What is Anomaly Detection?

Anomaly detection is the process of identifying data points that deviate significantly from the norm. These "anomalies" could be:

- **Point anomalies**: single data points that are too far from the rest.
- **Contextual anomalies**: abnormal in a specific context (e.g., high CPU at night).
- **Collective anomalies**: a group of values behaving unusually together (e.g., a DDoS attack).
- **Other forms**
    - **Behavioral anomalies**: when a user's behavior suddenly changes (e.g., clicks, navigation).
    - **Dimensional anomalies**: where the anomaly only appears in a subset of features/dimensions.
    - **Semantic anomalies**: data that is statistically normal but logically impossible (e.g., a birthdate in the future).

🔹 1. Point Anomalies

A single data point that significantly differs from the rest.

📌 Examples:

1. A credit card transaction of $10,000 when most are under $100.
2. A sudden temperature spike to 70°C in a room usually kept at 22°C.
3. A person's weight recorded as 2000 kg—clearly a sensor error.
4. One stock in a portfolio dropping 40% in a day while others stay stable.
5. A heart rate reading of 180 BPM while the person is resting.

🔸 2. Contextual Anomalies

Anomalies that are only unusual **given a context**—often time-based.

📌 Examples:

1. A high CPU usage at 2 AM (normal during work hours, but not at night).
2. High electricity usage on a public holiday (unusual compared to typical usage patterns).
3. Increased website traffic at 3 AM (odd for a local business).
4. Sudden drop in sales during a usually busy season.
5. A person’s step count hitting 20k on a day marked as “rest day”.

These usually appear in **time-series** or **geo-spatial** datasets.

▼ 3. Collective Anomalies

A group of data points that are only anomalous **together**, not individually.

📌 Examples:

1. A rapid sequence of login attempts—indicating a brute-force attack.
2. A block of zero readings in a sensor stream—might indicate a malfunction.
3. An unusual drop in multiple server metrics at once—possible system outage.
4. A group of users suddenly buying the same item in bulk—potential bot or fraud.
5. A cluster of patients reporting similar rare symptoms in the same hospital—could be an outbreak.

Often seen in **logs, sequences, or grouped events**.

▼ 4. Other Forms

There are also nuanced variations or special cases:

- **Behavioral anomalies**: when a user's behavior suddenly changes (e.g., clicks, navigation).
- **Dimensional anomalies**: where the anomaly only appears in a subset of features/dimensions.
- **Semantic anomalies**: data that is statistically normal but logically impossible (e.g., a birthdate in the future).

□ 1. Behavioral Anomalies

When a user, system, or entity behaves **differently** than their usual patterns—even if the behavior is not unusual in general.

📌 Explanation:

- Not a point anomaly, because the behavior might be "normal" *for others*.
- Not a contextual anomaly, because the context (e.g., time) might be fine.
- It's the *change in behavior* of the **specific user/entity** that stands out.

🔍 Examples:

1. A user who usually checks email 2–3 times a day suddenly sends 100 emails in an hour → potential spammer or hacked account.
2. A customer who usually buys small items suddenly purchases high-end electronics → could be fraud.

3. A cloud server that idles most days suddenly starts sending large outbound traffic → data exfiltration?
4. A gamer who usually plays solo starts interacting with multiple accounts rapidly → account farming or bot behavior.
5. An employee who rarely accesses HR files starts downloading large chunks of sensitive data → insider threat?

---

2. Dimensional Anomalies (Subspace/Feature-wise Anomalies)

A data point appears normal in **most features**, but is **anomalous in a specific dimension** or subset of the data.

📌 Explanation:

- These are **harder to detect** because traditional methods often assume global similarity.
- Often arise in **high-dimensional data**, like network logs, sensor arrays, or gene expressions.

🔍 Examples:

1. In a dataset of product reviews, a review might be normal in length and rating, but contains a rare keyword (like a hidden promo code).
2. A network packet has normal size and timing, but an odd destination port or header flag.
3. A car engine sensor reports all normal values except for cylinder #4 temperature being 30°C higher.
4. An email has regular metadata (sender, time), but the email body contains encrypted or obfuscated content.
5. A customer record seems normal, but the billing country doesn't match shipping or IP geolocation.

