufacturing processes. Salem et al [6] authors tested 288 approaches to classifying the SECOM dataset using various stages for data imputation, data imbalance, feature selection, and classification. The results showed that LR was the best classification model, and SMOTE was the best technique for synthetic data generation. Moldovan et al [8] tested their method on SECOM and SETFI datasets, and found that using the CSO algorithm to determine the number of nodes in each hidden layer increased the weighted precision of the MPC.

Feature Elimination enhances model performance by selecting pertinent features and noise, resulting in improved accuracy and efficiency. Salem et al [6] the authors used SELECTFDR to be effective for feature selection, and "Inpainting KNN-Imputation" was the best data imputation method. These discoveries have the potential to improve the accuracy and reliability of fault detection in semiconductor manufacturing processes. Extensive analysis of research papers revealed a focus on reducing training time by identifying relevant features. Increased feature count was consistently linked to longer training times. Consequently, feature elimination techniques were utilized to select informative features, reducing training time while maintaining model performance. Furthermore, apart from the aforementioned techniques, there exists a range of other methods for feature elimination, such as Principal Component Analysis (PCA), Multivariate Adaptive Regression Splines (MARS), and NOVA [11]. These approaches provide alternative strategies to select important features, thereby enhancing the performance of classification models across diverse domains and industries [12].

Classification is the crucial stage of evaluating product quality, and classification models have been extensively utilized. Various studies have explored different approaches, Kim et al [5] authors developed fault detection prediction models using logistic regression (LR), artificial neural networks (ANN), decision trees (DT), and random forests (RF)]. Kerdprasop and Kerdprasop [7] researchers have also proposed data mining-based fault detection methods, with Naive Bayes achieving a 90% fault detection rate but an 80% false alarm rate. To address this, boosting techniques were employed, resulting in improved precision for tree-based models while maintaining a low false alarm rate. Multilayer Perceptron Classifier (MPC) and Chicken Swarm Optimization (CSO) were employed in other studies by Moldovan et al [8]. Additionally, classification performance has been explored using support vector machines, logistic regression, artificial neural networks, and decision tree algorithms by Munirathinam [9] and Karthigaikumar [10]. These studies highlight the continuous efforts to develop effective models for product quality prediction and fault detection in the semiconductor manufacturing industry.

III. Evaluating the Dataset

SECOM Dataset was discovered by authors Michael McCann and Adrian Johnston and was donated to the UCI Machine Learning Repository on the 19th of November, 2008. This research utilizes the SECOM (Semiconductor Manufacturing) dataset, which includes both manufacturing operation data and semiconductor quality data, to determine the quality of semiconductors produced in the industry [13]. The dataset comprises 1567 instances with two classes, 104 fails, and some missing values.

The SECOM dataset is composed of 1567 examples that originate from a wafer manufacturing production line. Each example in the dataset is represented by a vector of 590 sensor measurements and includes an identification for pass-fail testing. Out of the total examples, only 104 are marked positive as failed cases, coded as 1, while the majority of examples pass the test and are marked negative, coded as -1. This significant class imbalance poses a challenge in achieving a good balance between the precision and recall of the classifier. The SECOM Dataset poses a two-class problem, but there is an imbalance in the distribution with a 14:1 skew of the pass to fail. With a total of 590 features, this dataset presents a significant number of variables. However, missing data and incomplete Feature information are prevalent throughout the dataset. Additionally, a few columns consist of constant values, further adding to the dataset's unique characteristics [14].The dataset also includes a date-time stamp corresponding to the functionality test. The dataset contains null values, which vary in intensity depending on the individual features. These null values correspond to data points with no recorded measurements in the original metrology data.

IV. Framework Building & Model Assessment

As we know, there are two methods for balancing a dataset: oversampling and undersampling. We initially used the undersampling technique, specifically the random undersampling, to balance our data by reducing the number of case conditions. However, we found that this approach was not particularly helpful since fewer case conditions result in lower prediction accuracy [15]. Therefore, we decided to try out oversampling techniques and tested various options such as

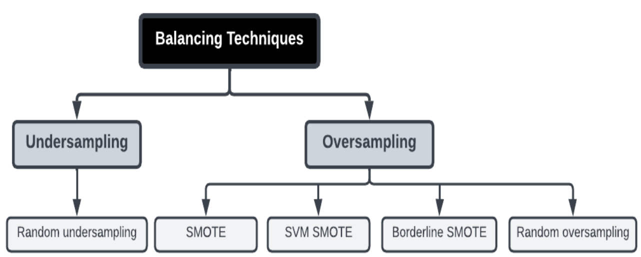

Fig. 3. Different Types of Balancing Techniques.

