Comparative Analysis of Feature Selection and Imbalance Handling for Semiconductor Defect Detection on the SECOM Dataset

Sidorela Uku
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
Sidorela.Uku@ijs.si

Klemen Kenda
Jožef Stefan Institute
Ljubljana, Slovenia
Klemen.Kenda@ijs.si

Abstract

Semiconductor manufacturing demands reliable automated defect detection, yet applying machine learning to real-world fabrication data is complicated by severe class imbalance, high dimensionality, and missing values. This paper presents a comparative study of feature selection and imbalance handling strategies for defect prediction on the SECOM benchmark dataset (1,567 samples, 591 features, 6.6% defect rate). Unlike much prior work, we enforce strict data integrity by performing train/test splitting before any label-dependent preprocessing, preventing information leakage that can inflate reported performance. We evaluate six experimental flows spanning three feature selection paradigms: a Pearson filter, the Boruta wrapper, and FASTENER, an evolutionary search applied here to SECOM for the first time, combined with SMOTE, ADASYN, and class-weighting strategies using LightGBM and SVM classifiers. Every pipeline is additionally assessed by five-fold cross-validation with within-fold preprocessing. On the held-out split the strongest pipeline reaches a geometric mean (GM) of 0.745, but five-fold cross-validation shows single-split estimates to be optimistic by approximately 0.08 GM, and the three selectors become statistically indistinguishable once the split is varied. Threshold choice matters more than feature selection, with class-weighting and threshold tuning raising GM from 0.524 to 0.721. These findings highlight the importance of methodological rigor and principled pipeline design for industrial defect detection.

Keywords

Semiconductor Manufacturing, Defect detection, SECOM, Class imbalance, Feature selection, LightGBM

1 Introduction

Semiconductor manufacturing generates high-dimensional sensor data throughout hundreds of sequential process steps. Ensuring low defect rates is critical, as faulty products lead to significant economic losses and resource waste. Machine learning (ML) offers the potential to automate defect detection by learning complex patterns from sensor measurements, replacing costly and time-consuming manual inspection. However, applying ML to real-world fabrication data presents several interrelated challenges: severe class imbalance, high dimensionality, missing values, and noisy or irrelevant features.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.125

The SECOM (SEmiCOnductor Manufacturing) dataset [10] is one of the most popular public benchmarks for studying these challenges. It contains 1,567 samples with 590 sensor features and 1 timestamp column, of which only 104 (6.6%) represent defective products. Approximately 4.5% of values are missing, and 116 features have constant values, offering no discriminative information. This combination of properties makes SECOM representative of the difficulties encountered in industrial production data more broadly, and it has been identified as one of the three most commonly used manufacturing datasets in ML research [3]. Nevertheless, a systematic review of publicly available manufacturing datasets found that well-documented datasets with clearly labeled features remain scarce, and SECOM's anonymized feature names limit process-level interpretability [6].

Despite a growing body of literature applying ML to the SECOM dataset, existing studies differ substantially in their preprocessing pipelines, algorithm choices, resampling strategies, feature selection methods, and evaluation protocols. These inconsistencies make a direct comparison of the reported results difficult and open the question of which methodological choices truly contribute to improved defect detection. This paper presents a comparative analysis of six pipelines using three feature selection paradigms and three imbalance handling strategies. This work contributes to existing literature in four ways. First, we show that threshold tuning accounts for a GM difference larger than feature selection. Second, we quantify the optimism of a single-split evaluation in comparison to five-fold split. Third, we separate two forms of selector stability: the Pearson filter recovers an identical feature set under markedly different missing-value handling, yet retains fewer than half its features across resampled splits, locating the instability in the scarcity of defective samples rather than in preprocessing choices. Lastly, we report the first application of FASTENER to SECOM, which gives comparable result to the other selectors nor matches their stability, retaining only one feature across all five folds.

2 Related Work

Research on defect detection for semiconductor manufacturing has progressed in different areas of ML, including classifier selection, feature engineering, and methods to handle imbalanced data. Table 1 summarises prior ML and the design choices for each case. It can be observed that tree ensembles and SVM dominate as the classifier and SMOTE as the most used resampling technique. However, reported results present a span of ROC AUC from 0.72-0.95 on the same dataset, which reflects also that the difference is not just change of methods but also the evaluation protocols.

