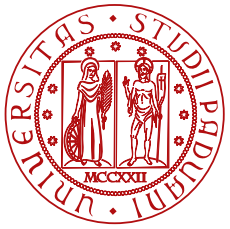

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

MASTER THESIS IN CONTROL SYSTEM ENGINEERING
UNIVERSITY OF PADUA
DEPARTMENT OF INFORMATION ENGINEERING

Anomaly detection in unsupervised industrial settings, developing human-centric diagnostic and prognostic tools to improve decision-making and transparency

MASTER CANDIDATE

Ali Darvishi

Student ID 2109822

SUPERVISOR

Prof. Gian Antonio Susto

University of Padova

CO-SUPERVISOR

Chiara Masiero, PhD

Statwolf Data Science S.r.l.

ACADEMIC YEAR
2025/2026
DATE : JULY 7, 2026

To my family, whose support made this possible, and to my supervisors, for their guidance and mentorship throughout this journey

Abstract

The increasing adoption of sensor-rich, automated production systems under Industry 4.0 and Industry 5.0 frameworks has created both the opportunity and the necessity for intelligent, real-time anomaly detection. In complex industrial environments, unexpected deviations in process data can precede equipment failures, quality defects, and unplanned downtime, yet the rarity and absence of labeled anomalies make conventional supervised approaches largely impractical. This thesis addresses the challenge of developing unsupervised anomaly detection methods that are interpretable, computationally efficient, and aligned with human-centric operational requirements. The research was conducted in collaboration with an industrial bakery company, where multivariate sensor data capturing temperature, mixer speeds, conveyor velocities, and other process variables were collected across multiple production lines. Three complementary approaches were developed: a statistical threshold-based method for abrupt signal deviations, an Isolation Forest for multivariate anomalies in an unsupervised manner, and a CatBoost regression–residual framework for signal-level predictive monitoring under weak supervision. Methods were integrated with Explainable AI (XAI) techniques, including SHAP and AcME, to provide operators with actionable, human-interpretable insights into detected anomalies. AcME emerged as the preferred tool due to its computational efficiency and suitability for near-real-time deployment. The CatBoost framework achieved strong predictive performance with R^2 values exceeding 0.96, while the Isolation Forest demonstrated complementary strengths in detecting anomalies arising from multivariate interactions.

The results show that combining lightweight machine learning models with fast, interpretable XAI methods yields a practical and transferable framework for industrial anomaly detection, supporting human-in-the-loop decision-making without requiring labeled fault data. Limitations and directions for future work are discussed.

Sommario

La crescente adozione di sistemi di produzione automatizzati e ricchi di sensori nellámbito dei paradigmi Industry 4.0 e Industry 5.0 ha creato sia lópportunit'a che la necessit'a di un rilevamento intelligente delle anomalie in tempo reale. In ambienti industriali complessi, deviazioni inattese nei dati di processo possono precedere guasti alle apparecchiature, difetti di qualit'a e fermi non pianificati; tuttavia, la rarit'a e lássenza di anomalie etichettate rendono gli approcci supervisionati convenzionali in gran parte impraticabili. Questa tesi affronta la sfida di sviluppare metodi di rilevamento delle anomalie non supervisionati e debolmente supervisionati, che siano interpretabili, computazionalmente efficienti e allineati ai requisiti operativi incentrati sullúomo.

La ricerca 'e stata condotta in collaborazione con unázienda industriale del settore bakery, dove dati sensoriali multivariati,che acquisiscono temperatura, velocit'a dei miscelatori, velocit'a dei nastri trasportatori e altre variabili di processo, sono stati raccolti su pi'u linee di produzione. Sono stati sviluppati tre approcci complementari: un metodo statistico basato su soglie per le deviazioni abrupte del segnale, un modello Isolation Forest per le anomalie multivariate in modalit'a non supervisionata, e un framework CatBoost basato su regressione e analisi dei residui per il monitoraggio predittivo a livello di segnale in regime di supervisione debole.

Tutti e tre i metodi sono stati integrati con tecniche di Intelligenza Artificiale Spiegabile (XAI), tra cui SHAP e AcME, per fornire agli operatori informazioni interpretabili e fruibili sulle cause delle anomalie rilevate. AcME 'e emerso come lo strumento XAI preferito grazie alla sua efficienza computazionale e allídoneit'a per il deployment in near-real-time. Il framework CatBoost ha raggiunto prestazioni predittive elevate con valori di R^2 superiori a 0.96, mentre lÍsolation Forest ha dimostrato punti di forza complementari nel rilevare anomalie derivanti da interazioni multivariate.