💡 **Detection Tip**: Subspace anomaly detection algorithms like **HiCS**, **LODA**, or **Feature Bagging in PyOD** help here.

---

3. Semantic Anomalies (Logical/Domain Anomalies)

The data is **statistically normal**, but **logically or semantically invalid** based on domain knowledge.

📌 Explanation:

- These anomalies **don't stand out statistically**, which makes them tricky!
- You need **business rules**, **domain logic**, or **external context** to detect them.

🔍 Examples:

1. A person has a birthdate in 2050 → Impossible unless you're in a time-travel dataset 😂
2. A transaction timestamp shows purchase at 03:00, delivery at 01:00 the same day → Logical inconsistency.
3. A car listed as manufactured in 1890 → There were no Toyotas then!
4. A medical record shows a male patient listed as "pregnant" → Biologically invalid.
5. An employee is marked as "terminated" but still has active system access → Possible data/process flaw or breach risk.

💡 Summary Table:

Type	Key Trait	Needs Context?	Common In
Behavioral	Change in *individual's* normal behavior	Yes	User analytics, security
Dimensional	Anomaly in *subset* of features	Sometimes	High-dimensional data
Semantic	*Logically* invalid even if statistically okay	Yes (domain)	Data integrity, QA, compliance

---

Methods for Anomaly Detection in Python

1. Statistical Methods

Good for simpler datasets or when assumptions like normality hold.

- **Z-score / Standard Deviation Method**
- `from scipy.stats import zscore`
- `anomalies = data[(zscore(data) > 3) | (zscore(data) < -3)]`

- **IQR (Interquartile Range)**
- `Q1 = data.quantile(0.25)`
- `Q3 = data.quantile(0.75)`
- `IQR = Q3 - Q1`
- `anomalies = data[(data < Q1 - 1.5 * IQR) | (data > Q3 + 1.5 * IQR)]`

---

2. Machine Learning-based Methods

These are more flexible and powerful, especially with higher-dimensional data.

🌲 Isolation Forest

- **Best for:** High-dimensional datasets
- **How it works:** Randomly isolates data points; anomalies are isolated faster

```
from sklearn.ensemble import IsolationForest

clf = IsolationForest(contamination=0.1, random_state=42)
y_pred = clf.fit_predict(data)  # -1 for anomaly, 1 for normal
```

🌀 One-Class SVM

- **Best for:** Detecting anomalies with a compact normal class
- **How it works:** Learns a boundary that encompasses normal data

```
from sklearn.svm import OneClassSVM

clf = OneClassSVM(nu=0.1, kernel="rbf", gamma=0.1)
y_pred = clf.fit_predict(data)  # -1 for anomaly, 1 for normal
```

□ Local Outlier Factor (LOF)

- **Best for:** Local density-based anomaly detection
- **How it works:** Compares local density of a point with its neighbors

```
from sklearn.neighbors import LocalOutlierFactor

clf = LocalOutlierFactor(n_neighbors=20, contamination=0.1)
y_pred = clf.fit_predict(data)  # -1 for anomaly, 1 for normal
```

□ DBSCAN (Density-Based Spatial Clustering)

- **Best for:** Density-based anomaly detection in arbitrary shapes
- **How it works:** Points in sparse regions (noise) are flagged as anomalies

```
from sklearn.cluster import DBSCAN

clf = DBSCAN(eps=1.2, min_samples=10)
labels = clf.fit_predict(data)
y_pred = (labels == -1).astype(int)  # 1 = anomaly, 0 = normal
```

📐 ABOD (Angle-Based Outlier Detection)

- **Best for:** High-dimensional small datasets
- **How it works:** Uses angle variance between a point and others. High variance implies anomaly.

```
from pyod.models.abod import ABOD
```

```
clf = ABOD()
clf.fit(data)
y_pred = clf.labels_  # 1 = anomaly, 0 = normal
```

👥 KNN (K-Nearest Neighbors for Outlier Detection)

- **Best for:** Intuitive distance-based detection
- **How it works:** Points far from neighbors are more likely to be anomalies

```
from pyod.models.knn import KNN

clf = KNN()
clf.fit(data)
y_pred = clf.labels_  # 1 = anomaly, 0 = normal
```

---

3. Deep Learning-based Methods

Used when dealing with sequences or complex, nonlinear data.