Upon thorough examination of several papers, we acquired knowledge about a range of algorithms that exhibited superior performance in this particular area. Armed with this information, we proceeded to implement these high-performing algorithms in conjunction with a select few others, which are detailed below [16].

- **Logistic Regression:**
  Logistic regression is performed when dealing with a classification issue. It is employed to determine the relationship between characteristics and the likelihood of a specific result. In our instance, the possibility will fall between 1 and 0 (pass or fail). To determine the probability, we generally employ the logistic function or the sigmoid function.
- **Decision Tree Classifier:**
  A decision tree is a visual representation of choices and their outcomes, typically depicted as a tree-like graph. The graph consists of nodes, which represent a specific event or choice, and edges, which provide the conditions or rules necessary for making decisions. In every decision tree, there are branches and nodes, with each branch indicating a possible value for the node, and each node representing a set of characteristics that must be classified.
- **Support Vector Machine (SVM):**
  In order to classify the data points into two groups, the SVM algorithm searches for the optimal line to do so. SVM examines the data used in classification and regression analysis. Support vector machines (SVMs) can effectively perform non-linear classification as well as linear classification by implicitly converting their inputs into high-dimensional feature spaces.
- **Extreme Gradient Boost Classifier (XG Boost):**
  A decision tree that learns from the residuals of earlier predictor variables is the Extreme Gradient Boost Classifier. XGBoost works by training a variety of decision trees. Each tree is trained using a portion of the data. The final forecast is then obtained by combining the predictions from each tree. It is the best machine-learning tool for problems like regression, classification, and ranking and provides a parallel tree boosting.
- **Random Forest Classifier:**
  The Random Forest model is an ensemble learning approach that, during the training phase, creates several decision trees. In a random forest, each decision tree is built using a subsample of features rather than each decision tree is built using all characteristics. Trees then predict a class outcome, and the model's final class prediction is based on the trees' consensus.
- **Neural Networks:**
  Neural networks imitate the way the brain finds connections and patterns in data. Combining various numerical inputs yields a single numerical output with adjustable coefficient weights. Unlike conventional models, neural networks can modify their parameters in response to varying inputs, allowing them to produce optimal outputs without the need to adjust the output criterion.

V. Proposed Methodology

This research paper looks at some of the most prominent machine learning and deep learning algorithms for sensor-based manufacturing approaches. The proposed methodology is outlined, and the corresponding implementations are summarized in Figure 2. The dataset we worked with comprises 1567 records and 592 properties. We used a range of tools, including NumPy, pandas, sklearn, matplotlib, and many more, to conduct in-depth research. The msno function was used to check for missing values once we discovered them when cleaning the data. A graphical depiction of missingness patterns in a dataset, known as a matrix plot for missing values, enables you to quickly and accurately identify trends in missingness across several variables. The primary advantage of using a matrix plot for missing values is that it can aid researchers in developing better-informed decisions regarding handling missing data, resulting in more reliable and accurate data analyses.We found missing values in the dataset, which might significantly affect our conclusions. In order to identify any data imbalances, we displayed the pass/fail predictions. Then, we used the smote library to balance the dataset, and the results were then shown.

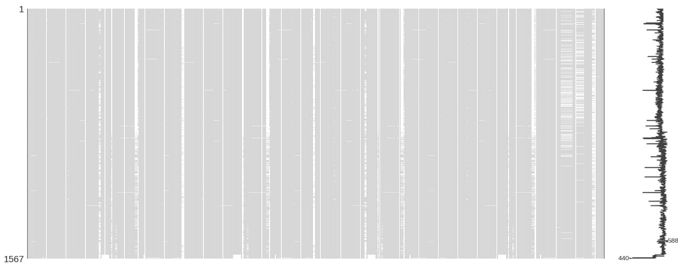

Fig. 4. Before removing missing values.