I risultati mostrano che la combinazione di modelli di machine learning leggeri con metodi XAI veloci e interpretabili produce un framework pratico e trasferibile per il rilevamento di anomalie industriali, supportando il processo decisionale human-in-the-loop senza richiedere dati di guasto etichettati. Vengono discussi i limiti e le direzioni per il lavoro futuro.

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**
1.1 What is Anomaly Detection? 1
1.2 Anomaly Detection Approaches 2
1.3 The Importance of Explainable AI 3
1.4 Problem Statement 4
1.5 Research Context 5
1.6 Research Objectives 6
1.7 Methodological Approach 6

2 Background and Literature Review **9**
2.1 Introduction 9
2.2 Industrial Anomaly Detection 10
2.3 Relevant Modelling Approaches 11
2.3.1 Statistical and Classical Machine Learning Methods 12
2.3.2 Deep Learning Methods 14
2.4 Challenges in Industrial Applications 16
2.5 Evaluation Metrics and Validation Strategies 18
2.5.1 Evaluation Strategies for Unsupervised Anomaly Detection 19
2.5.2 Regression-Based Anomaly Detection and Residual Analysis 21

3 Data Description **23**
3.1 Data Source and Collection Process 23
3.1.1 Historical Data Extraction (Phase I) 23

3.1.2 Validation and Testing Set (Phase II) . . . 23
3.1.3 Deployment and Connectivity Roadmap . . . 23
3.2 Description of Sensors, Variables, and Operational Context . . . 24
3.2.1 F1 . . . 24
3.2.2 F2 . . . 25
3.2.3 F3 . . . 26
3.2.4 BL . . . 27
3.2.5 C1 . . . 28
3.2.6 C2 . . . 29
3.3 Data Preprocessing . . . 29
3.3.1 Overview: Concept and Motivation . . . 29
3.3.2 Removal of Irrelevant Features . . . 30
3.3.3 Removal of Duplicate Timestamps . . . 31
3.3.4 Target Signal Selection . . . 31
3.3.5 Temporal Indexing and Resampling . . . 31
3.3.6 Handling Missing Values . . . 33
3.4 Clean Data Summary . . . 33
3.4.1 F1_131 . . . 33
3.4.2 F2_132 . . . 35
3.4.3 F3_133 . . . 36
3.4.4 BL_140 . . . 38
3.4.5 IS_141, C1_138, and C2_139 . . . 40

4 Methodology **41**
4.1 Feature Engineering . . . 41
4.1.1 Overview . . . 41
4.1.2 Categorical Feature Engineering . . . 42
4.1.3 Alarm-Based Features . . . 43
4.1.4 Rolling Window Statistics . . . 45
4.1.5 Cross-Signal Interaction Features . . . 46
4.1.6 Frequency-Domain Features . . . 47
4.1.7 Time-Based and Calendar Features . . . 49
4.2 Simple Statistical Approach for Time-Series Anomaly Detection . . . 50
4.2.1 Z-Score-Based Anomaly Detection . . . 50
4.2.2 Median Absolute Deviation (MAD) . . . 51
4.2.3 Applicability and Limitations . . . 51

4.3 Residual-Based Anomaly Detection Using CatBoost Regression . 52
4.3.1 CatBoost for Modeling Normal System Behavior 52
4.3.2 Residual Thresholding for Anomaly Detection 53
4.3.3 Interpretability and Industrial Relevance 53
4.4 Isolation Forest for Unsupervised Anomaly Detection 54
4.4.1 Principle of Operation 54
4.4.2 Applicability to Industrial Data 55
4.4.3 Strengths and Limitations 55
4.4.4 Role in Human-Centric Industrial Monitoring 56
4.5 Explainable Artificial Intelligence (XAI) 56
4.5.1 Goals of XAI 56
4.5.2 Importance of XAI in Industrial and Unsupervised Settings 57
4.6 SHapley Additive exPlanations (SHAP) 57
4.6.1 Principle of SHAP 58
4.6.2 Global SHAP Explanations 59
4.6.3 Local SHAP Explanations 59
4.7 AcME-AD 60
4.7.1 Overview of AcME-AD 60
4.7.2 What AcME-AD Is 60
4.7.3 How AcME-AD Works 60
4.7.4 Goal of AcME-AD 61
4.7.5 Benefits of AcME-AD 62
4.7.6 Summary 62