➤ *Autoencoders (Keras/TensorFlow or PyTorch)*

Reconstructs data and flags items with high reconstruction error.

```
from keras.models import Model
from keras.layers import Input, Dense

input_dim = X_train.shape[1]
input_layer = Input(shape=(input_dim,))
encoded = Dense(32, activation='relu')(input_layer)
decoded = Dense(input_dim, activation='sigmoid')(encoded)
autoencoder = Model(inputs=input_layer, outputs=decoded)

autoencoder.compile(optimizer='adam', loss='mse')
autoencoder.fit(X_train, X_train, epochs=50, batch_size=32)
```

Then calculate MSE between input and reconstructed input.

---

4. Time Series Anomaly Detection

Useful when data is sequential (e.g., IoT, logs).

- **Facebook Prophet**
- **ARIMA**
- **LSTM-based Autoencoders**

- **Twitter's AnomalyDetection (in R, but has ports)**

Example with **Prophet**:

```
from prophet import Prophet

df.rename(columns={'timestamp': 'ds', 'value': 'y'}, inplace=True)
model = Prophet()
model.fit(df)
forecast = model.predict(df)
```

---

5. Outlier Detection Libraries

To make life easier:

- `PyOD`: Combines many algorithms (LOF, IF, ABOD, etc.)
- `from pyod.models.knn import KNN`
- `clf = KNN()`
- `clf.fit(data)`
- `y_pred = clf.labels_`
- `Scikit-learn`: Good for traditional ML methods
- `River`: Great for real-time anomaly detection on streams

Why use PyOD?

Because it saves you from coding individual outlier algorithms manually. It includes models from multiple categories:

Algorithm Type	Examples
Distance-based	KNN, HBOS, Mahalanobis
Density-based	LOF, COPOD
Ensemble-based	Isolation Forest, XGBOD
Neural Network-based	AutoEncoder (Keras), SO-GAAL

Anomaly Detection in Various Domains- 1

1. Fraud Detection in Financial Transactions

- **Scenario**: Detecting fraudulent activities such as abnormal credit card transactions or financial transfers.
- **Application**: Use outlier detection to identify transactions that deviate from normal spending behavior, such as sudden large withdrawals, abnormal purchases, or transactions from unusual locations.
- **Algorithms**: KNN, LOF (Local Outlier Factor), ABOD (Angle-Based Outlier Detection).

2. Network Intrusion Detection

- **Scenario**: Detecting unusual patterns in network traffic, such as DDoS attacks or unauthorized access attempts.
- **Application**: Anomalous patterns such as sudden spikes in traffic, strange IP addresses, or unusual data packets can be flagged as potential threats.
- **Algorithms**: KNN, IForest (Isolation Forest), LOF.

3. Manufacturing Quality Control

- **Scenario**: Detecting defective products in a production line based on sensor data.
- **Application**: Outlier detection can be used to identify defective products or out-of-tolerance measurements during manufacturing, such as abnormal temperature, pressure, or speed readings in machinery.
- **Algorithms**: KNN, LOF, IForest.

4. Anomaly Detection in IoT Sensor Data

- **Scenario**: Detecting malfunctions in IoT devices or sensors (e.g., smart meters, health trackers).
- **Application**: If a sensor starts producing values that significantly deviate from the normal pattern, it might indicate a fault in the device or sensor.
- **Algorithms**: KNN, IForest, ABOD.

