Efficient Data Stream Anomaly Detection

This project simulates a real-time data stream and detects anomalies using three algorithms: Z-Score, Isolation Forest, and One-Class SVM. The algorithms monitor the stream and identify unusual patterns or outliers in the data. A visualization displays the data stream and highlights detected anomalies in real-time. This system offers flexible anomaly detection, balancing simplicity and computational efficiency depending on the chosen algorithm.

Algorithms:

For this project i have used 3 algorithms - Z-Score , Isolation Forest and One-Class SVM. The reason for that is to demonstrate the use of 3 of the powerful and widely accepted anomaly detection algorithms in a modular format. This approach can allow flexibility, as we can switch between algorithms and even use to to compare them each other.

Z-Score: This method calculates the mean and standard deviation of a sliding window of data points. It then computes the Z-Score for each incoming point (how far the point is from the mean, in terms of standard deviations). If the Z-Score exceeds a predefined threshold, the point is considered an anomaly.

Isolation Forest: This is an ensemble-based unsupervised learning algorithm that isolates anomalies by randomly partitioning the data. Anomalies are isolated faster because they are few and different, making them easier to separate in fewer steps.

One-Class SVM: This is a variation of the Support Vector Machine (SVM) that is trained to distinguish normal data from anomalies. It tries to find a boundary that separates normal data from outliers, using only normal data to learn.

Data Stream Simulation:

The data stream simulation in this project generates a sequence of data points that mimic a real-time stream. It uses a sine wave pattern with added random noise to simulate normal data. Occasionally, anomalies are introduced by adding a large, random value to simulate abnormal behavior. The simulation runs for a specified number of iterations (`n=1000` by default) and yields one data point at a time, making it ideal for testing anomaly detection in a streaming context.

Anomaly Detection

Used 3 kinds of Anomaly Detection Techniques:

1) **Z-Score Anomaly Detector**: Uses a sliding window to compute mean and standard deviation, and detects anomalies based on the Z-Score.

2) **Isolation Forest Anomaly Detector**: Uses Isolation Forest to identify anomalies based on the assumption that anomalies are few and different from the rest of the data.

3) **One-Class SVM Detector**: Uses Support Vector Machines to identify anomalies by learning from the data and separating the normal points from anomalies.

Optimization for Speed and Efficiency

Z-Score:

- **Speed**: Very fast for small datasets, as it only requires updating the mean and standard deviation within a sliding window.
- **Efficiency**: Computationally light but less sophisticated, so it might miss complex patterns or be sensitive to changes in the data stream.

Isolation Forest:

- **Speed**: Reasonably fast, especially for large datasets. However, the initial model fitting can be slow for high-dimensional data.
- **Efficiency**: More robust for detecting a wide variety of anomalies but requires tuning for contamination levels and data size.

One-Class SVM:

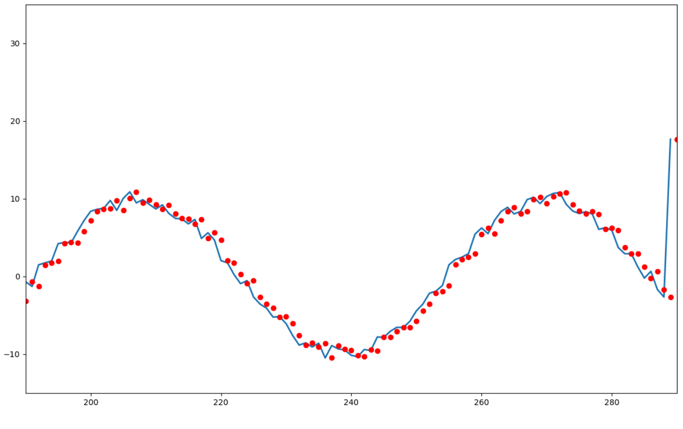
- **Speed**: Slower than Z-Score and Isolation Forest, especially when fitting the model to data, as SVMs are computationally expensive.
- **Efficiency**: Good for handling complex anomalies, but can be sensitive to the choice of kernel and hyper parameters, which might affect its speed and accuracy.

Visualization

The project uses `matplotlib` to visualize streaming data in real-time. A line plot updates continuously as new data arrives, with red scatter points marking anomalies. The plot scrolls to keep the latest 100 points visible, making anomalies easy to spot.

Results:

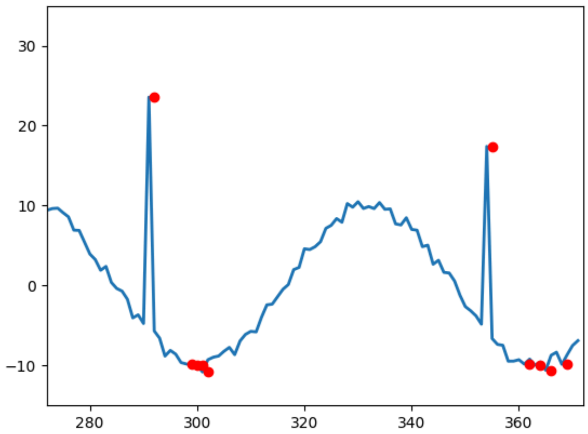
Z-Score:



- This approach detects anomalies primarily based on the deviation from the mean, making it good for periodic or seasonal data patterns.

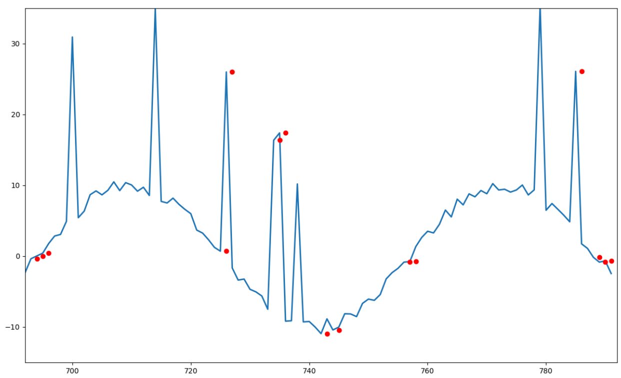
- However, it can sometimes miss subtler anomalies and is dependent on how you set the threshold (typically a trade-off between false positives and false negatives).

Isolation Forest:



- Captured anomalies at significant spikes and dips.
- Suitable for detecting global anomalies that deviate considerably from the regular data pattern.
- The red points correspond well with large changes in the data stream, showing it handles large variations effectively.

One-Class SVM:



- This method appears to capture both smaller and larger anomalies.
- More sensitive than Isolation Forest, detecting a broader range of points, possibly including some borderline outliers.
- It may be overly sensitive in some regions, flagging several consecutive points as anomalies.

Comparisons

For major deviations: Isolation Forest performs best as it robustly captures anomalies without flagging too many consecutive points.

For smaller, continuous deviations: One-Class SVM is more sensitive, but it may result in more false positives, as seen from more red dots clustered together.

For periodic data: Z-Score performs consistently for typical seasonal data but may miss sharp anomalies like the other methods.