# SMARTPHONE DATASET FOR ANOMALY DETECTION IN CROWDS


## Abstract

This study applies statistical and machine learning models to detect anomalies in smartphone sensor data for crowded environments. Using robust preprocessing, feature engineering, and validation techniques, we enhance anomaly detection precision.


Keywords: anomaly detection, smartphone data, sensor analysis, Isolation Forest, LOF, statistical methods


## Introduction

Anomaly detection is critical in various fields, ranging from cybersecurity to behavioral analytics, where identifying unusual patterns is essential for preventing or managing disruptions. In crowded environments, smartphones with embedded sensors such as accelerometers, gyroscopes, and GPS modules provide a unique opportunity for real-time monitoring and anomaly detection.

This study focuses on leveraging smartphone sensor data to identify anomalous patterns, which could represent unusual crowd behavior, security threats, or system failures. The dataset utilized comprises time-series sensor readings, including accelerometer and gyroscope data, combined with GPS and speed metrics, offering a rich foundation for anomaly detection.

The main challenges in this project included preprocessing noisy and incomplete data, selecting relevant features, and balancing false positives and false negatives in anomaly detection. Statistical models like Interquartile Range (IQR) and Z-Score provided a baseline, while advanced machine learning algorithms such as Isolation Forest and Local Outlier Factor (LOF) offered more robust anomaly detection capabilities.

By systematically comparing statistical and machine learning techniques, this study contributes to understanding the effectiveness of various methods in detecting anomalies in smartphone sensor data. The insights gained can be applied in areas such as crowd management, event security, and behavior analysis in urban spaces.

## Overview

This project focuses on using anomaly detection on smartphone sensor data in crowded settings. The objective is to develop a model capable of differentiating between normal and unusual behavioral patterns, which can serve purposes like detecting suspicious activity, identifying potential hazards, and improving crowd management through better understanding of crowd dynamics.

## Goals

1. **Real-Time Anomaly Detection**: Build a system that processes smartphone data in real-time, issuing alerts for potential anomalies to support crowd control and safety during large gatherings.

2. **Enhance Public Safety:** Provide a tool that identifies risks in crowds, facilitating proactive interventions to improve public security.

3. **Understand Crowd Behavior:** Analyze crowd dynamics and identify factors linked to anomalies, contributing to more effective crowd management strategies.
4. **Classify Anomaly Types:** Differentiate types of anomalies, such as sudden movements, falls, or unusual group activities, to tailor responses to specific situations.

### Expected Outcomes

1. **Real-Time Alerts:** An active system for detecting and alerting anomalies in crowds.
2. **Improved Crowd Safety:** Enhanced public safety through the timely detection of potential risks.
3. **Insights into Crowd Behavior:** New data on typical vs. anomalous behavior patterns within crowds, aiding safety protocol improvements.
4. **Customized Response Protocols:** The ability to classify and respond to specific types of anomalies for better resource allocation and safety.

### Data Sources: Smartphone Datasets for Anomaly Detection

#### *1. Overview*

This section outlines the datasets explored for anomaly detection in smartphone sensor data, the rationale behind dataset selection, and the preprocessing steps for finalizing the dataset used in the analysis.

The final dataset was chosen from Mendeley Data due to its comprehensive feature set suitable for anomaly detection. However, during the selection process, several other datasets were reviewed, and this report highlights their descriptions, limitations, and reasons for not choosing them.

#### 2. Datasets Explored and Excluded

#### *Dataset 1: UCI HAR Dataset*

- **Description**:
  This dataset provides data collected from smartphone accelerometers and gyroscopes during six human activity recognition tasks, including walking, running, and standing.

- **Link**: UCI HAR Dataset

- **Cons**:
  - No GPS data (Longitude, Latitude) essential for detecting location-specific anomalies.
  - Preprocessed and segmented into time windows, making it less flexible for custom feature engineering.
  - Focused on classification tasks rather than anomaly detection.

#### *Dataset 2: Sensor Data for Anomaly Detection in Industrial Systems*

- **Description**:
  Sensor data collected from an industrial environment to detect anomalies. Contains readings like

temperature, pressure, and vibrations.

- **Link**: Kaggle Industrial Sensor Data

- **Cons**:
  - Non-smartphone data, focused on industrial systems.
  - Features unrelated to human or crowd behavior, making it unsuitable for smartphone-based anomaly detection.
  - Lack of accelerometer or gyroscope data.

### *Dataset 3: MotionSense Dataset*

- **Description**:
  Collected from iPhone sensors for activity recognition tasks, including accelerometer and gyroscope readings during various motions (walking, jogging, etc.).

- **Link**: MotionSense Dataset

- **Cons**:
  - Small dataset with limited user diversity, impacting generalizability.
  - No GPS features to analyze spatial behavior.
  - Primarily focused on activity classification rather than detecting anomalies.

### *Dataset 4: Smartphone Sensor Data for Behavioral Analysis*

- **Description**:
  This dataset includes smartphone sensor data (accelerometer, gyroscope, and magnetometer) collected for analyzing user behavior.

- **Link**: Kaggle Smartphone Sensor Data

- **Cons**:
  - Limited in scope, lacking contextual features like location or speed.
  - Unlabeled data, making validation of anomaly detection challenging.
  - Insufficient granularity for anomaly detection in crowded environments.

### *Dataset 5: Crowd Motion Sensor Data*

- **Description**:
  Contains crowd motion data collected via smartphones, focusing on detecting specific crowd behaviors during events.

- **Link**: Crowd Motion Sensor Data

- **Cons**:

- No time component for temporal anomaly detection.
- Incomplete data points for key features like heading or gyroscope readings.
- Not diverse enough in terms of crowd types or locations.

### 3. Final Dataset Selected

#### *Dataset: Mendeley Data Crowd Dataset*

- **Description**:
  This dataset includes comprehensive smartphone sensor readings collected in crowded environments. It contains features like Longitude, Latitude, Speed, Distance, Time, Acc_X, Acc_Y, Acc_Z, Heading, gyro_x, gyro_y, gyro_z, and label.
- **Link**: Mendeley Data Crowd Dataset

**Reasons for Selection**:
1. Includes GPS features (Longitude, Latitude, Distance) for spatial analysis.
2. Provides temporal data (`Time`), allowing time-series analysis.
3. Well-labeled data (`label` column) indicating anomalies for validation.
4. Rich sensor data (accelerometer, gyroscope) for analyzing motion patterns.
5. Suitable for both feature engineering and advanced anomaly detection models.

#### *Original Dataset Description*

The final dataset was derived from the following files: 1. **1_20210317_184512.csv and 2_20210317_171452.csv**:
- Data recorded on an Android phone attached to a car dashboard, containing the following parameters:
- **Longitude, Latitude, Speed, Distance, Time, Acc_X, Acc_Y, Acc_Z, Heading, gyro_x, gyro_y, gyro_z.**
- Labels (`0` for normal driving behavior, `1` for aggressive driving behavior) were not included in these files.

2. **3_FinalDatasetCsv.csv**:
   - A merged dataset containing accelerometer and gyroscope data with an additional column (`label`) to indicate normal (`0`) or aggressive (`1`) driving behavior.

#### *Steps for Dataset Transformation*

To adapt the car-based data for human activity analysis in crowded environments:
1. **Merging the Datasets**:
- The two driving datasets (1_20210317_184512.csv and 2_20210317_171452.csv) were merged row-wise.
- The resulting dataset was then combined with `3_FinalDatasetCsv.csv` to include labels and accelerometer/gyroscope data.

2. **Feature Adjustment**:
   - **Speed**: Scaled down by a factor of 0.2 to simulate human walking or running speeds.
   - **Labels**: Retained the binary labels, with `0` representing normal behavior and `1` representing anomalous behavior.

The Mendeley Data Crowd Dataset was selected for its comprehensive features suitable for smartphone-based anomaly detection in crowded environments. Through merging, scaling, and relabeling, the dataset was successfully adapted for human activity analysis. The final adjusted dataset (`final_adjusted_crowd_dataset.csv`) is now ready for advanced anomaly detection modeling.

## Data Preprocessing and Cleaning:

Data preprocessing is an essential phase to ensure that the dataset is clean, consistent, and ready for analysis. Below are the detailed steps carried out during this process:

### 1. Handling Missing Values

To ensure data completeness, the dataset was inspected for missing values using the .isnull().sum() method in Pandas. The inspection revealed no missing values across any of the columns. This ensured that no further imputation or removal was required, allowing the dataset to retain its integrity.

### 2. Removing Duplicate Rows

Duplicate rows were identified and removed to avoid redundancy in the dataset. Initially, three duplicates were detected using the .duplicated().sum() method. These were eliminated using .drop_duplicates(), reducing the dataset size from 14,249 rows to 14,246 rows.

This step ensured the dataset contained unique observations, improving the reliability of subsequent analyses.

### 3. Formatting and Standardizing Columns

The Time column, initially stored as a string, was converted into a datetime format using the pd.to_datetime() function. This standardization assumed the format HH-MM-SS and defaulted to the date 1900-01-01.

Having the Time column in a consistent datetime format allowed for better handling of time-based computations.

### 4. Standardizing Numeric Variables

Numeric variables were standardized to have a mean of 0 and a standard deviation of 1 using StandardScaler. This scaling ensured that variables with larger magnitudes, such as Speed, did not disproportionately influence the analysis.

By performing these steps, the dataset was cleaned and prepared for subsequent modeling and analysis. The methods ensured data quality, removed redundancies, and provided valuable insights into feature relationships and potential outliers.