5 Experiments and Results **65**
5.1 Isolation Forest 65
5.1.1 F1 65
5.1.2 BL 67
5.2 CatBoost Regression 69
5.2.1 F1 69
5.2.2 BL 72
5.3 Simple Statistical Approach 75
5.3.1 F1 75
5.4 Comparison of Models 76

6 Discussion **77**

7 Future Work **81**
7.1 Improved and More Accurate Data Collection 81
7.2 Real-Time Deployment and Online Monitoring 82
7.3 Model Improvements and Methodological Enhancements 83
7.4 Extension to Additional Plant Modules 84

References **85**

Acknowledgments **89**

List of Figures

3.1 10-Minute Sampling Intervals 32
3.2 Hourly Sampling Intervals 32
3.3 set_attivo_temperatura - F1 34
3.4 temperatura_visualizzata - F1 35
3.5 set_attivo_temperatura - F2 36
3.6 temperatura_visualizzata - F2 36
3.7 set_attivo_temperatura - F3 38
3.8 temperatura_visualizzata - F3 38
3.9 iVelSpiraManuale1 - BL 39
3.10 iVelSpira - BL 39

5.1 set_attivo_temperatura - Anomaly 66
5.2 temperatura_visualizzata - Anomaly 66
5.3 iVelSpiraManuale1 - Anomaly 68
5.4 iVelSpira - Anomaly 68
5.5 temperatura_visualizzata - Main Data 69
5.6 temperatura_visualizzata – Test Data 70
5.7 Local SHAP analysis 70
5.8 set_attivo_temperatura – Main Data 71
5.9 set_attivo_temperatura – Test Data 71
5.10 iVelSpiraManuale1 - Main Data 73
5.11 iVelSpiraManuale1 - Test Data 73
5.12 iVelSpira - Test Data 74
5.13 set_attivo_temperatura – simple approach - main Data 75
5.14 set_attivo_temperatura – simple approach - Test Data 76

List of Tables

3.1 F1_131 24
3.2 F2_132 25
3.3 F3_133 26
3.4 BL_140 27
3.5 C1_138 28
3.6 C2_139 29
3.7 Example of Duplicated Samples 31
3.8 Categorical EDA - F1 34
3.9 Numerical EDA - F1 34
3.10 Categorical EDA - F2 35
3.11 Numerical EDA - F2 36
3.12 Categorical EDA - F3 37
3.13 Numerical EDA - F3 37
3.14 Categorical EDA - BL 39
3.15 Numerical EDA - BL 39

5.1 Performance metrics - temperatura_visualizzata 70
5.2 Performance metrics - set_attivo_temperatura 72
5.3 Performance metrics - iVelSpiraManuale1 73
5.4 Performance metrics - iVelSpira 74

1 Introduction

In modern industrial manufacturing, production systems are characterized by widespread digital transformation, interconnected machinery, and pervasive sensor networks that continuously generate large volumes of operational data. As manufacturing evolves through Industry 4.0, marked by automation, cyber-physical systems, and the Industrial Internet of Things (IIoT) [16, 21], and toward Industry 5.0, which emphasizes human-centricity, resilience, and sustainability [8], the ability to monitor processes in real time and detect abnormal behavior has become essential for ensuring reliability, efficiency, and effective human-machine collaboration. Within these complex, highly digitalized, and sensor-driven systems, anomaly detection plays a critical role [5]. Unexpected deviations in sensor data often precede costly equipment failures, unplanned downtime, quality defects, or cybersecurity threats [6]. Early identification of such deviations is essential for preventing failures, avoiding costly downtime, maintaining product quality, and ensuring safety. It enables predictive maintenance strategies, optimized resource usage, improved decision-making, and enhanced system resilience [3].

1.1 What is Anomaly Detection?

In data science, anomaly detection refers to the process of identifying data patterns or observations that deviate significantly from expected behavior [5]. These unusual instances, often called anomalies, outliers, or novelties, typically indicate events of interest such as system faults, operational deviations, security

breaches, or data quality issues [29]. The goal is to automatically distinguish normal behavior from unexpected patterns without relying on explicit rule-based definitions of what constitutes an abnormal event. Anomaly detection encompasses a broad set of statistical, machine learning, and signal processing techniques designed to model normal system behavior and flag deviations from that model [25]. In practice, it is applied across numerous domains, including manufacturing, cybersecurity, finance, healthcare, and sensor-based monitoring [5], playing a crucial role in detecting emerging problems early, preventing failures, improving safety, and supporting data-driven decision-making.

