Real-time Data Stream and Anomaly Detection

This project demonstrates real-time data stream processing and anomaly detection using a **Kalman Filter**. The data stream includes a trend, seasonal component, and noise, with anomalies introduced at random points.

Features

- Simulate a data stream with trend, seasonality, and noise.
- Introduce positive and negative anomalies to the data stream.
- Use a Kalman Filter for real-time prediction.
- Detect anomalies based on a threshold.
- Visualize the data stream, predictions, and anomalies in real-time.

Requirements

- Python 3.x
- NumPy
- Matplotlib

Installation

1. Clone the repository:

```python

git clone https://github.com/yourusername/repo-name.git
```

cd repo-name

2. Install the required packages:

```python

pip install -r requirements.txt

```

3. Run the script:

```python

python main.py

```

Algorithm Selection:

The Kalman Filter is chosen for anomaly detection due to its ability to adapt to concept drift and seasonal variations. It provides real-time predictions and updates based on incoming data points, making it suitable for dynamic environments.

The ***Kalman filter*** is an efficient recursive algorithm that provides estimates of the hidden state of a dynamic system from a series of noisy measurements. It operates in two main steps: prediction and update.

- **Prediction**: Based on the system's previous state and known dynamics (modeled by equations), the Kalman filter predicts the next state of the system. This step also predicts the uncertainty of this state estimate.
- **Update**: Once new measurements (observations) are available, the filter corrects its prediction. It does this by computing a weighted average of the predicted state and the new measurement. The weights are determined by the uncertainties in the prediction and the measurements, with more weight given to the more accurate source (less noisy).

Data Stream Simulation

The data stream is simulated with a trend, seasonal component, and noise. Anomalies are introduced at random points.

Anomaly Detection

A Kalman Filter is initialized and used for real-time prediction. Anomalies are detected based on a threshold.

Optimization

The algorithm is optimized for both speed and efficiency as all of the new data points are updated in O(1).

> **Note**: A delay was added only to mimic the real time processing.

Visualization

The plot is initialized in interactive mode to update in real-time, displaying the true data, predictions, and detected anomalies

Key Kalman Filter Parameters for Adaptability:

Process Noise Covariance (Q):

`Q` models the uncertainty in the system dynamics. A larger value of `Q` makes the Kalman filter more responsive to changes, as it assumes that the system is more uncertain or dynamic.

- Increase `Q` to make the filter adapt more quickly to abrupt changes in the system or data. This reduces the filter's reliance on the past and increases responsiveness to new measurements.

Measurement Noise Covariance (R):

`R` models the uncertainty in the observations.

- A larger `R` tells the filter that the measurements are noisy and should be trusted less.
- Conversely, a smaller R gives more trust to the incoming data.
  Decrease R if you want the filter to trust new observations more, making the filter adapt faster to real changes.

Error Covariance Matrix (P):

`P` represents the uncertainty in the state estimate. Although it's not usually tuned dynamically, its initialization can affect the initial response.

- A larger initial `P` makes the filter less confident about its initial estimate, leading to quicker adaptation in the beginning.

Tuning Strategy:

Increase Adaptation (Fast Response):

- **Increase Q (higher process noise)**: to allow for faster tracking of changes in the system.
- **Decrease R (lower measurement noise)**: to trust the measurements more and adapt to sudden changes quickly.

Decrease Adaptation (Slow Response):

- **Decrease Q (lower process noise)**: to make the filter smoother, less responsive to sudden changes.
- **Increase R (higher measurement noise)**: to trust the measurements less, allowing the filter to focus on the system model rather than new data points.