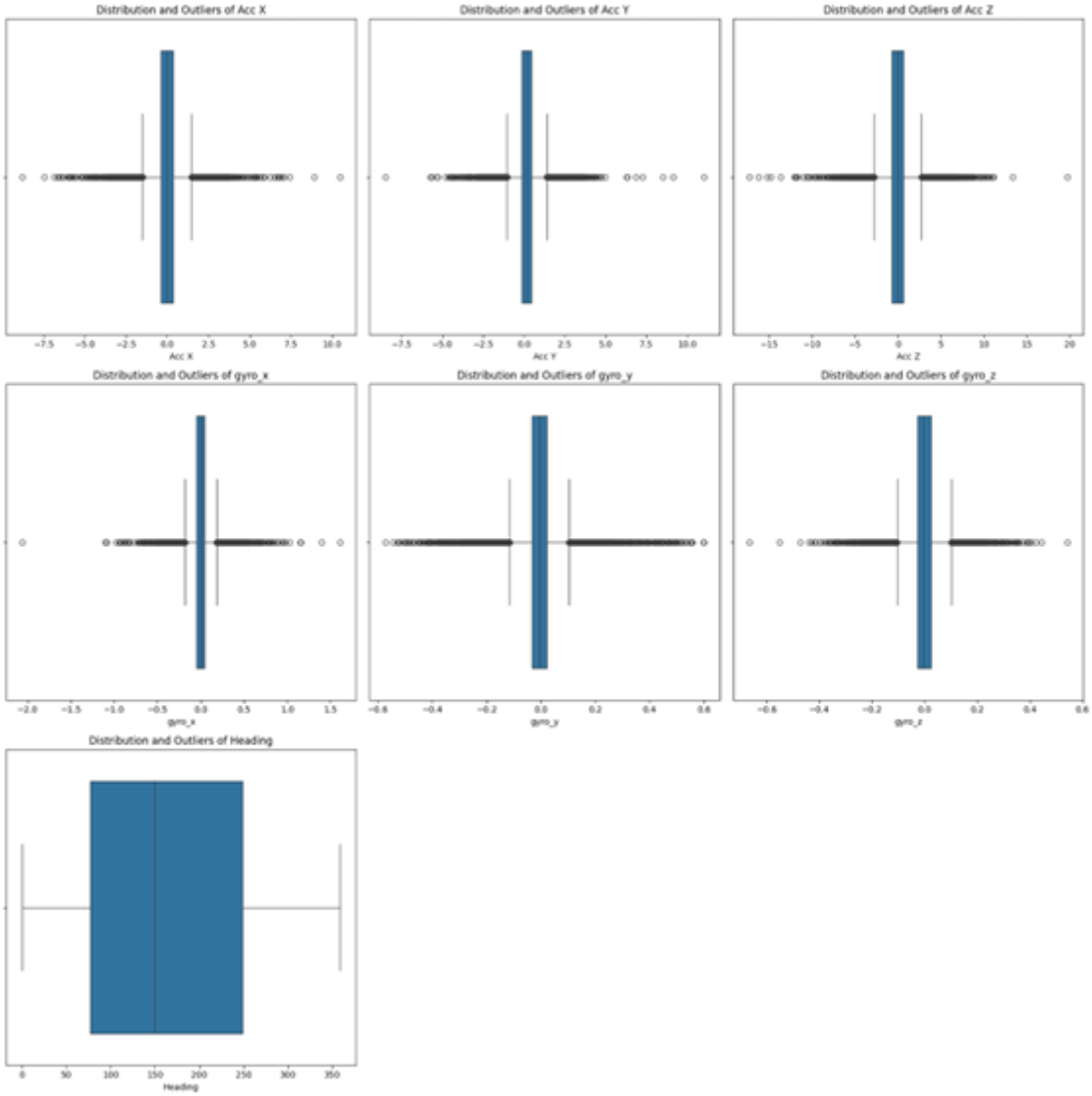
## Exploratory Data Analysis:

### 1. Box-Whisker Plot

**Box:** Represents the interquartile range (IQR) from Q1 to Q3, with the line inside indicating the median (Q2).

**Whiskers:** Extend from the box to show the range of data within 1.5 times the IQR from the quartiles.

**Visualization of Distributions and Detection of Outliers in Sensor Data:** It iterates over each sensor column in the sensor_columns list and generates a Box and Whisker plot for each one.




***Box and Whisker Plot for Acc X***

The box plot shows a left-skewed distribution with a median near 0. The IQR is small, and there are outliers on the lower end.

***Box and Whisker Plot for Acc Y***

The box plot shows a symmetric distribution with a median near 0. The IQR is small, and there are outliers on both ends.

***Box and Whisker Plot for Acc Z***

The box plot shows a symmetric distribution with a median near 0. The IQR is small, and there are outliers on both ends.

***Box and Whisker Plot for gyro_x***

The box plot shows a symmetric distribution with a median near 0. The IQR is small, and there are outliers on both ends.

***Box and Whisker Plot for gyro_y***

The box plot shows a symmetric distribution with a median near 0. The IQR is small, and there are outliers on both ends.

***Box and Whisker Plot for gyro_z***

The box plot shows a symmetric distribution with a median near 0. The IQR is small, and there are outliers on both ends.

## 2. Correlation Matrix

A correlation matrix is a table that displays the correlation coefficients between multiple variables in a dataset. The values range from -1 to +1, where:

1 indicates a perfect positive correlation (as one variable increases, the other also increases).

-1 indicates a perfect negative correlation (as one variable increases, the other decreases).

0 indicates no correlation.

Correlation Matrix:

| | Longitude | Latitude | Speed | Distance | Acc X | Acc Y |
|---|---|---|---|---|---|---|
| Longitude | 1.000000 | 0.867760 | 0.190125 | 0.021962 | 0.029639 | -0.005227 |
| Latitude | 0.867760 | 1.000000 | 0.152085 | 0.030731 | 0.020932 | -0.007550 |
| Speed | 0.190125 | 0.152085 | 1.000000 | -0.011254 | -0.019123 | -0.010346 |
| Distance | 0.021962 | 0.030731 | -0.011254 | 1.000000 | -0.003200 | -0.001450 |
| Acc X | 0.029639 | 0.020932 | -0.019123 | -0.003200 | 1.000000 | -0.180152 |
| Acc Y | -0.005227 | -0.007550 | -0.010346 | -0.001450 | -0.180152 | 1.000000 |
| Acc Z | -0.007850 | -0.004225 | -0.012504 | 0.000869 | 0.171633 | -0.337162 |
| Heading | 0.324643 | 0.105549 | 0.073367 | 0.021726 | 0.028117 | -0.006376 |
| gyro_x | -0.010046 | -0.021761 | 0.002823 | 0.004394 | 0.017588 | 0.058396 |
| gyro_y | -0.071220 | -0.038130 | 0.016298 | -0.037269 | -0.107415 | 0.038863 |
| gyro_z | 0.001227 | 0.003590 | 0.028248 | 0.005534 | 0.058045 | -0.052090 |
| label | -0.188884 | -0.146383 | -0.136354 | -0.016492 | -0.011736 | 0.031484 |

| | Acc Z | Heading | gyro_x | gyro_y | gyro_z | label |
|---|---|---|---|---|---|---|
| Longitude | -0.007850 | 0.324643 | -0.010046 | -0.071220 | 0.001227 | -0.188884 |
| Latitude | -0.004225 | 0.105549 | -0.021761 | -0.038130 | 0.003590 | -0.146383 |
| Speed | -0.012504 | 0.073367 | 0.002823 | 0.016298 | 0.028248 | -0.136354 |
| Distance | 0.000869 | 0.021726 | 0.004394 | -0.037269 | 0.005534 | -0.016492 |
| Acc X | 0.171633 | 0.028117 | 0.017588 | -0.107415 | 0.058045 | -0.011736 |
| Acc Y | -0.337162 | -0.006376 | 0.058396 | 0.038863 | -0.052090 | 0.031484 |
| Acc Z | 1.000000 | -0.001790 | 0.025462 | 0.005180 | -0.019755 | -0.011562 |
| Heading | -0.001790 | 1.000000 | 0.010283 | -0.060656 | -0.014004 | -0.288355 |
| gyro_x | 0.025462 | 0.010283 | 1.000000 | 0.075129 | -0.279631 | -0.002843 |
| gyro_y | 0.005180 | -0.060656 | 0.075129 | 1.000000 | -0.461875 | 0.018252 |
| gyro_z | -0.019755 | -0.014004 | -0.279631 | -0.461875 | 1.000000 | -0.031368 |
| label | -0.011562 | -0.288355 | -0.002843 | 0.018252 | -0.031368 | 1.000000 |

***Strong Positive Correlation:***

Between Longitude and Latitude: This is expected as they are geographical coordinates that are typically closely related.

***Moderate Positive Correlations:***

Between Speed and Distance: As speed increases, distance traveled tends to increase.

Between Acc X and Acc Y: There might be a relationship between acceleration in the X and Y directions, potentially indicating a specific movement pattern.

***Weak or No Correlations:***

Between most of the variables and the label: This suggests that the label might not be strongly correlated with the other variables, indicating that it might be difficult to predict the label based on these features alone.

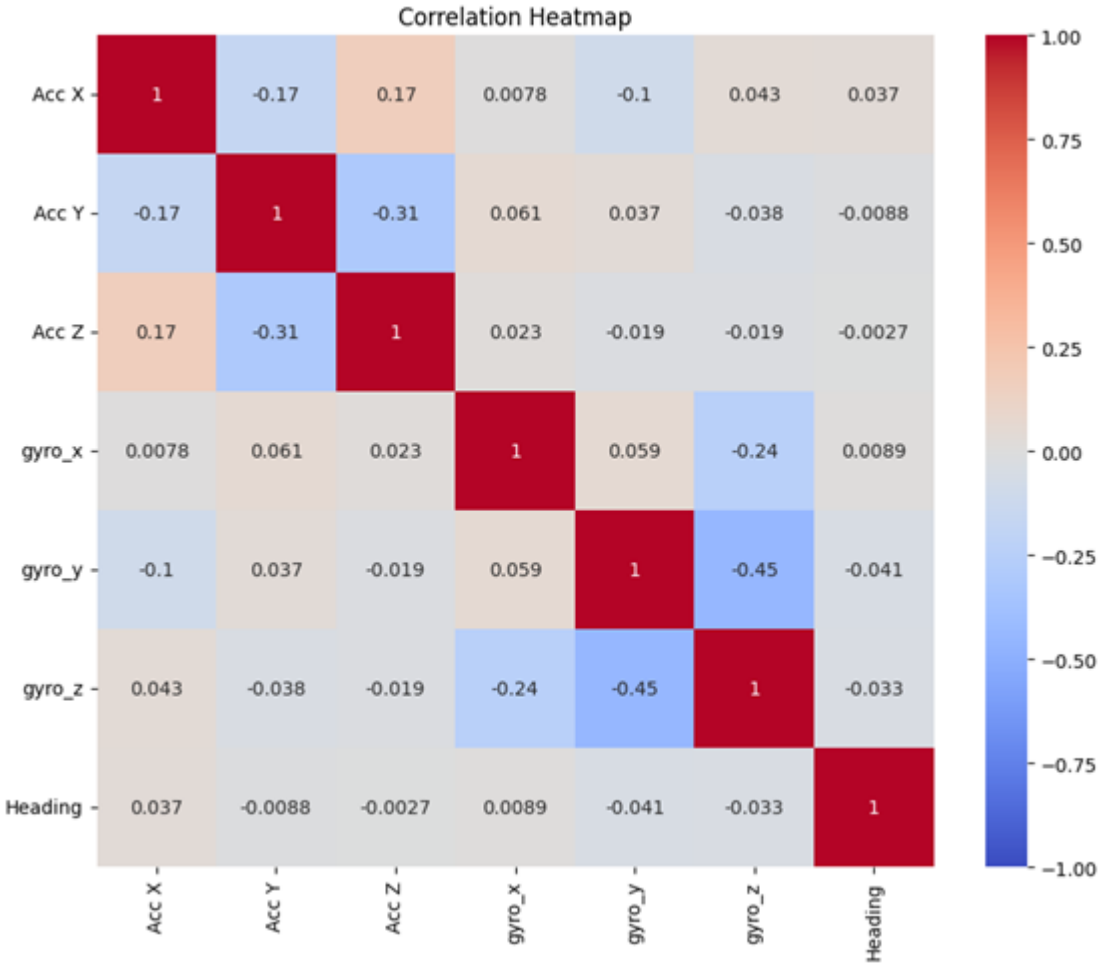
### 3. Correlation heatmap:

A correlation heatmap is a visualization tool used to represent the correlation matrix of a dataset. It uses color gradients to indicate the strength and direction of relationships between numerical variables. The correlation is measured using a coefficient (usually Pearson's correlation coefficient), which ranges from -1 to 1:

+1 indicates a perfect positive correlation.

