

A Study on a Random Forest-Based Algorithm for Predicting Yield Rates in Process-Oriented Production Lines

Weihong Chen

School of Mathematics and Statistics, Fujian Normal University, Fuzhou, Fujian, 350100, China

Abstract: Industrial production line data poses three intertwined difficulties: the features are high-dimensional and anonymized, the process is strongly sequential, and the class distribution is extremely skewed (171:1). This paper presents a Process-based Cascaded Random Forest (P-RF) algorithm tailored to these conditions. Instead of flattening the feature space, P-RF maps the physical order of workstations onto a hierarchy of forest modules and lets a suspicion probability flow forward along the line, so that cumulative error effects are captured rather than ignored. To cope with the severe imbalance, SMOTE resampling, cost-sensitive learning, and grid search are combined into a single optimization scheme. Experiments show that the final model reaches an overall accuracy of 91.21% with a macro-average F1 score of 0.4839, while cutting false alarms to 29,361—a trade-off that favors low disruption on high-volume lines. A global feature-importance traceback further confirms that the transmission probabilities from preceding workstations dominate model decisions, revealing the “error snowball effect” at work. This not only supports the soundness of the P-RF architecture, but also gives production managers a concrete handle for quality control and precise traceability.

Keywords: Yield Rate Prediction; Random Forest; Process-based Cascaded Architecture; Highly Imbalanced Data; Hybrid Resampling.

1. Introduction

1.1. Research Background and Problem Statement

Modern manufacturing is going digital, and production lines now collect far more data than engineers can digest by hand. Predicting yield from that data has become one of the core tasks in smart manufacturing. The dataset used here comes from the Bosch production line, and it is anything but easy to model. Three properties stand out. First, the features are high-dimensional and anonymized: physical formulas cannot be applied directly, and the “curse of dimensionality” is a real concern. Second, the process is strongly sequential—the yield of any station depends on what happened upstream. Third, the classes are extremely imbalanced, with a yield-to-defect ratio as high as 171:1.

Most off-the-shelf machine learning algorithms treat features as a flat, unordered pile. Process-related information is lost, and generalization suffers badly under the 171:1 imbalance. The approach proposed here is “process-oriented modeling combined with joint correction for data imbalance.” In plain terms: respect the order of the workstations, and fix the imbalance from several sides at once. The goal is a predictor that is accurate enough to use in practice, and transparent enough to support cost reduction, efficiency gains, and precise traceability on the shop floor.

1.2. Review of Related Research

Random Forests (RF) have earned their place in data mining largely because they resist overfitting and learn features on their own. Related work is reviewed below in roughly chronological order.

Chang Yuqing’s team at Northeastern University (2021) [1] put forward the Mutual Information-Weighted Random Forest (MIWRF), which reduces redundancy among decision

trees when the data is high-dimensional. Around the same time, Adetunji et al. (2022) [2] used RF to fuse multi-source features for house price prediction—a very different domain, but the same lesson: RF handles non-standardized, heterogeneous features well. Both lines of work, however, still compress features through global statistics. They do not exploit the physical layout of a production line to organize features into refined “buckets.” That gap motivates the preprocessing scheme in this paper, which derives workstation IDs such as L0_S1 and L0_S2 from feature names by regular expressions.

More recent work has shifted toward raw performance and automated tuning in complex industrial settings. Salman et al. (2024) [3] showed that RF ensembles scale well to large datasets and pattern recognition tasks, which speaks to their computational feasibility. Zhao Xiaohui’s team at Xi’an University of Engineering (2026) [4] went a step further, pairing PCA with the Whale Optimization Algorithm (WOA) to automate hyperparameter search for industrial equipment prediction. Progress, in other words, has been real—but the architectures themselves have barely changed. Almost all of them still feed “full-feature flat input” into the model, so the spatiotemporal order between workstations is never represented. On top of that, the extreme 171:1 imbalance is rarely handled in a way that respects the process structure, which keeps generalization limited. The strategy developed in this paper—Process-Oriented Cascaded Forests (P-RF) plus SMOTE with cost-sensitive joint regularization—targets exactly these two gaps.

2. Experimental Design

2.1. Data Sources and Preprocessing Methods

The experiments use data collected by low-level sensor monitoring nodes on the Bosch production line. As noted

earlier, the dataset is high-dimensional, highly anonymized, and strongly sequential, and good products outnumber defective ones by roughly 171:1.

