Data Quality as a Service Framework for AI-Enabled Industrial Internet of Things

Arda Goknil
SINTEF
Oslo, Norway
arda.goknil@sintef.no

Phu Nguyen
SINTEF
Oslo, Norway
phu.nguyen@sintef.no

Erik Johannes Husom
SINTEF
Oslo, Norway
erik.johannes.husom@sintef.no

Sagar Sen
SINTEF
Oslo, Norway
sagar.sen@sintef.no

Simeon Tverdal
SINTEF
Oslo, Norway
simeon.tverdal@sintef.no

Flavien Peysson
PREDICT
Nancy, France
flavien.peysson@predict.fr

Dimitra Politaki
IRT SystemX
Paris, France
dimitra.politaki@irt-systemx.fr

Roberto González-Velázquez
TEKNIKER
Eibar, Spain
roberto.gonzalez@tekniker.es

Abstract

In the era of the Industrial Internet of Things (IIoT), the integration of Artificial Intelligence (AI) and Machine Learning (ML) into industrial processes has significantly enhanced operational efficiency and decision-making capabilities. However, the effectiveness of AI-enabled IIoT systems heavily depends on the quality of the underlying data. Ensuring data integrity, accuracy, and reliability in IIoT is challenging due to the heterogeneous nature of data sources and the dynamic operational conditions. Existing approaches often focus on individual data quality techniques, lacking a unified and comprehensive framework. To address this need, we propose Data Quality as a Service (DQaaS), a novel framework designed to provide scalable and reusable data quality solutions as services for AI-enabled IIoT applications. DQaaS encompasses a suite of modular services for data monitoring and repair, all orchestrated through a centralized platform. This framework leverages advanced ML algorithms and state-of-the-art processing techniques to ensure robust data quality management across the edge-cloud continuum.

Keywords

Industrial Internet of Things, Data Quality as a Service, Edge-Cloud Continuum, Data Quality Management

ACM Reference Format:
Arda Goknil, Phu Nguyen, Erik Johannes Husom, Sagar Sen, Simeon Tverdal, Flavien Peysson, Dimitra Politaki, and Roberto González-Velázquez. 2024. Data Quality as a Service Framework for AI-Enabled Industrial Internet of Things. In *14th International Conference on the Internet of Things (IoT 2024), November 19–22, 2024, Oulu, Finland*. ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3703790.3703794

IoT 2024, November 19–22, 2024, Oulu, Finland

ACM ISBN 979-8-4007-1285-2/24/11
https://doi.org/10.1145/3703790.3703794

1 Introduction

The Industrial Internet of Things (IIoT) transforms industries by enhancing how products are manufactured, improved, and distributed. Companies integrate Artificial Intelligence (AI) and Machine Learning (ML) into their operations to exploit the extensive data generated by IIoT systems. This integration boosts decision-making, optimizes production lines, and enables predictive maintenance by forecasting tool lifespan [5, 61]. However, the effectiveness of AI-enabled IIoT systems is hindered by data quality and consistency issues. Neglecting data quality leads to unstructured, untagged "dark data" [10] and biases [1]. Addressing these challenges requires comprehensive data quality management techniques to improve and maintain data quality in various IIoT scenarios [18, 23, 52, 53, 55].

Addressing data quality problems spans diverse research fields, each offering various interpretations and solutions. In relational databases, it involves data normalization [13], while in signal processing, it addresses signal-to-noise ratios. Data scientists have developed numerous techniques to enhance data quality before applying ML pipelines. Despite the importance of data quality in IIoT, improving it remains challenging due to two main reasons. First, sensor measurements in IIoT are often corrupted or missing due to factors like electromagnetic interference, packet loss, or signal processing faults. Second, IIoT data travels along the edge-cloud continuum, making it susceptible to various quality issues. This journey includes data collection by sensors, processing by programmable logic controllers (PLCs), transfer to edge devices via industrial protocols [8, 43], and transmission to the cloud via API protocols [16, 36, 59]. Consequently, IIoT systems must address erroneous values, missing values, noise, and data drift while ensuring data continuity throughout the edge-cloud continuum.

Effective data quality management requires strategies for both *online* (real-time) and *offline* (historical datasets in the cloud) contexts. This necessitates configurable data quality services. Our research aims to develop independent, adaptable data quality services to meet the evolving needs of AI-enabled IIoT systems.

Extensive research has been conducted on data quality techniques for IIoT systems [18], focusing on various aspects such as data monitoring, cleaning, and repair. Notable contributions include methods for repairing data using sensory substitution [49], addressing data corruption through computational dependencies [31], and utilizing clustering techniques for outlier detection [11, 57]. Other approaches employ noise filters [62] and sampling techniques [29] to enhance data robustness, and ML pipelines [22, 24] to identify and validate process behavior patterns. Despite these advancements, existing solutions are often isolated and lack a unified framework that coordinates these techniques as composable and reusable services. Moreover, there is a significant gap in the design and implementation of data quality techniques that can operate seamlessly across the edge-cloud continuum.

In this paper, we propose and implement Data Quality as a Service (DQaaS), a novel framework offering data quality solutions for AI-enabled IIoT applications. DQaaS addresses the complex data quality challenges of IIoT by providing scalable, on-demand services to ensure data integrity, accuracy, and reliability. It integrates advanced ML techniques to deliver a comprehensive range of services, from real-time anomaly detection to historical data validation and repair, accessible via a unified cloud-based platform. We explore both the theoretical foundations and practical implementations of DQaaS, demonstrating its effectiveness in improving data quality management within DQaaS. The data quality tasks in DQaaS are designed with modularity at their core, enabling the creation of encapsulated functionalities that function as distinct yet integrated services. These services can operate independently or collaboratively, tailored to meet diverse user requirements.

The DQaaS framework currently offers five critical data quality services, each designed to address specific data quality challenges in AI-enabled IIoT applications. The Generic Data Monitoring Service provides real-time evaluation of data integrity across various industrial environments. The Machine-tool Condition Data Monitoring Service focuses on monitoring the health and performance of machine tools, enhancing Overall Equipment Effectiveness (OEE) through predictive maintenance. The Anomaly Detection Data Monitoring Service leverages advanced models to identify point and collective anomalies in manufacturing data, preventing unexpected failures. The Generic Data Repair Service employs machine learning techniques to automatically repair erroneous sensor data, ensuring the continuity and accuracy of data-driven predictions. Finally, the Historical Data Quality Validation Service formalizes machine tool concepts and utilizes a semi-supervised pattern recognition approach for validation of historical manufacturing data.