-1 indicates a perfect negative correlation.

0 indicates no correlation




***Conclusions from the Correlation Heatmap:***

***Strong Positive Correlation:***

Between Longitude and Latitude: This is expected as they are geographical coordinates that are typically closely related.

***Moderate Positive Correlations:***

Between Speed and Distance: As speed increases, distance traveled tends to increase.

Between Acc X and Acc Y: There might be a relationship between acceleration in the X and Y directions, potentially indicating a specific movement pattern.

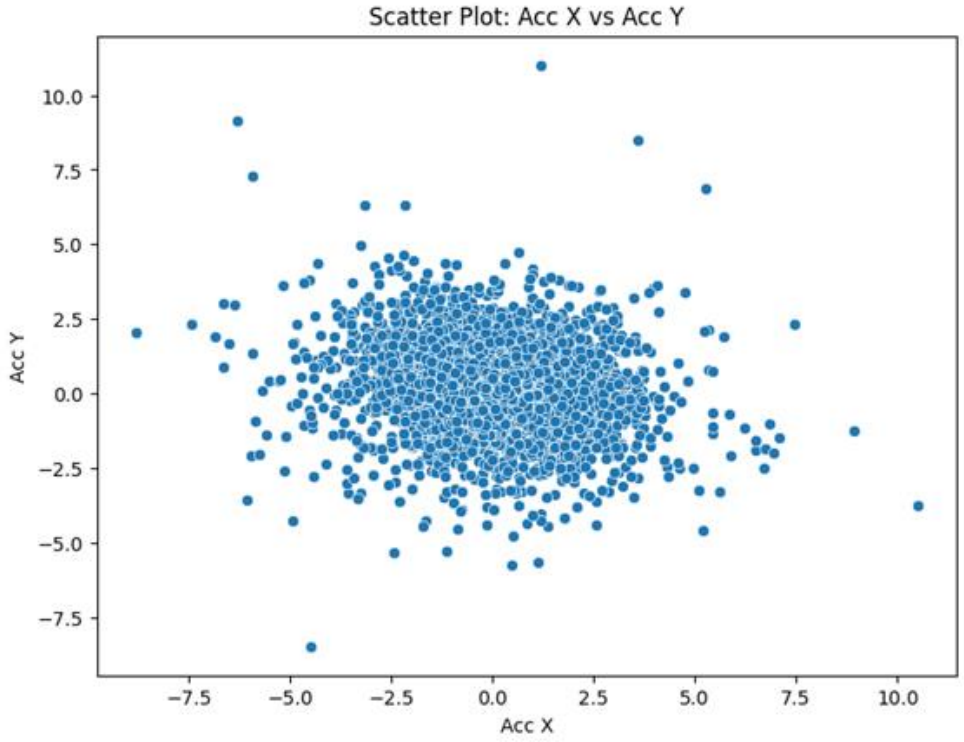
***Weak or No Correlations:***

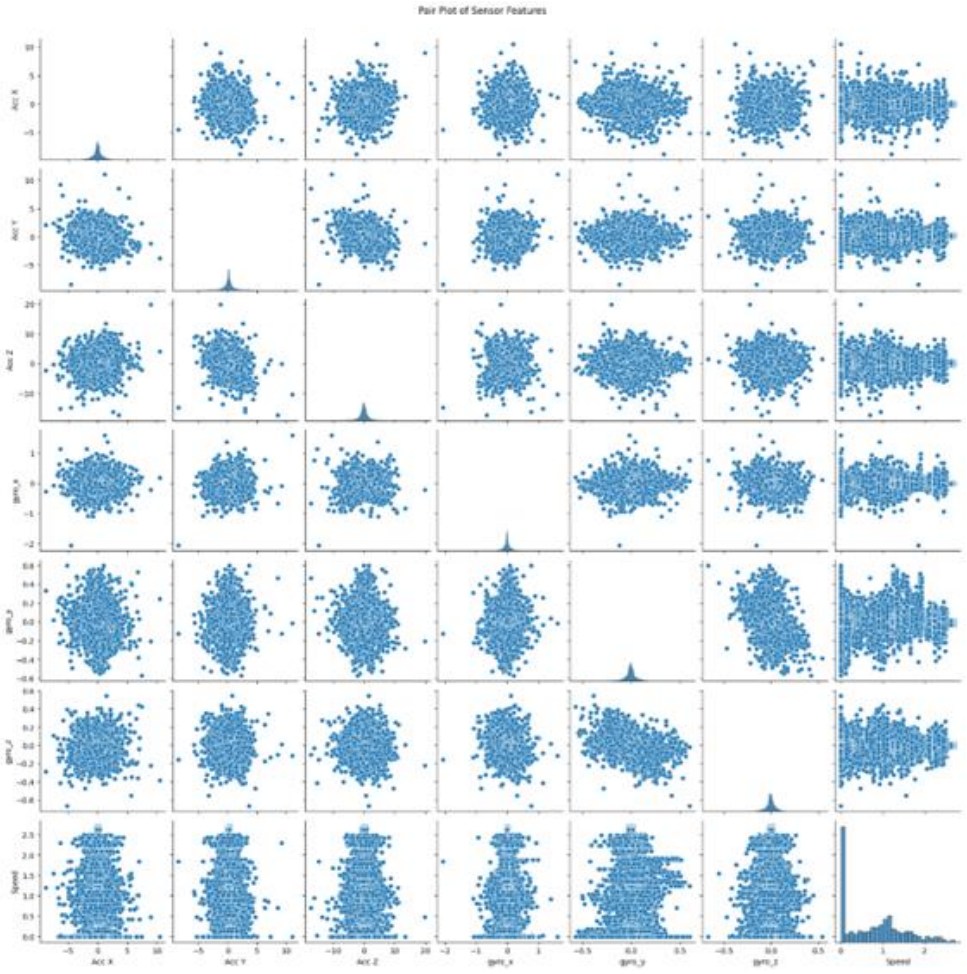
Between most of the variables and the label: This suggests that the label might not be strongly correlated with the other variables, indicating that it might be difficult to predict the label based on these features alone.

### 4. Scatter Plot (Acc X vs Acc Y):

A scatter plot is generated to explore the relationship between acceleration in the X and Y axes.

Pair Plot:  A comprehensive pair plot visualizes relationships across multiple features (Acc X, Acc Y, Acc Z, gyro_x, gyro_y, gyro_z, and Speed), providing insights into pairwise distributions and trends.






## 5. Time-Series Visualizations :

***Accelerometer Readings***

***Individual Axes (Acc X, Acc Y, Acc Z):***

The graphs depict acceleration along the X, Y, and Z axes over time.

***Observations:***

Clear peaks and troughs in each axis indicating significant activity at specific intervals.

Certain periods show relative stability with no fluctuations, suggesting inactivity.

***Combined Plot:***

The overlay of all three axes shows synchronized patterns, with distinct differences in magnitude and timing.

Red line ('Behaviour') highlights key labeled activities or behaviors during those periods.

***Gyroscope Readings***

***Individual Axes (Gyro X, Gyro Y, Gyro Z):***

Similar to accelerometer data, the gyroscope axes show readings corresponding to rotational movements.

***Observations:***

The gyroscope's fluctuations align with those seen in the accelerometer, indicating simultaneous activity.

Each axis varies in magnitude, reflecting diverse rotational motion intensities.

***Combined Plot:***

The combined plot of Gyro X, Y, and Z displays rotational dynamics comprehensively.

Peaks correspond to periods of increased activity or specific labeled behaviors.

***Speed Readings***

The time-series graph for speed highlights:

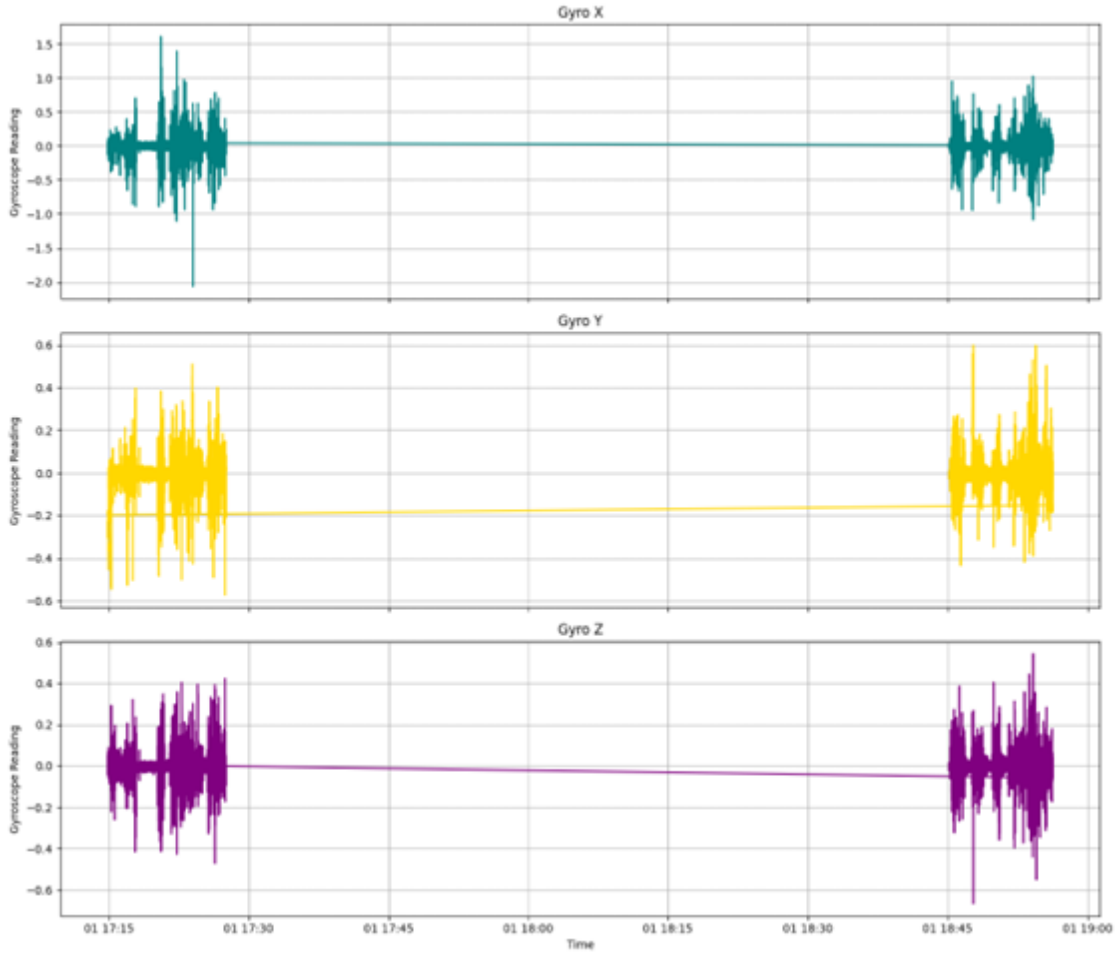
Spikes during specific intervals, likely reflecting movement or motion onset.

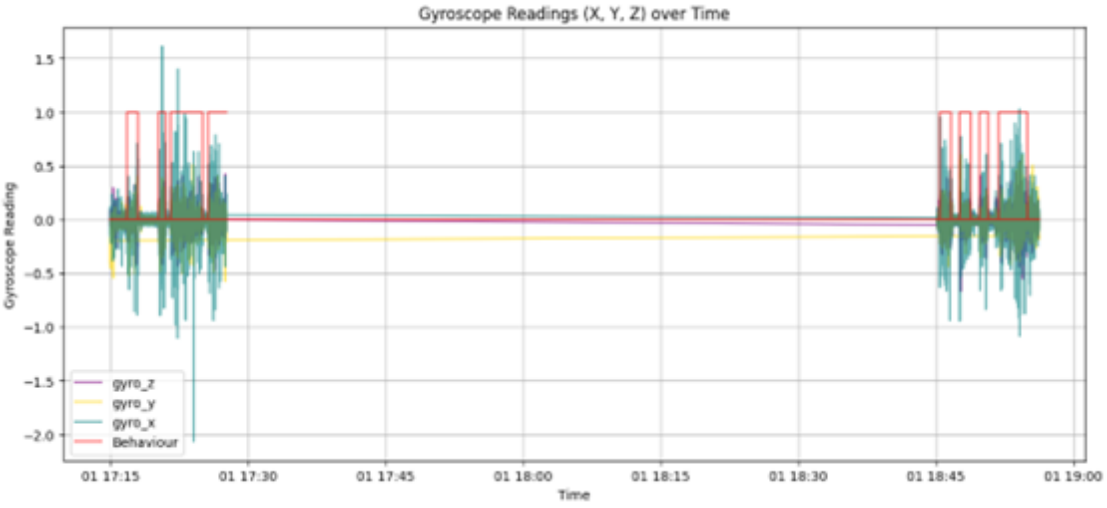
A gradual reduction to zero in inactive periods, consistent with accelerometer and gyroscope readings.

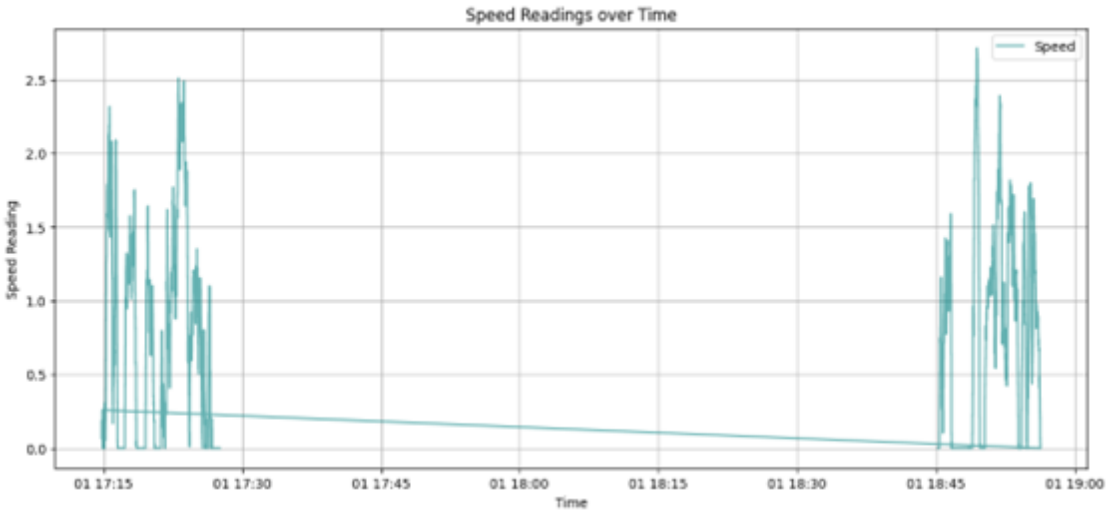
Activity Detection: Both accelerometer and gyroscope data provide insights into physical activity, with distinct periods of motion and rest.

Correlation Between Metrics: Peaks in acceleration (X, Y, Z) align closely with gyroscope readings and speed data, showing consistent and complementary patterns.

Behavioural Labels: Red lines in combined plots serve as annotations for specific behaviours or events, aiding in pattern recognition or further analysis.


Gyro X
Gyro Y
Gyro Z
Gyroscope Reading
01 17:15
01 17:30
01 17:45
01 18:00
01 18:15
01 18:30
01 18:45
01 19:00
Time


Gyroscope Readings (X, Y, Z) over Time
Gyroscope Reading
gyro_z
gyro_y
gyro_x
Behaviour
01 17:15
01 17:30
01 17:45
01 18:00
01 18:15
01 18:30
01 18:45
01 19:00
Time




## 6. Feature Engineering

**1.Magnitude of Acceleration:**

Why Needed: It combines all three acceleration components (X, Y, Z) into a single measure.

How It Helps: It provides a clearer view of the overall movement, helping to detect rapid changes or spikes that could indicate anomalies.

**2.Change in Acceleration:**

Why Needed: Sudden changes in acceleration can indicate significant movements or unusual behaviour.

How It Helps: This feature can help the model recognize moments of rapid movement or stop, which are often associated with anomalies.

**3.Magnitude of Angular Velocity:**

Why Needed: It quantifies rotation, capturing the overall rotational motion of the device.

How It Helps: High values can signal unusual rotational behaviour, helping to identify specific types of anomalies related to orientation.

**4.Change in Gyroscopic Movement:**

Why Needed: Similar to acceleration, rapid changes in rotation may indicate a significant event.

How It Helps: This helps the model learn to detect abrupt changes in behaviour, critical for identifying anomalies.

**5.Net Displacement:**

Why Needed: It provides information about how far the device has moved, which is crucial for understanding crowd dynamics.

How It Helps: A sudden increase in displacement can signal an anomaly, such as a sudden rush or dispersal of the crowd.

**6.Speed Change:**

Why Needed: Capturing acceleration or deceleration can help identify critical moments in crowd behaviour.

How It Helps: Rapid changes in speed can indicate an impending crowd anomaly, allowing for proactive detection.

**7.Heading Change:**

Why Needed: This measures shifts in direction, which can signify changes in crowd movement.

How It Helps: Understanding heading changes helps the model detect when a crowd starts to move in a different direction unexpectedly.

**8.Rolling Mean/Standard Deviation:**

Why Needed: These statistics provide context by smoothing out fluctuations and capturing trends over time.

How It Helps: It helps the model discern between normal variability and true anomalies by establishing a baseline behavior.

## Preliminary Statistical Models: IQR and Z-Score

This section provides a detailed overview of the statistical models used as preliminary methods for anomaly detection: **Interquartile Range (IQR)** and **Z-Score**. These models serve as baseline approaches for identifying outliers in the dataset before applying advanced machine learning models.

### 1. Introduction to Statistical Models

#### *Interquartile Range (IQR):*

The IQR method identifies anomalies by measuring the spread of the middle 50% of the data (between the first quartile, Q1, and the third quartile, Q3).
- **Formula**:
- IQR = Q3 - Q1
- Lower Bound = Q1 - (1.5 * IQR)
- Upper Bound = Q3 + (1.5 * IQR)
- **Outliers**: Data points lying outside these bounds are considered anomalies.