1.2 Anomaly Detection Approaches

Depending on the availability of labeled data, anomaly detection approaches may be supervised, unsupervised, or semi-supervised, each with distinct strengths and limitations [29]. Supervised anomaly detection relies on labeled training data where both normal and abnormal instances are explicitly identified. The model learns a decision boundary separating normal behavior from known types of anomalies using classification algorithms such as decision trees, support vector machines, or deep neural networks [25]. However, this approach faces significant limitations in real-world industrial settings: anomalies occur infrequently and are highly diverse, making it difficult to collect sufficient labeled examples; data are highly imbalanced with normal instances vastly outnumbering anomalies [12]; annotation requires costly expert knowledge and manual inspection; and supervised models typically detect only the types of anomalies present in the training set, failing to recognize novel or emerging faults [5]. Due to these challenges, supervised learning is often impractical in complex industrial systems where anomalies are rare, diverse, and unpredictable. Unsupervised anomaly detection does not require labeled data. Instead, models are trained exclusively on normal operating data, learning its underlying structure or distribution. Instances that deviate significantly from this learned pattern are flagged as anomalies. Methods include clustering techniques, statistical models, autoencoders, generative models, and one-class classification approaches [30, 29]. This approach is well-suited for industrial environments because normal behavior is abundant while anomalies are rare or unknown, the system can detect previously unseen faults by identifying deviations rather than relying on predefined labels, no expensive annotation is required, and it supports contin-

uous monitoring as new data can be incorporated without requiring new labels [25]. Because anomalies are rare and often undocumented or poorly classified in industrial settings, unsupervised or weakly supervised methods are needed to learn normal behavior directly from data without relying on labeled examples [6]. Nevertheless, unsupervised methods present their own challenges, including sensitivity to noise, difficulty distinguishing rare-but-valid events from true faults, and the inherent challenge of evaluating performance without labeled anomalies [5].

1.3 The Importance of Explainable AI

While unsupervised anomaly detection is well-suited for industrial environments where anomalies are rare, diverse, and costly to label, it also introduces significant challenges related to interpretability and trust [29]. Since unsupervised models operate without explicit supervision or predefined anomaly categories, their outputs may appear opaque or difficult for human operators to understand. This is where Explainable AI (XAI) becomes critically important [1]. XAI enhances transparency by providing insights into why a data instance is identified as anomalous [1]. In conventional unsupervised models,such as autoencoders, clustering algorithms, or density estimators,the decision logic is often hidden in complex latent representations or statistical thresholds [25]. Without interpretability, operators may struggle to determine whether an alert reflects an actual fault, a sensor error, or a harmless deviation. XAI helps bridge this gap by highlighting influential features, visualizing reconstruction errors, or explaining deviations in terms that domain experts can evaluate [22]. In human-centric industrial environments, such as those envisioned in Industry 5.0 [8], human operators remain central to decision-making, and AI tools are expected to assist rather than replace them [37]. The necessity of human-centric anomaly detection is clear: models must be interpretable and transparent, support human decision-making, and provide actionable diagnostic insights [1]. Transparent anomaly detection systems improve situational awareness, reduce cognitive load, and allow operators to validate or reject model outputs with confidence [7]. When explanations are meaningful and aligned with domain knowledge, humans are more likely to trust the system and intervene appropriately in critical situations [27]. Furthermore, XAI supports diagnostic and prognostic decision-making [18]. While unsupervised models can identify abnormal patterns, they

often do not specify the underlying cause or severity of the deviation. Explainability methods can uncover patterns indicating equipment degradation, process drift, or potential safety issues, enabling maintenance teams to take targeted actions [3]. Finally, XAI is essential for model validation and continuous improvement, allowing engineers to examine flagged cases, identify systematic errors, and refine model architectures [1].

