Feature Engineering and Data Augmentation Report

Introduction

The dataset used for anomaly detection in crowds via smartphone data was enhanced through feature engineering and data augmentation techniques. This report documents the newly created features, their purposes, and the results from implementing statistical analysis to explore their relationships.

1. New Features Created

Several new features were engineered based on the existing accelerometer, gyroscope, and location data. These features help to capture more detailed information about movement, behavior, and anomalies in the dataset. Below is a detailed description of each feature:

1.1 Speed Change

- **Purpose**: Captures the change in speed between consecutive data points to detect acceleration and deceleration patterns.
- **Formula**:

```
python
Copy code
df['Speed_Change'] = df['Speed'].diff()
```

1.2 Direction Change

- **Purpose**: Identifies abrupt changes in the movement direction by calculating the difference in heading between consecutive points.
- **Formula**:

```
python
Copy code
df['Direction_Change'] = df['Heading'].diff().fillna(0)
```

1.3 Time Change

- **Purpose**: Computes the time difference between consecutive data points for time-sensitive feature engineering (e.g., rate of change in speed or acceleration).
- **Formula**:

```
python
Copy code
df['Time'] = pd.to_datetime(df['Time'], format='%H-%M-%S')
df['Time_Change'] = df['Time'].diff().dt.total_seconds().fillna(0)
```

1.4 Acceleration Rate

- **Purpose**: Measures the rate at which speed changes, helping to identify acceleration and braking events.
- **Formula**:

```
python
Copy code
df['Acceleration_Rate'] = df['Speed_Change'] / df['Time_Change']
```

1.5 Braking Intensity

- **Purpose**: Determines the intensity of braking by focusing on negative values of acceleration rate (i.e., deceleration).
- **Formula**:

```
python
Copy code
df['Braking_Intensity'] = df['Acceleration_Rate'].apply(lambda x: x
if x < 0 else 0)
```

1.6 Jerk

- **Purpose**: Calculates the rate of change of acceleration (jerk), which can be indicative of sudden movements or stops.
- **Formula**:

```
python
Copy code
df['Jerk'] = df['Acc_Magnitude'].diff() / df['Time_Change']
```

1.7 Cumulative Distance

- **Purpose**: Keeps a running total of the distance covered over time, aiding in trajectory analysis.
- **Formula**:

```
python
Copy code
df['Cumulative_Distance'] = df['Distance'].cumsum()
```

1.8 Speed Variance

- **Purpose**: Measures the variance in speed over a rolling window, providing insights into the steadiness of the movement.
- **Formula**:

```
python
Copy code
df['Speed_Variance'] = df['Speed'].rolling(window=5).var()
```

2. Time-Based Features

2.1 Rolling Mean of Accelerometer X (Acc X)

- **Purpose**: Computes the rolling average of the accelerometer's X-axis data over a 5-sample window, which helps smoothen out fluctuations and reveal underlying trends.
- **Formula**:

```
python
Copy code
df['Rolling_Mean_AccX'] = df['Acc X'].rolling(window=5).mean()
```

2.2 Moving Variance of Gyroscope X (Gyro X)

- **Purpose**: Calculates the variance of the gyroscope's X-axis over a 5-sample window to detect variations in rotational motion.
- **Formula**:

```
python
Copy code
df['Variance_GyroX'] = df['gyro_x'].rolling(window=5).var()
```

3. Engineered Feature

3.1 Total Acceleration

- **Purpose**: Computes the total magnitude of acceleration from the X, Y, and Z components of the accelerometer data, providing a comprehensive measure of motion intensity.
- **Formula**:

```
python
Copy code
df['Total_Acc'] = np.sqrt(df['Acc X']**2 + df['Acc Y']**2 + df['Acc
Z']**2)
```

4. Feature Importance Analysis

A correlation matrix was computed to identify the relationships between the various features. Significant observations include:

- **Speed** is positively correlated with **Heading** and **Cumulative Distance**.
- **Total Acceleration** is strongly correlated with **Acceleration Magnitude**, suggesting that these features provide similar information regarding motion intensity.
- **Jerk** is moderately correlated with **Acceleration Magnitude**, which is expected as it measures the rate of change in acceleration.
- Features like **Braking Intensity** and **Speed Variance** exhibit low correlations with most other features, indicating their unique contribution to the dataset.