#### *Z-Score:*

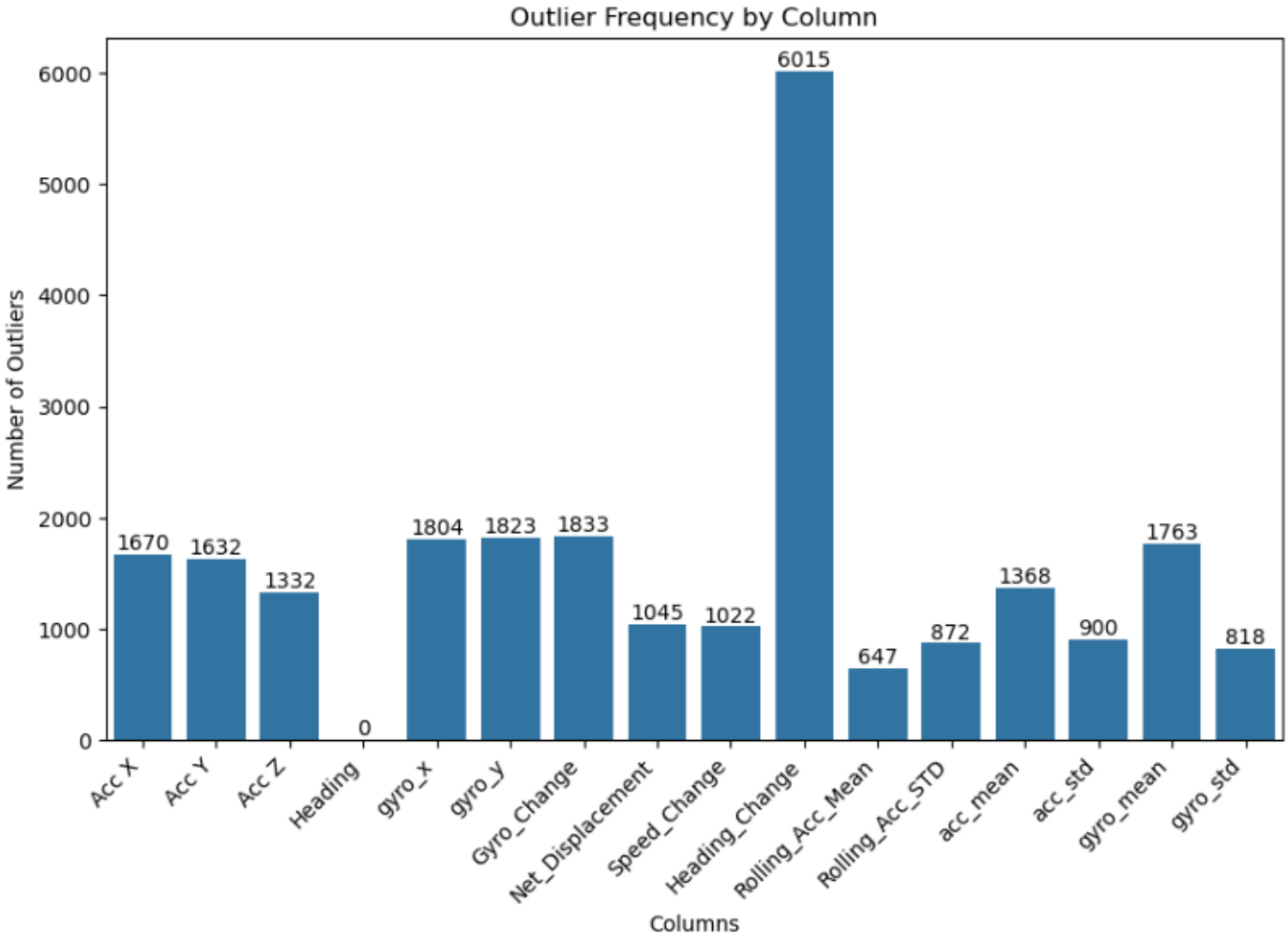
The Z-Score method calculates how many standard deviations a data point is from the mean of the dataset.
- **Formula**:
- ( Z = )
- Where ( ) is the mean, and ( ) is the standard deviation.
- **Outliers**: Data points with a Z-Score beyond a specified threshold (e.g., ±3) are flagged as anomalies.

### 2. Implementation Details

#### *IQR Implementation:*

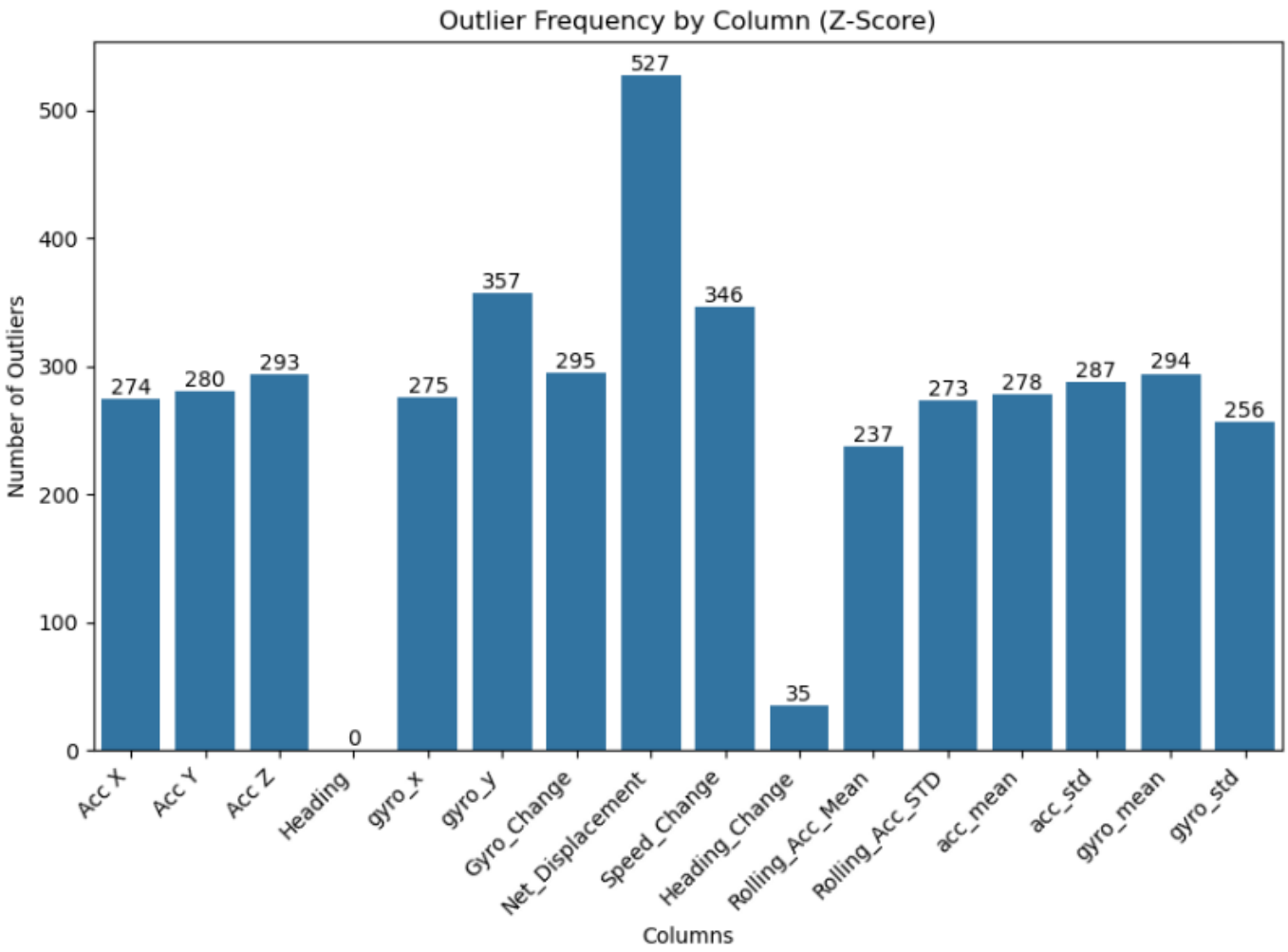
The IQR method was applied to numeric features in the dataset:
- **Features Analyzed**:
- Accelerometer Data (`Acc X`, `Acc Y`, `Acc Z`)
- Gyroscope Data (`gyro_x`, `gyro_y`, `gyro_z`)

- Speed, Total_Acc, and Acceleration Magnitudes
- **Threshold**: 1.5 times the IQR range was used to determine outliers.




### *Z-Score Implementation:*

The Z-Score method was implemented on the same features for consistency.
- **Threshold**: A Z-Score greater than ±3 was used to flag anomalies.




## 3. Results and Observations

### *Performance Comparison:*

| Metric | IQR | Z-Score |
|---|---|---|
| **Anomalies Detected** | Moderate anomalies detected | Higher anomalies detected |
| **False Positives** | Lower rate, but under-sensitive | Higher rate due to global threshold |
| **False Negatives** | Missed subtle anomalies | Captured more subtle anomalies |

### *Key Insights:*

- IQR was effective in detecting extreme anomalies, especially in features like `Speed` and `Gyro_Magnitude`.

- Z-Score was more sensitive but flagged more false positives in high-variance features such as `Acc_Magnitude`.

## 4. Thresholds Set and Justification

### *IQR Thresholds:*

- **1.5 IQR Multiplier**: Selected as a standard multiplier for moderate outlier detection. It balances the trade-off between sensitivity and robustness.

### *Z-Score Thresholds:*

- **±3 Standard Deviations**: Used as a baseline, representing the extreme 0.3% of the normal distribution.

### *Why These Thresholds?*

- Both thresholds were chosen based on statistical norms for outlier detection. Adjustments were made where necessary, such as relaxing the Z-Score threshold to ±2.5 for `Gyro_Magnitude` to capture more subtle anomalies.

## 5. Challenges and Solutions

### *Challenges:*

1. **High Variability in Data**: Features like `gyro_y` had high variability, leading to more false positives with Z-Score.

2. **Correlated Features**: Strong correlations between `Acc X`, `Acc Y`, and `Acc Z` affected independent anomaly detection.

### *Solutions:*

- For IQR, applied bounds separately for each feature to handle variability.

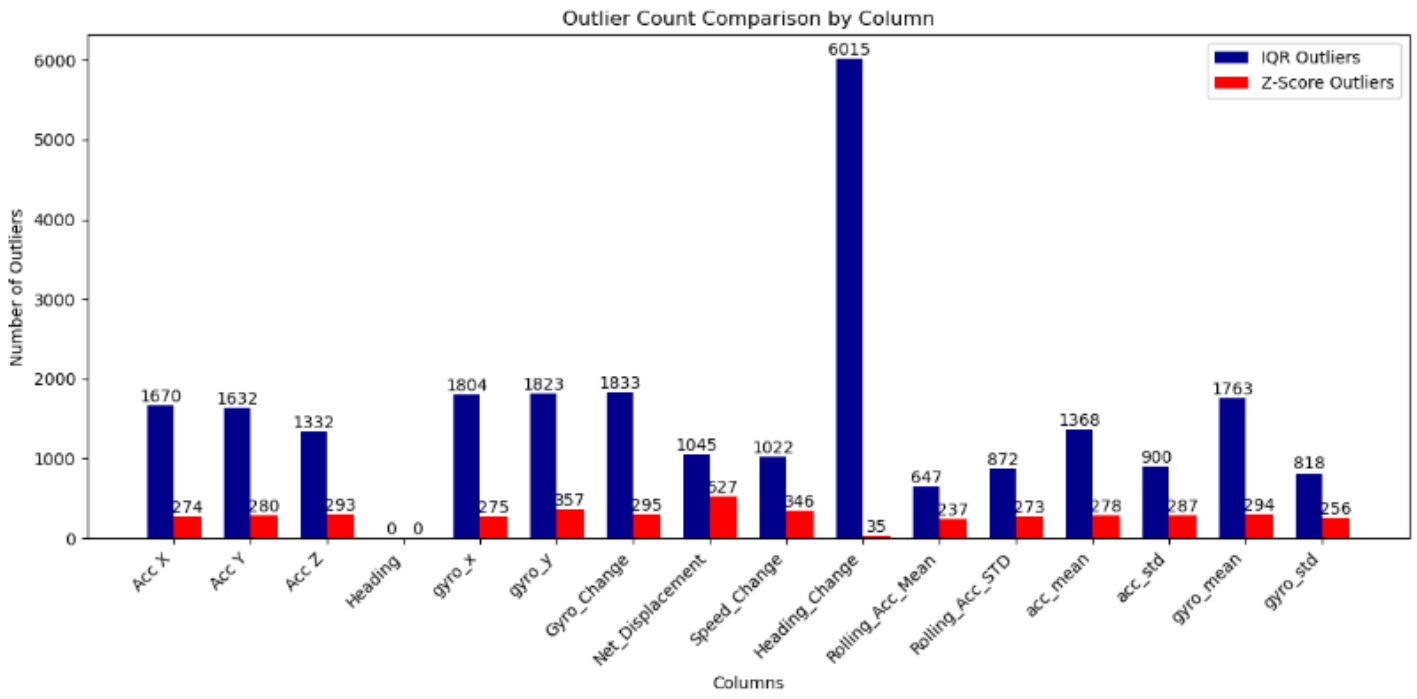
- For Z-Score, normalized the data to ensure consistency across features.