1.4 Problem Statement

Implementing effective anomaly detection in real industrial settings remains challenging despite its critical importance [6]. Traditional monitoring approaches rely on simple threshold-based rules that do not capture relationships between multiple sensors, miss subtle anomalies that manifest across multiple variables, and produce many false alarms that overwhelm operators [36]. These limitations highlight the inadequacy of conventional methods for complex, multivariate industrial systems. Industrial datasets present fundamental challenges: data are highly imbalanced, with normal data vastly outnumbering anomaly instances [12]; anomalies are rare, diverse, and difficult to label due to the high cost of expert annotation, sensitivity of operational data, or the unpredictable nature of real-world faults [5]; and labels are scarce or unavailable because anomalies are often undocumented or poorly classified [25]. These factors limit the feasibility of supervised learning approaches and highlight the need for unsupervised or weakly supervised methods. The high-dimensional nature of sensor streams and complex interdependencies between process variables make it difficult to define what constitutes "normal" behavior [5]. Defining normal behavior is particularly challenging because operating conditions change with different product recipes, environmental conditions vary over time and seasons, and equipment experiences natural aging [17]. Normal operational conditions can vary significantly, while true anomalies,such as equipment degradation or process deviations,may manifest subtly across multiple sensors before becoming critical failures [18]. Deep learning-based anomaly detection models have shown strong performance in controlled environments [25], yet their adoption in industry is constrained by several critical limitations: the need for large labeled datasets that are unavailable in industrial settings, high computational costs that may be impractical for real-time monitoring, and lack of interpretability, as "black-box" models provide no insight into why anomalies are flagged [1]. In

safety-critical or regulated industrial domains, black-box predictions are insufficient for operational decision-making, as engineers require transparent, human-understandable insights to trust and act upon model outputs [27]. Therefore, the central research problem addressed in this thesis is the development of unsupervised or weakly supervised, interpretable anomaly detection methods tailored to real industrial constraints and aligned with Industry 4.0 and Industry 5.0 principles [16, 8]. These methods must be capable not only of identifying abnormal patterns but also of explaining their causes, providing actionable diagnostics, and supporting effective maintenance strategies [3].

1.5 Research Context

This research was conducted in collaboration with an industrial bakery company specializing in bread and cake production. The production line is a highly automated system equipped with numerous sensors measuring temperature, humidity, pressure, mixing speeds, conveyor velocities, oven conditions, and ingredient flow rates [21]. These sensors generate continuous, high-frequency data streams that collectively describe the state of the production process, reflecting a complex and interdependent physical process. The food manufacturing production environment presents unique challenges that make this setting particularly demanding for anomaly detection [6]. Anomalies are rare and unlabeled, with no comprehensive historical record of fault types. Data are noisy and multivariate, captured from multiple interdependent sensors across different process stages [5]. Small deviations in early process stages can propagate through the system and have cascading effects on downstream operations or final product quality [36]. Each product type may have different optimal operating conditions, equipment operates continuously with gradual wear over time [17], and environmental factors can influence process behavior. Deviations from normal operation can lead to quality issues in the final product or indicate impending equipment failures requiring maintenance intervention [3]. The industrial setting introduces several inherent difficulties that reinforce the need for unsupervised and weakly supervised methods capable of learning patterns of normal behavior directly from data [29]. Because anomalies are rare, diverse, unlabeled, and occur in a noisy multivariate environment, purely supervised approaches are unsuitable for this real-world application [25].

1.6 Research Objectives

The primary objective of this thesis is to develop an effective anomaly detection system for industrial bakery production that addresses the challenges of unlabeled data and multivariate complexity while supporting human decision-making [25]. Specifically, this research aims to:

- Develop robust anomaly detection models capable of identifying equipment failures and process deviations in an unsupervised or weakly supervised manner, suitable for deployment in environments with limited labeled data [5].
- Handle high-dimensional multivariate data from production line sensors, capturing complex relationships between process variables that traditional univariate methods cannot detect.
- Create interpretable and transparent monitoring tools that not only detect anomalies but also provide operators and maintenance personnel with actionable insights into the nature and potential causes of detected deviations [1].
- Design human-centric interfaces including visualization dashboards and alert systems that support effective decision-making for maintenance scheduling and operator intervention, balancing detection sensitivity with practical usability [7].
- Provide both diagnostic and prognostic capabilities, enabling the system to identify current anomalies while also detecting early warning signs of developing issues before they impact production [17].