- **Before SMOTE:**

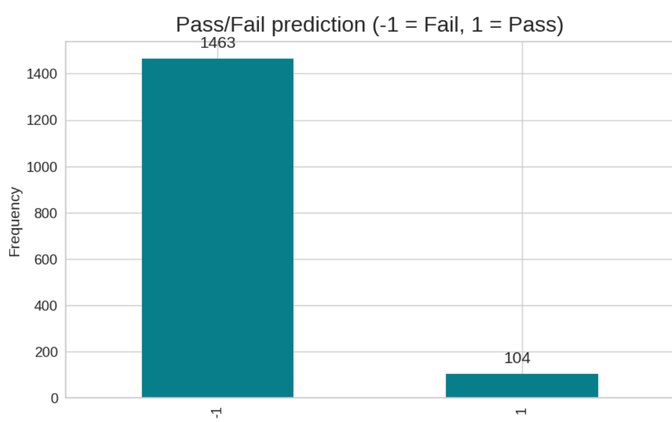

Fig. 5. Before removing missing values.

- **After SMOTE:**

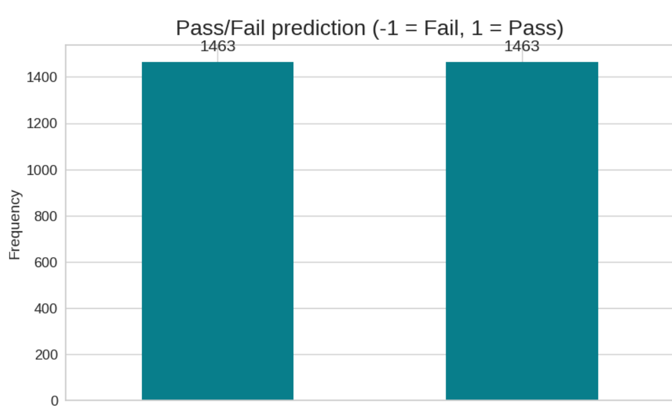

Fig. 6. After removing missing values.

To prepare the data for analysis or modeling, we performed some preprocessing steps on the dataset. One of these processes was to divide the "timestamp" column, which enabled us to turn the timestamp values into a numerical format that could be expressed as a decimal integer. Moreover, we discovered and deleted some unnecessary columns, such as "start time." Using a unique function, we were able to retrieve a special weekday, year, month, and date. Here, we displayed various histograms, such as

- **Month Vs probability of Pass/Fail:**

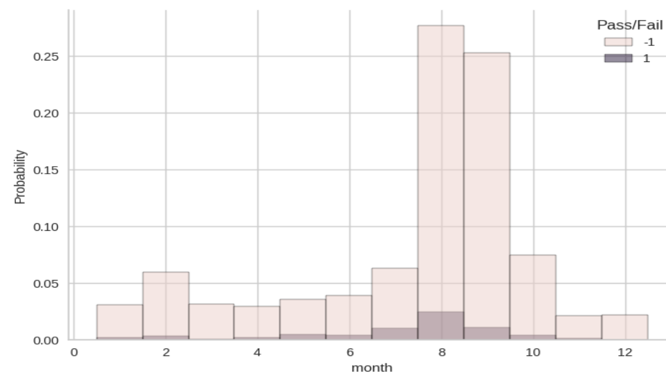

Fig. 7. Pass/Fail Rates by Month.

- **Date Vs probability of Pass/Fail:**

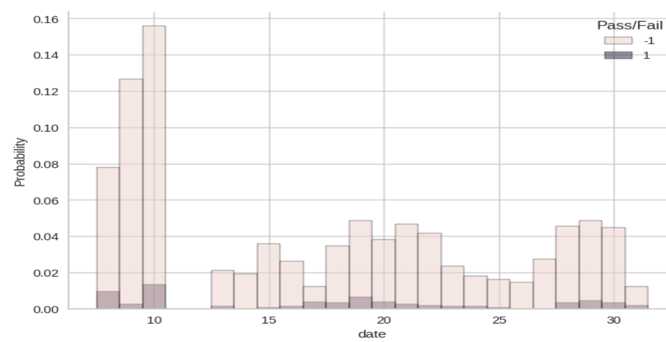

Fig. 8. Pass/Fail Rates by Date.

