DQAT: An Online Machine Learning Framework for Real-Time Data Quality Assurance in IoT

Marcos Lima Romero and Ricardo Suyama

Abstract— **The Internet of Things (IoT) revolutionizes agriculture, but the quality of the generated data often hinders reliable decision making. This study introduces the Data Quality Assurance Tool (DQAT), an open-source event-driven framework tailored for real-time data assessment in IoT systems. DQAT's modular architecture enables seamless integration with existing applications and facilitates end-to-end scenario simulations. Using online machine learning algorithms like Half-Space Trees and Support Vector Machines, DQAT detects anomalies in streaming data, outperforming traditional batch methods. Evaluation with agricultural datasets demonstrates DQAT's ability to monitor critical data quality dimensions, including accuracy, completeness, timeliness, and availability. This research directly contributes to the improvement of the trustworthiness and utility of data for informed decision making in the IoT sector.**

Keywords— **Data Quality, Machine Learning, Data Stream, Online Learning, IoT**

I. Introduction

The enthusiasm for Artificial Intelligence (AI) solutions has sparked questions about how it will impact individuals, society, and the economy, driving enormous investments. Nevertheless, to achieve the desired positive return of these investments and benefit for both individuals and society, attention to Data Quality is fundamental. AI solutions rely in general on high-quality data inputs for success [1]. Poor data quality has led to catastrophic consequences around the world, such as the NASA Challenger space shuttle explosion in 1986, where flaws in Data Quality dimensions, such as accuracy, completeness, consistency, relevance, and relevance, were determining factors of the disaster [2]. Another example of the significant role that data quality plays in society was the 2008 financial crisis, which was exacerbated by poor data quality in the subprime mortgage market. Inaccurate risk assessments and lack of transparency in mortgage data contributed to poorly informed investment decisions, leading to the collapse of major financial institutions [3].

In the context of Brazilian agriculture, the Internet of Things (IoT) generates vast amounts of data from sensors, weather stations, and other sources. Although these data have the potential to transform farming practices, the quality of these data is often compromised. Inaccurate sensor readings, missing data points, and delays in data transmission can lead to erroneous predictions, suboptimal resource allocation, and ultimately reduced crop yields. For instance, inaccurate soil moisture data could result in over- or under-irrigation, while delayed pest infestation alerts could miss critical intervention windows [4].

Several articles have highlighted the complex challenges of ensuring data quality in IoT systems [5]. These challenges arise from the inherent variety and heterogeneity of data sources, ranging from diverse sensor types to mobile devices, each producing data in different formats and structures. The dynamic nature of IoT environments, with devices constantly joining and leaving the network, further complicates data quality assurance. Furthermore, the lack of standardization and interoperability between different devices and IoT platforms exacerbates these issues [6]. Despite these recognized challenges, the issue of inadequate Data Quality in IoT often remains overlooked, potentially leading to significant financial, operational, and even safety consequences if left unaddressed.

A major challenge in real-world IoT systems is maintaining data quality in real-time. Although previous initiatives have attempted to automate data quality management, most have not fully applied online machine learning algorithms that can adapt to data drift and concept drift over time. Data drift occurs when the statistical properties of the input data change over time. These changes can be due to intentional malicious actions, such as attackers altering their behavior to evade detection, or unintentional circumstances such as data quality issues [7]. Concept drift can be described as the changes in the pattern and relations of the target in a data stream, thus existing models trained offline become rapidly obsolete [8]. Although several works have addressed concept drift in IoT systems, including anomaly and intrusion detection, none have specifically focused on the impact of concept drift on data quality dimensions [7], [9], [10], [11].

To address these challenges, this study introduces the Data Quality Assurance Tool (DQAT)[1], an open-source, event-driven software framework designed to seamlessly integrate with real-world IoT applications. DQAT leverages on-line machine learning algorithms, implemented using Python libraries River[2] [12] (a dedicated framework for online machine learning and data stream mining) and PySAD[3] [13], such as Half-Space Trees (HST) and One-Class Support Vector Machines (OCSVM), for real-time anomaly detection in streaming data, offering advantages over traditional batch methods. The modular architecture of the tool is a key feature, enabling flexibility and customization, allowing it to adapt to the dynamic nature of IoT environments and diverse data sources. By focusing

Marcos Lima Romero, e-mail: marcos.romero@ufabc.edu.br; Ricardo Suyama, e-mail: ricardo.suyama@ufabc.edu.br; CECS, UFABC, Santo André-SP

[1] https://github.com/RomeroCode/DQAT

[2] https://riverml.xyz/

[3] https://pysad.readthedocs.io/

on the detection, interpretation, and adaptation to concept drift, DQAT aims to enhance the reliability and stability of anomaly detection models in IoT systems. The tool was designed to be agnostic to data inputs and to function with minimal configuration requirements. It also performs online data profiling, updating with each new data entry, and provides crucial information on availability and accuracy, and other relevant data quality dimensions.