1.7 Methodological Approach

This thesis adopts a multi-level methodological strategy to address the challenges of anomaly detection in industrial bakery production, balancing computational efficiency, detection capability, and human-centric interpretability. Three complementary modeling approaches are explored. First, a simple statistical anomaly detection method is employed as a baseline approach. This method relies on lightweight statistical measures and thresholds to detect obvious peaks, drops, or abrupt changes in sensor signals. Due to its low computational cost and ease of deployment, it is well suited for fast initial monitoring and for identifying clear abnormal events at early stages of analysis. Although limited in its ability to capture complex dependencies, this approach

provides a practical and interpretable starting point for anomaly detection in industrial environments [36]. Second, Isolation Forest is investigated as an unsupervised, tree-based anomaly detection technique [20]. Isolation Forest exploits the principle that anomalies are rare and easier to isolate than normal observations. In addition to its suitability for unlabeled data, this method is capable of capturing multivariate relationships between sensor signals, allowing it to detect anomalies that emerge only through interactions among multiple process variables rather than through univariate deviations alone. This makes it particularly appropriate for complex industrial systems with interconnected processes. Third, a CatBoost regression–residual framework is developed as a predictive anomaly detection approach [26]. In this framework, a regression model is trained to learn the expected system behavior under normal operating conditions. Anomalies are identified by analyzing residuals, defined as deviations between predicted and observed sensor values. This approach supports weak supervision and offers enhanced interpretability by indicating which variables deviate from expected behavior and to what extent. All three methods are evaluated using real multivariate sensor data collected from the industrial bakery production line. The evaluation considers not only anomaly detection performance but also suitability for weakly supervised settings, computational efficiency and deployment feasibility, and compatibility with interpretability and explainability requirements [29]. Particular emphasis is placed on the quality and actionability of explanations provided to human operators, in line with the human-centric principles of Industry 5.0 [8].

2

Background and Literature Review

2.1 Introduction

Anomaly detection in industrial environments has become a central component of intelligent manufacturing systems, contributing to predictive maintenance, quality assurance, and real-time monitoring [5]. As production lines become increasingly sensorized and interconnected through the proliferation of Industrial Internet of Things (IIoT) technologies [21], large volumes of multivariate time-series data are generated continuously. This data-rich environment provides significant opportunities for data-driven insights, enabling organizations to optimize operations, reduce costs, and improve reliability. However, it also introduces new methodological challenges related to data volume, velocity, variety, and the complexity of extracting meaningful patterns from high-dimensional sensor streams. The transition from reactive to proactive maintenance strategies, driven by the Industry 4.0 paradigm [16], has positioned anomaly detection as a critical enabling technology. Rather than responding to failures after they occur or relying on fixed maintenance schedules, modern industrial systems increasingly leverage continuous monitoring and automated anomaly detection to identify emerging issues before they escalate into costly failures or safety hazards [3]. This shift requires robust, scalable, and interpretable methods capable of operating in environments where labeled failure data is scarce, system dynamics are complex, and real-time decision support is essential. This chapter reviews the relevant foundations of industrial anomaly detection, discusses established modeling approaches ranging from classical

statistical methods to contemporary deep learning architectures, and outlines the main challenges associated with real-world applications. Particular attention is given to the trade-offs between model complexity, interpretability, and practical deployability in industrial settings. The chapter also examines evaluation strategies in contexts where labeled anomalies are unavailable, which is a common situation in industrial environments where failures are rare and inconsistently documented [25].

2.2 Industrial Anomaly Detection

Industrial anomaly detection refers to the identification of unusual or unexpected patterns in process or equipment behavior based on sensor data collected from machines, production lines, or industrial assets [5]. Unlike anomaly detection in other domains, such as fraud detection in finance or intrusion detection in cybersecurity, industrial anomaly detection must contend with the physical constraints, temporal dependencies, and multivariate interdependencies inherent in manufacturing processes [6]. The ultimate goal is not merely to flag deviations, but to provide actionable insights that support maintenance decisions, quality control interventions, and process optimization.

In manufacturing environments, anomaly detection supports predictive maintenance by identifying deviations in sensor patterns that may indicate wear, malfunction, or emerging faults in equipment components [18]. Detecting subtle changes in vibration signatures, temperature profiles, or power consumption before catastrophic failure occurs allows organizations to schedule maintenance interventions proactively, minimizing unplanned downtime and extending equipment lifespan [3]. Anomaly detection also contributes to quality control by identifying deviations in process parameters, such as temperature fluctuations, pressure variations, or timing inconsistencies, which can directly influence product consistency and quality. Early detection enables rapid corrective action, reduces waste, and ensures compliance with regulatory standards. Continuous process monitoring further enables organizations to detect operational shifts that might compromise stability, efficiency, or safety. Anomalies may indicate changes in raw material properties, environmental conditions, operator behavior, or gradual equipment degradation, allowing adaptive responses to maintain optimal performance [36].