5. Healthcare Monitoring (Outlier Detection in Patient Data)

- **Scenario**: Detecting unusual health parameters (e.g., heart rate, blood pressure) for early warning signs of medical issues.
- **Application**: Use outlier detection on patient data to flag sudden changes in medical parameters, such as a sharp increase in heart rate or blood pressure, which could indicate a potential emergency.
- **Algorithms**: KNN, LOF, IForest.

6. E-commerce: Product Review Anomaly Detection

- **Scenario**: Detecting fake or fraudulent reviews for products on an e-commerce platform.
- **Application**: Outlier detection can help identify suspicious patterns in product reviews, such as unusually positive or negative reviews from a small group of users, which could indicate manipulation.
- **Algorithms**: KNN, LOF, IForest.

7. Supply Chain Anomaly Detection

- **Scenario**: Identifying abnormal supply chain behavior (e.g., delivery delays, stockouts).
- **Application**: Outlier detection can help track unusual shipping delays or unexpected supply shortages that could disrupt the supply chain.
- **Algorithms**: KNN, LOF, IForest.

8. Energy Consumption Monitoring

- **Scenario**: Detecting unusual energy consumption patterns (e.g., in industrial plants, offices, or homes).
- **Application**: Detecting large spikes or drops in energy consumption that could indicate a malfunction, equipment failure, or unauthorized usage.
- **Algorithms**: KNN, LOF, IForest.

9. Social Media Sentiment Analysis

- **Scenario**: Detecting anomalies in social media content (e.g., sudden spikes in mentions of a brand, unusual sentiment).
- **Application**: Identify outlier posts or trends that represent spikes in sentiment, such as sudden interest in a brand, product, or public figure, which could indicate an emerging issue or opportunity.
- **Algorithms**: KNN, LOF, IForest.

10. Customer Behavior Analysis in Retail

- **Scenario**: Identifying unusual customer purchasing behavior or shopping patterns.
- **Application**: Use outlier detection to identify customers who deviate significantly from typical shopping behavior (e.g., large purchases that do not fit the customer's usual patterns).
- **Algorithms**: KNN, LOF, ABOD.

11. User Behavior Analytics (UBA)

- **Scenario**: Identifying unusual user behavior that could indicate compromised accounts.
- **Application**: Login at odd hours, access from multiple locations in a short time, or accessing sensitive data not typical for the user.
- **Algorithms**: KNN, IForest, LOF, Autoencoders.

12. Email Spam or Phishing Detection

- **Scenario**: Identifying malicious or phishing emails.
- **Application**: Detect outlier email patterns based on sender, content, and frequency.
- **Algorithms**: One-Class SVM, Isolation Forest, NLP + Anomaly Detection.

13. Medical Imaging Anomalies

- **Scenario**: Detecting unusual patterns in X-rays, MRIs, or CT scans.
- **Application**: Identify tumors or abnormalities that deviate from normal tissue structure.
- **Algorithms**: Autoencoders, CNN-based Anomaly Detection, GANs.

14. Clinical Trial Monitoring

- **Scenario**: Identifying abnormal patient responses or data entry errors.
- **Application**: Flag patient records with unusual trends in vitals or side effects.
- **Algorithms**: LOF, IForest.

15. Aircraft Engine Monitoring

- **Scenario**: Detecting early signs of engine failure.
- **Application**: Monitor sensor data (vibration, temperature, pressure) for deviations.
- **Algorithms**: IForest, Autoencoders.

16. Fleet Management

- **Scenario**: Monitoring vehicle performance or fuel usage for anomalies.
- **Application**: Identify vehicles with unexpected fuel consumption or unusual routes.
- **Algorithms**: KNN, Time-Series Models with Anomaly Detection.

17. Payroll Fraud Detection

- **Scenario**: Identifying abnormal payroll disbursements.
- **Application**: Detect duplicate payments, ghost employees, or irregular raises.
- **Algorithms**: IForest, LOF.

18. Expense Claim Fraud

- **Scenario**: Spotting fraudulent employee reimbursements.
- **Application**: Identify unusually high or repeated expense claims.
- **Algorithms**: KNN, Autoencoders.

