**Real-time Anomaly Detection System**

**Algorithm Overview**

This system implements a real-time anomaly detection algorithm with the following key components:

1. **Data Generation**

* Generates synthetic data with seasonal patterns
* Includes random noise following a Gaussian distribution
* Introduces anomalies with 5% probability

1. **Anomaly Detection Method**

* Uses absolute threshold boundaries
* Implements direct anomaly flagging for generated anomalies
* Validates data points against predefined normal ranges

1. **Optimization Features**

* Employs **Welford's algorithm** for stable statistical calculations
* Implements efficient memory management using deque data structures
* Includes automatic data cleanup for off-screen points

**Effectiveness**

The algorithm is particularly effective because:

1. **Memory Efficiency**

* Automatically removes old data points
* Uses fixed-size data structures
* Implements efficient data cleaning routines

1. **Accuracy**

* Prevents false positives after anomalies
* Maintains clear distinction between normal and anomalous data
* Provides robust detection of genuine anomalies

1. **Performance**

* Optimized for real-time processing
* Minimal computational overhead
* Efficient visualization updates

**Error Handling**

The system includes comprehensive error handling:

1. **Data Validation**

* Input boundary checking
* Null value handling
* Type verification

1. **Exception Management**

* Try-except blocks in critical sections
* Graceful error recovery
* Proper resource cleanup

1. **Resource Management**

* Proper matplotlib cleanup
* Memory management
* Buffer overflow prevention