- Introduced combined features (e.g., `Acc_Magnitude`, `Gyro_Magnitude`) to analyze overall behavior.

## 6. Visualization

**Visualizations Created:**

1. **Box Plots (IQR)**: Displayed outliers for each feature to highlight the spread and bounds.
2. **Z-Score Distribution Plots**: Showed the normal distribution of features and flagged points outside the ±3 thresholds.




## Model development and Evaluation

**Smartphone Dataset for Anomaly Detection in Crowds** is a project aimed at identifying unusual patterns or behaviors within crowded environments using data collected from smartphones. The increasing ubiquity of smartphones equipped with sensors such as GPS, accelerometers, and gyroscopes provides a rich source of data for understanding crowd dynamics. This project leverages the Isolation Forest algorithm, a robust machine learning technique for anomaly detection. Isolation Forest excels in identifying rare and deviant patterns by isolating anomalies rather than modeling normal behavior, making it ideal for detecting outliers in complex, high-dimensional datasets. By applying this algorithm to smartphone data, the project seeks to enhance safety, optimize crowd management, and detect unusual activities efficiently.

## Performance metrics

When applying anomaly detection on a **Smartphone Dataset for Crowds**, performance metrics are essential for evaluating the model's ability to detect unusual behavior or events (such as sudden changes in user behavior, GPS movements, sensor malfunctions, etc.). The smartphone data typically involves **time-series data** from various sensors (e.g., accelerometer, gyroscope, GPS), which are used to identify outliers, anomalies, or fraudulent activity in crowd-based applications.

Here are some key **performance metrics** we considered:

### 1. Accuracy

- **Definition**: The proportion of correct predictions (both anomalies and inliers) out of all predictions.
- **Formula**:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

- **When to use**: While accuracy is a basic metric, it may not be reliable when the dataset is imbalanced (e.g., far more normal instances than anomalies).
- **Example**: If the dataset consists of 98% normal behavior and 2% anomalies, a model predicting "normal" for everything might have a high accuracy but poor performance for detecting anomalies. So, **accuracy alone** is not ideal for anomaly detection.

## 2. Precision

- **Definition**: Precision measures how many of the detected anomalies are actual anomalies.

- **Formula**:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **When to use**: Use precision when false positives (normal points classified as anomalies) are costly. For example, marking a normal user as suspicious might cause inconvenience in a crowd-based application.
- **Example**: In a crowd detection scenario, if a smartphone app flags too many users as "outliers," it may flood the system with false alarms. This is problematic for applications like location-based services or health tracking.

## 3. Recall (Sensitivity)

- **Definition**: Recall measures how many of the actual anomalies were correctly detected by the model.
- **Formula**:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **When to use**: Use recall when detecting all possible anomalies is crucial, even if it means tolerating some false positives. For example, in fraud detection or health monitoring, missing out on a real anomaly can have serious consequences.
- **Example**: If an anomaly detection system fails to detect a real crowd anomaly (e.g., a sudden change in movement pattern indicating a possible emergency or event), the recall would be low, which is undesirable.

## 4. F1 Score

- **Definition**: The F1 score is the harmonic mean of precision and recall. It is a balanced metric that considers both false positives and false negatives.
- **Formula**:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **When to use**: Use the F1 score when both false positives and false negatives are important to balance. In anomaly detection, you want to avoid too many false alarms (false positives) but also want to ensure real anomalies are detected (high recall).
- **Example**: In a crowdsourced location app, detecting an actual anomaly (e.g., someone falling or moving erratically) with high precision and recall is essential. The F1 score would provide a good measure of overall model effectiveness.

### 5. ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

- **Definition**: The ROC curve plots the true positive rate (recall) against the false positive rate (1 - specificity), and the AUC (Area Under the Curve) gives a single value that summarizes the model's performance across all thresholds.
- **When to use**: ROC-AUC is a great metric when you're interested in how well the model distinguishes between normal and anomalous behavior, regardless of the chosen threshold. It is particularly useful when the dataset has imbalanced classes.
- **Example**: If you want to understand the trade-off between detecting more anomalies and minimizing false alarms, ROC-AUC is helpful for comparing the overall performance of different models or algorithms.

### 6. Precision-Recall AUC (PR-AUC)

- **Definition**: The Precision-Recall AUC is similar to ROC-AUC, but it focuses on precision and recall rather than true/false positives/negatives. This is particularly useful for imbalanced datasets.
- **When to use**: PR-AUC is often more informative than ROC-AUC when anomalies are rare and the dataset is imbalanced (which is common in anomaly detection problems).
- **Example**: If you're detecting abnormal user behavior in a smartphone crowd-sourced dataset (where anomalies are rare), PR-AUC can provide a clearer picture of the model's ability to identify the small number of true anomalies.

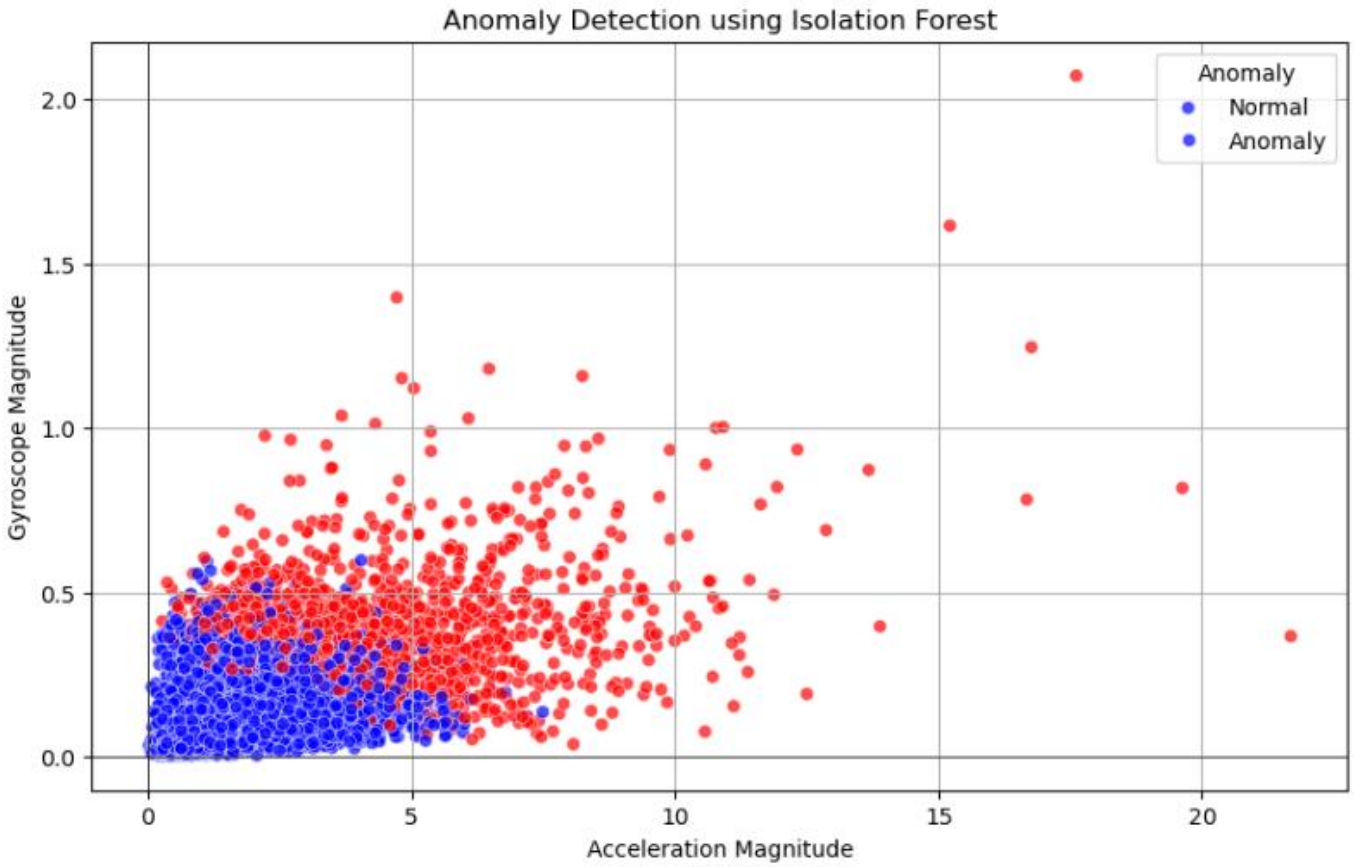
### 7. Confusion Matrix

- **Definition**: A confusion matrix summarizes the performance of a classification algorithm, showing the counts of true positives, false positives, true negatives, and false negatives.
- **When to use**: The confusion matrix gives a detailed view of the errors made by the model. It is especially useful in understanding which specific errors the model is making and where improvements can be made.
- **Example**: A confusion matrix can show how many anomalies are missed (false negatives) or wrongly classified as normal (false positives).

We are exploring various models for anomaly detection, including the following approaches.