Because the raw data is massive, preprocessing had to be handled carefully. The pipeline developed here has three stages: low-level optimization, logical restructuring, and quality restoration. At the computational level, chunked loading plus a forced downsampling of numeric precision to float32 cut memory use by about half without sacrificing the core information. Next came the restructuring stage. Regular expressions were applied to feature names to extract workstation IDs such as L0_S1 and L0_S2; the flat, anonymous feature set was thus reorganized into relatively independent “workstation feature buckets.” The last stage targets sensor noise: local median interpolation fills missing readings, and a denoising step suppresses outlier-driven fluctuations. Together these measures made the underlying data noticeably cleaner and more robust.

2.2. Design of a Process-Based Cascading Forest Model

The P-RF architecture follows the physical, temporal logic of the Bosch line. The idea is straightforward: mirror the

sequence of workstations in the model itself, so that errors accumulating along the real machining process have a counterpart in the algorithm. Each station gets its own random forest module, and the modules are chained in the order of the line. The three layers are described below.

- **Input Layer:** Each workstation’s feature bucket, as produced by the preprocessing stage, is fed into its own random forest module and processed independently.
- **Cascade Layer (Probability Propagation):** Once trained, the module for the N th workstation outputs P_n , the probability that the current sample is defective. Rather than being used as a verdict, P_n is appended to the feature vector of the $(N+1)$ th workstation as a new feature. Prior suspicion, in other words, travels downstream with the product.
- **Output Layer:** At the last workstation, the module combines the sensor features of the whole line with the cumulative defect probability passed down from every preceding station, and issues the final classification.

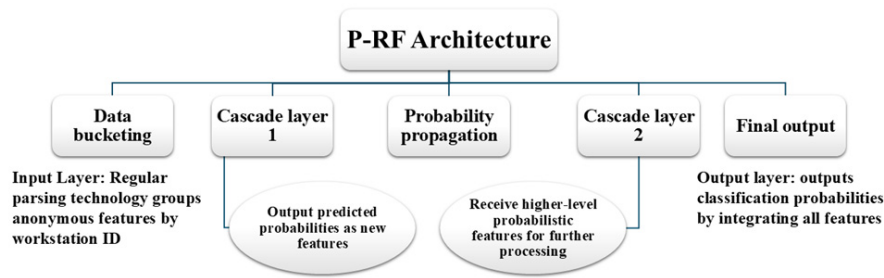


Figure 1. Schematic diagram of the P-RF Architecture

2.3. Handling and Training Strategies for Highly Imbalanced Data

The extreme imbalance was attacked from three directions at once: data augmentation, algorithm calibration, and parameter tuning. For augmentation, SMOTE was applied at a restrained 10% expansion ratio to synthesize virtual defective samples. This relieves the sparsity of the minority class without the memory blow-ups and overfitting risks that full oversampling usually brings. For calibration, a cost-sensitive factor was built into the splitting criterion of the random forest, raising the penalty for false negatives so that the model is pushed to take minority samples seriously during training. With these two mechanisms in place, grid search then tunes the core hyperparameters—maximum tree depth, minimum leaf size, and the like—to find the best compromise between catching defects and keeping false alarms affordable.

3. Experimental Results and In-Depth Analysis

A stepwise ablation study was designed to validate each optimization strategy in turn. The experiments start from a basic cascade model, add hyperparameter tuning, and finally integrate the SMOTE hybrid resampling strategy; a global feature-importance analysis at the end makes the model’s behavior interpretable.

3.1. Experimental Environment and Establishment of Evaluation Criteria

All experiments were implemented in Python 3.13, relying mainly on Scikit-learn and Imbalanced-learn. Metric choice deserves a word here. With good products outnumbering defective ones 171 to 1, plain accuracy is almost meaningless—a model that predicts “good” for everything would look excellent under accuracy. The evaluation therefore uses the confusion matrix, precision, recall, the precision-recall curve (PRC), and the macro-F1 score, which together reflect how well the model actually finds the rare defective samples.

3.2. Phase 1: Comparison of Basic Cascade Construction and Hyperparameter Tuning

Phase 1 began with a baseline P-RF cascade equipped with cost-sensitive penalization (`class_weight=‘balanced’`). A dynamic threshold search on the test set found 0.47 as the best preliminary decision threshold. Not surprisingly for such an imbalanced dataset, this base model landed in the familiar dilemma of high recall paired with a flood of false positives. Grid search was then introduced, taking Macro-F1 as the objective and tuning the maximum tree depth, the leaf node count, and related hyperparameters via three-fold cross-validation. Tables 1 and 2 compare the baseline cascade against the tuned model.