2 Background: Data Quality for IIoT

Data quality is defined in ISO/IEC 25012:2008 [25] as *a degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions.* **Data quality metrics** are the measurements by which you assess your data. They benchmark how complete, valid, accurate, timely, and consistent the data is and help differentiate between high- and low-quality data. They can be obtained from **data quality dimensions**, i.e., the measurement attributes of data, which we can assess and improve. Data accuracy and completeness are two data quality dimensions addressed by quality metrics. **Data completeness** refers to the degree to which all parts of the data are given with no missing information [60]; **data accuracy** is the degree of similarity of a measured quantity to its actual value.

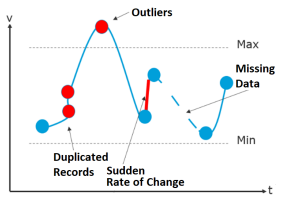

Figure 1: Example data quality problems.

Data quality requirements describe the needs or conditions that high-quality data should meet. They are checked on the input data to compute the corresponding metrics. The data quality requirement violation indicates a data quality problem/issue. Figure 1 shows some data quality problems on time-series data. **Missing data** refers to cases when a variable or attribute has no value. **Outliers** are extreme values that deviate from other observations of data. **Duplicated records** are two or more adjacent data points in the same timestamp. A **sudden rate of change** refers to cases where a variable changes unrealistically over a period of time.

Data quality management techniques (in short, data quality techniques) improve and maintain data quality across system components. There are three types of data quality techniques:

- ***Data Monitoring*:** Data is monitored to check **data quality requirements** for detecting quality issues such as outliers.
- ***Data Cleaning*:** It entails the removal of corrupt and unusable data, e.g., those affected by environmental noise or extreme operating conditions such as high temperature.
- ***Data Repair*:** It restores data that has been lost, accidentally deleted, corrupted, or made inaccessible, e.g., by using simulation data or data from redundant sources (other sensors).

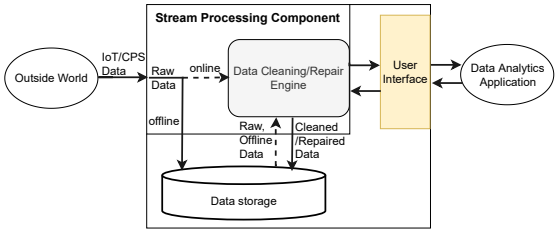

Figure 2: Architecture for data quality systems [28, 50].

Data quality techniques can be **online** (real-time) and **offline** (for large historical datasets). Karkouch et al. [28] adopt an architecture (see Figure 2) proposed by Sathe et al. [50] to depict the distinction between online and offline techniques (IoT layers edge and cloud are not explicit in the architecture).

3 Related Work

Data quality techniques involve various approaches, technologies, and tools to detect and rectify data quality issues. While some literature groups data cleaning and repair together, we differentiate between them due to their distinct treatments of these issues (as discussed in Section 2). Data monitoring, which identifies quality issues, is a prerequisite and integral part of both data repair and cleaning. Notably, existing data cleaning and repair methods inherently support data monitoring.

Significant research has focused on developing data quality techniques for IIoT data, including monitoring, cleaning, and repair [18].

For instance, Russel et al. [49] propose repairing camera-sensed data using raw data from ambient sensors through sensory substitution to enhance robustness and dependability. Lin et al. [31] address data corruption by replaying dependent computations in a distributed IoT environment to fix degradation caused by hardware malfunctions, software bugs, or network issues. Syafrudin et al. [57] and Corrales et al. [11] use Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [14] for outlier detection. Yu et al. [62] apply noise filters based on computation unit distances to eliminate long- and short-duration noise. Kong et al. [29] use sampling techniques to reduce noise in datasets used for state prediction in machine tools. Johannes et al. [24] introduce an unsupervised machine learning pipeline, UDAVA, to identify and validate recurring process behavior patterns in sensor data. While these methods address specific data quality issues, they do not design and implement data quality techniques as reusable services to meet the evolving challenges of IIoT systems over time.

Despite extensive research, limited attention has been given to developing a comprehensive solution that coordinates multiple data quality techniques as services for IIoT data. Sen et al. [54] address this gap with a decentralized edge-cloud AI pipeline architecture supporting two ML-based data quality pipelines. This architecture is an initial step toward a generic solution utilizing data quality services. However, it does not introduce or implement composable data quality services to meet the unique requirements of IIoT data. To complement this architecture, Tverdal et al. [58] implement edge-based data monitoring and repair as a service for IoT. However, this edge-based service does not fully integrate with other data quality services needed for comprehensive IIoT data management.

Distinct from the approaches given above, DQaaS presents a unified and flexible framework for managing data quality in IIoT systems. Unlike existing studies focusing on isolated techniques, DQaaS provides a comprehensive suite of composable and reusable services tailored to address the diverse and evolving data quality challenges in IIoT environments. Our framework encompasses robust services for data monitoring, cleaning, and repair, all orchestrated through a centralized platform. This orchestration ensures seamless integration and coordination of data quality processes, significantly enhancing overall data reliability and operational efficiency. Furthermore, DQaaS is designed to support scalability and adaptability, making it adept at managing the dynamic and complex nature of IIoT systems over time.

4 DQaaS Approach

In this section, we introduce Data Quality as a Service (DQaaS), a framework designed to provide data quality solutions as online services tailored for IIoT applications. DQaaS offers reusable services that simplify the complexities of traditional data reliability procedures. Figure 3 illustrates the DQaaS architecture, which integrates IIoT with data quality services via edge and cloud computing. Built on cloud-native, services-based architecture principles [3], DQaaS incorporates key design paradigms such as the API Gateway/Backend for Frontend (BFF) pattern, inspired by the reference architecture of Microsoft.Net [37]. The architecture includes client and cloud-based server layers, as well as IIoT and Edge layers, which supply live data streams to the backend of the data quality services.

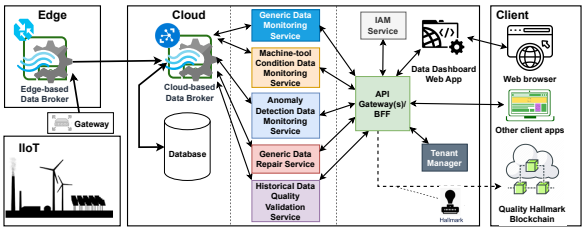

Figure 3: Cloud-based Data quality as a service architecture.