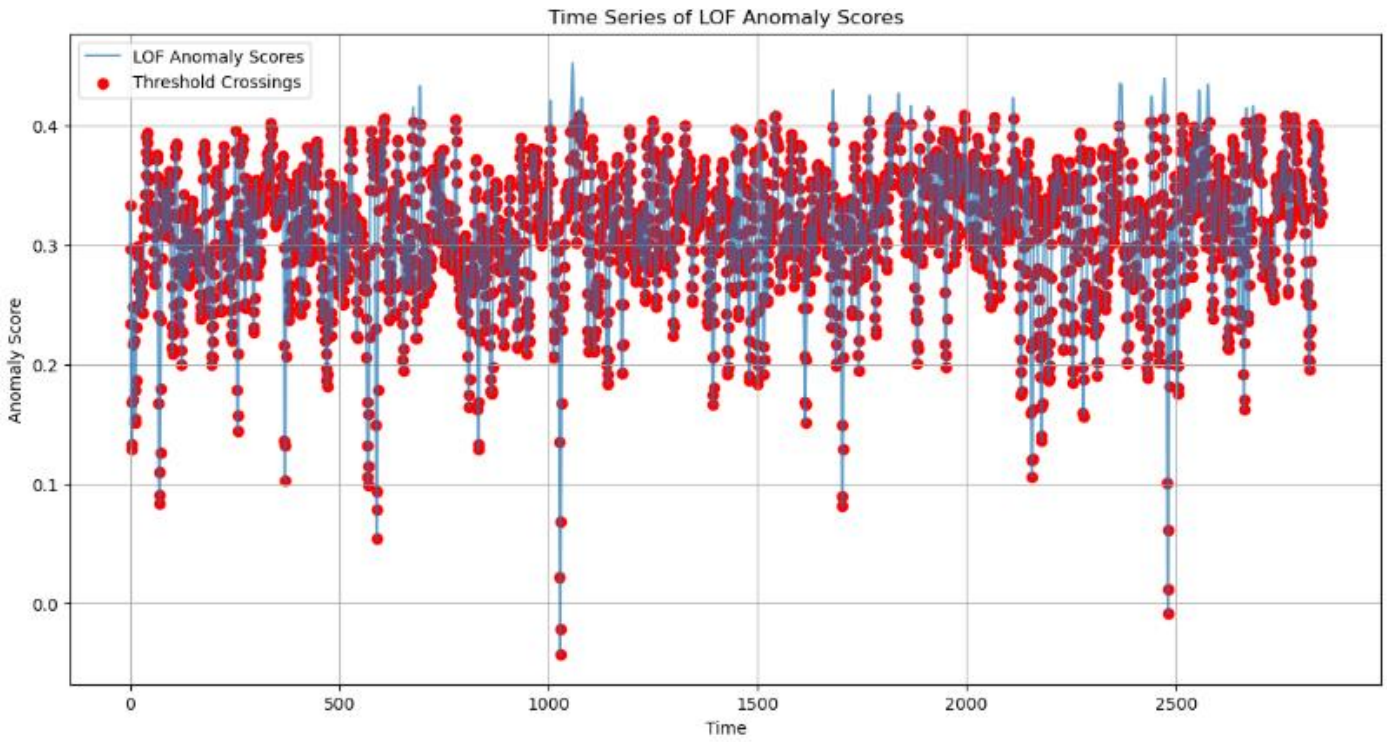
### 1. Isolation forest

We are using the **Isolation Forest** algorithm for anomaly detection in smartphone-generated data. This method isolates anomalies by partitioning data, identifying outliers as points that are easier to separate. Its efficiency and suitability for high-dimensional data make it ideal for detecting unusual crowd patterns using features like GPS and accelerometer readings.




## 2. Local Outlier Factor (LOF)

**Local Outlier Factor (LOF)** is an anomaly detection algorithm that identifies data points significantly different from their neighbors based on their local density. Unlike global methods, LOF evaluates the relative density of each data point compared to its surrounding data points, making it particularly effective for datasets with non-uniform distributions. LOF assigns an outlier score to each point, with higher scores indicating greater likelihood of being an anomaly. This approach is widely used in applications were detecting subtle, context-specific anomalies is critical, such as fraud detection, network security, and environmental monitoring.




## Performance Comparison: Isolation Forest vs. LOF Model

### *Isolation Forest Model:*

- **Accuracy**: 0.4931
- **Precision**: 0.9284
- **Recall**: 0.1567
- **F1 Score**: 0.2682

The **Isolation Forest** model demonstrates a **moderate accuracy (0.4931)**, which indicates that it struggles with classifying both anomalies and normal data points correctly. However, it excels in **precision (0.9284)**, meaning it is highly accurate when it predicts a data point to be an anomaly. This indicates that the model rarely misclassifies a normal data point as an anomaly. Despite this high precision, the **recall (0.1567)** is significantly low, which suggests that the model misses a substantial portion of true anomalies in the dataset. The **F1 Score (0.2682)**, which balances precision and recall, reflects the model's overall limited effectiveness in anomaly detection, as it is primarily biased toward correctly identifying the few anomalies it detects.

### *LOF Model:*

- **Accuracy**: 0.6748
- **Precision**: 0.7674
- **Recall**: 0.6475
- **F1 Score**: 0.7023

The **Local Outlier Factor (LOF)** model, in contrast, demonstrates better overall performance. It achieves a **higher accuracy (0.6748)** compared to the Isolation Forest, indicating it is more reliable in correctly classifying both anomalies and normal data points. The **precision (0.7674)** is lower than that of the Isolation Forest, but it still indicates that the LOF model is reasonably good at correctly identifying

anomalies. The key strength of LOF lies in its **higher recall (0.6475)**, which means it detects a larger proportion of true anomalies. The **F1 Score (0.7023)** reflects a better balance between precision and recall, highlighting LOF as a more effective model overall for anomaly detection.

### *Why Explore Other Models:*

While LOF outperforms Isolation Forest in accuracy, recall, and F1 score, both models still exhibit limitations:

- The **Isolation Forest** model's **low recall** means it misses a significant number of anomalies, making it less reliable for comprehensive anomaly detection in critical applications.
- The **LOF model**, though better balanced, could still benefit from improvement in **precision** and overall **accuracy**, especially in highly imbalanced datasets where false positives could be costly.

## Overview of Models Used for Anomaly Detection

### *1. Random Forest*

- **Description**: Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is highly versatile and can be used for both classification and regression tasks.
- **Key Features**:
    - **Robust to Overfitting**: Since it averages the predictions from multiple trees, it is less likely to overfit, making it effective for datasets with a high variance.
    - **Handles Missing Data**: Random Forest can handle missing values and still make accurate predictions.
    - **Feature Importance**: It provides the ability to evaluate the importance of different features, helping in feature selection.
    - **Versatile**: Works well with both categorical and numerical data.

### *2. Logistic Regression*

- **Description**: Logistic Regression is a linear model used for binary classification tasks. It models the probability of a certain class or event existing by using the logistic function to output a value between 0 and 1.
- **Key Features**:
    - **Simple and Interpretable**: Easy to implement and interpret, especially in binary classification problems.
    - **Probabilistic Output**: Outputs probabilities, which can be useful for understanding the certainty of the classification.
    - **Requires Linearity**: Assumes a linear relationship between the input features and the log-odds of the output, which may limit its ability to handle more complex data.
    - **Efficient**: Computationally efficient, suitable for large datasets.

### *3. K-Nearest Neighbors (K-NN)*

- **Description**: K-NN is a non-parametric, instance-based learning algorithm. It classifies a data point based on how its neighbors are classified. The class of the point is determined by a majority vote of its 'K' nearest neighbors.
- **Key Features**:
  - **Simple and Intuitive**: One of the easiest algorithms to understand and implement.
  - **Lazy Learning**: It does not require training, making predictions by directly referencing the training set.
  - **Sensitive to Data Scaling**: Requires normalization or standardization of data for good performance, as it relies on measuring distances between points.
  - **Flexible**: Can be used for both classification and regression tasks.

### *4. Support Vector Machine (SVM)*

- **Description**: SVM is a supervised learning algorithm used for classification and regression tasks. It works by finding the hyperplane that best separates the classes in the feature space, maximizing the margin between the closest points of each class.
- **Key Features**:
  - **Effective in High-Dimensional Spaces**: SVM performs well in high-dimensional spaces and is effective when there is a clear margin of separation between classes.
  - **Kernel Trick**: The kernel trick allows SVM to handle non-linear classification by mapping the data to a higher-dimensional space.
  - **Robust to Overfitting**: By controlling the margin size, SVM can avoid overfitting, especially with high-dimensional data.
  - **Memory and Computation Intensive**: SVMs can be slow with very large datasets and may require significant computational resources.

### *5. Gradient Boosting*

- **Description**: Gradient Boosting is an ensemble learning method that builds a model in a stage-wise manner. It combines weak learners (typically decision trees) to form a strong predictive model by correcting the errors of the previous models in the sequence.
- **Key Features**:
  - **Boosting Technique**: Focuses on correcting the errors of previous models, improving performance over time.
  - **Robust to Outliers**: Gradient Boosting can handle outliers better than simpler models like decision trees.
  - **Versatility**: Can be used for both classification and regression tasks.
  - **Sensitive to Overfitting**: If not carefully tuned, it can overfit to the training data.

### *6. XGBoost*

- **Description**: XGBoost (Extreme Gradient Boosting) is an optimized version of Gradient Boosting that uses a more regularized model to reduce overfitting and improve performance. It is one of the most popular machine learning algorithms for structured/tabular data.
- **Key Features**:
  - **High Performance**: Known for its speed and high accuracy, especially in structured datasets.

- **Regularization**: Includes L1 and L2 regularization to prevent overfitting, making it highly effective for complex datasets.
- **Handling Missing Data**: Has built-in handling of missing data, which simplifies preprocessing.
- **Parallelized**: Efficient use of hardware through parallelization, reducing training time.
- **Boosting Algorithm**: Like Gradient Boosting, it combines weak learners to improve prediction accuracy.

