Machine Learning Anomaly Detection in Semiconductor Manufacturing Processes Utilizing Sensor Signals: A SECOM Dataset-Based Integrated Pipeline Approach with SMOTE, PCA, and Statistical Feature Selection

Seongjin Park
University of Wisconsin-Madison
Computer Science & Data Science
(seongjinpark99@gmail.com)

https://github.com/Seongjin74/Efficient-Semiconductor-Anomaly-Detection

1. Introduction

In current manufacturing processes, the importance of effective analysis and prediction using limited and heterogeneous data is increasingly recognized. In particular, data from manufacturing processes come with vast amounts of information collected from sensors, but also with issues such as missing values, noise, and multicollinearity. These challenges in data preprocessing and transformation significantly impact the performance of the final prediction model. Consequently, efficiently preprocessing the given data and applying appropriate analytical techniques plays a key role in product quality management and anomaly detection.

Semiconductor manufacturing processes are especially noteworthy from this perspective. In semiconductor manufacturing, managing product quality and detecting defects or anomalies early—based on numerous sensor data—is essential. However, sensor data face several limitations, including class imbalance, high dimensionality, missing values, noise, and multicollinearity, which, if unaddressed, can lead to degraded model performance and difficulties in interpretation.

2. Related Work

To address issues such as imbalance, noise, and high dimensionality in manufacturing process data, previous studies have proposed various preprocessing and modeling techniques. For instance, to mitigate the problem of data imbalance, Chawla et al. [2] proposed SMOTE (Synthetic Minority Over-sampling Technique), a representative methodology that artificially augments minority class data to help models learn these classes more effectively. Subsequent studies have applied SMOTE along with various modifications to real manufacturing process data to validate its effectiveness, and He & Garcia [5] comprehensively discussed the severity of imbalanced data issues and various approaches to resolve them.

On the other hand, sensor data collected from manufacturing processes include a large number of features, which brings about challenges such as the "curse of dimensionality" and noise. Consequently, Principal Component Analysis (PCA) is widely used as a dimensionality reduction technique to preserve the primary information in the data while eliminating unnecessary noise. Jolliffe [3] provided a detailed overview of the theoretical background and applications of PCA. More recently, to overcome the limitations of simple PCA, nonlinear and cluster-based dimensionality reduction techniques such as hierarchical PCA have been proposed, attracting attention for their ability to more effectively reflect complex interactions within the data.

Furthermore, it has been observed that a single preprocessing technique is often insufficient to fully resolve the inherent complexities in manufacturing process data, leading to active research on hybrid and ensemble methodologies that combine various preprocessing techniques and classifiers. Breiman [4] demonstrated that ensemble learning techniques like Random Forest can improve prediction performance by compensating for the weaknesses of individual models. Similarly, Van Hulse et al. [6] and Blagus & Lusa [7] showed that combining statistical feature selection with dimensionality reduction techniques can effectively extract meaningful features from complex manufacturing process data.

Building on these previous studies, this research proposes an integrated pipeline for the SECOM dataset. The pipeline incorporates missing value imputation, class imbalance resolution using SMOTE, dimensionality reduction via PCA combined with statistical feature selection, and model optimization using GridSearchCV. GridSearchCV performs cross-validation on various hyperparameter combinations for different models to identify the optimal settings, thereby maximizing model performance. Through this approach, the study aims to overcome the limitations of existing methods in manufacturing process anomaly detection and contribute to the development of more reliable prediction models.

3. Proposed Method

This study integrates previous research to improve a machine learning model that predicts anomaly detection using signal data. Specifically, for the SECOM dataset, preprocessing techniques such as missing value imputation, removal of constant values and noise, and alleviation of multicollinearity are applied, and the class imbalance issue is addressed using SMOTE. Subsequently, hierarchical PCA and statistical feature selection are employed to enhance the efficiency of the data. Six classifiers are individually trained and evaluated to establish baseline performance, followed by an optimal classifier selection process using GridSearchCV.