4.1 Client Layer

End-users can access DQaaS data quality services through various client applications, e.g., web browsers, mobile apps, and desktop applications, via the API Gateway. These applications provide robust user interfaces, including web-based dashboards, for monitoring and managing data quality within IIoT systems. Web browsers offer the most common access through standard web technologies, enhancing user engagement and control. In addition, mobile and desktop apps, including Single Page Applications (SPAs), connect seamlessly to DQaaS, offering specialized functionalities and expanding the accessibility and versatility of the data quality services.

The Quality Hallmark Blockchain in the Client layer ensures IIoT data traceability, trust, and security across industrial and inter-organizational supply chains. Acting as a trusted node, this blockchain-based system records, stores, and verifies data quality metrics, certifications, and transactions immutably. Leveraging blockchain technology, it provides a transparent, tamper-proof mechanism for maintaining trustworthy records of data quality hallmarks (i.e., data quality metrics logged by running the DQaaS services on the data) by the DQaaS framework. Access to the Quality Hallmark Blockchain is optional for each tenant.

4.2 Cloud Layer

The Cloud layer enables end-users to access data quality services via the Data Dashboard Web App, typically through web browsers (see Figure 3). We do not cover other client apps in this paper. The Data Dashboard Web App can be built with any modern web technology (e.g., ASP.Net Core [7]) and interacts with data quality services through APIs provided by the API Gateway, using access tokens from the Identity and Access Management (IAM) service.

Figure 4 illustrates the interactions between clients and the cloud-based server in the DQaaS framework, highlighting the flow of requests and responses to ensure secure and efficient access to data quality services. The end-user initiates a request from a web browser, which is directed to the Web App and then forwarded to the API Gateway. The API Gateway sends the request to the IAM service for authentication, which verifies credentials and issues access tokens. These tokens are validated by the Tenant Manager to confirm the tenant's subscription and authorization. The request (along with tokens) is forwarded to the data quality services, which query the Cloud-based Data Broker for the required data. The Data Broker processes and returns the data to the data quality services, which send the results back through the API Gateway to the Web App and finally to the end-user. This sequence ensures secure, authenticated access, enabling efficient IIoT data management through a robust cloud-centric framework.

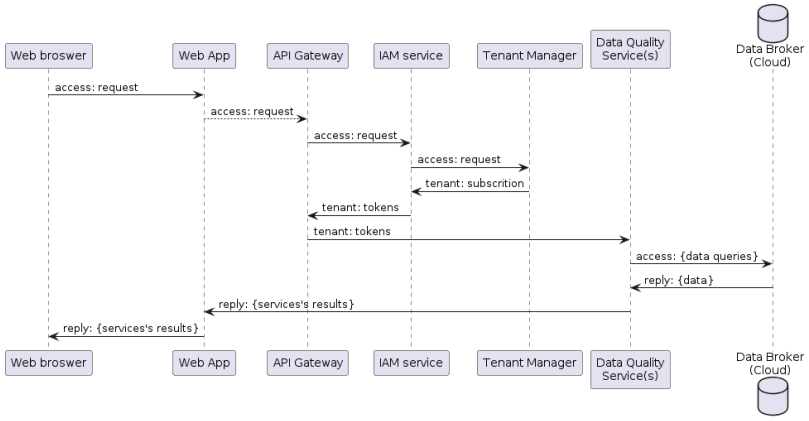

Figure 4: A simplified sequence diagram of client requests and the Cloud-based server responses.

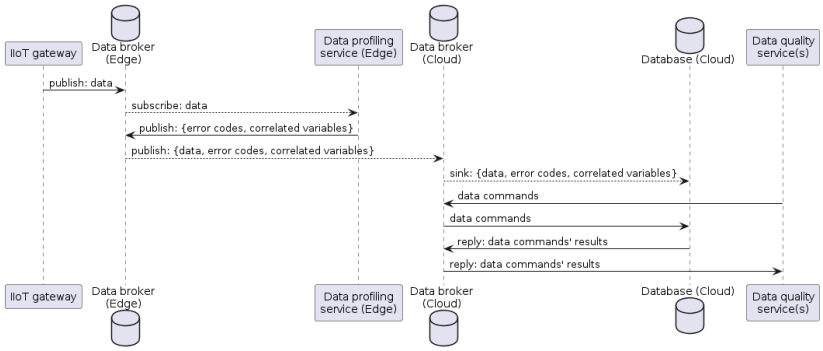

Figure 5: A simplified sequence diagram of data flows from IIoT to Edge to Cloud.

The API Gateway is the entry point for clients to access DQaaS services, managing interactions, rate limiting, analytics, and the Data Quality Hallmark. It enhances security, performance, and scalability by centralizing authentication and authorization with the IAM service and aggregating API usage analytics. It adapts and filters messages, optimizes performance by caching frequent requests, and abstracts backend complexities, providing a simplified, secure interface for clients. This approach leverages the cloud-based software-as-a-service (SaaS) model, ensuring a robust and efficient cloud service infrastructure.

The IAM service manages permissions and access control, ensuring secure interactions within DQaaS. It handles authentication, authorization, user management, access control, single sign-on, identity management, and compliance reporting. Working with the API Gateway and microservices, the service ensures secure and compliant interactions with the system. Clients must authenticate via the IAM Service as an OpenID Connect [42] or OAuth 2.0 Identity provider [41] before accessing data quality services. This approach uses bearer tokens for fine-grained access control, securing interactions from web apps, SPAs, and mobile apps, and safeguarding sensitive IIoT data throughout the edge-cloud continuum.

The Tenant Manager manages end-user subscriptions to backend data quality services, allowing each tenant to subscribe based on their needs and payment. Together with the IAM service, the Tenant Manager enables multi-tenancy on the DQaaS platform, offering customized services for different tenants [39, 40, 56]. It enforces authorization and access control for end-users of each tenant and manages service administration and monitoring at runtime. Frequent interaction with the IAM service suggests that the Tenant Manager could be a local component within IAM, but it is more effective as a standalone microservice in the cloud. This standalone implementation, accessible via the API Gateway, allows for runtime scaling to enhance performance.

