Open-source Technologies and Stream Mining joint Project Documentation

Table of contents

Smart City Air Quality Monitoring with Real-Time Stream Analytics (SCAir-IoT)

Abstract

lorem ipsum...

Keywords: *Anomaly detection, Forecasting, Stream mining, Open-source technologies*

Authors

Albazzal, Houmam
Balogh, Máté
Földvári, Ádám
Lahmar, Abderraouf
Nagy, Zsuzsanna

Mentored by

Loubna Seddiki

1 Introduction

2 Background and Literature Review

In stream mining, we are limited to a portion of data and make decisions real-time in memory. As *Wares, Isaacs, and Elyan (2019)* highlight, in traditional machine learning contexts, data is referred to as batch data which can be loaded into memory in its entirety. According to the authors,

> "this is of stark contrast to stream mining, where data streams produce elements in a sequential, continuous fashion, and may also be impermanent, or transient, in nature … This means stream data may only be available for a short time."

The authors refer to *Babcock et al. (2002)*, highlighting that

> "once an element from a data stream has been processed, it is discarded or archived. It cannot be retrieved easily unless it is explicitly stored in memory, which is small relative to the size of data streams."

3 Dataset

The dataset is the *UCI Air Quality* dataset *Vito, S. (2008)* which includes responses of gas sensor devices deployed in an Italian city. Besides these device readings, each gas measurement has a counterpart feature which denotes the gas concentration recorded by a co-located certified analyzer. Additionally, readings related to temperature along with absolute and relative humidity are included in the dataset.

The records span 1 year from March 2004 to February 2025, and are present in hourly aggregated form. Missing values are denoted with the value of `-200`.

4 System architecture

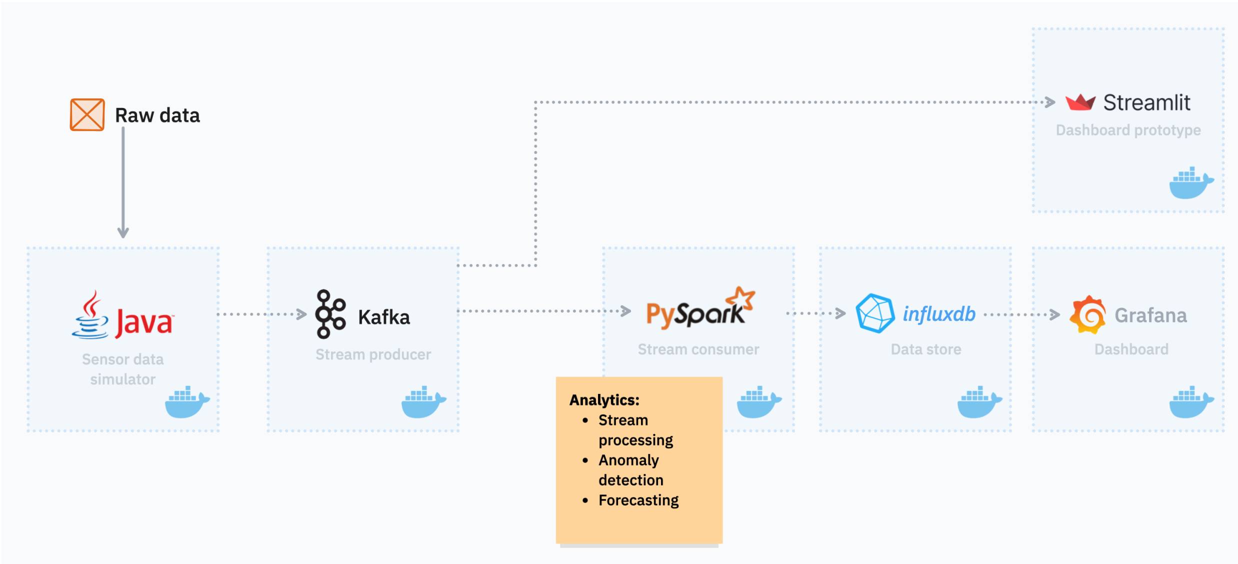

Figure 1: High-level view of the architecture with the utilized open-source technologies denoted for each component.