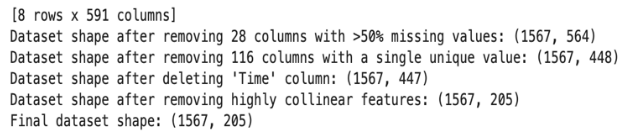
This approach is expected to maximize the efficiency of data utilization in quality management and anomaly detection in manufacturing processes, thereby contributing to the development of more reliable prediction models. In this study, we comprehensively evaluate the impact of preprocessing and optimization processes on the final model's performance by comparing each stage of the proposed integrated pipeline—from Stage 1 (original data) to Stage 2 (SMOTE applied), Stage 3 (SMOTE + PCA), Stage 4 (SMOTE + statistical feature selection + PCA), and Stage 5 (SMOTE + statistical feature selection + PCA + GridSearchCV).

The SECOM dataset, provided on November 18, 2008, consists of semiconductor manufacturing process data with 1,567 examples and 591 features. Each example includes sensor signals corresponding to a production unit along with a simple Pass/Fail label (with 104 failures). This dataset contains a variety of signals collected from actual processes, highlighting the critical importance of meaningful feature selection and noise removal [1].

4. Experiments

4.1 Data Preprocessing

Data preprocessing is a critical step that directly impacts model performance. Various techniques were applied to address missing values, noise, unnecessary features, and class imbalance issues present in the raw data.



```
[8 rows x 591 columns]
Dataset shape after removing 28 columns with >50% missing values: (1567, 564)
Dataset shape after removing 116 columns with a single unique value: (1567, 448)
Dataset shape after deleting 'Time' column: (1567, 447)
Dataset shape after removing highly collinear features: (1567, 205)
Final dataset shape: (1567, 205)
```

Figure 1: Data Metrics After Preprocessing

First, manufacturing process data often exhibit frequent missing values during sensor measurement, which can introduce noise during data analysis and model training. Therefore, columns containing over 50% missing values were removed, as they were deemed to provide unreliable information. For the remaining missing values, forward fill and backward fill methods—considering the order of the data—were used to preserve data continuity. This approach reduced unnecessary noise by removing features with excessive missing values and minimized data loss through the filling technique. However, since the sequential filling method assumes temporal characteristics in the data, it may have limitations compared to other interpolation methods when data order or seasonality is not reflected.

Additionally, columns with a constant value across all samples provide no distinguishing information for the model and only add to the computational burden. Thus, these constant features were removed, improving learning efficiency and reducing unnecessary dimensionality. The 'Time' column, which is not directly related to the analysis objective, was also removed since it does not affect model performance. Moreover, high correlations among many sensor data features can indicate redundant information when the correlation coefficient exceeds 0.7. In such cases, only one representative feature was retained while the others were eliminated to mitigate multicollinearity. Although this process enhanced model interpretability and reduced the risk of overfitting by removing redundant information, careful consideration of the correlation threshold is necessary to avoid losing significant information.

Given the varied scales of features, StandardScaler was used to normalize all features to have a mean of 0 and a standard deviation of 1. This normalization ensured that each feature was compared on the same scale, improving model training stability and convergence speed. However, it should be noted that normalization may not significantly impact some tree-based algorithms.

Finally, visualization confirmed that the original data were highly imbalanced, with approximately 93.36% normal (Pass) and 6.64% defective (Fail).

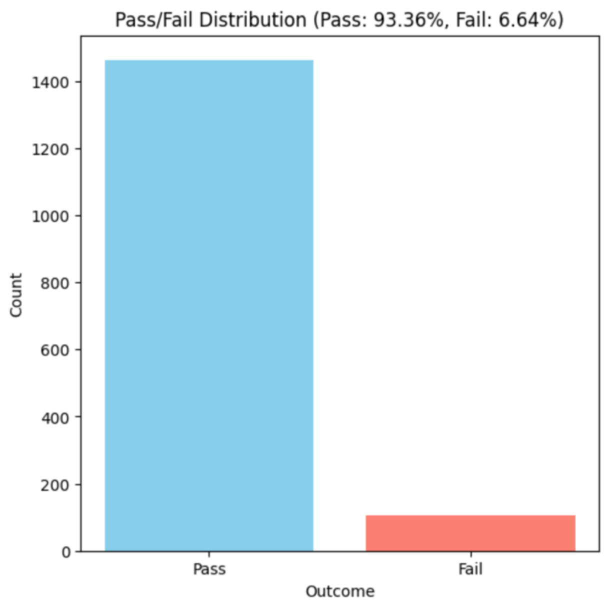


Figure 2: Pass/Fail Distribution Visualization

Visualizing the target variable distribution provided an intuitive understanding of the imbalance problem, highlighting the necessity for oversampling techniques like SMOTE in subsequent steps.