The Cloud layer's backend includes **the Cloud-based Data Broker**, which stores historical time-series data and supports live streams for the online services. The broker enables two-way data synchronization between the Edge, IIoT layers, and specialized data quality services. DQaaS currently offers five data quality services:

- **Generic Data Monitoring Service:** This service offers a generic solution for real-time data quality monitoring across IIoT environments. It continuously evaluates incoming data against predefined metrics like completeness, accuracy, and consistency, identifying anomalies and errors to ensure data integrity throughout the data lifecycle.
- **Machine-tool Condition Data Monitoring Service:** This service monitors machine-tool health and performance in real-time within industrial settings, enhancing Overall Equipment Effectiveness (OEE) [12]. It operates in both online and offline modes, supporting Condition-Based Maintenance (CBM) by continuously assessing data to detect degradation indicators and data quality anomalies.

- **Anomaly Detection Data Monitoring Service:** This service identifies point and collective anomalies in manufacturing time-series data to prevent failures and downtime. Using prediction-based and reconstruction-based models, it detects anomalies through four methods: out-of-limit, proximity-based, prediction-based, and reconstruction-based detection. This enhances manufacturing reliability by anticipating and addressing unusual data behaviors.
- **Generic Data Repair Service:** This service automates the repair of erroneous sensor data to ensure continuity and accuracy in AI-enabled IIoT applications. Using ML techniques, it learns sensor correlations to predict missing values and replace corrupted data. Integrated into an ML pipeline, it leverages redundant sensor inputs to determine the most accurate values, enhancing data-driven predictions' reliability.
- **Historical Data Quality Validation Service:** This service validates historical manufacturing data, especially for machine tools. It formalizes concepts and establishes business rules for automated data analysis pipelines. Using a semi-supervised Dynamic Time Warping (DTW) [35] approach, it recognizes patterns with minimal labeled data, ensuring accurate validation and enhancing the reliability of data-driven insights in manufacturing.

Each service can validate industrial data and generate data quality hallmarks that can be shared via the Trusted Framework (Quality Hallmark Blockchain) [26]. More specifically, each service can include a special rule called *QualityHallmarkDoc*, which supports publishing the Quality Hallmark Document (QHD) to the Quality Hallmark Blockchain. The QHD, provided in JSON format, contains metadata and data quality metrics from various services. This QHD sharing is optional and only for tenants with blockchain settings.

DQaaS uses Docker technology to streamline data quality service distribution. Docker images package data quality rules and their execution environment for easy, license-free creation and distribution. These images can be shared and version-controlled on platforms like GitLab, enhancing accessibility and collaboration. This approach ensures a secure, efficient environment and promotes a shared ecosystem of tools and practices for diverse stakeholders.

4.3 Edge Layer

The Edge Layer, situated between the IIoT and Cloud Layers, processes data near its source, enabling low-latency, optimized bandwidth, and efficient management. It includes robust Edge-based Data Brokers that can handle data profiling, cleaning, compression, protocol translation, and preliminary analytics [58]. These brokers can perform initial data quality checks before sending data to the Cloud Layer for advanced processing.

4.4 IIoT Layer

The IIoT layer in the DQaaS architecture is the source of industrial data, bridging physical and digital interactions. It includes smart devices, sensors, and actuators embedded in industrial machinery that monitor, collect, and relay operational data. The Gateway connects IIoT devices to processing layers, performing data aggregation, initial processing (e.g., compression), protocol translation, and connectivity management. Although the Edge layer is optional,

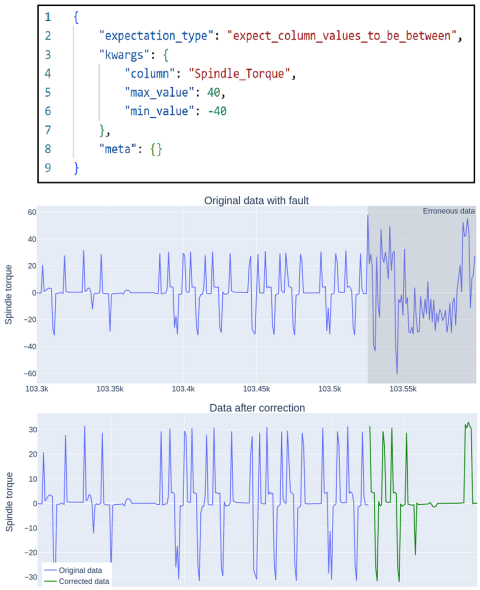

Figure 6: Spindle torque sensor data monitoring and correction in an IIoT environment. The JSON body in Great Expectations defines the expected data range for the sensor. The middle graph shows the original time series data, with erroneous measurements highlighted in grey due to environmental interference. The bottom graph displays the time series after erroneous data repair.

the Gateway can directly publish IIoT data to the Cloud layer via the Cloud-based Data Broker, which supports historical data access and online data streaming to data quality services, either directly or through the Edge layer. Figure 5 illustrates the data flow sequence from IIoT gateways through the Edge to the Cloud layer within the DQaaS framework. The IIoT gateway publishes raw data to the Edge Data Broker, which subscribes to and processes it. The Edge Data Profiling Service generates error codes and identifies correlated variables, and then the Edge Data Broker publishes this enriched data. The Cloud Data Broker subscribes to and transfers the enriched data to the Cloud Database. The Cloud Data Profiling Service processes the data further, responding to data commands from data quality services. These services query the Cloud Database, which processes commands and returns results.

5 Generic Data Monitoring Service

The service provides metrics to assess and improve data quality in IoT and CPS, ensuring data is fit for specific purposes. These metrics, aligned with data quality dimensions, help certify data sources. Despite their extensive study, these metrics are underutilized in IIoT, leading to the accumulation of dark data—unstructured, untagged, and unanalyzed. Prompt computation and feedback of these metrics can prevent data from becoming dark, enhance audibility, and encourage acquiring high-quality data for ML/AI products.

In DQaaS, data quality metrics are implemented using the Great Expectations (GE) library [20], a widely used Python library for data validation, documentation, and profiling. GE allows us to define Expectations—statements that describe verifiable data properties like missing data, duplicates, and value ranges. These Expectations

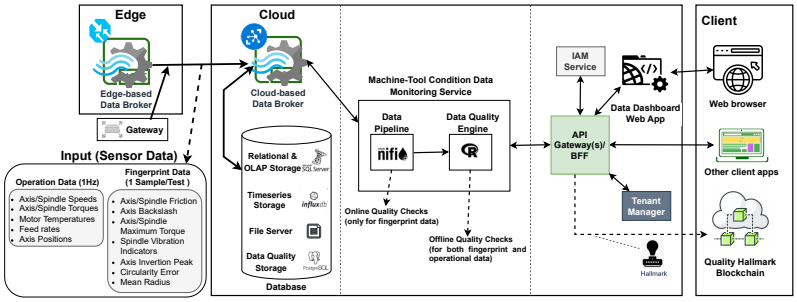

Figure 7: Deployment of the machine-tool condition data monitoring service.

evaluate data from various perspectives to ensure it meets quality standards. The service includes two categories of data quality rules:

Standard Rules: Utilizing common Expectations from GE to address typical data quality issues with mostly fixed input parameters, such as chi-square test, existence check, valid DateTime, valid type, Kolmogorov-Smirnov test, quantile ranges, and missing values.