- **Weekday Vs probability of Pass/Fail:**

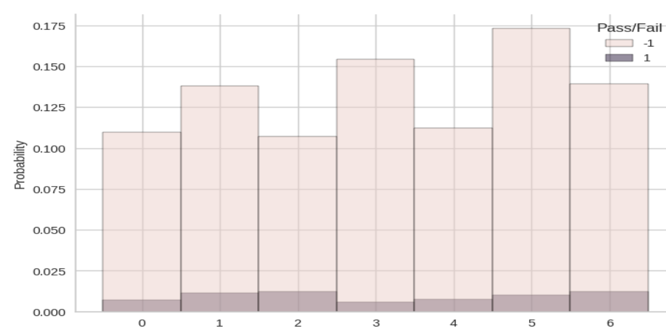

Fig. 9. Pass/Fail Rates by Weekday.

- **Hour Vs probability of Pass/Fail:**

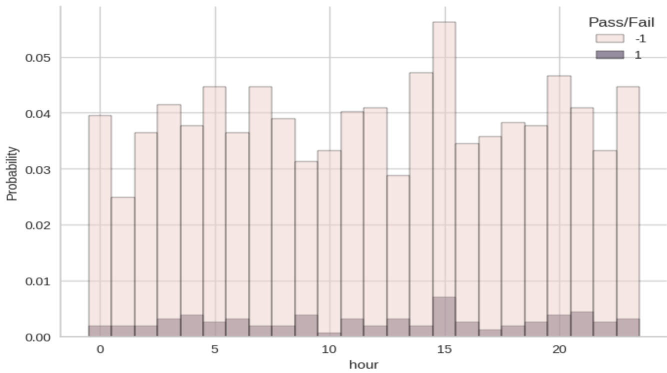

Fig. 10. Pass/Fail Rates by Hour.

After importing the dataset into a pandas data frame and analyzing its rows and columns, significant imbalances in the target values were found. To resolve this issue, the target values were balanced using the SMOTE method. The data was then split into 75:25 training and testing sets using the train-test split approach.

We used a dataset containing a variety of characteristics, the bulk of which were numerical. The information in a few columns, such as ”year”, ”month”, ”date”, ”weekday”, ”hour”, and ”min,” however, was not numerical. We carefully considered these features and determined that they were not necessary for our modeling or analysis, therefore we removed them from the dataset.

By implementing the Synthetic Minority Over-sampling Technique (SMOTE), multiple data points are added to the dataset, which leads to a significant enhancement in the efficiency and accuracy of the model. This is primarily attributed to the fact that the additional data points aid in improving the feature engineering process. It has been observed that incorporating SMOTE has resulted in a reduction in training time compared to not utilizing SMOTE.

In our dataset, we encountered the challenge of missing values, which can hinder data analysis and modeling. To overcome this issue, we have implemented the KNN imputer.

- **KNN Imputer:**
  We handled missing values in our dataset by using the K-nearest neighbor (KNN) imputation method. KNN imputation utilizes the values of the nearest neighboring data points to fill in missing values [17]. We also created a graph to visualize the missing data, but it was subsequently removed after imputation. Initially, we removed columns with more than 30% missing values, eliminating 52 features. For the remaining missing samples (less than 30%), KNN imputation was applied. To prevent data leakage, the imputed data was modified only in the testing dataset. This approach led to a reduction in training time.

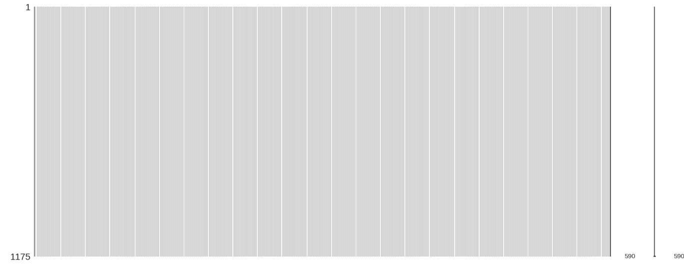

Fig. 11. After removing missing values.