19. Predictive Maintenance

- **Scenario**: Predicting machine breakdowns before they happen.
- **Application**: Detect irregularities in vibration, pressure, or noise patterns.
- **Algorithms**: IForest, RNNs with Anomaly Detection layers.

20. Pipeline Monitoring

- **Scenario**: Detecting leaks or tampering in oil, gas, or water pipelines.
- **Application**: Outlier detection on flow rate or pressure.
- **Algorithms**: Statistical Anomaly Detection, Time-Series Anomaly Detection.

21. Database Integrity Monitoring

- **Scenario**: Detecting outliers in structured data entries (e.g., missing or duplicate entries).
- **Application**: Spot inconsistent or corrupted records in critical databases.
- **Algorithms**: DBSCAN, LOF.

22. Sensor Calibration Errors

- **Scenario**: Identifying sensors giving faulty data.
- **Application**: Detect values that deviate consistently from similar sensors.
- **Algorithms**: Clustering + Anomaly Detection.

23. Website Activity Monitoring

- **Scenario**: Spotting bots or abnormal browsing patterns.
- **Application**: Detect scraping attempts, brute force logins, or spam.
- **Algorithms**: LOF, IForest, Time-Series models.

24. API Abuse Detection

- **Scenario**: Identifying overuse or malicious use of APIs.
- **Application**: Flag excessive or suspicious API requests.
- **Algorithms**: KNN, Isolation Forest, LSTM-based models.

25. Environmental Monitoring

- **Scenario**: Detecting abnormal pollution levels or climate readings.
- **Application**: Monitor air/water quality sensors for extreme or unexpected values.
- **Algorithms**: IForest, Autoencoders.

26. Smart City Infrastructure Monitoring

- **Scenario**: Detecting irregularities in urban systems (e.g., traffic lights, street lighting).
- **Application**: Find faults or unexpected patterns in public service systems.
- **Algorithms**: LOF, Time-series anomaly detection.

27. Call Detail Record (CDR) Anomaly Detection

- **Scenario**: Detecting fraud or unusual call/SMS patterns.
- **Application**: Identify SIM card cloning or spam calls.
- **Algorithms**: KNN, Autoencoders.

28. Bandwidth Abuse Detection

- **Scenario**: Identifying excessive or unauthorized usage of internet services.
- **Application**: Spot sudden spikes in bandwidth that don't match user behavior.
- **Algorithms**: LOF, Time-Series Forecasting with Outlier Detection.

29. Model Drift Detection

- **Scenario**: Identifying changes in model input/output behavior over time.
- **Application**: Flag input data distributions or predictions that deviate from training data.
- **Algorithms**: Distribution monitoring + Anomaly Detection.

30. Training Data Anomalies

- **Scenario**: Catching mislabeled or noisy data points.
- **Application**: Improve dataset quality by removing statistical or semantic outliers.
- **Algorithms**: ABOD, Isolation Forest, PCA-based detection.

Anomaly Detection in Various Domains- 2

1. Telecom & Communications

- **Call Detail Record (CDR) Anomaly Detection**
  - **Scenario**: Detecting fraud or unusual call/SMS patterns.
  - **Application**: Identify SIM card cloning or spam calls.
  - **Algorithms**: KNN, Autoencoders.
- **Bandwidth Abuse Detection**
  - **Scenario**: Identifying excessive or unauthorized usage of internet services.
  - **Application**: Spot sudden spikes in bandwidth that don't match user behavior.
  - **Algorithms**: LOF, Time-Series Forecasting with Outlier Detection.
- **Website Activity Monitoring**
  - **Scenario**: Spotting bots or abnormal browsing patterns.
  - **Application**: Detect scraping attempts, brute force logins, or spam.
  - **Algorithms**: LOF, IForest, Time-Series models.
- **API Abuse Detection**
  - **Scenario**: Identifying overuse or malicious use of APIs.
  - **Application**: Flag excessive or suspicious API requests.
  - **Algorithms**: KNN, Isolation Forest, LSTM-based models.