Custom Rules: Developing domain-specific Expectations using GE templates to verify project-specific data. These rules allow user-defined parameters, including correlation coefficient, existence check, max value, value kurtosis, noise, skewness, standard deviation, duplicate records, missing records, and six sigma deviation.

Figure 6 demonstrates the generic data monitoring service's functionality in a manufacturing environment, focusing on spindle torque sensor data. The top panel displays a JSON configuration in Great Expectations, setting the expected data range (-40 to 40). The middle graph shows the original time series data with erroneous measurements highlighted in grey, indicating deviations caused by environmental interference.

6 Machine-tool Condition Data Monitoring

This service focuses on machine-tool condition monitoring [4], enhancing Overall Equipment Effectiveness (OEE) [12] in manufacturing. It operates in both online and offline modes (see Section 2) to prevent unexpected production interruptions due to equipment failures, aligning with the zero-defect manufacturing paradigm [2]. Initial performance validation tests are conducted during setup and are rarely repeated. A Condition-Based Maintenance (CBM) strategy monitors machine condition and detects degradation indicators timely [15]. Our service identifies data quality anomalies, improving overall maintenance. It involves three stages:

Stage 1: Data Acquisition. Machine data is gathered through built-in sensors and external systems, capturing variables like position, speed, vibration, and temperature. Monitoring software on an edge device collects data from the machine's PLC at 50Hz during CNC program execution. A rapid test known as *Fingerprint* [6] assesses machine performance without production interference.

Stage 2: Fingerprint Test Cycle. A predefined sequence of movements is executed to test machine axes and spindles quickly and automatically, minimizing productivity impact. Data collected is processed and uploaded to a cloud-based maintenance platform. The test uses a specific CNC program customized for each machine tool configuration and controller.

Stage 3: KPI Generation and Condition Monitoring. Acquired data is post-processed to calculate Fingerprint KPIs using a digital twin of the machine's components. These KPIs diagnose potential machine condition issues. Regular production data is also processed to generate usage metrics, providing a comprehensive assessment of the machine's health.

The service utilizes eleven distinct metrics (completeness, completeness by observation, completeness by variables, time uniqueness, range, consistency, typicality, moderation, timeliness, name, and format), categorized into five data quality dimensions (completeness, uniqueness, accuracy, timeliness, and conformity), to evaluate data quality [34]. These metrics produce values ranging from 0 to 1, where 0 represents poor data quality and 1 represents excellent quality. Figure 7 illustrates the deployment of the service for both online and offline modes within DQaaS.

The process begins with the collection of operation and fingerprint data from machine tools, transmitted via an edge device to the Cloud-based data broker and then to the Apache NiFi data pipeline. Online quality checks are performed on fingerprint data within the pipeline before storage in repositories, including relational (Microsoft SQL Server), time-series (InfluxDB), and file servers. The data quality engine conducts offline quality checks on both operational and fingerprint data, with results stored in a PostgreSQL database. These results are accessible through the API gateway, which interacts with the IAM service for secure access. They are presented to end-users via a web browser or other client apps.

Online monitoring starts when new fingerprint data arrives, using two of eleven metrics (completeness and accuracy) in the Apache NiFi data pipeline. Completeness ensures all expected Fingerprint KPIs are present, while accuracy identifies outliers by comparing each KPI to a predefined range (mean ± five times the standard deviation). If either metric fails, the data is not stored, and a notification prompts a retest. Offline monitoring occurs at scheduled intervals, using the data quality engine to analyze archived data with all eleven metrics, generating a comprehensive quality index through arithmetic or weighted averaging.

7 Anomaly Detection Data Monitoring Service

Manufacturing systems often face anomalies that lead to unexpected failures, increased downtime, diminished product quality, and economic loss. The vast amount of time-series data generated necessitates effective anomaly detection techniques to anticipate and prevent breakdowns. This service uses prediction-based and reconstruction-based models to identify point and collective anomalies in manufacturing datasets, defined as unusual behaviors at specific time points or periods. The service focuses on detecting:

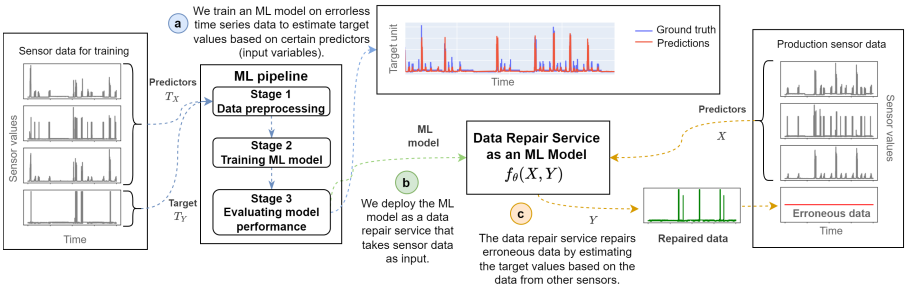

Figure 8: Machine learning pipeline to produce models as an erroneous data repair service.

point anomalies (a single data point significantly deviates from the norm) and **collective anomalies** (a sequence of data points is anomalous, even if individual points appear normal). The service uses four detection methods:

- **Out-of-limit-based detection:** Points are classified as anomalous based on predefined rules.
- **Proximity-based detection:** Points are classified as anomalous based on their unusual distance from other points.
- **Prediction-based detection:** Points are considered anomalous if their actual value significantly deviates from predicted values based on past data.
- **Reconstruction-based detection:** Points are considered anomalous if their actual value deviates significantly from reconstructed values after down-sampling and up-sampling.

State-of-the-art anomaly detection techniques for manufacturing uses statistical prediction and shallow-learning techniques [27, 32]. However, recent work has shown that deep learning methods, such as Generative Adversarial Networks (GANs) [17], outperform "traditional" statistical methods in handling non-linearity in complex temporal correlations [45]. Based on this, the service includes:

Stage 1: Point Anomaly Detection. The first stage of the service focuses on point anomaly detection for two main reasons. First, a single anomaly in a time series can indicate a process defect, making prompt detection crucial. Second, analyzing occurrence, characteristics, and frequency of point anomalies can help predict larger, collective anomalies. To detect point anomalies, the service employs ARIMA [38], a baseline model well-suited for this purpose.