Table 1. Comparison of Confusion Matrices for Base Models and Grid-Search-Optimized Models

Actual category	Base model predicts good products (0)	Base model predicts defective products (1)	The tuned model predicts good products (0)	The tuned model predicts defective products (1)
Actual good units (0)	104,476 (TN)	248,585 (FP)	239,364 (TN)	113,697 (FP)
Actual defective items (1)	598 (FN)	1,466 (TP)	1,368 (FN)	696 (TP)

Table 2. Comparison of Detailed Evaluation Metrics Between the Base Model and the Model Optimized via Grid Search

Key Performance Indicators	Performance of the base model	Performance of the tuned model	Performance Evolution
Macro-F1	0.2339	0.4091	significant improvement
Accuracy	29.83%	67.60%	sharp rebound
Recall	71.03%	33.72%	Strategic pullback
Precision	0.0059	0.0061	Still sluggish

Analysis of Experimental Results: The numbers in Tables 1 and 2 tell a clear story. The unconstrained base model is good at finding defects—recall reaches 71.03%—but it pays for that with 248,585 false positives. Translated into factory terms, the “better safe than sorry” strategy would flood the manual re-inspection line with good products, and overall accuracy sits at just 29.83%. That level of operational disruption is simply not viable at scale.

With deep hyperparameter constraints applied through grid search, the model settles into a much more practical operating point. Recall for defective products drops to 33.72%—still intercepting 696 defective units—while false positives fall to 113,697. That is a reduction of more than 130,000 unnecessary re-inspections of genuine good products. Overall accuracy climbs to 67.60%, and Macro-F1 improves to 0.4091.

The same numbers, however, expose the limits of tuning alone. Against a 171:1 skew, hyperparameter search had clearly hit a ceiling—the precision for defective items stayed stuck around 0.6%. The implication is hard to miss: no amount of algorithmic adjustment can compensate for a data distribution this lopsided. The next phase therefore moves to the data source itself, using SMOTE to reshape the feature space before training.

3.3. Phase 2: Evolution of the Final Model Under a Hybrid Resampling Strategy

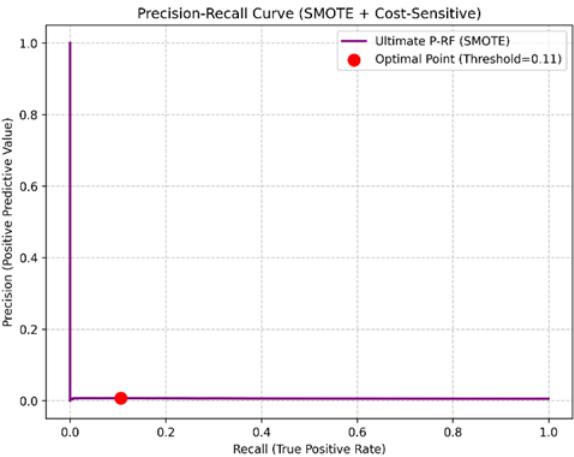


Figure 2. PR curve of the model using the SMOTE hybrid strategy

The earlier phases showed that algorithmic intervention alone has hit its ceiling. The final phase therefore goes upstream, to the data. The strategy is a hybrid one: block-based, memory-limited compression followed by SMOTE. Ten percent of the training set’s defective samples are synthesized as virtual, high-quality examples, which reshapes the feature-space distribution seen by the model.

	Prediction: Good (0)	Prediction: Defective products (1)
Actual: Good (0)	323,700(TN)	29,361(FP)
Actual: Defective items (1)	1,845(FN)	219(TP)

Figure 3. Prediction Confusion Matrix for the SMOTE Ensemble Model

Category/Evaluation Criteria	Precision	Recall	F1-Score
Good (0)	0.9943	0.9168	0.9540
Defective items (1)	0.0074	0.1061	0.0138
Accuracy	-	-	0.9121
MacroAvg	0.5009	0.5115	0.4839
WeightedAvg	0.9886	0.9121	0.9486

Figure 4. Detailed Metrics for Predictions Using the SMOTE Ensemble Model

Analysis of Experimental Results: The final report shows what SMOTE changed. False positives dropped to 29,361, overall accuracy reached 91.21%—the best figure in the whole experiment—and Macro-F1 climbed to 0.4839.

Why did it work? SMOTE widened the feature diversity of the defective class, so the model no longer had to learn from a handful of examples; the decision boundary became considerably more stable. It is true that recall narrowed to 10.61%, which translates to 219 critical defects intercepted. But recall is only half the story on a line where false alarms dominate re-inspection costs: the final model keeps production disruption at an extremely low level. Under a 171:1 disparity, that is a defensible trade-off, and it is the kind of trade-off that large-scale, consumer-grade manufacturing can actually live with.

