Efficient Real-Time Anomaly Detection Using LSTM Autoencoder

# Table of Contents

1. Introduction  
2. Data Stream Simulation  
3. Chosen Algorithm: LSTM Autoencoder  
4. Implementation Explanation  
5. Thresholding Mechanism: EMA  
6. Real-Time Visualization  
7. PDF Export and Reporting  
8. Conclusion

# 1. Introduction

This project aims to detect anomalies in a continuous data stream using a real-time mechanism. The data stream simulates time-series data, incorporating seasonal patterns, random noise, and rare anomalies. The focus is on identifying anomalies efficiently using an LSTM-based autoencoder and dynamically adjusted thresholds.

# 2. Data Stream Simulation

The data stream is simulated by combining several components:  
- Seasonal Patterns: Simulated using a sinusoidal function.  
- Noise: Random noise is added to make the stream more realistic.  
- Anomalies: Rare anomalies are randomly introduced into the data.  
- Long-term Trend: A small upward trend is added over time.  
  
This provides a realistic simulation of metrics that could represent financial transactions or system metrics.

# 3. Chosen Algorithm: LSTM Autoencoder

The core algorithm used for anomaly detection is an LSTM Autoencoder. The LSTM (Long Short-Term Memory) network is well-suited for time-series data, as it captures long-term dependencies and learns patterns over time. The autoencoder reconstructs the input data and compares the reconstruction error to detect anomalies.

## Effectiveness

- LSTM Strength: Capable of learning complex patterns over time and detecting deviations from normal patterns.  
- Autoencoder Role: Learns to reconstruct normal patterns in the data, and anomalies manifest as high reconstruction errors.  
- Adapts to Concept Drift: The model continuously adapts to changing patterns, making it suitable for environments where data characteristics evolve.

# 4. Implementation Explanation

Training  
The LSTM autoencoder is trained on a sliding window of the data stream. The window captures a subset of the time series, and the model is trained to reconstruct this data. The model comprises:  
- An encoder that compresses the input data into a lower-dimensional representation.  
- A decoder that reconstructs the data from the encoded representation.  
The model is trained using Mean Squared Error (MSE) as the loss function.

## Anomaly Detection

Once the model is trained, it continuously monitors the data stream. For each sliding window, the model calculates the reconstruction error. If the error exceeds a dynamic threshold, the current point is flagged as an anomaly.

# 5. Thresholding Mechanism: EMA

The threshold for anomaly detection is not fixed. Instead, it is dynamically adjusted using an Exponential Moving Average (EMA). This allows the threshold to adapt to changes in the data stream over time. The formula for updating the EMA is:  
threshold\_ema = 0.9 \* previous\_ema + 0.1 \* current\_mse  
  
This ensures that the threshold is sensitive to short-term spikes but does not overreact to minor fluctuations.

# 6. Real-Time Visualization

The project uses Plotly to provide real-time visualization of the data stream and detected anomalies. The stream is plotted continuously, and anomalies are marked with red dots for easy identification. You can also zoom, download and see each and every plot in full screen.

# 7. PDF Export and Reporting

After the simulation completes, users can export the results to a PDF report. The report summarizes:  
- Total number of data points  
- Total anomalies detected  
- A detailed listing of each data point and whether it was flagged as an anomaly.

# 8. Conclusion

The LSTM autoencoder, combined with the EMA thresholding mechanism, provides a robust and adaptive method for real-time anomaly detection in continuous data streams. The project demonstrates the algorithm's effectiveness in identifying unusual patterns while adapting to concept drift and seasonal variations.