Stage 2: Collective Anomaly Detection. This stage utilizes an ensemble approach with multiple models to increase precision and recall. The models selected are (i) ARIMA [38] (a baseline, computationally efficient forecasting model), (ii) LSTM with dynamic threshold [21] (sets anomaly thresholds dynamically using historical error values), (iii) LSTM autoencoder [33] (detects anomalies by reducing and reconstructing data dimensionality, suited for local data patterns), (iv) Dense-based autoencoder [33] (computationally cheaper, learns global data patterns, and reduces data dimensionality), and (v) GAN (TadGAN) [9] (combines an LSTM-autoencoder generator with an LSTM-based anomaly classifier).

8 Generic Data Repair Service

This service automates repairing erroneous sensor data, ensuring the continuity and accuracy of data-driven predictions in AI-enabled IIoT applications. Using ML techniques, it learns correlations among sensors to predict missing values and replace corrupted data. Integrated into an ML pipeline, it leverages inputs from redundant sensors to determine the most representative values for data repair, enhancing the reliability of data used in IIoT applications.

Figure 8 presents the ML pipeline for producing and deploying ML models as an erroneous data repair service. The pipeline trains the ML model $f_\theta(X, Y)$, optimizing parameters θ to find the most accurate predictions. Input includes multivariate time series data T_X from candidate sensors C and univariate time series data T_Y from a target sensor s_i. The model learns the relationship between T_X and T_Y to predict target sensor values. High-quality training data, characterized by completeness, accuracy, timeliness, and validity, is essential for effective model training. These datasets are ideally fault-free and obtained from successful production cycles. The learning process involves selecting model parameters θ, such as the model type (DNN/FCNN, CNN, LSTM [19]), *window sizes*, and data split percentages for training and evaluation. The pipeline input goes through three stages:

Stage 1: Data pre-processing. The input data undergoes several processing stages for neural network use: *data profiling*, *cleaning*, *feature engineering*, *data scaling*, and *splitting into training/test sets and sub-sequences*. Data profiling computes *the maximum information coefficient* [48] and *Pearson's correlation coefficient* [51] to identify sensor correlations. Statistical metrics highlight zeros or missing values, guiding data cleaning to remove constant or null columns. Feature engineering extracts statistical properties from raw data, ensuring noise invariance and efficient classification. The data is then split into training and test sets, with the training set used for model training and hyper-parameter tuning, and the test set kept isolated for unbiased evaluation.

Datasets with varying sensor measurements are scaled [30] for comparable influence during training. Both training (including validation) and test datasets are restructured into input and output sub-sequences based on a specified window size, as predictions depend on time-varying observations from input sensors and the desired window of output values.

Stage 2: Training ML model. Input and output sub-sequences, tailored to the desired input and output *window sizes* determined in Stage 1, are used to train the ML model, allowing flexibility in learning parameters and model types, including DNN/FCNN, CNN, and LSTM. Before training, 20% of the training data is set aside for validation. During training, the pipeline monitors prediction error on the validation set, stopping if no improvement occurs to prevent overfitting. The trained model is then saved for evaluation.

Stage 3: Evaluating model performance. The pipeline uses the test dataset to evaluate the model's performance and detect any

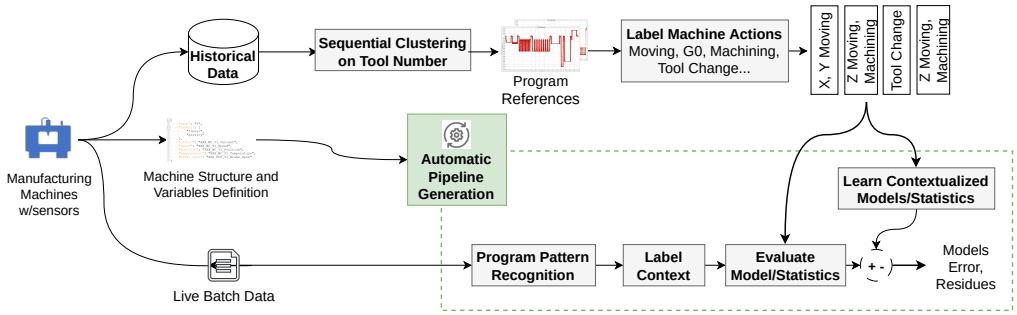

Figure 9: Overview of the historical data validation service.

bias from hyper-parameter tuning in Stage 2. By comparing the model's output with ground truth values, we assess its prediction accuracy. The pipeline generates visual plots of predictions and employs metrics such as Mean Squared Error (MSE), R^2 score, and Mean Absolute Percentage Error (MAPE) to quantify performance.

The ML model becomes a service via the API Gateway in Figure 3. The API processes time series data from input sensors and returns target sensor values with timestamps. Since the model is trained on scaled, feature-engineered data, it cannot directly use raw input sequences. Therefore, the feature engineering steps, scaler, and model, along with necessary ML libraries (e.g., Scikit-learn [46] and Pandas [44]), are encapsulated in a Docker container for inference.

The generic data repair service works seamlessly with the generic data monitoring service to ensure data integrity in IIoT environments. As shown in Figure 6, the monitoring service identifies erroneous measurements that deviate from acceptable ranges. Upon detection, the repair service corrects the faulty data. Together, these services maintain accurate and reliable sensor data, essential for effective data-driven decision-making in IIoT. The bottom graph in Figure 6 illustrates the corrected data (in green) within the expected range, demonstrating the effectiveness of this collaboration.

9 Historical Data Validation Service

In numerical machining, computational programs perform repetitive tasks in cyclical patterns, with each component undergoing the same sequence of operations. These include sub-patterns, like varying spindle speeds, reflecting machine acceleration or deceleration. Analyzing these patterns provides a robust assessment of machine dynamics under consistent conditions. The service uses supervised ML to identify predefined patterns, requiring a labeled training set for classification.

Figure 9 illustrates the service for validating historical data, leveraging machining industry expertise. The approach has two primary workflows. **Offline Workflow** depicted in the upper part of Figure 9 utilizes historical data to create a contextualized machining model. This stage integrates AI algorithms and an automated pipeline based on a knowledge base that includes physical laws and production rules. **Online Workflow** given in the lower part of Figure 9 applies the offline-constructed model to real-time data to assess the quality metrics of historical data.

