# LSTM Autoencoder Anomaly Detection Workflow

1. **Data Cleaning Phase** - Uses the Confidence Detector to iteratively clean training data
2. **Model Training Phase** - Trains on the cleaned data to learn normal patterns
3. **Threshold Optimization Phase** - Uses synthetic anomalies to find optimal detection threshold
4. **Evaluation Phase** - Applies trained model to detect anomalies in new data

A screenshot of a computer

AI-generated content may be incorrect.

**Implementation Details**

1. **Confidence Detector (5 iterations)**

* Trains initial model on raw data
* Finds anomalies using statistical confidence intervals
* Removes/replaces detected anomalies with interpolated values
* Retrains model on cleaned data
* Repeats process to progressively clean the data

1. **LSTM Autoencoder Training**

* Trains on the cleaned data to learn normal patterns
* Creates both global model and station-specific models

1. **Threshold Optimization**

* Uses synthetic anomalies to find the best threshold
* Calculates optimal threshold using precision/recall metrics
* This is now enabled with optimal\_threshold\_tuning=True

1. **Anomaly Detection**

* For each window in new data:
* Compares reconstruction error against threshold
* Flags points with errors above threshold as anomalies

Current issues:

Can find synthetic anomalies but not actual anomalies in the data.

Solution: Iterative approach? Semi-supervised? Prefilter these?

A graph on a white background

AI-generated content may be incorrect.

Merge together periods of consecutive anomalies?

Identify start and end of each anomaly.