Industrial processes, including those in food production such as baking,

generate complex and interdependent sensor measurements spanning multiple physical domains. Temperature sensors monitor oven conditions and ambient environment, humidity sensors track moisture levels critical to product texture, pressure sensors measure pneumatic systems and ingredient delivery, speed sensors monitor conveyors, mixers, and rotating equipment, and flow sensors track ingredient dosing and energy consumption. These signals collectively reflect nonlinear dynamics, thermal inertia, mechanical coupling, and evolving system states that change gradually over time due to equipment wear, seasonal variations, and operational adjustments [17].

The multivariate and temporal nature of industrial sensor data presents significant challenges for anomaly detection. Individual sensors rarely provide complete information about system health, and anomalies often manifest as subtle changes in the relationships between multiple variables [5]. The temporal evolution of these relationships provides critical context for distinguishing genuine anomalies from natural process variations. Manual monitoring is impractical and error-prone, as human operators cannot simultaneously track dozens of variables, recognize complex multivariate patterns, or maintain sustained vigilance over extended periods [7].

Explainable AI plays a critical role in industrial anomaly detection [1]. Since anomalies can be subtle, multivariate, and impact critical operations, operators need to understand why a specific instance is flagged as anomalous. Explainable methods provide transparency by highlighting influential features, explaining deviations in human-understandable terms, and supporting effective decision-making and maintenance interventions [22].

Key data challenges in industrial anomaly detection include the scarcity of labeled anomalies and the severe imbalance between normal and abnormal data [12]. These factors limit the feasibility of supervised learning approaches and highlight the necessity of unsupervised or weakly supervised methods capable of learning normal process behavior directly from historical data [29].

2.3 Relevant Modelling Approaches

The literature on anomaly detection presents a diverse array of modeling approaches, each with distinct assumptions, capabilities, and limitations. This section reviews the main categories of methods relevant to industrial applications, examining their theoretical foundations, practical considerations, and suitabil-

ity for environments characterized by unlabeled data, multivariate complexity, and interpretability requirements.

2.3.1 Statistical and Classical Machine Learning Methods

Traditional statistical approaches have formed the foundation of industrial process monitoring for decades. Control charts, developed initially by Walter Shewhart in the 1920s and extended by subsequent researchers, monitor individual process variables by detecting observations that fall outside statistically defined control limits. Statistical Process Control (SPC) techniques, including Shewhart charts, CUSUM (cumulative sum) charts, and EWMA (exponentially weighted moving average) charts, provide simple, interpretable methods for detecting shifts in process mean or variance [24]. These approaches are well-established, computationally efficient, and widely understood by industrial practitioners. However, classical statistical methods face significant limitations when applied to modern industrial systems. They typically assume that process variables follow specific probability distributions (often Gaussian), that observations are independent and identically distributed, and that the process is in a state of statistical control. These assumptions frequently do not hold in complex production systems where sensor measurements exhibit autocorrelation, non-stationarity, multivariate dependencies, and non-Gaussian distributions [36]. Furthermore, univariate control charts applied independently to each sensor fail to capture the correlations and interactions between variables that are often essential for detecting subtle process anomalies. Distance-based anomaly detection methods address some of these limitations by measuring the dissimilarity between observations in multivariate space. Techniques such as k-nearest neighbors (k-NN) and Local Outlier Factor (LOF) identify anomalies based on their distance or density relative to neighboring points [2]. These methods do not require distributional assumptions and can handle complex, nonlinear relationships. However, they are sensitive to the choice of distance metric, may perform poorly in high-dimensional spaces due to the curse of dimensionality, and can be computationally expensive for large datasets. More flexible classical machine learning methods provide improved adaptability to complex industrial data. Clustering algorithms such as k-means, DBSCAN, and hierarchical clustering partition data into groups representing different operational modes or process states, with instances far from any cluster center flagged as anomalies.