VI. Feature Elimination

Our main objective was to significantly shorten the training period, therefore we used a number of techniques to do so. As part of our plan, the number of characteristics dropped, which significantly reduced training time [18]. The methods we used to do this are as follows:

- **Normalization:**
  Normalization was applied as a preprocessing step before implementing the variance threshold. The purpose of normalization is to adjust the scale of numerical data, reducing its sensitivity to different feature scales. It rescales the data to fit within the range of 0 to 1. Normalization plays a crucial role in enhancing the performance of machine learning models. Standardizing the feature scale ensures that no single feature dominates the learning process based solely on its magnitude. This helps prevent biased learning and improves the overall efficiency and accuracy of the models. This is particularly important when using distance-based algorithms, such as KNN or clustering algorithms, where the calculation of distances is highly influenced by the scale of features.
- **Variance Threshold:**
  We noticed repeated values in corresponding rows, which hindered progress. To address this, we used the variance check method with a specific threshold (e.g., zero) to eliminate low-variance features. We imported a normalizer from sklearn and applied it to both the training and testing data to prevent data leakage. Features with a zero variance, indicating constant values that remained the same in each row, were removed. After this process, we were left with 590 features. We then utilized KNN imputation and reevaluated training time and outcomes.
- **Correlation:**
  To evaluate the correlations between features, we examined correlation coefficients (-1 to 1). Weakly correlated features (those close to 0) were deemed unimportant, whilst strongly correlated features (those near 1 or -1) were eliminated as redundant. We will exclude independent characteristics with a high absolute correlation since they are identical to other features and do not enhance our model. This left 264 features remaining after 214 features were removed, which then caused 214 further features to be removed. After this procedure, 264 important qualities were left.

- **Correlation With target:**
  Following the previous steps, we assessed the correlation between the remaining features and the target column. Features with a correlation below 5% were dropped. As a result, we identified 40 correlated features, while removing 224 features from the dataset.
- **RFE Work:**
  The RFE method recursively removes features until the desired number is achieved for feature selection. It starts by building a model with all components, then removes less crucial features and evaluates performance. RFE is efficient for high-dimensional datasets with few observations. We used RFE to obtain a reduced feature set. Applying K-nearest neighbor imputation significantly reduced training time compared to the "correlation with target" approach. The graph visually represents the recursive feature selection process and identifies the optimal number of features for the logistic regression model.
  - **Before SMOTE:**

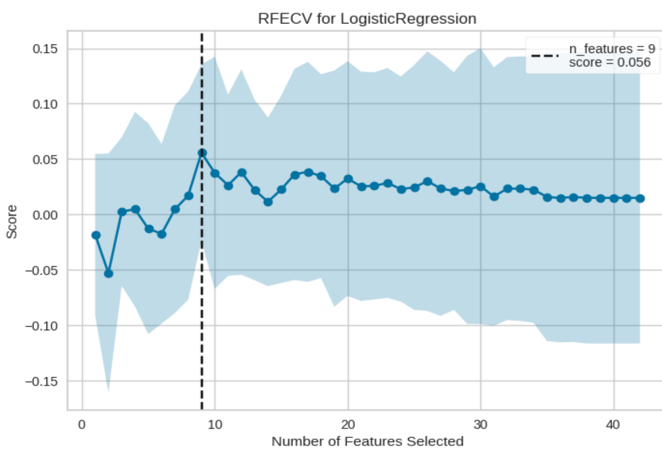

Fig. 12. Number of features selected before smote

  - **After SMOTE:**

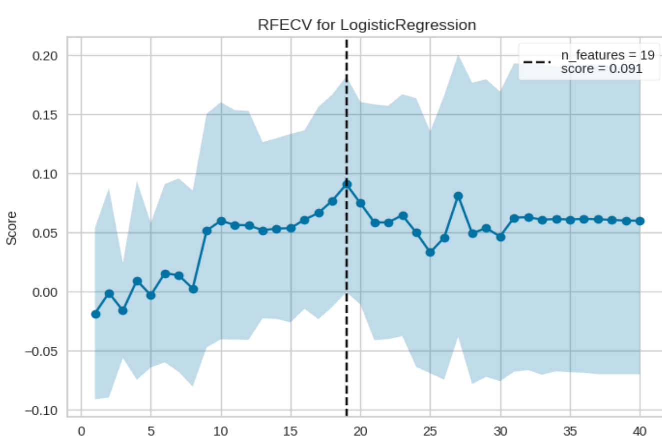

Fig. 13. Number of features selected after smote

Initially, the analysis concentrated on examining 9 attributes prior to incorporating SMOTE. However, after the integration of SMOTE, the selection process identified a total of 19 features. Upon implementing the undersampling technique, the feature count remained unchanged compared to the scenario without undersampling, attributed to the reduced case conditions. Likewise, upon employing various oversampling techniques and enhancing the case conditions, the feature count remained constant across all oversampling techniques. A visual representation is provided below, outlining the associated training time at each stage of the feature engineering procedure. Despite the reduction in training time, the model's MCC Score and F1 Score have remained unchanged.