Park et al. [12] addresses this in their work, on the importance of building a pipeline that ensures no data leakage. They argue that train/test splitting must be carried out before any

Table 1: Prior ML studies on SECOM. Reported results are not directly comparable: metrics, splits, and preprocessing order differ across studies. CW + thr. denotes class weighting with threshold tuning.

Study	Feature selection	Imbalance handling	Classifier	Best reported
[11]	Boruta, MARS, PCA	SMOTE, undersampling	LR, RF, GBT	Prec. 0.915
[1]	MARS/BGT	Oversampling	SVM or NN	Prec. 0.99, TPR= 0.918
[7]	ANOVA	SMOTE, undersampling	RF/MLP/RUSBT/SVC/GDBT	ROC AUC 0.72
[4]	12-method voting (≥3)	under + oversampling	LR/SVM/DTC/RF/GBC/XGB	ROC AUC 0.95
[5]	None/Zero-variance	None/SMOTE-ENN	LR/RF/SVM/NN/EEC/BRFC	Prec. 0.26
[13]	missingness-aware repr.	CW + thr. selection	LR, RF, LightGBM	PR-AUC 0.221
[12]	Pearson $\|r\| \geq 0.10$	ADASYN/SMOTE or KM/BL/SVM+SMOTE	LR/SVC/KNN/DT/RF/XGB	GM 0.73
This work	Pearson, Boruta, FASTENER	SMOTE, ADASYN, CW + thr.	LightGBM, SVM	GM 0.746 (0.660 CV)

preprocessing steps. Since handling missing values or reducing dimensionality on full dataset leaks test information into training. We adopt their pipeline and reproduce their best performing case as our baseline.

Closest to our work, Park and Baek [13] evaluate Logistic Regression, RF and LightGBM with validation-based threshold selection, using SECOM as a supplementary dataset to compare imputed against missingness-aware feature representations. They report that on SECOM the validation-selected threshold yielded zero recall on the held-out test set, attributing this to operating-point instability when only 21 defective samples are available per split. Our study takes SECOM as the primary dataset, compares feature *selection* paradigms rather than feature representations, and quantifies that instability directly through repeated cross-validation.

Recent work has also begun to reformulate the task itself. Bae et al. [2] replace binary pass/fail classification with autoencoder-based sub-class discovery, using four jointly optimised loss components (separation, clustering, repulsion, balance) to decompose defect labels into mechanism-aligned sub-classes; combined with Prototypical Networks this gave a 47.8% relative improvement over a binary few-shot baseline on SECOM, with further validation on proprietary wafer test data.

3 Methodology

This study investigates optimal combinations of feature selection and imbalance handling methods for defect prediction using the SECOM dataset. Following prior recommendation, we ensure data integrity by performing train/test splitting (70/30, stratified) *before* any label-dependent preprocessing. This prevents information leakage that can artificially inflate performance metrics.

3.1 Preprocessing Pipeline

Before splitting the data, we identified and removed features with zero variance (constant values across all samples) and entirely empty features. This step was label-independent, as variance is computed from feature values alone. A total of 116 features were removed, reducing the feature space from 590 to 474. The data were then split using stratified sampling into training (1,096 samples: 1,023 Pass, 73 Fail) and test (471 samples: 440 Pass, 31 Fail) sets. Features with more than 50% missing values in the training data were removed (24 features), and the same mask was applied to the test set. As an additional check, we used LightGBM feature importance to confirm that none of the dropped features ranked in the top quartile of importance.

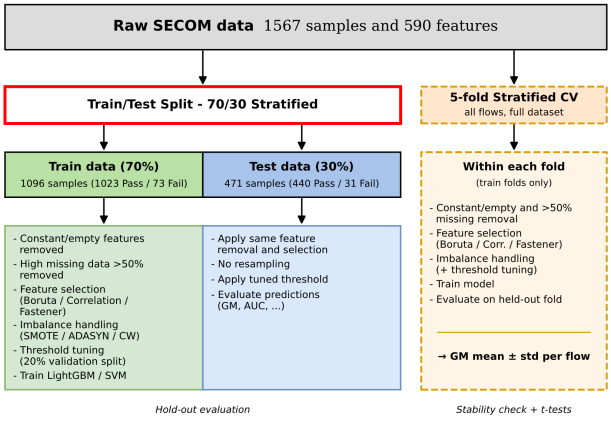


Figure 1: Preprocessing and experimental pipeline overview.