4.2 Model Training and Evaluation

After data preprocessing, the entire dataset was split into training and testing sets to evaluate the model's generalization performance. Six classifiers—Decision Tree, Naive Bayes, Logistic Regression, K-NN, SVM, and Neural Network—were compared at each stage to analyze performance changes in detail.

First, in **Step 1**, models were trained on the preprocessed but still imbalanced original data.

```
----- Step 1: Original Data (Imbalanced) -----
                 Model  Accuracy       TPR       FPR  F1 Score
0        Decision Tree  0.872611  0.083333  0.062069  0.511208
1          Naive Bayes  0.433121  0.750000  0.593103  0.369136
2  Logistic Regression  0.888535  0.250000  0.058621  0.597539
3                 K-NN  0.926752  0.041667  0.000000  0.520929
4                  SVM  0.923567  0.000000  0.000000  0.480132
5       Neural Network  0.914013  0.083333  0.017241  0.541903

Step 1 Average Results:
Accuracy    0.826433
TPR         0.201389
FPR         0.121839
F1 Score    0.503475
dtype: float64
```

Figure 3: Average Model Performance Metrics for Step 1

At this stage, the average Accuracy was relatively high at approximately 0.8264. However, due to the imbalanced nature of the data, predictions for the minority class (Fail) were insufficient, with an average TPR (True Positive Rate) of about 0.2014 and an average F1 Score of 0.5035.

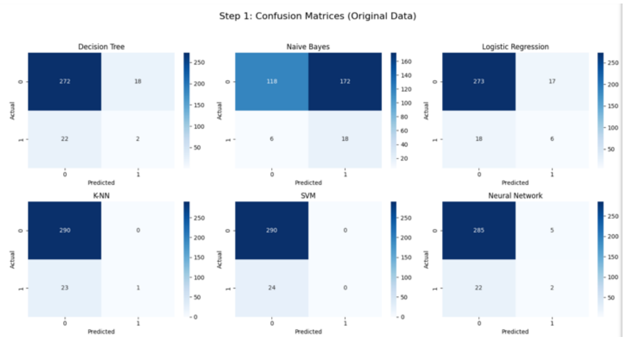


Figure 4: Step 1 – Confusion Matrices

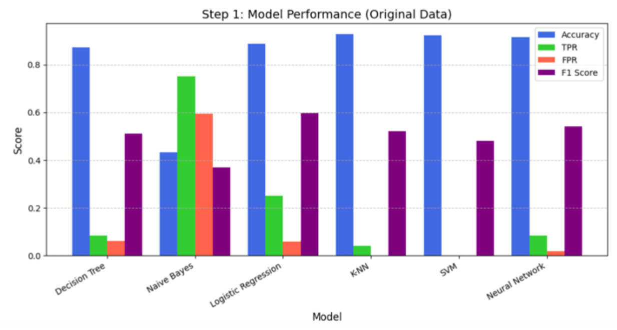


Figure 5: Step 1 – Model Performance

For example, the Naive Bayes model showed a TPR of around 75% but an FPR (False Positive Rate) as high as 59%, indicating a tendency for some models to excessively misclassify the normal class. K-NN and SVM, in particular, failed to recognize the minority class adequately. These results clearly demonstrate that relying solely on Accuracy is insufficient for evaluating the true defect detection performance of the models.

Next, in **Step 2**, SMOTE was applied to artificially augment the minority class data.

```
----- Step 2: SMOTE Applied -----
Shape after rebalancing training data: (2346, 204)
                 Model  Accuracy       TPR       FPR  F1 Score
0        Decision Tree  0.843949  0.083333  0.093103  0.495127
1          Naive Bayes  0.484076  0.625000  0.527586  0.392345
2  Logistic Regression  0.796178  0.416667  0.172414  0.560224
3                 K-NN  0.289809  0.916667  0.762069  0.273533
4                  SVM  0.920382  0.041667  0.006897  0.516238
5       Neural Network  0.891720  0.125000  0.044828  0.546088

Step 2 Average Results:
Accuracy    0.704352
TPR         0.368056
FPR         0.267816
F1 Score    0.463926
dtype: float64
```

Figure 6: Average Model Performance Metrics for Step 2

As a result, while the average Accuracy dropped to about 0.7044, the average TPR improved to approximately 0.3681, indicating enhanced defect detection capability. However, applying SMOTE also increased the average FPR—the rate of misclassifying normal samples—to around 0.2678, slightly reducing the overall F1 Score to 0.4639.

