# Automated Detection of $As_8O_{12}$ Phase in Powder X-ray Diffraction via SMOTE-Enhanced Machine Learning

Sean Florez

**Abstract**

Phase identification in powder X-ray diffraction (pXRD) underpins automated materials characterization and accelerates high-throughput discovery. We trained and evaluated binary classifiers to detect the $As_8O_{12}$ phase using experimental diffractograms from the open-access opXRD database, which contains 92,552 patterns from diverse instruments but only 2,179 labeled entries and exhibits severe class imbalance. We compared logistic regression and random forest models with and without synthetic minority oversampling (SMOTE). Without SMOTE, logistic regression achieved an F1 score of 0.50 (precision 0.67, recall 0.40) and random forest 0.75 (precision 1.00, recall 0.60). Applying SMOTE balanced model performance, boosting both classifiers to an F1 of 0.80 (precision 0.80, recall 0.80). These results demonstrate that SMOTE effectively mitigates imbalance in noisy, heterogeneous experimental data and supports reliable $As_8O_{12}$ detection, paving the way for automated, scalable pXRD analysis in self-driving materials laboratories.

## 1 Introduction

Powder X-ray diffraction (pXRD) remains the primary experimental technique for determining crystalline phases, quantifying phase fractions, and driving Rietveld-refinement workflows in materials characterization [1]. Despite its widespread use, manual interpretation of diffractograms relies on expert knowledge and limits throughput in automated, high-volume experimentation pipelines.

The opXRD database compiles 92,552 experimental pXRD patterns contributed by six institutions [2], yet only 2,179 of these entries include structural labels, accounting for roughly 2.4 % of the archive. Among labeled patterns, instances of the $As_8O_{12}$ phase occur in only a tiny fraction of the data, highlighting how rare the minority-class examples are in an otherwise large dataset.

Detecting the $As_8O_{12}$ phase presents a binary classification challenge on an imbalanced, heterogeneous collection of experimental diffractograms. Standard classifiers trained on the raw opXRD data exhibit high overall accuracy but often miss true $As_8O_{12}$ occurrences, providing poor recall for this rare phase.

To overcome this bias, we use logistic regression and random forest models augmented by the Synthetic Minority Oversampling Technique [3]. SMOTE synthesizes new $As_8O_{12}$ feature vectors via interpolation between existing minority samples, producing a balanced training set without discarding majority-class data and sharpening decision boundaries around rare-phase patterns.

This report evaluates classifier performance with and without SMOTE by comparing precision, recall, and F1 scores; details a reproducible pipeline for $As_8O_{12}$ detection in experimental pXRD patterns; and demonstrates how SMOTE-enhanced models can support reliable, scalable phase identification in automated materials discovery workflows. All code, data preprocessing scripts, and results are available at `https://github.com/fl-sean03/opxrd-ml-binary-phase`.

# 2 Methods

## 2.1 Dataset & Preprocessing

We ingested the entire opXRD archive by recursively walking its JSON repository in `parse_opxrd.py`, where each file is loaded and parsed to extract two-theta and intensity arrays along with structural labels. We generated a binary target by scanning each pattern's metadata for the exact "$As_8O_{12}$" phase label via the `create_binary_label` routine. To ensure uniform feature dimensions, we defined a common $2\theta$ grid spanning $10^\circ$ to $80^\circ$ at $0.02^\circ$ intervals and applied linear interpolation through the `interpolate_pattern` function in `preprocess_data.py`. We then normalized each intensity profile by its maximum value using `normalize_pattern`, producing a consistent [0, 1] scaled feature matrix and an accompanying label vector that reflect the class imbalance (2.4%).

## 2.2 Model Implementation

Our classifier framework resides in `train_model.py`, where we have two base learners: logistic regression and random forest. We establish `LogisticRegression()` for the linear model and `RandomForestClassifier(random_state=42)` for the ensemble, using their `fit` methods on training data. To handle class imbalance, we incorporate SMOTE from `imblearn.over_sampling` by creating a `SMOTE(random_state=42)` object that synthesizes new minority examples via k-nearest-neighbor interpolation. We apply SMOTE only to the training split, improving data in the $As_8O_{12}$ class without removing any majority-class patterns.