The stream mining pipeline includes components of *simulator*, *producer*, *consumer*, *data store*, and *dashboards* by which the raw `csv` dataset file was ingested. We utilized open-source technologies of *Java*, *Kafka*, *PySpark*, *Influxdb*, *Streamlit*, and *Grafana* provisioned in a containerized environment via *Docker*. The responsibility of each component is summarized as follows:

Raw data: Static file containing sensor measurements related to air quality.

Sensor data simulator: Reads the raw data file and simulates flow of sensor data.

Kafka producer: Streamlines the simulated sensor data into Kafka topics in their datetime order.

PySpark consumer: Listens to the streamlined data and creates mini batches to call analytical functions—such as Anomaly Detection and Forecasting—on this windowed data.

Influxdb: Data is then persisted in the database including the online predictions and the original incoming data.

Streamlit: Streamlit is used to create dashboard prototypes without the necessity of a database storage and connection. Kafka messages are directly consumed by this component to display simple line charts and anomaly detection related information and alerts.

Grafana: Dashboard visualization component that periodically fetches the database for new data to show the latest insights in real-time.

5 Modeling and Predictions

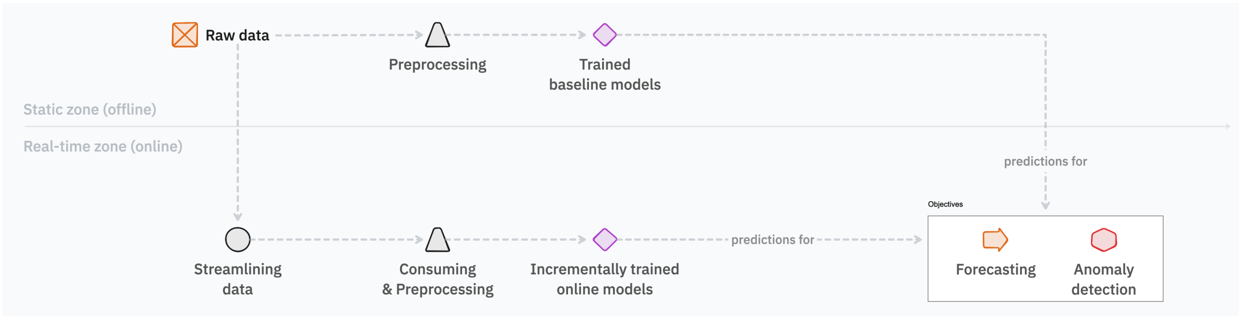

Figure 2: High-level view of the offline and online machine learning pipeline.

For *Anomaly Detection*, three simple yet essential methods were utilized by following a general novelty detection procedure:

1. Data transformation, to better reflect properties of interest
2. Obtain a novelty function
3. Apply peak picking algorithm on the novelty function.

Values of `-200` were replaced with `NA`. First, readings with missing values—meaning no value for a sensor for an entire hour—were detected. As a second method, missing values were imputed with the median of an 8-sample window, derivative operator was applied on this data, and positive peaks were considered by clipping negative changes to zero, only to apply a traditional IQR-based outlier detection on this obtained novelty function where `NA`s were re-inserted after the derivative function. As for a third approach, the same novelty detection was followed but percentiles were calculated based on an iteratively updated and maintained *T-digest* algorithm accounting for a global opinion on abnormal readings.

6 Experiments and testing

7 References

Wares, S., Isaacs, J. and Elyan, E. (2019). Data stream mining: methods and challenges for handling concept drift. SN Applied Sciences, 1(11), 1412.

Babcock, B., Babu, S., Datar, M., Motwani, R. and Widom, J. (2002, June). Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (pp. 1–16).

Vito, S. (2008) Air Quality Dataset. UCI Machine Learning Repository. Available at: https://archive.ics.uci.edu/ml/datasets/Air+Quality (Accessed: 25 November 2025).