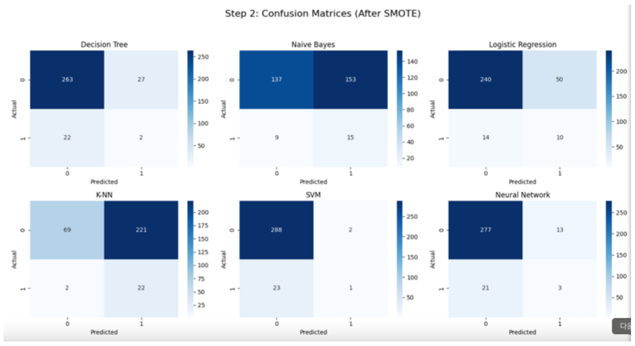


Figure 7: Step 2 – Confusion Matrices

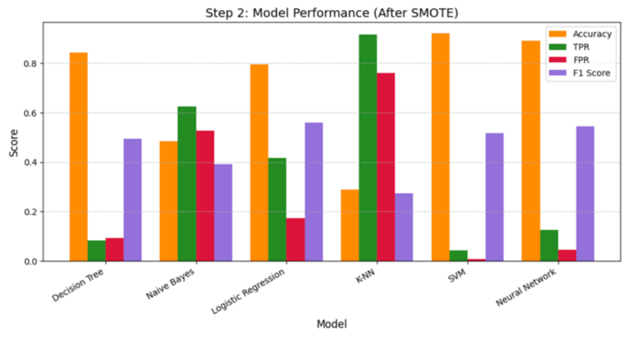


Figure 8: Step 2 – Model Performance

Notably, the K-NN model exhibited an extremely high TPR of 91.67%, but its FPR also soared to 76.21%, suggesting that while SMOTE enhanced augmentation, it also risked learning excessive noise, leading to predictions overly skewed toward the minority class.

In **Step 3**, PCA was applied to the SMOTE-augmented data, reducing the dimensionality to 50 components. This process helped reduce unnecessary noise and computational cost.

```
----- Step 3: SMOTE + PCA (Dimensionality Reduction) -----
Shape after rebalancing training data (SMOTE applied): (2346, 204)
X_train shape after PCA: (2346, 50)
                 Model  Accuracy       TPR       FPR  F1 Score
0        Decision Tree  0.828025  0.291667  0.127586  0.554727
1          Naive Bayes  0.802548  0.208333  0.148276  0.513689
2  Logistic Regression  0.726115  0.333333  0.241379  0.496682
3                 K-NN  0.859873  0.375000  0.100000  0.606292
4                  SVM  0.907643  0.083333  0.024138  0.536236
5       Neural Network  0.894904  0.125000  0.041379  0.548909

Step 3 Average Results:
Accuracy    0.836518
TPR         0.236111
FPR         0.113793
F1 Score    0.542756
dtype: float64
```

Figure 9: Average Model Performance Metrics for Step 3

The average Accuracy recovered to approximately 0.8365, and the average FPR dropped significantly to 0.1138. However, the average TPR decreased somewhat to 0.2361 compared to the SMOTE stage, resulting in an F1 Score improvement to 0.5428. While PCA might cause some information loss, it generally contributed to stabilizing overall model performance.