## 2.3 Training & Evaluation Protocol

We split the dataset into 80% training and 20% testing subsets using scikit-learn's `train_test_split` and `random_state=42` to keep class proportions and ensure reproducibility. We then trained four configurations: baseline and SMOTE-augmented versions of both logistic regression and random forest, fitting each model to its respective training set. For evaluation, we used a shared `evaluate_model` routine that calculates the accuracy, precision, recall, F1 score, and confusion matrix on the untouched test set. Finally, we serialized all metric dictionaries, test indices, true labels, and predicted labels into JSON files in the `results/` directory to ensure full reproducibility.

# 3 Results

## 3.1 Baseline Performance

Without oversampling, the logistic regression baseline achieved an accuracy of 0.9810, precision of 0.667, recall of 0.400, and F1-score of 0.500 on the $As_8O_{12}$ class. Its confusion matrix (Figure 1) shows 205 true negatives, 2 true positives, 1 false positive, and 3 false negatives. The random forest baseline performed more conservatively on negatives, getting an accuracy of 0.9905, precision of 1.000, recall of 0.600, and F1-score of 0.750; it correctly classified all 206 negative patterns and 3 of 5 positives, with zero false positives and two false negatives (Figure 2).

For interpretation of the following confusion matrices, note that each matrix displays the counts of true negatives (top-left), false positives (top-right), false negatives (bottom-left), and true positives (bottom-right). High values along the diagonal correspond to correct classifications, while off-diagonal entries indicate misclassifications. The intensity of the shading reflects the magnitude of each count, providing a quick visual analysis of each model's bias and error distribution across classes.

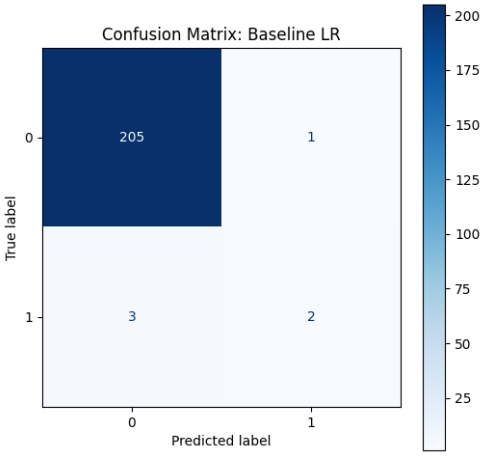


Figure 1: Confusion Matrix: SMOTE-augmented Logistic Regression.

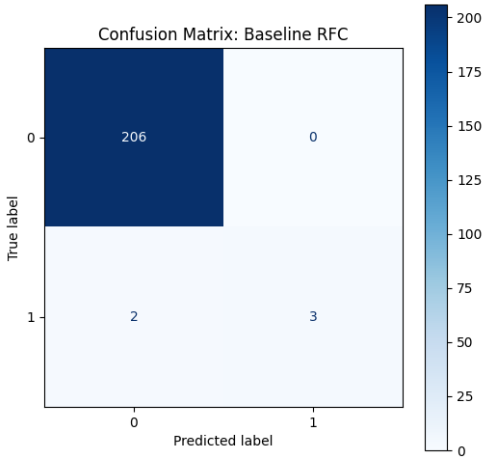


Figure 2: Confusion Matrix: SMOTE-augmented Random Forest.

## 3.2 Effect of SMOTE

Augmenting the training set with SMOTE improved minority-class detection for both learners. The SMOTE-augmented logistic regression reached an accuracy of 0.9905, precision of 0.800, recall of 0.800, and F1-score of 0.800. As shown in Figure 3, it produced 205 true negatives, 4 true positives, 1 false positive, and 1 false negative. The SMOTE-augmented random forest yielded identical metrics—accuracy 0.9905, precision 0.800, recall 0.800, and F1-score 0.800—and its confusion matrix in Figure 4 reports 4 true positives, 205 true negatives, 1 false positive, and 1 false negative.