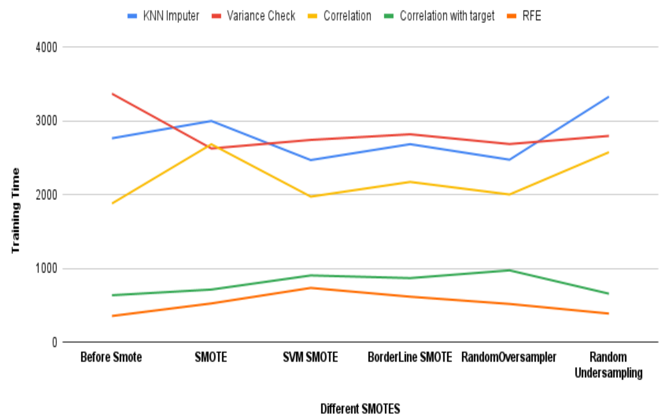

Fig. 14. Training Time with respect to Feature Elimination Techniques and different SMOTES

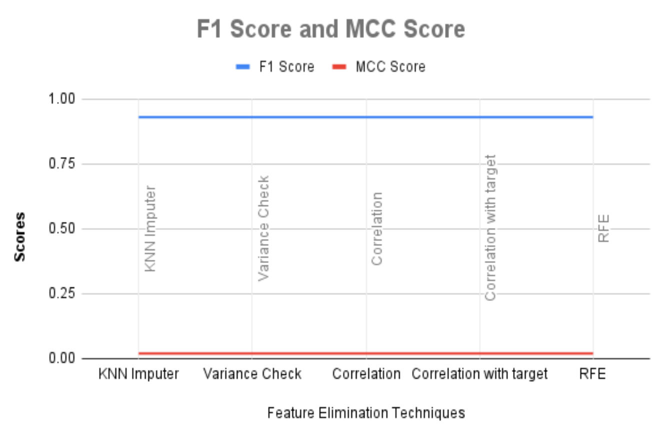

Fig. 15. F1 And MCC Score with respect to Feature Elimination Techniques and different SMOTES

VII. In-Depth Examination of Results

The main objective of this research paper is to examine and forecast the manufacturing process of semiconductor devices using several machine learning and deep learning techniques and compare the results to select the best-performing model.

TABLE I
ACCURACY OF DIFFERENT CLASSIFICATIONS FOR DIFFERENT SMOTE TECHNIQUES

Accuracy	**Random Undersampler**	**SMOTE**	**SVM SMOTE**	**Random Oversampler**	**Borderline SMOTE**	**Before SMOTE**
Logistic Regression	63.27%	52.81%	59.44%	63.52%	64.29%	76.53%
Decision Tree	68.11%	79.59%	89.54%	89.03%	46.68%	76.02%
XGBoost	77.81%	84.18%	90.05%	92.09%	55.36%	76.79%
Random Forest	68.88%	87.50%	92.35%	93.37%	45.41%	75.77%
SVM	69.39%	81.63%	88.27%	90.82%	49.23%	59.18%
Neural Networks	67.09%	88.27%	84.18%	44.90%	93.26%	73.98%

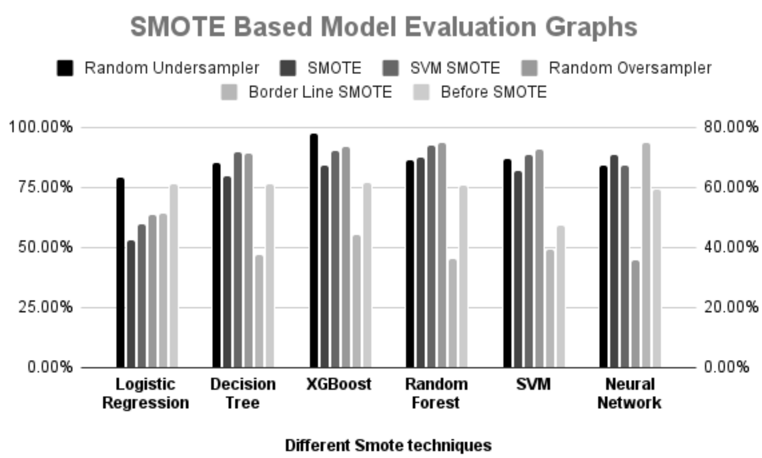

Fig. 16. Result Analysis of Performances.