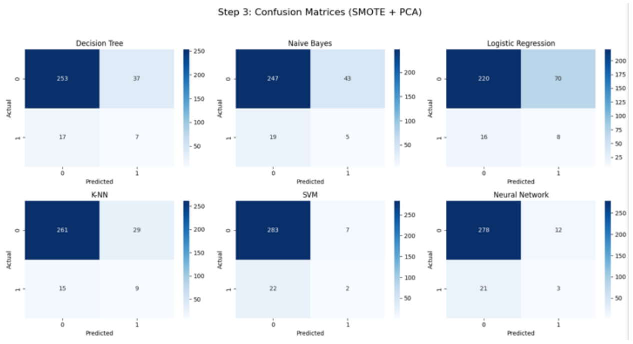


Figure 10: Step 3 – Confusion Matrices

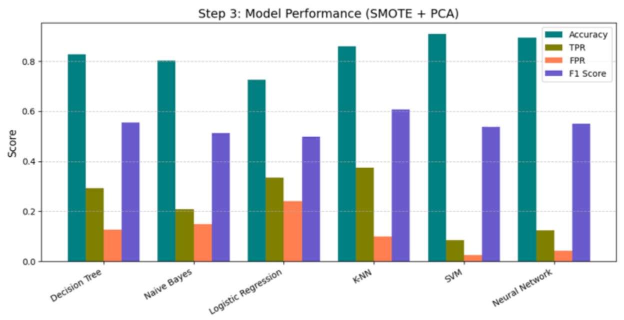
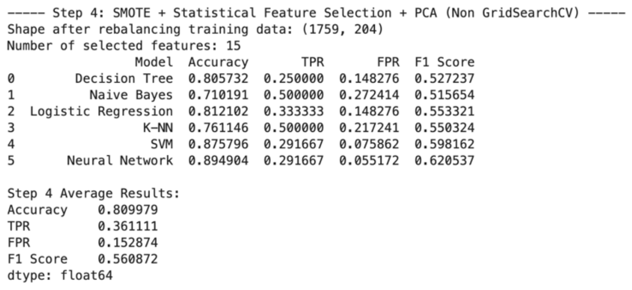


Figure 11: Step 3 – Model Performance

In **Step 4**, statistical feature selection and hierarchical PCA were introduced. At this stage, without applying GridSearchCV, the model was evaluated after selecting 15 meaningful features based on statistical criteria and extracting key information from each feature group through hierarchical PCA.


```
----- Step 4: SMOTE + Statistical Feature Selection + PCA (Non GridSearchCV) -----
Shape after rebalancing training data: (1759, 204)
Number of selected features: 15
                 Model  Accuracy       TPR       FPR  F1 Score
0        Decision Tree  0.805732  0.250000  0.148276  0.527237
1          Naive Bayes  0.710191  0.500000  0.272414  0.515654
2  Logistic Regression  0.812102  0.333333  0.148276  0.553321
3                 K-NN  0.761146  0.500000  0.217241  0.550324
4                  SVM  0.875796  0.291667  0.075862  0.598162
5       Neural Network  0.894904  0.291667  0.055172  0.620537

Step 4 Average Results:
Accuracy    0.809979
TPR         0.361111
FPR         0.152874
F1 Score    0.560872
dtype: float64
```

Figure 12: Average Model Performance Metrics for Step 4

The average TPR was approximately 0.3611, the average FPR was 0.1529, the average Accuracy reached 0.8100, and the average F1 Score increased to 0.5609. The combination of feature selection and dimensionality reduction improved the detection performance for the minority class and boosted the F1 Score, though a slight increase in FPR indicated a trade-off with an elevated rate of false positives for normal products.

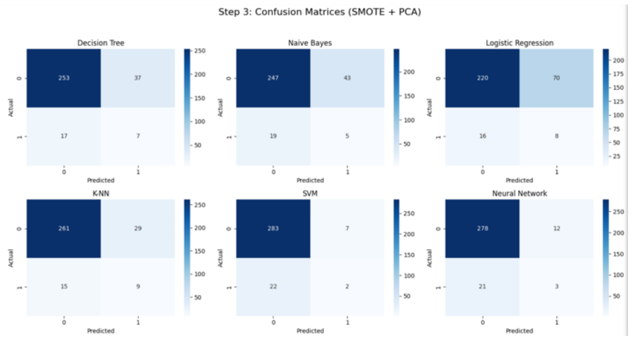


Figure 13: Step 4 – Confusion Matrices

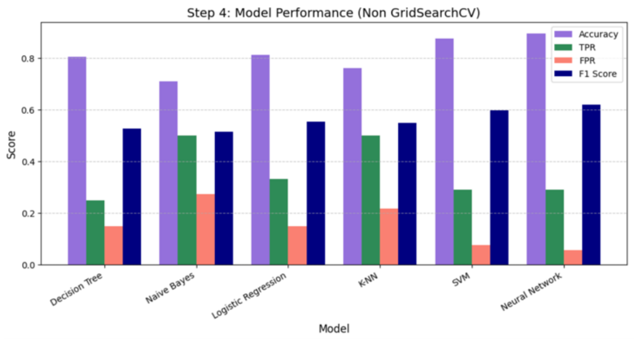
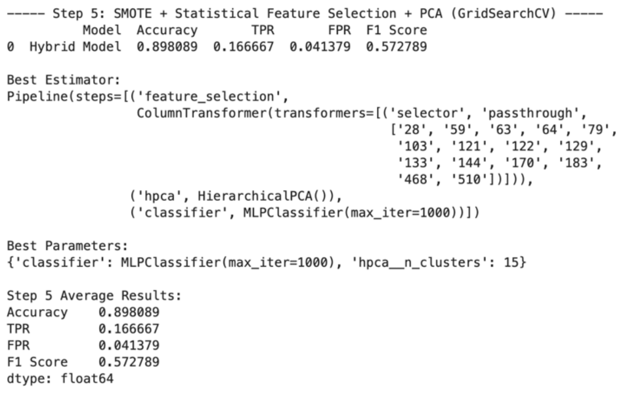