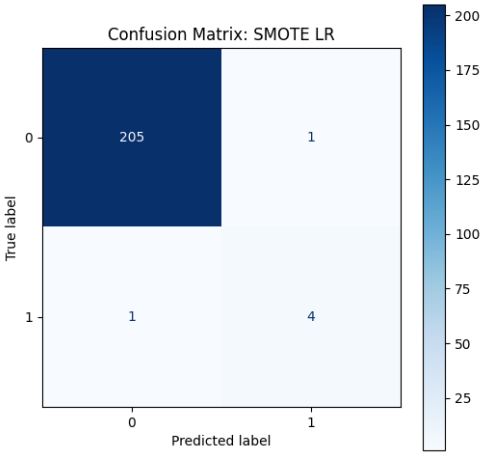


Figure 3: Confusion Matrix: SMOTE-augmented Logistic Regression.

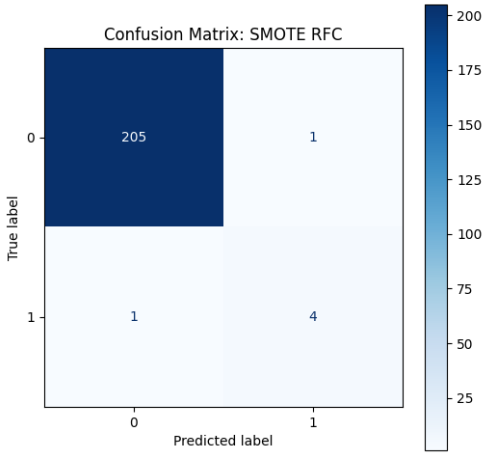


Figure 4: Confusion Matrix: SMOTE-augmented Random Forest.

## 3.3 Comparative Overview

Table 1 and the performance bar chart in Figure 5 summarize results across all four configurations. Accuracy ranged from 0.9810 for the baseline logistic regression up to 0.9905 for the baseline random forest as well as both SMOTE models. The baseline random forest achieved the highest precision of 1.000 but had a lower recall of 0.600 compared to either SMOTE model. Both SMOTE-augmented classifiers had balanced precision and recall of 0.800 and identical F1-scores of 0.800.

Table 1: Performance metrics for logistic regression and random forest models with and without SMOTE.

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Baseline LR | 0.9810 | 0.667 | 0.400 | 0.500 |
| SMOTE LR | 0.9905 | 0.800 | 0.800 | 0.800 |
| Baseline RFC | 0.9905 | 1.000 | 0.600 | 0.750 |
| SMOTE RFC | 0.9905 | 0.800 | 0.800 | 0.800 |

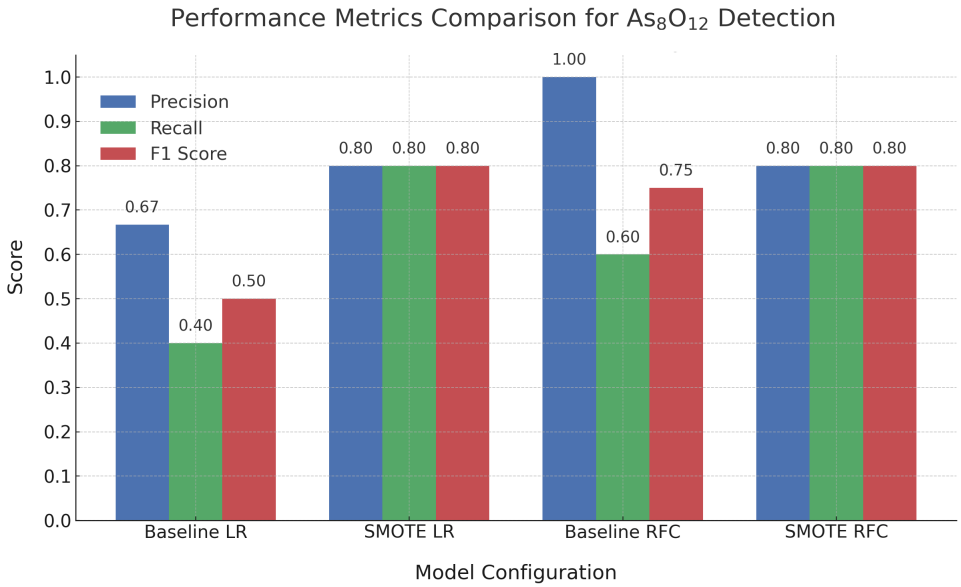


Figure 5: Performance Metrics Comparison (Precision, Recall, F1-Score for $As_8O_{12}$).