3.2 Feature Selection

We compared three paradigms. *Correlation-based* selection is a filter retaining features with $|r| \geq 0.10$, following Park et al. [12]. *Boruta* is a statistical wrapper using Random Forest (200 trees, max_depth=7, balanced class weights, 100 iterations, α=0.05) that retains features significantly more important than randomized shadow copies. *FASTENER* is an evolutionary wrapper [9] that searches subsets with a mutually guided genetic algorithm, using a base learner of decision-trees (max_depth = 6, balanced class weights) over 100 generations. We ran five seeds and kept features selected at least three times across the five runs. Boruta and correlation each selected 12 features, sharing only 5; FASTENER selected 19.

3.3 Imbalance Handling

We compared three approaches to address the severe 1:14 class imbalance, which differ in whether they modify the training data or not. SMOTE oversamples the minority class by interpolating between nearest neighbors, producing an approximately balanced training set. ADASYN, for comparability with [12], extends SMOTE by concentrating synthetic samples where minority instances are harder to classify. Class weighting build-in mechanism from LightGBM leaves the data unchanged. The scale_pos_weight parameter penalises minority misclassifications in proportion to imbalance ratio. Additionally, the decision

threshold is tuned on a validation split to maximise the geometric mean. All resampling and preprocessing were fitted on training data only, and full parameter settings are provided in code repository in Section 5.

3.4 Model Training and Experimental Design

LightGBM served as the primary classifier due to native NaN handling and efficiency on imbalanced tabular data (n_estimators=500, max_depth=5, learning_rate=0.05, regularization via L1=0.1 and L2=1.0). It uses Gradient-based One-Side Sampling (GOSS), which keeps high-gradient samples and randomly downsamples low-gradient ones to speed up training while preserving accurate split gains. It is also invariant to feature scaling so normalisation is not needed [8].

Six experimental flows were evaluated:

(1) Boruta + SMOTE + LightGBM
(2) Boruta + Class-Weighting + Threshold Tuning + LightGBM
(3) Correlation + MaxAbs + ADASYN + SVM (Park et al. [12])
(4) Correlation + Class-Weighting LightGBM
(5) Boruta + MaxAbs + ADASYN + SVM (isolate feature selection with SVM)
(6) FASTENER + Class-Weighting LightGBM

This design isolates feature selection effects (Flows 2 vs. 4 vs 6; Flows 3 vs. 5) and imbalance handling effects (Flows 1 vs. 2) independently. Since Park et al. [12] already evaluated 150 combinations and identified SVM + ADASYN + MaxAbs as the best-performing configuration, we adopted it directly instead of testing further SVM variants except the combination with Boruta which is new.

3.5 Evaluation

Each model is evaluated using a comprehensive set of metrics: Balanced Accuracy, Recall/Sensitivity, Specificity, Geometric Mean, and AUC. This work focuses on optimizing the Geometric Mean (GM) as a balanced performance of true positive rate (sensitivity) and true negative rate (specificity) as shown in equation 1. Additionally, five-fold stratified cross-validation with within-fold preprocessing validated stability for all flows.

$$\text{GM} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{1}$$

4 Results and Discussion

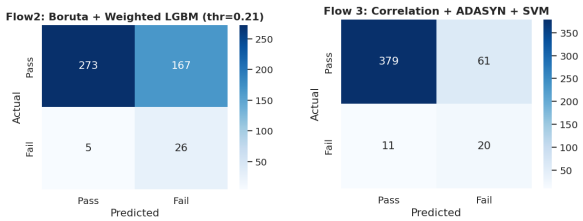
Table 2 summarizes the performance of the test set in all six experimental flows. The results reveal several key patterns regarding the relative importance of imbalance handling and feature selection strategies.

Impact of Imbalance Handling. Flows 1 and 2 share Boruta features and LightGBM but differ in two ways: from resampling method and decision threshold. Flow 2 detects 83.9% defects against 29.0% for Flow 1, and its lower accuracy reflects the 6.6% failure rate rather than a worse model. Additionally, we also tested Flow 2 without threshold tuning, which gave a GM value of 0.636 already better than using SMOTE which is reported in code repository. The GM gap (0.197) is more than four times the AUC gap (0.043) a gap that widens under cross-validation: both pipelines rank samples comparably and diverge mainly in where the boundary is placed. This suggests that the threshold accounts for most of the gain.