II. Background

A. Data Quality

The concept of data quality has been widely discussed in research on the topic. Data quality is defined by a degree of quality according to the purpose of the data for a particular need. In addition to this gradual concept, there is a dimensional approach to data quality, which means that there are several facets of data that must be observed, each with a level of adherence to overall quality.

Organizations such as the International Organization for Standardization (ISO) and Data Management Association International (DAMA) have initiatives to standardize concepts and practices involving data quality. ISO 8000 [14] defines data quality as the degree to which the inherent characteristics of the data meet demands; ISO 25012 [15] defines it as the degree to which the characteristics of the data satisfy the stated and implied needs when used under specified conditions; DAMA-DMBOK2 [16] defines data quality as how well the data meet the expectations and requirements of those who use them.

Furthermore, Karkouch et al. [17] specifically described the data quality for the IoT as appropriate device data collected to provide ubiquitous services to users of the IoT. The definitions used in these references differ in some terms, but all agree that there is a degree of adherence in which data must meet to be considered of high quality, the degree of data fitness for the application. Various authors point out many different dimensions, but, specifically for IoT, the main data quality dimensions cited are:

- **Accuracy:** The degree to which data reflects the true state of the entities it represents. Inaccurate data can lead to incorrect conclusions and faulty decisions.
- **Completeness:** The extent to which all required data elements are present. Missing data points can hinder analysis and create gaps in understanding.
- **Confidence:** The degree to which data contains a real value within a range.
- **Timeliness:** The degree to which data is available and up-to-date when needed. Outdated data can lead to missed opportunities or incorrect decisions.
- **Availability:** The extent to which data is available and retrievable.

A previous systematic survey obtained 667 software dedicated to data quality [18]. They found that more than half of the existing tools are domain-specific and none addressed all the most important data quality dimensions simultaneously (accuracy, consistency, timeliness, and completeness). Among open-source tools, Apache Griffin is the most similar tool to the proposed implementation, but was reported to have a lot of dependencies, relies on Apache Spark, does not have an event-oriented architecture and still needs some SQL abstraction.

B. Anomaly Detection in Data Streams

Several methods have been proposed for anomaly detection in scenarios with static data, i.e., data are available beforehand to train the model. Some of the proposed methods include Isolation Forest [19], that work by isolating anomalies in the dataset rather than modeling the normal data points by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature to isolate the anomaly; Local Outlier Factor (LOF) [20], which identifies anomalies by comparing the local density of a point to its neighbors; One-Class Support Vector Machines (OCSVM) [21], that operates separating normal data from anomalies by identifying a hyperplane that maximizes the margin between the data points and the origin; Clustering algorithms, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), can also be used labeling points in sparse regions as anomalies [22]; k-d Trees, which organize points in a k-dimensional space for nearest neighbor searches, aiding density-based anomaly detection. These methods focus on detecting anomalies using data distribution and space-partitioning techniques.

Static data learning and batch learning face challenges in addressing the dynamic nature of IoT environments, which continuously generate massive amounts of data. Often, applications lack sufficient data to train machine learning models, and limited computing resources can make storing training data problematic. Machine learning in data streams, on the other hand, presents distinctive features, including the continuous generation of potentially unbounded data over time, which requires single-pass processing due to the impracticality of storing the entire stream [23]. Anomalies, being rare occurrences, often require learning from predominantly normal instances only, with models needing to adapt to evolving data distributions and concept drifts. Moreover, memory and computational efficiency are paramount, with algorithms needing to operate within limited memory and in real-time, while maintaining scalability with data volume. Robustness against noise and outliers, along with parameter insensitivity for stable performance across diverse datasets, are essential. Incremental learning enables continuous model updates without complete retraining, facilitating autonomous operation with minimal human intervention. These characteristics are important for effective anomaly detection in dynamic and large-scale data environments such as network traffic monitoring and real-time fraud detection.