---

2. Finance & Business Operations

- **Fraud Detection in Financial Transactions**
  - **Scenario**: Detecting fraudulent activities such as abnormal credit card transactions or financial transfers.
  - **Application**: Use outlier detection to identify transactions that deviate from normal spending behavior, such as sudden large withdrawals, abnormal purchases, or transactions from unusual locations.
  - **Algorithms**: KNN, LOF (Local Outlier Factor), ABOD (Angle-Based Outlier Detection).
- **Payroll Fraud Detection**
  - **Scenario**: Identifying abnormal payroll disbursements.
  - **Application**: Detect duplicate payments, ghost employees, or irregular raises.
  - **Algorithms**: IForest, LOF.
- **Expense Claim Fraud**
  - **Scenario**: Spotting fraudulent employee reimbursements.
  - **Application**: Identify unusually high or repeated expense claims.
  - **Algorithms**: KNN, Autoencoders.
- **Customer Behavior Analysis in Retail**
  - **Scenario**: Identifying unusual customer purchasing behavior or shopping patterns.

    - **Application**: Use outlier detection to identify customers who deviate significantly from typical shopping behavior (e.g., large purchases that do not fit the customer's usual patterns).
    - **Algorithms**: KNN, LOF, ABOD.

---

3. Healthcare & Life Sciences

- **Healthcare Monitoring (Outlier Detection in Patient Data)**
    - **Scenario**: Detecting unusual health parameters (e.g., heart rate, blood pressure) for early warning signs of medical issues.
    - **Application**: Use outlier detection on patient data to flag sudden changes in medical parameters, such as a sharp increase in heart rate or blood pressure, which could indicate a potential emergency.
    - **Algorithms**: KNN, LOF, IForest.
- **Medical Imaging Anomalies**
    - **Scenario**: Detecting unusual patterns in X-rays, MRIs, or CT scans.
    - **Application**: Identify tumors or abnormalities that deviate from normal tissue structure.
    - **Algorithms**: Autoencoders, CNN-based Anomaly Detection, GANs.
- **Clinical Trial Monitoring**
    - **Scenario**: Identifying abnormal patient responses or data entry errors.
    - **Application**: Flag patient records with unusual trends in vitals or side effects.
    - **Algorithms**: LOF, IForest.

---

4. Manufacturing & Industrial Applications

- **Manufacturing Quality Control**
    - **Scenario**: Detecting defective products in a production line based on sensor data.
    - **Application**: Outlier detection can be used to identify defective products or out-of-tolerance measurements during manufacturing, such as abnormal temperature, pressure, or speed readings in machinery.
    - **Algorithms**: KNN, LOF, IForest.
- **Predictive Maintenance**
    - **Scenario**: Predicting machine breakdowns before they happen.
    - **Application**: Detect irregularities in vibration, pressure, or noise patterns.
    - **Algorithms**: IForest, RNNs with Anomaly Detection layers.
- **Pipeline Monitoring**
    - **Scenario**: Detecting leaks or tampering in oil, gas, or water pipelines.
    - **Application**: Outlier detection on flow rate or pressure.
    - **Algorithms**: Statistical Anomaly Detection, Time-Series Anomaly Detection.

---

5. Energy & Environment

- **Energy Consumption Monitoring**

    - **Scenario**: Detecting unusual energy consumption patterns (e.g., in industrial plants, offices, or homes).
    - **Application**: Detecting large spikes or drops in energy consumption that could indicate a malfunction, equipment failure, or unauthorized usage.
    - **Algorithms**: KNN, LOF, IForest.
- **Environmental Monitoring**
    - **Scenario**: Detecting abnormal pollution levels or climate readings.
    - **Application**: Monitor air/water quality sensors for extreme or unexpected values.
    - **Algorithms**: IForest, Autoencoders.
- **Smart City Infrastructure Monitoring**
    - **Scenario**: Detecting irregularities in urban systems (e.g., traffic lights, street lighting).
    - **Application**: Find faults or unexpected patterns in public service systems.
    - **Algorithms**: LOF, Time-series anomaly detection.