In offline workflow, historical data is clustered based on tool numbers, generating program references that categorize the operation sequences (Sequential Clustering on Tool Number). A unique reference for each cluster is constructed using an averaging algorithm to represent the average pattern for a set of reference sequences. The Dynamic Time Warping Barycenter Averaging (DBA) method [47] refines an initial sequence to minimize its squared DTW distance, reducing inertia until all barycenters are calculated. Machine actions are labeled using supervised ML (Label Machine Actions), identifying different states such as X-Y Moving, Z Moving, Machining, and Tool Change. By using labeled data, the service learns contextualized models and statistics that describe typical machine behavior (Learn Contextualized Models/Statistics). Errors and residuals are calculated to refine the models, ensuring accuracy (Models Error, Residues).

In online workflow, live batch data is structured and variable definitions are established, setting the stage for real-time analysis (Machine Structure and Variables Definition). An automated pipeline is generated to process live data, integrating knowledge from the offline workflow (Automatic Pipeline Generation). The pipeline recognizes patterns in the live data based on established program references (Program Pattern Recognition). It labels the identified contexts in real-time, enabling immediate classification of machine actions (Label Contexts). It continuously evaluates the models and statistics against the live data, ensuring the accuracy and relevance of the validation process (Evaluation of Models/Statistics).

10 Conclusions

In this paper, we have introduced Data Quality as a Service (DQaaS), a novel framework designed to address the diverse and evolving data quality challenges inherent in AI-enabled IIoT environments. DQaaS integrates modular, reusable data quality services, leveraging advanced machine learning (ML) algorithms and state-of-the-art processing techniques. This comprehensive framework ensures robust data quality management across the edge-cloud continuum, enabling secure, efficient, and reliable data handling from IIoT devices to cloud-based services.

Acknowledgments

The work has been conducted as part of the ENFIELD project (101120657) and the InterQ project (958357) funded by the European Commission within the HEU Programme and the H2020 Programme.

References

[1] Shahriar Akter, Grace McCarthy, Shahriar Sajib, Katina Michael, Yogesh K Dwivedi, John D'Ambra, and KN Shen. 2021. Algorithmic bias in data-driven