There are a few open-source libraries that implement machine learning methods for stream setting, such as PySAD [13], Creme [24], scikit-multiflow [25], and River [12], which combines the two aforementioned projects in a new architecture and expands their functionality. These improvements include support for mini-batches, processing time improvements, new metrics for classification, regression, and clustering, additional clustering methods, etc. In this work, we

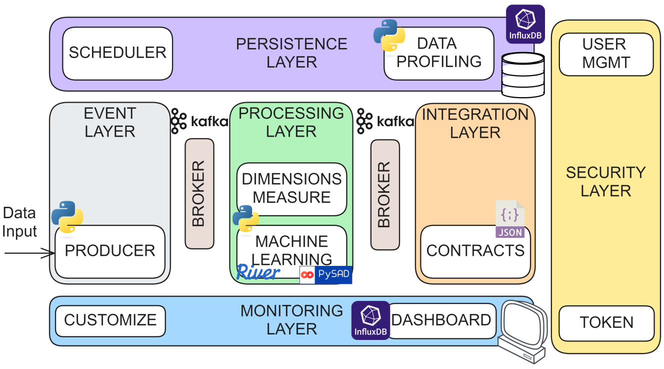


Fig. 1. DQAT basic architecture.

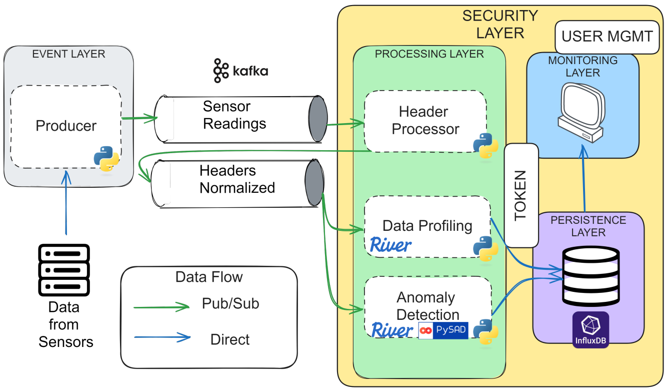


Fig. 2. Data flow between the different layers and components of DQAT.

implemented the PySAD and River methods, conducted a benchmark evaluation, and chose to employ two anomaly detection River methods: Half-Space Trees (HST) [26], which is a method used for anomaly detection, particularly in high-dimensional data, based on the same idea followed in isolation forests; and a stochastic implementation of the One-Class SVM (OCSVM) algorithm.

III. Proposed Framework

We addressed the data quality problem by developing the Data Quality Assurance Tool (DQAT), an open-source software framework designed for seamless integration with real-world IoT applications. DQAT leverages freely available tools like Kafka[4] for message brokering, InfluxDB[5] for time-series data storage, and Python's River and PySAD libraries for online machine learning. This enables DQAT to efficiently load data from event producers, normalize headers, apply real-time anomaly detection algorithms, perform data profiling, and store results in the database. The modular architecture of the tool facilitates integration with existing systems and its user-friendly dashboard provides a comprehensive view of data quality metrics.

A. Architecture

Considering the challenge of ensuring data quality inside the complex IoT scenario in near real time, we suggest an event-driven architecture [27], as shown in Fig. 1.

The layers are described as follows:

- **Event Layer:** This layer serves as the entry point for data into DQAT. It deals with events from various data sources, such as IoT sensors, databases, or message queues. The event layer uses event producers to publish events in a standardized format. The tool then utilizes Kafka topics to distribute these events to the appropriate processing components.
- **Processing Layer:** This layer houses the core data quality assessment logic, implementing River or PySAD libraries. It processes incoming events, performs headers normalization and check, applies anomaly detection algorithms, and updates the statistics to data profiling.
- **Integration Layer:** This layer suggests the interaction between DQAT and external systems, that has not been yet implemented. It could define data contracts, specifying the level of data quality exchanged between DQAT and other applications. Data contracts can be implemented using JSON or custom interfaces.
- **Persistence Layer:** This layer stores data profiles, anomaly detection results, and other relevant metadata generated by DQAT. It also handles the scheduling of actions related to the periodic check and sending alert events. The persistence layer uses InfluxDB.
- **Monitoring Layer:** This layer provides visibility into the data quality status of the system. It generates dashboards, reports, and alerts based on the data quality metrics calculated by DQAT. The monitoring layer uses InfluxDB's embedded dashboards. It also send alerts through log files.
- **Security Layer:** This layer ensures the security and integrity of DQAT and its data. It manages user authentication, authorization, and access control to different functionalities within the tool.