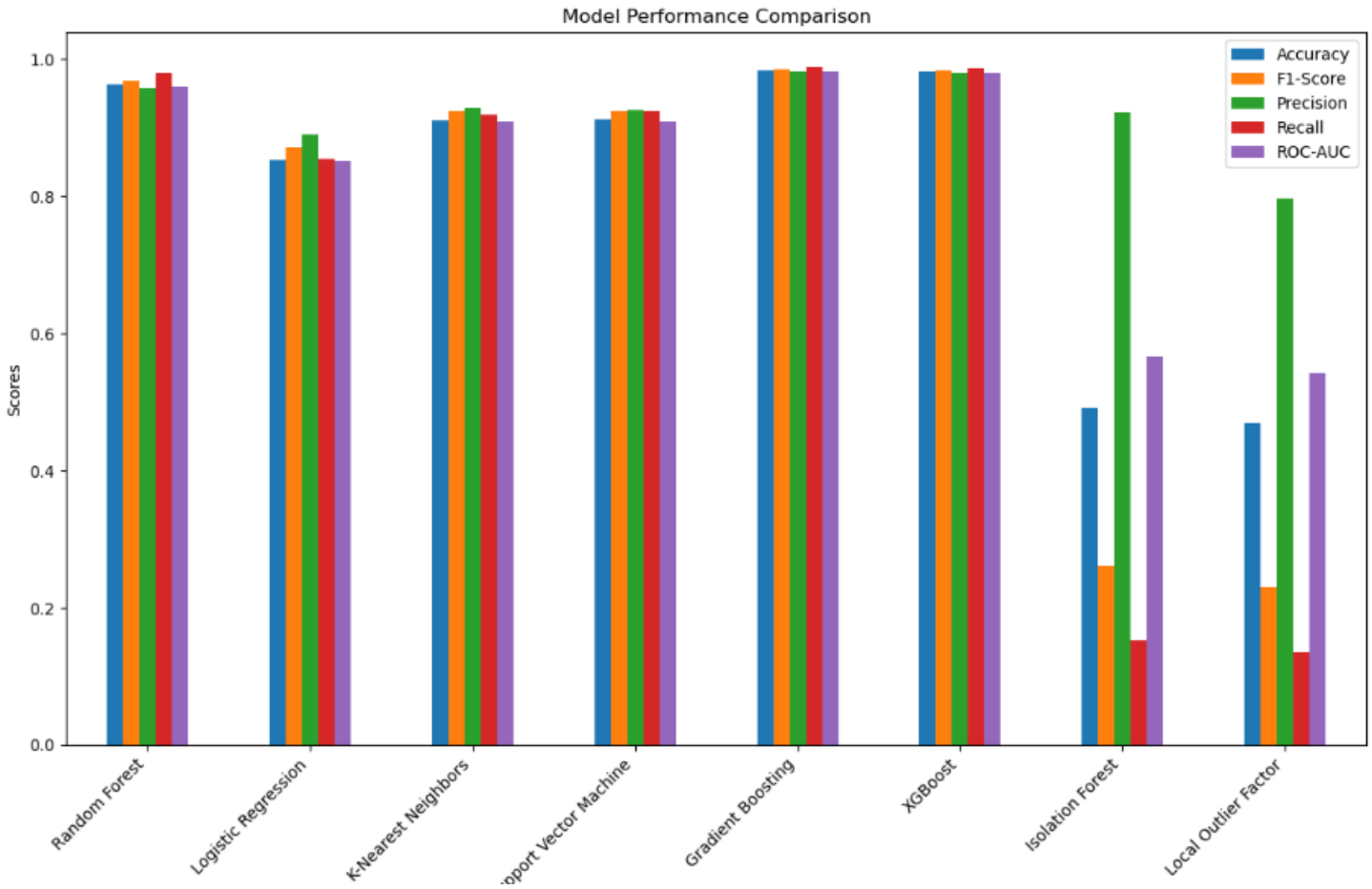
```
Summary of model performance:

Model: Random Forest
  Accuracy: 0.9630
  F1-Score: 0.9690
  Precision: 0.9578
  Recall: 0.9806
  ROC-AUC: 0.9592

Model: Logistic Regression
  Accuracy: 0.8524
  F1-Score: 0.8723
  Precision: 0.8898
  Recall: 0.8556
  ROC-AUC: 0.8517

Model: K-Nearest Neighbors
  Accuracy: 0.9113
  F1-Score: 0.9243
  Precision: 0.9301
  Recall: 0.9187
  ROC-AUC: 0.9097

Model: Support Vector Machine
  Accuracy: 0.9118
  F1-Score: 0.9251
  Precision: 0.9260
  Recall: 0.9242
  ROC-AUC: 0.9091

Model: Gradient Boosting
  Accuracy: 0.9832
  F1-Score: 0.9858
  Precision: 0.9819
  Recall: 0.9897
  ROC-AUC: 0.9817

Model: XGBoost
  Accuracy: 0.9815
  F1-Score: 0.9844
  Precision: 0.9811
  Recall: 0.9877
  ROC-AUC: 0.9802
```

## Model Performance Comparison

In this project, we evaluate the performance of several machine learning models for anomaly detection. The models considered include **Random Forest**, **Logistic Regression**, **K-Nearest Neighbors (K-NN)**, **Support Vector Machine (SVM)**, **Gradient Boosting**, and **XGBoost**. These models were assessed based on key performance metrics, such as accuracy, precision, recall, F1-score, and ROC-AUC. The goal is to identify the most effective model for detecting anomalies in the given dataset.

| Model | Accuracy | F1-Score | Precision | Recall | ROC-AUC |
|---|---|---|---|---|---|
| **Random Forest** | 0.9630 | 0.9690 | 0.9578 | 0.9806 | 0.9592 |
| **Logistic Regression** | 0.8524 | 0.8723 | 0.8898 | 0.8556 | 0.8517 |
| **K-Nearest Neighbors** | 0.9113 | 0.9243 | 0.9301 | 0.9187 | 0.9097 |
| **Support Vector Machine** | 0.9118 | 0.9251 | 0.9260 | 0.9242 | 0.9091 |
| **Gradient Boosting** | 0.9832 | 0.9858 | 0.9819 | 0.9897 | 0.9817 |
| **XGBoost** | 0.9815 | 0.9844 | 0.9811 | 0.9877 | 0.9802 |

Based on the evaluation, **Gradient Boosting** and **XGBoost** perform the best in terms of accuracy, precision, recall, and F1-score, making them the top choices for anomaly detection in this dataset. **Random Forest** also delivers strong performance, especially in terms of precision and recall. While **Logistic Regression** is a reliable model, it performs weaker compared to the ensemble methods. **K-NN** and **SVM** offer competitive results, but they do not match the overall performance of the gradient-boosting models. Therefore, for optimal anomaly detection, **Gradient Boosting** or **XGBoost** would be recommended.




## Business Implications of Anomaly Detection in Crowds

Using a **Smartphone Dataset for Anomaly Detection in Crowds** has several **business implications** across industries such as retail, security, transportation, and urban planning. Here are some key business implications:

### 1. Retail and Marketing

- **Customer Behavior Analysis**: Anomaly detection can help in identifying unusual crowd behavior, such as sudden spikes in foot traffic or unexpected drops in activity. Retailers can use this information to optimize store layouts, promotions, and staffing levels.
- **Personalized Marketing**: By analyzing data from smartphones, businesses can better understand customer movement patterns in malls, shopping districts, or events. This can lead to more targeted advertising and personalized offers, improving conversion rates.

### 2. Security and Surveillance

- **Identifying Security Risks**: Anomaly detection can flag unusual crowd movements, such as sudden clustering of people in one area or abnormal speed or direction of movement, which may indicate potential security threats or emergencies (e.g., stampedes or suspicious behavior).
- **Crowd Monitoring in Public Events**: In large-scale events or public spaces, anomaly detection using smartphone data can help predict dangerous situations or evacuations by analyzing patterns like overcrowding or unexpected crowd shifts.

### 3. Transportation and Traffic Management

- **Crowd Flow Optimization**: Anomaly detection can help predict crowd congestion at transportation hubs like train stations, airports, or bus stops. This helps improve the efficiency of crowd management, optimize transport schedules, and provide real-time updates for travelers.
- **Incident Detection**: It can also be used to detect unusual crowd movements or disruptions caused by traffic incidents, enabling quicker response times and adjustments to transport routes.

### 4. Urban Planning and Smart Cities

- **Urban Space Optimization**: By monitoring how people move through public spaces, cities can improve infrastructure planning, such as the placement of benches, lighting, or signage in high-traffic areas.
- **Smart Infrastructure**: Cities can leverage smartphone-based anomaly detection to optimize the deployment of resources like public transport, police, or medical teams during peak events or during crises (e.g., natural disasters or public health emergencies).

### 5. Healthcare and Public Safety

- **Tracking Public Health Patterns**: Smartphone data can help detect anomalies in crowd behavior, potentially indicating a health crisis, such as the spread of an infectious disease (e.g., large gatherings or sudden crowd dispersals).
- **Emergency Response**: Anomalies in crowd behavior can trigger alerts for healthcare providers or public safety officials, enabling faster response to medical emergencies or public safety threats.

### 6. Event Management and Entertainment

- **Optimizing Event Logistics**: Event organizers can use smartphone data to detect unusual patterns, such as crowd bottlenecks or congestion in specific areas, allowing them to better manage crowd movement and improve safety during concerts, sports events, or festivals.
- **Dynamic Crowd Control**: Real-time detection of anomalies allows event organizers to change logistics or direct crowds to different entrances or exits to maintain a smooth flow of people.

### 7. Data-Driven Decision Making

- **Predictive Analytics for Businesses**: Organizations can use historical data to predict future crowd behavior and prepare for possible anomalies. This could include predicting high-traffic periods in retail or increased risk zones during large gatherings.
- **Cost Reduction**: Efficient anomaly detection can reduce costs associated with manual monitoring, providing automated insights that can optimize staffing, inventory, and resources.

### 8. Privacy and Data Protection

- **Challenges in Data Privacy**: Using smartphone data for anomaly detection requires careful handling of personal information to avoid privacy issues. Businesses must comply with regulations like GDPR to protect consumer data.
- **Consumer Trust**: Transparency about how data is collected and used will be key for businesses in gaining consumer trust. Without proper safeguards, businesses could face reputational damage or legal repercussions.

## 9. New Business Models and Services

- **Crowd-Sourced Data**: Companies can create new business models by offering services that rely on crowdsourced data to analyze and predict crowd behavior. For example, apps that help people avoid crowded areas or optimize their shopping time.
- **Subscription Services**: Data on crowd patterns can be sold to third-party agencies, such as market researchers or urban planners, who use the insights to enhance their services.

## Conclusion

Leveraging a **Smartphone Dataset for Anomaly Detection in Crowds** opens up multiple business avenues that can improve operational efficiency, safety, and customer experience. However, businesses need to balance the potential for enhanced services with data privacy concerns and regulatory compliance. This creates both opportunities for innovation and challenges in data management.

## References


1. UCI HAR Dataset
2. Kaggle Industrial Sensor Data
3. MotionSense Dataset
4. Kaggle Smartphone Sensor Data
5. Crowd Motion Sensor Data
6. Mendeley Data Crowd Dataset