TABLE II
SMOTE VS MODELS

SMOTE Vs Models	**Best Models**	**Accuracy**
Before SMOTE	SVM	93.26%
SMOTE	XGBoost	89.54%
SVM SMOTE	Random Forest	92.09%
Borderline SMOTE	Random Forest	90.83%
Random Oversampler	Random Forest	93.37%
Random Undersampler	Neural Networks	76.53%

After implementation of the machine learning techniques such as Logistic Regression, Decision Tree Classifier, Support Vector Machine, Random Forest Classifier, and XGBoost Classifier, as well as deep learning techniques such as Neural Networks to the SECOM dataset, we are presented the results in a tabular and graphical format which makes it easy to understand in a concise manner. The graph provided above illustrates the evaluation of multiple classifiers, both before and after employing the SMOTE. Initially, only one classifier demonstrated a significant improvement in performance compared to the others. Specifically, the Random Forest classifier achieved the highest accuracy rate of 93

Among the various SMOTE methods tested, the random oversampling yielded the best results when compared to Borderline SMOTE and SVM SMOTE. Additionally, machine learning surpassed deep learning in terms of accuracy. However, the introduction of SMOTE led to a substantial improvement in the number of well-performing classifiers. Notably, the Random Forest (RF) and XGBoost classifiers exhibited superior performance, achieving accuracy rates of 93% and 92% respectively, outperforming the other classifiers.

TABLE III
BEST SCORES IN MACHINE LEARNING AND DEEP LEARNING

	Machine Learning	**Deep Learning**
Accuracy	93.37%	70.41%
Recall	100.00%	95.62%
Precision	93.37%	71.58%
F1 Score	96.57%	81.88%

Implementing SMOTE techniques showed a slight improvement in accuracy, though not significantly impacting neural network performance. The comparison table highlights the best-performing machine learning technique against deep learning, emphasizing the potential of SMOTE to enhance model performance.

VIII. Comprehensive Performance Measures

The performance of each model is assessed based on four major factors, namely Accuracy, Recall Accuracy, Precision Accuracy, and F1 Score [19].

1) **Classification Accuracy:** True Positives occur when positive instances are accurately predicted, while True Negatives occur when negative examples are correctly predicted. False Positives refer to incorrect predictions of positive cases, and False Negatives refer to inaccurate predictions of negative instances.

   Accuracy= Accurate Predictions / All Predictions

2) **Precision:** Every successful prediction made by the model has a precision score of 1, which is perfect. A score of 0 on the other hand means that no accurate positive predictions were generated by the model.

   Precision = True Positives / (True Positives + False Positives)

3) **Recall:** A score of 0 means that no positive cases were correctly identified, whereas a perfect recall score of 1 means that all of them were.

   Recall = True Positives / (True Positives + False Negatives)

4) **F1 Score:** The F1 score, ranging from 0 to 1, is a balanced statistic that assesses classification algorithm performance, with 1 indicating strong performance and 0 indicating weak performance, making it useful for model evaluation and comparison.

   F1 Score = 2(Precision * Recall) / (Precision + Recall)

IX. Conclusion

This study investigates how the combination of different SMOTE techniques and machine learning methods can effectively detect defects in manufacturing processes. The research findings highlight that employing various SMOTE methods significantly enhances the accuracy, precision, recall, and f-score of the classifiers by creating superior features. Prior to implementing SMOTE, only the RF classifier demonstrated satisfactory results, but after applying SMOTE techniques, particularly the random oversampling approach, both the RF and XGBoost classifiers exhibited the most promising performance in predicting manufacturing defects. These findings emphasize the crucial role of SMOTE and feature selection methodologies in improving the effectiveness of machine learning algorithms for detecting faults in manufacturing processes, leading to greater efficiency and accuracy.