Figure 14: Step 4 – Model Performance

Finally, in **Step 5**, GridSearchCV was employed to optimize key parameters—such as the number of clusters in hierarchical PCA and various classifier hyperparameters—resulting in the final hybrid model.


```
----- Step 5: SMOTE + Statistical Feature Selection + PCA (GridSearchCV) -----
          Model  Accuracy       TPR       FPR  F1 Score
0  Hybrid Model  0.898089  0.166667  0.041379  0.572789

Best Estimator:
Pipeline(steps=[('feature_selection',
                 ColumnTransformer(transformers=[('selector', 'passthrough',
                                                  ['28', '59', '63', '64', '79',
                                                   '103', '121', '122', '129',
                                                   '133', '144', '170', '183',
                                                   '468', '510'])])),
                ('hpca', HierarchicalPCA()),
                ('classifier', MLPClassifier(max_iter=1000))])

Best Parameters:
{'classifier': MLPClassifier(max_iter=1000), 'hpca__n_clusters': 15}

Step 5 Average Results:
Accuracy    0.898089
TPR         0.166667
FPR         0.041379
F1 Score    0.572789
dtype: float64
```

Figure 15: Average Model Performance Metrics for Step 5

Comparing candidate models (Decision Tree, Naive Bayes, Logistic Regression, K-NN, SVM, Neural Network), the ultimately selected model was a hybrid pipeline that combined statistical feature selection and hierarchical

PCA as preprocessing steps with an MLPClassifier (max_iter=1000) as the classifier. The optimal parameters obtained through GridSearchCV set the number of clusters in hierarchical PCA to 15, and the MLPClassifier was configured with a maximum of 1000 iterations for training.

As a result of this optimization, the final model achieved an average Accuracy of 0.8981, an average FPR of 0.0414, an average TPR of 0.1667, and an average F1 Score of 0.5728. The notably low FPR is particularly significant for quality management in manufacturing processes. However, the relatively low TPR can be interpreted as a trade-off made to minimize false positives in normal products, reflecting a cost balance that may require further improvement in future research.