B. Data Flow

The DQAT data flow starts by reading raw sensor data from CSV files within the data folder, as shown, for example, in Fig. 2. The producer module emulates real-time data streams, processing the data (e.g., replacing commas, handling non-float values) and converting it into JSON format. Headers are dynamically inferred, and events are produced as Kafka topics for downstream processing.

The header processor module consumes events from the Kafka topic, ensuring consistent header formats across various data sources. It checks for the presence of expected headers informed previously in the configuration files, applies normalization where necessary, and logs errors for missing headers. The standardized data are then forwarded to another Kafka topic.

Two parallel processes operate on the standardized data stream:

[4] Apache Kafka is an open-source stream message processing platform aiming to delivery high throughput and low latency for mission-critical applications. https://kafka.apache.org

[5] InfluxDB is a time-series open-source database developed by InfluxData®most used in real time analysis. https://www.influxdata.com/get-influx/

- **Data Profiling:** The data profile module, leveraging the River library for online learning, continuously calculates rolling statistics (e.g., maximum, minimum, mean, variance) for each sensor parameter. These statistical profiles are stored in an InfluxDB time series database, facilitating real-time monitoring and analysis.
- **Anomaly Detection:** The anomaly detection module also uses River for online learning, implementing the HST and OCSVM algorithms for anomaly detection. These algorithms identify unusual patterns or outliers in the incoming data stream, identifying potential data quality issues. Detected anomalies are logged and stored in InfluxDB along with the data profiles.

InfluxDB serves as the central repository for both data profiles and anomaly detection results. This allows for efficient storage and retrieval of time series data, enabling historical analysis and the creation of informative dashboards. The DQAT modular architecture facilitates integration with other tools, allowing stakeholders to monitor data quality metrics and anomalies in real time.

IV. Results

A. Dataset

To assess the effectiveness of DQAT, we used a real-world data set[6] from a remotely monitored aquaponic fish pond water quality management system developed at the University of Nigeria, Nsukka. This dataset provided a rich source of labeled data from conventional and aquaponic catfish ponds, encompassing various water quality parameters and fish growth metrics.

The aquaponics dataset comprised sensor readings collected from June to mid-October 2021 in 12 catfish ponds. Each pond's IoT unit housed six sensors that measure temperature, turbidity, dissolved oxygen (DO), pH, ammonia, and nitrate levels. Data were collected at 5-second intervals, resulting in more than 170,000 instances per unit at the time of analysis. Additional attributes included the population, length, and weight of the fish in each pond.

B. Metrics

Addressing the results of the availability dimension presented in Fig. 3 reveals insights into the operational status of the IoT monitoring system deployed in various fish ponds (IoTPond1 - IoTPond11). IoTPond1, IoTPond2, IoTPond3, IoTPond4, IoTPond6, IoTPond8 and IoTPond9 exhibit near-complete data availability during the observed time frame. IoTPond10, IoTPond11, and IoTPond12 exhibit very low data availability, indicating potential intermittent failures or gaps in data collection during the period. This could be due to temporary sensor malfunctions, connectivity issues, or other factors that impact system performance. The interruption or failure in monitoring these ponds can be attributed to sensor damage, power outages, or other critical issues.

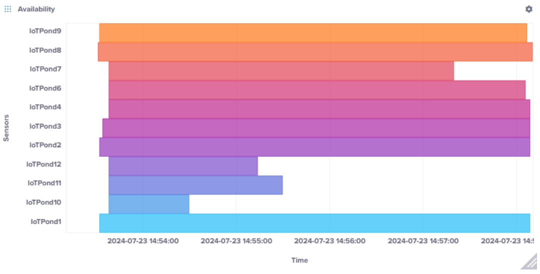


Fig. 3. Availability from InfluxDB dashboard.