5. Results Summary

The new features provide deeper insights into the behavior of the data and allow for more advanced analysis of movement, including sudden stops, changes in direction, and speed variations. The combination of time-based rolling statistics and engineered features will

improve anomaly detection, leading to better performance in security and crowd management applications.

6. Conclusion

The feature engineering and data augmentation efforts have enhanced the dataset by creating more informative variables that better represent the underlying dynamics of movement. These features will be useful for training machine learning models for real-time anomaly detection and understanding crowd behavior in various environments.

7. Saved Dataset

The updated dataset with all new features has been saved as `augmented_dataset.csv`.

Anomaly Detection Report: Z-Score vs IQR Comparison

1. Introduction

Anomaly detection plays a crucial role in identifying unusual patterns or outliers that deviate from the normal behavior of data. This report compares two widely used techniques for anomaly detection: Z-Score and Interquartile Range (IQR). We applied these methods to various sensor-based parameters, including acceleration, speed, and gyroscope readings, using time-series and scatter plots to visualize the results.

2. Objective

The primary objective of this analysis is to:

Detect anomalies using Z-Score and IQR methods.

Visualize anomalies through time-series plots and scatter plots.

Compare the performance and outputs of both methods.

3. Dataset Overview

The dataset contains sensor readings, including acceleration along the X, Y, and Z axes, speed, and gyroscope data. Each entry is timestamped to allow temporal analysis. The key parameters used for anomaly detection include:

Acc X, Acc Y, Acc Z: Accelerometer data

Speed: Speed readings

gyro_x, gyro_y, gyro_z: Gyroscope data

4. Methods Used for Anomaly Detection

4.1. Z-Score Method

Formula:

Where is the mean and is the standard deviation.

Anomaly Detection:

A Z-Score above 3 or below -3 indicates an outlier (since it lies outside 99.7% of the data in a normal distribution).

Advantages:

- Useful when data follows a normal distribution.

- Easy to implement and interpret.

4.2. IQR (Interquartile Range) Method

Formula:

IQR = Q3 - Q1 (where Q1 and Q3 are the 25th and 75th percentiles, respectively).

Lower Bound = Q1 - 1.5 * IQR

Upper Bound = Q3 + 1.5 * IQR

Anomaly Detection:

Any value outside the lower or upper bounds is classified as an anomaly.

Advantages:

- Robust to non-normal data distributions.
- Less sensitive to extreme values compared to the Z-Score.

5. Code Implementation and Visualization

5.1. Z-Score Calculation and Visualization

Using Z-Score, we identified outliers for each parameter. A scatter plot with time on the X-axis and sensor readings on the Y-axis was generated for better visualization. The anomalies were highlighted with distinct colors.

```
# Z-Score Calculation

for param in parameters:

    calculate_z_scores(df, param)
```

5.2. IQR Calculation and Visualization

IQR-based anomalies were identified by checking if any values fell outside the IQR range. Similar scatter plots were created to visualize these anomalies.

5.3. Comparison of Z-Score and IQR Anomalies

We compared the results from both methods to understand the consistency between them. A joint report was generated showing anomalies detected by each method side-by-side.

6. Results and Observations

6.1. Z-Score Observations

Z-Score effectively captured extreme deviations from the mean.

It identified several outliers for parameters such as Speed and Acc Z.

However, the Z-Score method showed limitations in handling non-normal data distributions.

6.2. IQR Observations

IQR-based detection performed well in identifying anomalies in datasets with skewed distributions.

The method flagged different anomalies than the Z-Score approach for certain parameters, indicating its robustness to non-normality.

7. Visualization Insights

- Time-Series Plots:

  These plots provided temporal insights into when anomalies occurred, allowing for better understanding of trends and irregularities over time.

- Scatter Plots:

  Scatter plots clearly displayed anomalies identified by both methods. In some cases, IQR and Z-Score detected the same outliers, but in other instances, their results diverged.

8. Conclusion

Both Z-Score and IQR methods offer valuable insights for anomaly detection, but they have distinct strengths. Z-Score is more appropriate for normally distributed data, while IQR is robust for data with skewness or outliers. For best results, it is recommended to use both techniques in parallel and compare their outputs.