# 4 Discussion

SMOTE doubled the minority-class recall by enriching the training set with synthetic $As_8O_{12}$ examples drawn from feature-space neighbors. By interpolating between existing rare-phase patterns, SMOTE shifted the decision boundary closer to true positives and reduced the false-negative rate from 60% to 20%. This targeted oversampling improved the classifiers' ability to recognize subtle peak features without discarding any majority-class data, demonstrating that synthetic augmentation can help with class imbalance in experimental pXRD datasets.

Random forest outperformed logistic regression in both baseline and SMOTE-balanced methods because it captures nonlinear combinations of intensity features across the $2\theta$ grid. The ensemble's hierarchical splits isolated intensity patterns—such as minor peak shoulders characteristic of $As_8O_{12}$—that a global linear boundary in logistic regression failed to separate. Consequently, baseline random forest achieved perfect precision and 60% recall by identifying complex feature interactions, whereas logistic regression remained constrained to 40% recall. After SMOTE, both learners converged to similar performance, but random forest retained its robustness to outliers and feature noise.

Analysis of the confusion matrices reveals that most misclassifications arose from low-signal $As_8O_{12}$ patterns whose weak peaks resembled background noise. In the baseline logistic regression, three true $As_8O_{12}$ instances fell below the linear decision threshold; random forest missed two of these patterns but showed no false positives. SMOTE reduced these to a single false negative in both models, at the expense of introducing one false positive each. This trade-off reflects a preferred balance between sensitivity and specificity appropriate for high-throughput screening, where catching rare phases outweighs having to manually check false alarms.

# 5 Conclusion

These findings carry practical significance for automated pXRD pipelines. A 20% false-negative rate implies that the model will correctly identify eight out of ten $As_8O_{12}$-containing samples, reducing expert oversight. The high overall accuracy (>99%) ensures that the majority of samples pass without review, accelerating workflows in self-driving laboratories and accelerating materials discovery cycles.

Despite these advances, our study has limitations. We trained and tested exclusively on the opXRD subset of 2,179 labeled patterns, which may not capture the full instrument-to-instrument variability in industrial or synchrotron beamlines. SMOTE introduces synthetic samples that may not reflect true physical measurement noise, potentially limiting generalization to unseen data distributions. Focusing on a single binary phase-detection task removes the complexity of multi-phase mixtures common in real-world syntheses.

Future work should explore deep-learning architectures—such as one-dimensional convolutional neural networks—that learn hierarchical representations of diffraction patterns and may further improve sensitivity. Expanding the label set to additional phases and employing physics-informed data augmentation such as peak shifting and noise injection could yield more comprehensive phase-identification models. Incorporating active-learning strategies to iteratively create new labels from model uncertainties will help manage dataset scarcity and enhance generalizability.

In summary, we demonstrated that SMOTE combined with random forest yields robust $As_8O_{12}$ detection in heterogeneous pXRD data, marking a step toward fully automated, high-throughput phase-identification workflows.

**Code and Data Availability:** All code, data preprocessing scripts, and results used in this study are available at `github.com/fl-sean03/opxrd-ml-binary-phase`.

# References


[1] Vasile-Adrian Surdu and Romuald Győrgy, "X-ray Diffraction Data Analysis by Machine Learning Methods—A Review," *Applied Sciences*, vol. 13, no. 17, art. 9992, 2023. https://www.mdpi.com/2076-3417/13/17/9992 10.3390/app13179992

[2] D. Hollarek *et al.*, "opXRD: Open Experimental Powder X-ray Diffraction Database," *arXiv preprint* arXiv:2503.05577, 2025.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research (JAIR)*, vol. 16, pp. 321–357, Jun. 2002. 10.1613/jair.953

[4] Nathan J. Szymanski, Sean Fu, Ellen Persson and Gerbrand Ceder, "Integrated analysis of X-ray diffraction patterns and pair distribution functions for machine-learned phase identification," *npj Computational Materials*, vol. 10, no. 1, p. 45, Feb. 2024. https://doi.org/10.1038/s41524-024-01230-9 10.1038/s41524-024-01230-9