Table 2: Test Set Performance Across Experimental Flows. Hold-out versus five-fold cross-validated performance for all six flows, including the standard deviation for GM across folds. Δ the optimism of the single split (hold-out − CV).

	Hold-out				5-fold CV	
Flow	Rec	Spec	GM	AUC	GM mean	Δ
1	0.290	0.946	0.524	0.760	0.408 ± 0.065	+0.116
2	0.839	0.621	0.721	0.803	0.640 ± 0.063	+0.081
3	0.645	0.861	0.745	0.813	0.660 ± 0.088	+0.085
4	0.710	0.648	0.678	0.742	0.633 ± 0.027	+0.045
5	0.613	0.848	0.720	0.804	0.650 ± 0.040	+0.070
6	0.871	0.575	0.707	0.799	0.649 ± 0.042	+0.058

Impact of Feature Selection. Under class-weighted LightGBM (Flows 2, 4, 6) Boruta leads, followed by FASTENER and then Correlation . Under ADASYN and SVM (Flows 3, 5) the order of the first two reverses, with correlation ahead of Boruta. Neither selector is superior independently of the classifier it feeds, and the reversal is comparable in size to the gaps between selectors within either row. Boruta and correlation share only five of twelve features, and FASTENER converges on a third subset of different size. Notably, the correlation filter selected exactly the same 12 sensors as in Park's work, despite our flow following a different approach in handling of missing-values. Yet, under cross validation it retained only five of twelve across all folds. This shows that selection under this filter is robust to preprocessing, but not to training set composition. We additionally tested RFE, PCA and ANOVA within the Flow 1 configuration; all performed worse and are omitted from this report.



(a) Flow 2 (b) Flow 3

Figure 2: Confusion matrix for two of the flows to visualise the different outputs for high number of detected defects with Flow 2 and the highest GM value with Flow 3. Note that the highest recall is from Flow 6, in that case 27 defects were detected but the false positives increase with 20 compared to Flow 2, even though we only detected 1 more.

Moreover, the pipelines differ less in overall quality than in where they sit on the sensitivity vs specificity trade-off. Flow 2 achieves the highest recall (0.839) at the cost of substantially more false positives, while Flow 3 combines moderate recall (0.645) with high specificity (0.861) and correspondingly higher accuracy (Fig. 2). The choice between them depends on the relative cost of false positives and false negatives in the specific manufacturing environment, which also influences the tuning of the classifier towards a selected metric.

Cross-Validation Stability. Table 2 contrasts single-split and five-fold cross-validated performance. Three findings are identified. First, the single split is optimistic for all the pipelines with

less differences in Flow 4 and 6. Which shifts the position on feature selection performance for FASTENER being in the lead when combined with LightGBM with threshold tuning. Second, with the classifier held constant the cross-validated the three feature selectors do not show significant important differences with paired t-tests. Third, the selectors differ markedly in stability: correlation and Boruta retain 5 and 4 features across all five folds, whereas FASTENER retains 1, its subsets varying from 12 to 26 features per fold. Boruta selected 18.0 features on average against 12 on the single split, and the mean tuned threshold was 0.27 against 0.21, reflecting the sensitivity of both selection and calibration to training sample composition. Lastly, while we reported comparative results

Methodological Observations. A key contribution of this work is demonstrating the importance of proper preprocessing order. By performing all label-dependent operations—feature selection, imputation, oversampling, and threshold tuning—strictly within the training partition, we avoid the information leakage that has affected prior SECOM studies. Our results, while lower in absolute terms than some reported in the literature, more accurately reflect real-world generalization performance. False positives remain the dominant practical limitation. At the tuned operating point, Flow 2 flags 167 of 440 passing wafers to recover 26 of 31 defects, roughly six false alarms for every defect caught; Flow 3 trades recall for precision but still raises three false alarms per defect (Fig. 2). This is inherent to the 1:14 imbalance ratio and the small absolute number of defective samples, and suggests that future work should explore cost-sensitive learning frameworks that explicitly model the asymmetric misclassification costs in semiconductor production.