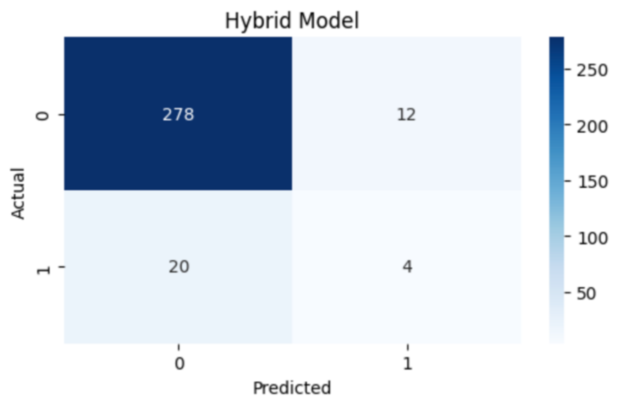


Figure 16: Step 5 – Confusion Matrices

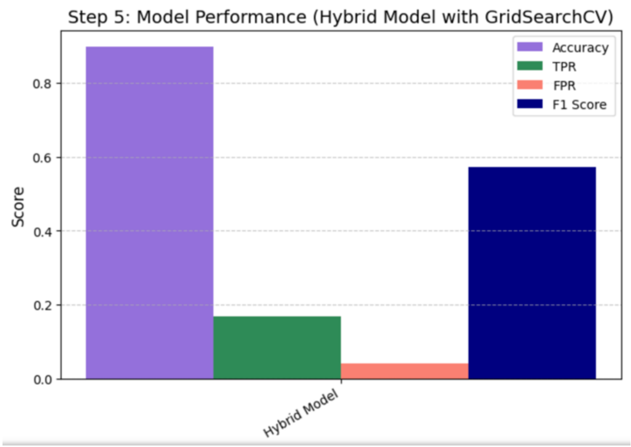


Figure 17: Step 5 – Model Performance

5. Experimental Evaluation

1. Necessity of Overcoming Imbalanced Data (SMOTE):

a. In the original data, although Accuracy appeared high, low TPR and F1 Score indicated insufficient defect (minority class) detection performance. With SMOTE, the augmentation of minority class data improved TPR, but it also resulted in an increase in FPR.

2. Role of Dimensionality Reduction (PCA):

a. The introduction of PCA reduced noise and computational cost while overall improving Accuracy and F1 Score. However, due to some information loss, TPR might decrease slightly.

3. Contributions of Feature Selection and Hierarchical PCA:

a. Statistical feature selection retained only the important features, and hierarchical PCA extracted the key information from each group, allowing the model to focus on more concentrated information, which resulted in improved TPR and F1 Score.

b. It should be noted that this process also exhibited a slight increase in FPR.

4. **Effect of Optimization via GridSearchCV:**

a. The final hybrid model was selected through GridSearchCV and achieved high Accuracy with a very low FPR, contributing significantly to “minimizing false positives” in manufacturing quality control.

b. However, the slightly lower TPR observed during the optimization process can be interpreted as a reflection of the trade-off between actual defect detection and cost.

4. Suitability of the Evaluation:

a. The F1 Score, which balances Precision and Recall, has the advantage of comprehensively capturing defect

detection performance in imbalanced data.
b. In scenarios like manufacturing process anomaly detection, where the cost of false positives and false negatives is critical, it is advisable to consider multiple evaluation metrics such as Accuracy, TPR, and FPR along with the F1 Score.

6. Conclusion and Future Work

In this study, an integrated pipeline is proposed for improving the performance of a machine learning anomaly detection model using sensor signals collected from semiconductor manufacturing processes, based on the SECOM dataset. The proposed pipeline sequentially applies data preprocessing, minority class augmentation via SMOTE, dimensionality reduction using PCA and hierarchical PCA, statistical feature selection, and optimization through GridSearchCV. This approach not only enhances key performance metrics—Accuracy and F1 Score—for manufacturing quality control but also achieves an extremely low false positive rate (FPR) for normal products. These results indicate that the method effectively overcomes various data quality and imbalance issues encountered in semiconductor manufacturing processes, thereby laying an important foundation for practical applications.

However, the results of this study also have several limitations. First, the artificial expansion of minority class data via SMOTE may introduce discrepancies with the actual data distribution, leading to increased FPR and excessive false positives in some models. Additionally, some significant information might be lost during the PCA and hierarchical PCA processes, resulting in a slightly reduced defect detection ability (TPR) that needs to be addressed. The feature selection step also relies on current statistical methodologies, which may not fully capture non-linear relationships or complex interactions between variables.

To address these limitations, future research should consider a comprehensive comparative analysis of various imbalance handling techniques such as ADASYN and undersampling to develop a more refined data augmentation method. Instead of PCA, applying non-linear dimensionality reduction techniques like autoencoders could be explored to more effectively preserve important non-linear features while reducing dimensionality. The feature selection process could also be improved by incorporating non-linear metrics such as Mutual Information to achieve more precise feature selection. Finally, although this study primarily focused on offline data, the nature of semiconductor manufacturing processes requires application in real-time data streaming environments. Future work should focus on integrating online learning techniques and real-time data processing systems to implement automated anomaly detection systems in field environments.

In summary, this study confirms that combining data preprocessing with optimization techniques is an effective approach for manufacturing quality control and anomaly detection. Further research into advanced data augmentation, preservation of non-linear features, precise feature selection, and the development of real-time processing systems is expected to lead to even more sophisticated automated anomaly detection systems.

Reference

[1] McCann, M. & Johnston, A. (2008). SECOM [Dataset]. UCI Machine Learning Repository.
[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.
[3] Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer.
[4] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
[5] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.
[6] Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. In Proceedings of the 2007 ACM symposium on Applied computing (pp. 984-988).
[7] Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics, 14(1), 106.