A chart with stars on it

AI-generated content may be incorrect.

**Algorithm Categories & Use Cases**

**1. Unsupervised Anomaly Detection**

**Isolation Forest (Your Current)**

* **Best for**: High-dimensional data, scalable
* **Strengths**: Fast, handles mixed data types
* **Payment use case**: General transaction anomalies

**One-Class SVM**

* **Best for**: High-dimensional, complex boundaries
* **Strengths**: Robust to outliers, kernel flexibility
* **Payment use case**: Fraud detection, complex patterns

**Local Outlier Factor (LOF)**

* **Best for**: Local density-based anomalies
* **Strengths**: Finds local outliers, adaptive
* **Payment use case**: Merchant-specific anomalies

**DBSCAN**

* **Best for**: Clustering-based anomalies
* **Strengths**: No assumptions about data distribution
* **Payment use case**: Transaction clustering, fraud rings

**2. Ensemble Methods**

**Voting Ensemble**

python

*# Combines multiple algorithms for better accuracy*

algorithms = ['isolation\_forest', 'one\_class\_svm', 'lof']

ensemble\_result = voting\_ensemble(data, algorithms)

**Weighted Ensemble**

python

*# Weights based on algorithm performance*

weights = {

'isolation\_forest': 0.3,

'one\_class\_svm': 0.25,

'lof': 0.25,

'statistical': 0.2

}

**3. Statistical Methods**

**Z-Score Detection**

* **Best for**: Gaussian distributed data
* **Formula**: z = (x - μ) / σ
* **Payment use case**: Amount-based anomalies

**IQR Method**

* **Best for**: Skewed distributions
* **Formula**: Outlier if x < Q1 - 1.5\*IQR or x > Q3 + 1.5\*IQR
* **Payment use case**: Transaction amount outliers

**Modified Z-Score**

* **Best for**: Non-Gaussian data
* **Uses**: Median Absolute Deviation (MAD)
* **Payment use case**: Robust to extreme values

**4. Payment Domain-Specific**

**Amount-Based Detection**

* Large transaction anomalies
* Micro-transaction patterns
* Round number bias
* Merchant-specific amount patterns

**Temporal Pattern Detection**

* Off-hours transactions
* Weekend high-value transactions
* Holiday patterns
* Velocity anomalies

**Settlement Pattern Detection**

* Long settlement delays
* Weekend settlements
* Inconsistent settlement patterns
* High processing fees

**🏆 Recommended Ensemble Strategy**

Production Ensemble Strategy

Code

# Production Ensemble Strategy for Payment Anomaly Detection class ProductionAnomalyEnsemble: """ Production-ready ensemble for British Airways payment anomaly detection """ def \_\_init\_\_(self): self.models = {} s

Retry

G

Continue

Edit

Production Ensemble Strategy

Code ∙ Version 2

# Merchant-specific recommendations high\_anomaly\_merchants = df.iloc[high\_anomalies]['merchant\_id'].value\_counts() if len(high\_anomaly\_merchants) > 0: top\_merchant = high\_anomaly\_merchants.index[0] reco

**🎯 Algorithm Selection Guide for British Airways**

**Production Recommendations:**

**Tier 1: Primary Detection (Real-time)**

python

*# For live transaction monitoring*

primary\_ensemble = {

'isolation\_forest': 0.4, *# Fast, scalable*

'statistical\_zscore': 0.3, *# Simple, interpretable*

'payment\_specific': 0.3 *# Domain knowledge*

}

**Tier 2: Secondary Analysis (Batch)**

python

*# For detailed investigation*

secondary\_ensemble = {

'one\_class\_svm': 0.35, *# Complex patterns*

'lof': 0.25, *# Local anomalies*

'dbscan': 0.2, *# Clustering*

'elliptic\_envelope': 0.2 *# Gaussian assumptions*

}

**Tier 3: Specialized Detection**

python

*# For specific use cases*

specialized\_detectors = {

'amount\_based': 'Large transactions, micro-payments',

'temporal\_based': 'Off-hours, weekend patterns',

'settlement\_based': 'Delay patterns, fee anomalies',

'merchant\_based': 'Merchant-specific baselines'

}

**📊 Performance Comparison Summary**

| **Algorithm** | **Speed** | **Accuracy** | **Scalability** | **Interpretability** | **Use Case** |
| --- | --- | --- | --- | --- | --- |
| **Isolation Forest** | ⭐⭐⭐⭐⭐ | ⭐⭐⭐⭐ | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ | General anomalies |
| **One-Class SVM** | ⭐⭐⭐ | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ | ⭐⭐ | Complex patterns |
| **LOF** | ⭐⭐ | ⭐⭐⭐⭐ | ⭐⭐ | ⭐⭐⭐⭐ | Local anomalies |
| **DBSCAN** | ⭐⭐⭐ | ⭐⭐⭐ | ⭐⭐⭐ | ⭐⭐⭐⭐ | Clustering |
| **Statistical** | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ | ⭐⭐⭐⭐⭐ | ⭐⭐⭐⭐⭐ | Simple outliers |
| **Ensemble** | ⭐⭐⭐ | ⭐⭐⭐⭐⭐ | ⭐⭐⭐⭐ | ⭐⭐⭐ | **Production** |

**🚀 Implementation Roadmap**

**Phase 1: Enhance Current System (Week 1)**

* Add ensemble voting to existing Isolation Forest
* Implement statistical Z-score detection
* Add payment-specific rules

**Phase 2: Advanced Algorithms (Week 2)**

* Integrate One-Class SVM for complex patterns
* Add LOF for merchant-specific anomalies
* Implement DBSCAN for clustering analysis

**Phase 3: Production Ensemble (Week 3)**

* Build weighted ensemble system
* Add performance monitoring
* Implement automatic retraining

**Phase 4: Specialized Detectors (Week 4)**

* Amount-based detection rules
* Temporal pattern analysis
* Settlement-specific algorithms