5 Conclusions and Future Work

This study presented a systematic comparison of feature selection and imbalance handling strategies for semiconductor defect detection on the SECOM dataset, with strict accordance to data integrity principles. From our findings, firstly, the choice of imbalance handling method has a substantially greater impact on defect detection performance than the choice of feature selection method. Class-weighting with threshold optimization raised GM from 0.524 to 0.721 under identical feature selection and classifier conditions, although the closely comparable AUC values indicate that the tuned decision threshold, rather than the resampling strategy, accounts for most of this gain. Second, the three selectors capture fundamentally different signal types—Boruta and correlation share only five out of twelve features, and FASTENER converges on a third subset—yet their cross-validated performance did not show a difference. Third, the selectors differ markedly in stability, the Pearson filter and Boruta retain 5 and 4 features across all five folds, whereas the evolutionary search retains none, its subsets varying from 12 to 29 features per fold. Lastly, single-split estimates proved optimistic by approximately 0.045 to 0.116 GM in five of six pipelines, with FASTENER the exception, demonstrating that strict leakage prevention alone is insufficient without repeated evaluation, and that the variance across folds reflects the inherent challenge of small defect sample sizes in industrial datasets.

Future work should address several limitations. The SECOM dataset's anonymized features prevent process-level interpretation of the selected sensors. Exploring ensemble strategies that combine all feature selection approaches may capture complementary signals. Cost-sensitive learning frameworks should be investigated to address the persistently low precision across all pipelines, and deeper integration with recent advances such as sub-class discovery within the defect population [2] may further improve detection in heterogeneous manufacturing environments.

Acknowledgements

This research is supported by EU funding HE MSCA Project Apriori (GA: 101073551) and ARIS Gravitacija (GC-0001).

Data and Code Availability

The source code and experimental configurations are publicly available at https://gitlab.com/Sidorela.Uku/ML_seminar. The SECOM dataset is available from the UCI Machine Learning Repository [10].

References

[1] Ionut Anghel, Tudor Cioara, Dorin Moldovan, Ioan Salomie, and Madalina Maria Tomus. 2018. Prediction of manufacturing processes errors: Gradient boosted trees versus deep neural networks. In *2018 IEEE 16th International conference on embedded and ubiquitous computing (EUC)*. IEEE, 29–36.

[2] Young-Mok Bae, Yingdong He, Zhen He, and Kwang-Jae Kim. 2026. Sub-class discovery in semiconductor defect detection through clustering and few-shot learning. *Expert Systems with Applications* (2026), 131353.

[3] Alican Dogan and Derya Birant. 2021. Machine learning and data mining in manufacturing. *Expert Systems with Applications* 166 (2021), 114060. doi:10.1016/j.eswa.2020.114060

[4] Abdelrahman Farrag, Mohammed-Khalil Ghali, and Yu Jin. 2024. Rare class prediction model for smart industry in semiconductor manufacturing. *arXiv preprint arXiv:2406.04533* (2024).

[5] Daksh Gandhi, Rajat Dandekar, Sreedath Panat, and Raj Dandekar. 2026. Enhancing Failure Detection in Semiconductor Manufacturing Using Balanced Random Forest Model. In *International Conference on Machine Learning and Soft Computing*. Springer, 625–635.

[6] Valentina Golendukhina, Bianca Wiesmayr, and Michael Felderer. 2024. A review of publicly available datasets from manufacturing systems. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–8.

[7] Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. 2022. Product failure detection for production lines using a data-driven model. *Expert Systems with Applications* 202 (2022), 117398.

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[9] Filip Koprivec, Klemen Kenda, and Beno Šircelj. 2020. Fastener feature selection for inference from earth observation data. *Entropy* 22, 11 (2020), 1198.

[10] Michael McCann and Adrian Johnston. 2008. SECOM. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54305.

[11] Dorin Moldovan, Tudor Cioara, Ionut Anghel, and Ioan Salomie. 2017. Machine learning for sensor-based manufacturing processes. In *2017 13th IEEE international conference on intelligent computer communication and processing (ICCP)*. IEEE, 147–154.

[12] Ha-Je Park, Yun-Su Koo, Hee-Yeong Yang, Young-Shin Han, and Choon-Sung Nam. 2024. Study on data preprocessing for machine learning based on semiconductor manufacturing processes. *Sensors* 24, 17 (2024), 5461.

[13] Jung Kyu Park and Youngmi Baek. 2026. Data-Driven Defect Prediction for Manufacturing Quality Monitoring Under Class Imbalance and Missing Data: A Performance–Efficiency Trade-Off Analysis. *Machines* 14, 7 (2026), 716.