3.4. Global Feature Importance Analysis and Error Traceback

Analysis of Experimental Results: The traceability results

bring out something worth dwelling on. The top five features driving the model’s decisions are all of the form Cascade_Prob_Input_L0_Sxx—that is, the defect probability

handed down from the preceding process step—and each carries an importance weight above 0.96. Nothing else comes close.

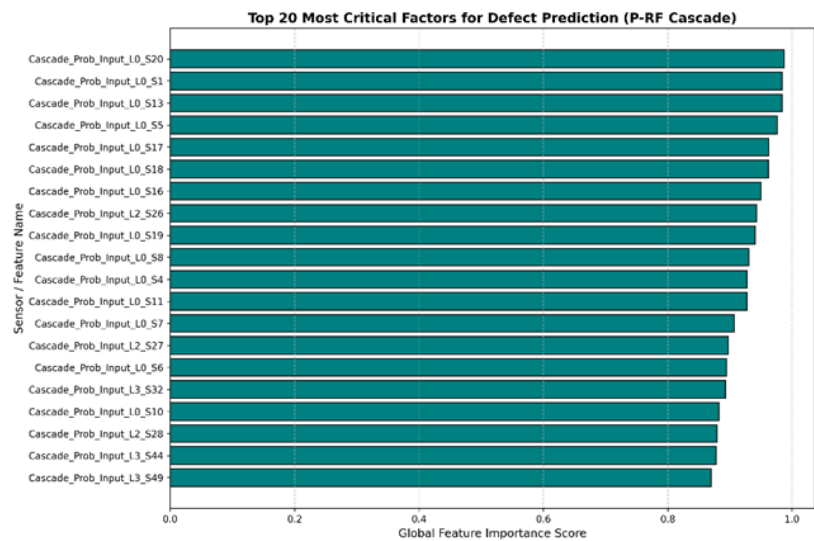


Figure 5. Top 20 Global Error Sources in the P-RF Cascaded Early Warning System

Two inferences follow from this pattern:
 First, defects in modern precision manufacturing rarely spring from one momentary sensor failure. More often, small anomalies are passed down the line and amplified stage by stage. The model’s heavy reliance on cumulative historical probabilities is, in effect, a learned version of this same physical principle.
 Second, the same data strongly supports the P-RF cascade design itself. Given a choice, the model trusts what happened upstream more than the current station’s own reading. That is precisely the behavior the architecture was built to produce, and it doubles as practical advice for factory management: calibrate the sensors at the early core workstations of the L0 line, and the snowball effect of errors propagating downstream can be cut off at its source.

4. Conclusion

This study set out to deal with data that is high-dimensional, anonymous, strongly sequential, and imbalanced at 171:1 on the Bosch line. The result is a validated yield prediction and traceability system built on Process-based Cascaded Random Forests (P-RF). The main conclusions are as follows.
 First, P-RF successfully maps the physical logic of the factory onto the logic of the algorithm. Propagating the defect probability of preceding workstations as a dynamic feature captures the “error accumulation effect” of the assembly line, something flat-input models cannot see. This cascade design, rather than any single trick, is what drives the performance gains reported here.
 Second, the coordinated “data-algorithm-parameters” scheme goes a long way toward taming the 171:1 imbalance.

With 10% SMOTE boosting, cost-sensitive penalization, and grid search working together, false alarms fall to 29,361 and overall accuracy reaches 91.21%, with a Macro-F1 of 0.4839. For a production line where the cost of false alarms dominates, this balance between defect interception and re-inspection cost is a practical one, and it is what makes the model deployable at scale.
 Third, feature tracing highlights the “error snowball effect.” The five most influential error-causing factors are all propagation probabilities from preceding stages, each weighted above 0.96. Because the model is interpretable in this white-box sense, the finding carries operational weight: for production managers, tightening sensor calibration and quality control at the early core processes is the most direct way to break the chain of defect propagation.

References

- [1] Chang, Y. Q., Sun, X. T., Zhong, L. S., & et al. (2021). Evaluation of industrial process operating conditions based on an improved random forest algorithm. *Acta Automatica Sinica*, 47(9), 2214-2225. <https://doi.org/10.16383/j.aas.c200341>.
- [2] Adetunji, A. B., Akande, O. N., Ajala, F. A., & et al. (2022). House price prediction using random forest machine learning technique. *Procedia Computer Science*, 199, 806-813. <https://doi.org/10.1016/j.procs.2022.01.101>.
- [3] Salman, H. A., Kalakeh, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning*, 2024, 69-79.
- [4] Zhao, X. H., Liu, L., Pu, J. P., & et al. (2026). Prediction of vibration in the lower housing of a coal mill based on modal analysis and PCA-WOA-RF. *Journal of Shandong University (Engineering Edition)*, 56(1), 150-157.