---

6. Security & Cybersecurity

- **Network Intrusion Detection**
    - **Scenario**: Detecting unusual patterns in network traffic, such as DDoS attacks or unauthorized access attempts.
    - **Application**: Anomalous patterns such as sudden spikes in traffic, strange IP addresses, or unusual data packets can be flagged as potential threats.
    - **Algorithms**: KNN, IForest (Isolation Forest), LOF.
- **User Behavior Analytics (UBA)**
    - **Scenario**: Identifying unusual user behavior that could indicate compromised accounts.
    - **Application**: Login at odd hours, access from multiple locations in a short time, or accessing sensitive data not typical for the user.
    - **Algorithms**: KNN, IForest, LOF, Autoencoders.
- **Email Spam or Phishing Detection**
    - **Scenario**: Identifying malicious or phishing emails.
    - **Application**: Detect outlier email patterns based on sender, content, and frequency.
    - **Algorithms**: One-Class SVM, Isolation Forest, NLP + Anomaly Detection.

---

7. Miscellaneous

- **E-commerce: Product Review Anomaly Detection**
    - **Scenario**: Detecting fake or fraudulent reviews for products on an e-commerce platform.
    - **Application**: Outlier detection can help identify suspicious patterns in product reviews, such as unusually positive or negative reviews from a small group of users, which could indicate manipulation.
    - **Algorithms**: KNN, LOF, IForest.
- **Supply Chain Anomaly Detection**

    - **Scenario**: Identifying abnormal supply chain behavior (e.g., delivery delays, stockouts).
    - **Application**: Outlier detection can help track unusual shipping delays or unexpected supply shortages that could disrupt the supply chain.
    - **Algorithms**: KNN, LOF, IForest.
- **Social Media Sentiment Analysis**
    - **Scenario**: Detecting anomalies in social media content (e.g., sudden spikes in mentions of a brand, unusual sentiment).
    - **Application**: Identify outlier posts or trends that represent spikes in sentiment, such as sudden interest in a brand, product, or public figure, which could indicate an emerging issue or opportunity.
    - **Algorithms**: KNN, LOF, IForest.
- **Database Integrity Monitoring**
    - **Scenario**: Detecting outliers in structured data entries (e.g., missing or duplicate entries).
    - **Application**: Spot inconsistent or corrupted records in critical databases.
    - **Algorithms**: DBSCAN, LOF.
- **Sensor Calibration Errors**
    - **Scenario**: Identifying sensors giving faulty data.
    - **Application**: Detect values that deviate consistently from similar sensors.
    - **Algorithms**: Clustering + Anomaly Detection.
- **Model Drift Detection**
    - **Scenario**: Identifying changes in model input/output behavior over time.
    - **Application**: Flag input data distributions or predictions that deviate from training data.
    - **Algorithms**: Distribution monitoring + Anomaly Detection.
- **Training Data Anomalies**
    - **Scenario**: Catching mislabeled or noisy data points.
    - **Application**: Improve dataset quality by removing statistical or semantic outliers.
    - **Algorithms**: ABOD, Isolation Forest, PCA-based detection.

Summary of Algorithms

- **KNN (K-Nearest Neighbors)**
- **LOF (Local Outlier Factor)**
- **IForest (Isolation Forest)**
- **ABOD (Angle-Based Outlier Detection)**
- **Autoencoders**
- **One-Class SVM**
- **DBSCAN**
- **RNNs (Recurrent Neural Networks)**

This classification should make it easier to identify and review anomaly detection scenarios based on the domain or industry. Let me know if you need further adjustments!

How to Choose the Best Scenario:

- **For detecting unusual behavior patterns in transaction data or network traffic**, **KNN** and **Isolation Forest (IForest)** work well.
- **For sensor or health data analysis**, where data can be continuous and highly dimensional, **LOF (Local Outlier Factor)** is useful.
- **For detecting anomalies in sparse or high-dimensional datasets**, **ABOD (Angle-Based Outlier Detection)** is often more effective.