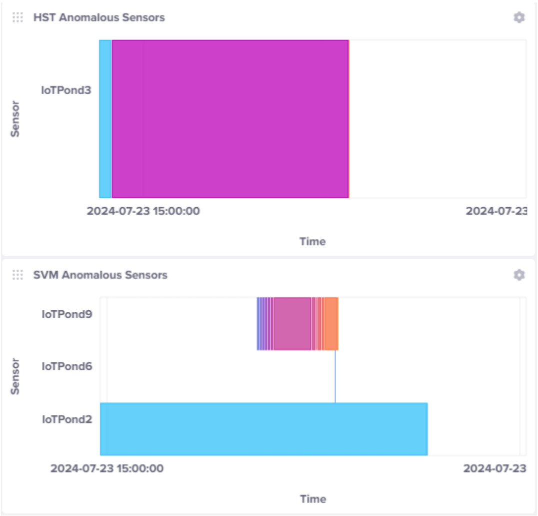


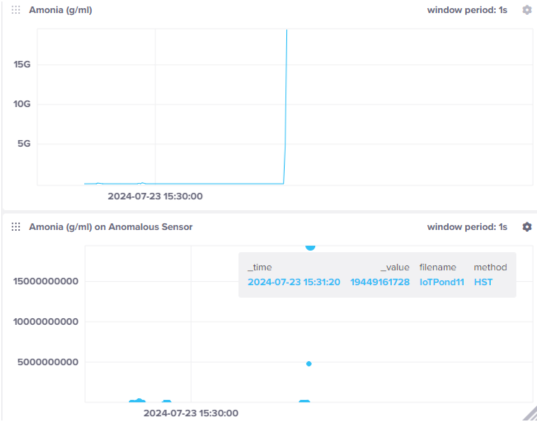
Fig. 4. Anomalous sensors detected by HST and OCSVM from InfluxDB dashboard.

Fig. 4 illustrates the results of anomaly detection in IoT sensors using two different algorithms: HST and OCSVM. At the time of simulation, according to the HST algorithm, only the IoTPond3 sensor exhibited anomalous behavior during this time frame. In contrast, the OCSVM algorithm identified anomalies in three sensors: IoTPond9, IoTPond6, and IoTPond2, with IoTPond2 showing a longer duration of anomalous behavior compared to the others. These findings demonstrate the potential for different algorithms to identify distinct sets of anomalies within the same dataset.

The time series data reveal an anomalous measurement of ammonia detected by the HST algorithm in the IoTPond11 sensor, as shown in Figure 5. IoTPond11 recorded an exceptionally high and inconsistent ammonia level exceeding $15x10^6$ g/ml, significantly deviating from the baseline measurements observed earlier in the time series. This abrupt and extreme increase in ammonia concentration suggests a possible anomaly in the sensor reading or an environmental event that affects ammonia levels. More research is needed to determine the cause of this unusual measurement.

DQAT leverages on-line learning algorithms that can continuously adapt to changing data patterns, improving the model's accuracy over time even without explicit labels. Although unsupervised anomaly detection offers several advantages, it also has limitations. Without labeled data, it can be challenging to assess the accuracy and precision of anomaly detection models. Furthermore, after a period of simulation, instances

[6] https://www.kaggle.com/datasets/ogbuokiriblessing/sensor-based-aquaponics-fish-pond-datasets



_time	_value	filename	method
2024-07-23 15:31:20	19449161728	IoTPond11	HST

Fig. 5. Highlight of sensor IoTPond11 with anomalous measurement of ammonia detected by HST.

of higher anomaly values tended to overshadow other less prominent anomalies. This phenomenon is likely attributed to the use of scaler preprocessors in both the HST and OCSVM algorithms.

V. Concluding Remarks

In conclusion, this study introduced DQAT, an open-source event-driven data quality assurance tool designed to address the challenges of real-time data monitoring in IoT systems. By focusing on critical data quality dimensions, DQAT enables proactive anomaly detection in complex, data-intensive environments.

Through the evaluation of a real-world dataset, we demonstrated DQAT's ability to identify anomalies in sensor readings, highlighting potential issues with data availability and accuracy. This showcases the potential of the tool to improve decision making. Although our initial experiments focused on a specific set of data quality dimensions and a particular dataset, DQAT's modular architecture and flexible design allow for its adaptation to diverse applications and a broader range of data quality concerns. Future iterations of DQAT can be improved by expanding the range of supported anomaly detection algorithms and incorporating advanced visualization techniques.

By providing a comprehensive, adaptable and open-source solution for data quality assurance, DQAT has the potential to empower stakeholders in various industries to harness the full potential of IoT data for informed decision-making.

Aknowledgements

This work was partially supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Financial Code 001, by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) - grant #2020/09838-0, and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - grant #311380/2021-2.

