**1️⃣ Observations from the results**

Looking at your CSV, the key metrics to focus on are **recall, F1-score, and precision**, because this is a **highly imbalanced fraud detection problem**. Here’s a summary:

| **Model** | **Method** | **Recall** | **Precision** | **F1-score** | **Observation** |
| --- | --- | --- | --- | --- | --- |
| RandomForestClassifier | SMOTE / SMOTE+Tomek | 0.444 | 0.085 | 0.143 | Detects actual frauds; moderate recall. Best balanced model among all. |
| DecisionTreeClassifier | Tomek | 0.667 | 0.073 | 0.132 | Slightly higher recall but lower F1 → predicts more frauds but more false positives. |
| RandomForestClassifier | Tomek | 0.889 | 0.052 | 0.099 | Extremely high recall but very low precision → almost everything predicted as fraud (impractical). |
| GradientBoostingClassifier / XGBoost / LogisticRegression | Various | 0.0 | 0.0 | 0.0 | Fail to predict any fraud; stuck predicting majority class. |
| XGBoostClassifier (baseline / SMOTE / Tomek) | Various | 0.333–0.666 | 0.038–0.056 | 0.068–0.104 | Detects some frauds with random search/early stopping, but very low F1 and precision. |

**Key points:**

1. **Most models fail to detect fraud**  
   Many configurations of XGBoost, GradientBoosting, and LogisticRegression result in recall = 0 and F1 = 0. These models are **biased toward the majority class** because the dataset is heavily imbalanced.
2. **Resampling techniques matter**
   * SMOTE and SMOTE+Tomek improved RandomForestClassifier performance significantly, allowing the model to detect actual frauds.
   * Tomek alone tends to increase recall drastically but lowers precision (RandomForest\_\_tomek → 0.889 recall, 0.052 precision). This is because it **overpredicts fraud**.
3. **XGBoost with early stopping and random search**
   * Even with hyperparameter tuning and early stopping, XGBoost didn’t outperform RandomForest.
   * Some minor improvements are seen (recall 0.333–0.666 in some runs), but F1 is still low (~0.1), which shows the model **detects frauds inconsistently**.
   * Likely cause: limited random search (5 iterations), and extreme class imbalance still dominates learning.
4. **Threshold tuning**
   * You applied threshold tuning, but the performance still remains poor for many models.
   * This indicates that **the models never learn meaningful signal for fraud**, so changing the threshold doesn’t help much.

**2️⃣ Why RandomForestClassifier with SMOTE/SMOTE+Tomek is still the best**

1. **Detects actual frauds** – it’s the only model consistently predicting minority class (fraud).
2. **Balanced recall and precision** – moderate recall (0.444) and better F1 (~0.143) compared to other models.
3. **Resampling works well** – SMOTE/SMOTE+Tomek addresses class imbalance better than Tomek alone or no resampling.
4. **Stable across experiments** – XGBoost, GradientBoosting, and LogisticRegression often fail to detect fraud entirely, even with tuning.

✅ **Conclusion:** RandomForestClassifier with SMOTE or SMOTE+Tomek remains the **most reliable model** for your dataset. It balances detecting frauds and keeping false positives at a reasonable level.

**3️⃣ Why some models failed despite hyperparameter tuning**

* **Extreme imbalance** → even RandomizedSearchCV or GridSearchCV cannot help if the signal for fraud is too weak.
* **Limited search space** → only 5 iterations of random search for XGBoost may not cover enough hyperparameter combinations.
* **Data preprocessing** → resampling is applied after scaling, but models like LogisticRegression or GradientBoosting are **more sensitive to feature scaling and imbalance**, so performance suffers.
* **Overfitting risk** → XGBoost with too many trees and small fraud examples can memorize training data but fail on test/validation.

**4️⃣ Insights for next steps**

1. Consider **more extensive hyperparameter search** or **Bayesian optimization (e.g., Optuna)** for XGBoost to see if it can match RandomForest.
2. Stick with **RandomForest + SMOTE/SMOTE+Tomek** as baseline.
3. Explore **ensemble methods** like RandomForest + XGBoost or stacking to improve F2-score.
4. Keep **F2-score and recall as main metrics**, because accuracy is misleading in imbalanced datasets.