Looking ahead, the results of this study hold considerable potential for the application of machine learning in the manufacturing industry. The incorporation of feature selection approaches successfully decreased the number of features, which resulted in a successful reduction in training time. By integrating various SMOTE techniques and feature selection methods, it becomes possible to develop more precise and efficient monitoring systems for manufacturing processes, thereby reducing defects and enhancing product quality. Additionally, the study suggests exploring alternative feature selection techniques to further enhance machine learning performance in detecting faults during manufacturing processes, offering new avenues for future research in this field.

References

[1] Dogan, A., Birant, D. (2021). Machine learning and data mining in manufacturing. Expert Systems with Applications, 166, 114060.

[2] Stuart, T. E. (2000). Interorganizational alliances and the performance of firms: a study of growth and innovation rates in a high-technology industry. Strategic management journal, 21(8), 791-811.

[3] Godina, R., Rodrigues, E. M. (2021). A Review of Data Mining Applications in Semiconductor Manufacturing. Processes, 9(2), 305.

[4] Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., Conde, J. G. (2009). Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. Journal of biomedical informatics, 42(2), 377-381.

[5] Kim, J., Han, Y., Lee, J. (2016). Data imbalance problem solving for smote-based oversampling: Study on fault detection prediction model in the semiconductor manufacturing process. Advanced Science and Technology Letters, 133, 79-84.

[6] Salem, M., Taheri, S., Yuan, J. S. (2018). An experimental evaluation of fault diagnosis from imbalanced and incomplete data for smart semiconductor manufacturing. Big Data and Cognitive Computing, 2(4), 30.

[7] Kerdprasop, K., Kerdprasop, N. (2010, March). Feature selection and boosting techniques to improve fault detection accuracy in the semiconductor manufacturing process. In World Congress on Engineering 2012. July 4-6, 2012. London, UK. (Vol. 2188, pp. 398-403). International Association of Engineers.

[8] Moldovan, D., Chifu, V., Pop, C., Cioara, T., Anghel, I., Salomie, I. (2018, September). Chicken swarm optimization and deep learning for manufacturing processes. In 2018 17th RoEduNet conference: networking in education and research (RoEduNet) (pp. 1-6). IEEE.

[9] Munirathinam, S., Ramadoss, B. (2016). Predictive models for equipment fault detection in the semiconductor manufacturing process. IACSIT International Journal of Engineering and Technology, 8(4), 273-285.

[10] Karthigaikumar, P. (2021). Industrial quality prediction system through data mining algorithm. Journal of Electronics and Informatics, 3(2), 126-137.

[11] Doke, O. (2020). Data Mining for Enhancing Silicon Wafer Fabrication (Doctoral dissertation, Dublin, National College of Ireland).

[12] Cioara, T., Anghel, I., Moldovan, D., Tomus, M. M., Salomie, I. Prediction of Manufacturing Processes Errors: Gradient Boosted Trees Versus Deep Neural Networks.

[13] I. Anghel, T. Cioara, D. Moldovan, I. Salomie and M. M. Tomus, "Prediction of Manufacturing Processes Errors: Gradient Boosted Trees Versus Deep Neural Networks," 2018 IEEE 16th International Conference on Embedded and Ubiquitous Computing (EUC), Bucharest, Romania, 2018, pp. 29-36, doi: 10.1109/EUC.2018.00012.

[14] H. Kuzuno and S. Otsuka, "Early Detection of Network Incident Using Open Security Information," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, Poland, 2018, pp. 18-23, doi: 10.1109/WAINA.2018.00057.

[15] Mohammed, R., Rawashdeh, J., Abdullah, M. (2020, April). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In 2020 11th international conference on information and communication systems (ICICS) (pp. 243-248). IEEE.

[16] Shinde, P. P., Shah, S. (2018, August). A review of machine learning and deep learning applications. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-6). IEEE.

[17] Keerin, P., Kurutach, W., Boongoen, T. (2012, October). Cluster-based KNN missing value imputation for DNA microarray data. In 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 445-450). IEEE.

[18] Chen, X. W., Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In Sixth international conference on machine learning and applications (ICMLA 2007) (pp. 429-435). IEEE.

[19] Shung, K. P. (2021). Accuracy, Precision, Recall or F1? 2018. URL: https://towardsdatascience. com/accuracy-precision-recall-or-f1-331fb37c5cb9 (3.7. 2021.).