References

[1] J. Mckendrick, "A data gap continues to inhibit artificial intelligence," january 2023. Retrieved February 20, 2023, from https://www.forbes.com/sites/joemckendrick/2023/01/22/a-data-gap-continues-to-inhibit-artificial-intelligence.

[2] C. W. Fisher and B. R. Kingma, "Criticality of data quality as exemplified in two disasters," *Information Management*, vol. 39, no. 2, pp. 109–116, 2001.

[3] L. Francis and V. R. Prevosto, "Data and disaster: the role of data in the financial crisis," in *Casualty Actuarial Society E-Forum, Spring 2010*, vol. 62, 2010.

[4] R. Togneri, G. Camponogara, J.-P. Soininen, and C. Kamienski, "Foundations of data quality assurance for iot-based smart applications," in *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–6, 2019.

[5] M. L. Romero and R. Suyama, "Data quality in iot applications: A scoping review," in *2023 15th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 1368–1373, IEEE, 2023.

[6] H. Li, H. Lu, C. S. Jensen, B. Tang, and M. A. Cheema, "Spatial data quality in the internet of things: Management, exploitation, and prospects," *ACM Comput. Surv.*, vol. 55, feb 2022.

[7] O. Abdel Wahab, "Intrusion detection in the iot under data and concept drifts: Online deep learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19706–19716, 2022.

[8] I. Žliobaitė, M. Pechenizkiy, and J. Gama, *An Overview of Concept Drift Applications*, pp. 91–114. Cham: Springer International Publishing, 2016.

[9] L. Xu, X. Ding, H. Peng, D. Zhao, and X. Li, "Adtcd: An adaptive anomaly detection approach toward concept drift in iot," *IEEE internet of things journal*, vol. 10, no. 18, pp. 15931–15942, 2023.

[10] L. Xu, Z. Han, D. Zhao, X. Li, F. Yu, and C. Chen, "Addressing concept drift in iot anomaly detection: Drift detection, interpretation, and adaptation," *IEEE transactions on sustainable computing*, pp. 1–12, 2024.

[11] R. Chu, P. Jin, H. Qiao, and Q. Feng, "Intrusion detection in the iot data streams using concept drift localization," *AIMS mathematics*, vol. 9, no. 1, pp. 1535–1561, 2024.

[12] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdessalem, and A. Bifet, "River: machine learning for streaming data in python," *J. Mach. Learn. Res.*, vol. 22, jan 2021.

[13] S. F. Yilmaz and S. S. Kozat, "Pysad: A streaming anomaly detection framework in python," *arXiv preprint arXiv:2009.02572*, 2020.

[14] ISO, "ISO 8000-1:2022 data quality — part 1: Overview," 2022. Retrieved April 17, 2023, from https://www.iso.org/standard/81745.html.

[15] ISO, "ISO 25012-1:2008 software engineering — software product quality requirements and evaluation (SQuaRE) — data quality model," 2008. Retrieved April 17, 2023, from https://www.iso.org/standard/35736.html.

[16] DAMA, *DAMA-DMBOK (2nd Edition): Data Management Body of Knowledge*. Technics Publications, paperback ed., 2017.

[17] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel, "Data quality in internet of things: A state-of-the-art survey," *Journal of Network and Computer Applications*, vol. 73, pp. 57–81, 2016.

[18] L. Ehrlinger and W. Wöß, "A survey of data quality measurement and monitoring tools," *Frontiers in Big Data*, vol. 5, 2022.

[19] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[20] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, p. 93–104, may 2000.

[21] Y. Wang, J. Wong, and A. Miner, "Anomaly intrusion detection using one class svm," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, pp. 358–364, 2004.

[22] M. Ali, P. Scandurra, F. Moretti, and H. H. R. Sherazi, "Anomaly detection in public street lighting data using unsupervised clustering," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4524–4535, 2024.

[23] A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer, *Machine Learning for Data Streams: with Practical Examples in MOA*. The MIT Press, Mar. 2018.

[24] M. Halford, G. Bolmier, R. Sourty, R. Vaysse, and A. Zouitine, "creme, a Python library for online machine learning," 2019.

[25] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018.

[26] S. C. Tan, K. M. Ting, and F. T. Liu, "Fast anomaly detection for streaming data," in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011* (T. Walsh, ed.), pp. 1511–1516, IJCAI/AAAI, 2011.

[27] B. M. Michelson, "Event-driven architecture overview," *Patricia Seybold Group*, vol. 2, no. 12, pp. 10–1571, 2006.