innovation in the age of AI. *International Journal of Information Management* 60 (2021), 102387.
[2] Michele Albano and Urko Zurutuza. 2022. *The MANTIS book: cyber physical system based proactive collaborative maintenance.* CRC Press.
[3] Nuha Alshuqayran, Nour Ali, and Roger Evans. 2016. A systematic mapping study in microservice architecture. In *2016 IEEE 9th international conference on service-oriented computing and applications (SOCA).* IEEE, 44–51.
[4] Nitin Ambhore, Dinesh Kamble, Satish Chinchanikar, and Vishal Wayal. 2015. Tool condition monitoring system: A review. *Materials Today: Proceedings* 2, 4-5 (2015), 3419–3428.
[5] Mihai Andronie, George Lăzăroiu, Mariana Iatagan, Cristian Uță, Roxana Ștefănescu, and Mădălina Cocoșatu. 2021. Artificial Intelligence-Based Decision-Making Algorithms, Internet of Things Sensing Networks, and Deep Learning-Assisted Smart Process Management in Cyber-Physical Production Systems. *Electronics* 10, 20 (2021), 2497.
[6] Mikel Armendia, Flavien Peysson, and Dirk Euhus. 2016. Twin-control: a new concept towards machine tool health management. In *PHM Society European Conference*, Vol. 3.
[7] ASP.NET Core. [n. d.]. https://github.com/dotnet/aspnetcore. Visited in 2024.
[8] National Marine Electronics Association. 2023. NMEA 0183 Protocol. https://www.nmea.org/nmea-0183.html
[9] Md Abul Bashar and Richi Nayak. 2020. TAnoGAN: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 1778–1785.
[10] Angelo Corallo, Anna Maria Crespino, Vito Del Vecchio, Mariangela Lazoi, and Manuela Marra. 2021. Understanding and Defining Dark Data for the Manufacturing Industry. *IEEE Transactions on Engineering Management* (2021).
[11] David Camilo Corrales, Juan Carlos Corrales, and Agapito Ledezma. 2018. How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning. *Symmetry* 10, 4 (2018).
[12] Bulent Dal, Phil Tugwell, and Richard Greatbanks. 2000. Overall equipment effectiveness as a measure of operational improvement–a practical analysis. *International journal of operations & production management* 20, 12 (2000), 1488–1502.
[13] Alan F Dutka and Howard H Hansen. 1991. *Fundamentals of data normalization.* Addison-Wesley Longman Publishing Co., Inc.
[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, Vol. 96. 226–231.
[15] Susana Ferreiro, Egoitz Konde, Santiago Fernández, and Agustín Prado. 2016. Industry 4.0: predictive intelligent maintenance for production equipment. In *PHM society European conference*, Vol. 3.
[16] Roy Fielding. 2023. REpresentational State Transfer and an architectural style (REST). https://restfulapi.net/
[17] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2020. Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 ieee international conference on big data (big data).* IEEE, 33–43.
[18] Arda Goknil, Phu Nguyen, Sagar Sen, Dimitra Politaki, Harris Niavis, Karl John Pedersen, Abdillah Suyuthi, Abhilash Anand, and Amina Ziegenbein. 2023. A Systematic Review of Data Quality in CPS and IoT for Industry 4.0. *ACM Comput. Surv.* 55, 14s, Article 327 (jul 2023), 38 pages. https://doi.org/10.1145/3593043
[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* MIT press.
[20] Great Expectations. [n. d.]. https://greatexpectations.io/. Visited in 2023.
[21] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 28, 10 (2016), 2222–2232.
[22] Erik Johannes Husom, Arda Goknil, Simeon Tverdal, Sagar Sen, and Phu H Nguyen. 2023. Automated Behavior Labeling for IIoT Data. In *Proceedings of the 13th International Conference on the Internet of Things.* 174–178.
[23] Erik Johannes Husom, Sagar Sen, Arda Goknil, Simeon Tverdal, and Phu H Nguyen. 2023. REPTILE: a Tool for Replay-driven Continual Learning in IIoT. In *Proceedings of the 13th International Conference on the Internet of Things.* 204–207.
[24] Erik Johannes Husom, Simeon Tverdal, Arda Goknil, and Sagar Sen. 2022. UDAVA: An unsupervised learning pipeline for sensor data validation in manufacturing. In *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI.* 159–169.
[25] ISO/IEC International. 2008. *ISO/IEC 25012:2008 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model.*
[26] Mauro Isaja, Phu Nguyen, Arda Goknil, Sagar Sen, Erik Johannes Husom, Simeon Tverdal, Abhilash Anand, Yunman Jiang, Karl John Pedersen, Per Myrseth, et al. 2023. A blockchain-based framework for trusted quality data sharing towards zero-defect manufacturing. *Computers in Industry* 146 (2023), 103853.
[27] Klaus Kammerer, Burkhard Hoppenstedt, Rüdiger Pryss, Steffen Stökler, Johannes Allgaier, and Manfred Reichert. 2019. Anomaly detections for manufacturing systems based on sensor data—insights into two challenging real-world production settings. *Sensors* 19, 24 (2019), 5370.
[28] Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. 2016. Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications* 73 (2016), 57–81.
[29] Tianxiang Kong, Tianliang Hu, Tingting Zhou, and Yingxin Ye. 2021. Data Construction Method for the Applications of Workshop Digital Twin System. *Journal of Manufacturing Systems* 58 (2021), 323–328.
[30] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade.* Springer, 9–48.
[31] Wei-Tsung Lin, Fatih Bakir, Chandra Krintz, Rich Wolski, and Markus Mock. 2019. Data Repair for Distributed, Event-Based IoT Applications. In *DEBS'19.* 139–150.
[32] Benjamin Lindemann, Fabian Fesenmayr, Nasser Jazdi, and Michael Weyrich. 2019. Anomaly detection in discrete manufacturing using self-learning approaches. *Procedia CIRP* 79 (2019), 313–318.
[33] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).
[34] Gómez-Omella Meritxell, Basilio Sierra, and Susana Ferreiro. 2022. On the evaluation, management and improvement of data quality in streaming time series. *IEEE Access* 10 (2022), 81458–81475.
[35] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.
[36] Bruce Jay Nelson. 1981. *Remote procedure call.* Carnegie Mellon University.
[37] .NET eshop Reference Architecture. [n. d.]. https://github.com/dotnet/eShop. Visited in 2024.
[38] Paul Newbold. 1983. ARIMA model building and the time series analysis approach to forecasting. *Journal of forecasting* 2, 1 (1983), 23–35.
[39] Phu H Nguyen, Hui Song, Franck Chauvel, Roy Muller, Seref Boyar, and Erik Levin. 2019. Using microservices for non-intrusive customization of multi-tenant SaaS. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering.* 905–915.
[40] Espen Tønnessen Nordli, Phu H Nguyen, Franck Chauvel, and Hui Song. 2020. Event-based customization of multi-tenant saas using microservices. In *International Conference on Coordination Languages and Models.* Springer, 171–180.
[41] OAuth 2.0. [n. d.]. https://oauth.net/2/. Visited in 2024.
[42] OpenID Connect. [n. d.]. https://openid.net/developers/how-connect-works/. Visited in 2024.
[43] The Modbus Organization. 2023. Modbus Protocol. https://www.modbus.org/
[44] The pandas development team. 2020. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134
[45] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)* 54, 2 (2021), 1–38.
[46] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[47] François Petitjean, Alain Ketterlin, and Pierre Gançarski. 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern recognition* 44, 3 (2011), 678–693.
[48] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. 2011. Detecting novel associations in large data sets. *science* 334, 6062 (2011), 1518–1524.
[49] Luke Russell, Felix Kwamena, and Rafik Goubran. 2019. Towards Reliable IoT: Fog-Based AI Sensor Validation. In *IEEE Cloud Summit.* 37–44.
[50] Saket Sathe, Thanasis G Papaioannou, Hoyoung Jeung, and Karl Aberer. 2013. A survey of model-based sensor data acquisition and management. In *Managing and mining sensor data.* 9–50.
[51] Philip Sedgwick. 2012. Pearson's correlation coefficient. *Bmj* 345 (2012).
[52] Sagar Sen, Erik Johannes Husom, Arda Goknil, Dimitra Politaki, Simeon Tverdal, Phu Nguyen, and Nicolas Jourdan. 2023. Virtual sensors for erroneous data repair in manufacturing a machine learning pipeline. *Computers in Industry* 149 (2023), 103917.
[53] Sagar Sen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, and Phu Nguyen. 2024. Uncertainty-aware Virtual Sensors for Cyber-Physical Systems. *IEEE Software* 41, 2 (2024), 77–87.
[54] Sagar Sen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, Phu Nguyen, and Iker Mancisidor. 2022. Taming Data Quality in AI-Enabled Industrial Internet of Things. *IEEE Software* 39, 6 (2022), 35–42.
[55] Sagar Sen, Simon Myklebust Nielsen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, and Leonardo Sastoque Pinilla. 2023. Replay-driven continual learning for the industrial internet of things. In *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN).* IEEE, 43–55.
[56] Hui Song, Phu H Nguyen, Franck Chauvel, Jens Glattetre, and Thomas Schjerpen. 2019. Customizing multi-tenant SaaS by microservices: a reference architecture. In *2019 IEEE International Conference on Web Services (ICWS).* IEEE, 446–448.
[57] Muhammad Syafrudin, Ganjar Alfian, Norma Latif Fitriyani, and Jongtae Rhee. 2018. Performance Analysis of IoT-Based Sensor, Big Data Processing, and

Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing. *Sensors* 18, 9 (2018).

[58] Simeon Tverdal, Arda Goknil, Phu Nguyen, Erik Johannes Husom, Sagar Sen, Jan Ruh, and Francesca Flamigni. 2023. Edge-based Data Profiling and Repair as a Service for IoT. In *Proceedings of the 13th International Conference on the Internet of Things*. 17–24.

[59] The World Wide Web Consortium (W3C). 2023. Simple Object Access Protocol (SOAP). https://www.w3.org/TR/soap/

[60] Y Richard Wang, Lisa M Guarascio, and Richard Wang. 1991. Dimensions of data quality: Toward quality data by design. (1991).

[61] Wenjin Yu, Tharam Dillon, Fahed Mostafa, Wenny Rahayu, and Yuehua Liu. 2019. A global manufacturing big data ecosystem for fault detection in predictive maintenance. *IEEE Transactions on Industrial Informatics* 16, 1 (2019), 183–192.

[62] Wenjin Yu, Tharam Dillon, Fahed Mostafa, Wenny Rahayu, and Yuehua Liu. 2019. Implementation of Industrial Cyber Physical System: Challenges and Solutions. In *ICPS'19*. 173–178.