Density-based detectors identify regions of low probability density as anomalous. Principal Component Analysis (PCA) and its variants reduce dimensionality by projecting data onto principal components capturing maximum variance, with anomalies detected based on reconstruction error or residuals in the principal component space [15]. Among classical machine learning methods, Isolation Forest has gained particular prominence in industrial anomaly detection due to its distinctive approach and practical advantages [20]. Proposed by Liu et al. (2008), Isolation Forest exploits the observation that anomalies are "few and different", they are both rare and have feature values that differ significantly from normal instances. Rather than explicitly modeling normal behavior, Isolation Forest isolates anomalies by recursively partitioning the feature space using random splits. Anomalies, being different, are isolated closer to the root of randomly generated trees and thus require fewer splits to be isolated. The algorithm constructs an ensemble of isolation trees, each built by randomly selecting a feature and a split value within the feature's range, continuing recursively until each instance is isolated in a leaf node. An anomaly score is computed based on the average path length required to isolate an instance across all trees; shorter paths indicate anomalies. This approach offers several advantages relevant to industrial applications:

- Unsupervised learning: No labeled anomaly examples are required for training.
- Efficiency: The algorithm scales linearly with the number of observations and handles high-dimensional feature spaces efficiently.
- Minimal assumptions: It does not assume specific data distributions or require distance metrics that may be problematic in high dimensions.
- Interpretability: The tree structure provides some inherent interpretability, as the features and split points used to isolate an instance can be examined.

Isolation Forest requires comparatively modest computational resources and offers a reasonable balance between performance and interpretability, making it suitable for many industrial settings where real-time processing and limited hardware resources are constraints [20]. Its ability to detect anomalies without labeled data makes it particularly valuable in contexts where failures are rare and varied.

2.3.2 Deep Learning Methods

Deep learning approaches have demonstrated strong performance in anomaly detection tasks across various domains, leading to growing interest in their application to industrial monitoring [25]. These methods leverage artificial neural networks with multiple layers to learn hierarchical representations of data, enabling them to model complex nonlinear relationships and temporal dependencies that classical methods may miss. Autoencoders are among the most widely studied deep learning architectures for unsupervised anomaly detection [29]. An autoencoder consists of an encoder network that compresses input data into a lower-dimensional latent representation and a decoder network that reconstructs the original input from this representation. When trained on normal data, the autoencoder learns to efficiently encode and reconstruct typical patterns. Anomalies, which differ from normal patterns, are poorly reconstructed, resulting in high reconstruction errors that serve as anomaly scores. Variants such as variational autoencoders (VAE) and denoising autoencoders provide additional regularization and robustness. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), are designed to model sequential dependencies in time-series data [13]. These architectures maintain internal memory states that capture temporal context, enabling them to learn patterns that unfold over time. In industrial anomaly detection, LSTMs can be trained to predict future sensor values based on historical observations; large prediction errors indicate deviations from expected temporal behavior. Encoder-decoder LSTM architectures combine reconstruction-based anomaly detection with temporal modeling. Transformer-based architectures, originally developed for natural language processing [35], have more recently been adapted for time-series anomaly detection. Transformers use self-attention mechanisms to weigh the importance of different time steps and features, potentially capturing long-range dependencies more effectively than RNNs. However, their application to industrial monitoring is still emerging and less mature than autoencoder or LSTM-based approaches. Generative Adversarial Networks (GANs) represent another category of deep learning methods for anomaly detection [10]. GANs consist of a generator network that creates synthetic data and a discriminator network that distinguishes real from generated data. When trained on normal data, GANs learn the distribution of normal behavior, and anomalies can be detected as

instances that the discriminator classifies as real but the generator cannot reproduce effectively. Despite their theoretical appeal and strong performance in controlled experimental settings, the practical use of deep learning methods in industrial anomaly detection is limited by several critical constraints: Data Requirements: Deep learning models typically require large labeled datasets for training to achieve good generalization performance [25]. In supervised or semi-supervised settings, this means substantial quantities of labeled anomalies representing diverse failure modes. Even in unsupervised settings, deep models may require large volumes of normal data to learn robust representations. However, anomalies in industrial environments are rare by definition, and comprehensive labeled datasets spanning all possible failure types are generally unavailable. The data that is available may be imbalanced, with normal operations vastly overrepresented compared to fault conditions [12]. Computational Requirements: Training deep neural networks demands significant computational resources, including powerful GPUs, substantial memory, and extended training time. Inference, while generally faster than training, may still impose latency constraints incompatible with real-time monitoring requirements. Many industrial facilities operate with limited computational infrastructure, and deploying deep learning models may necessitate costly hardware upgrades or cloud-based solutions that raise data security and latency concerns. Limited Interpretability: Deep learning models generally function as black-box systems, providing little insight into why a particular instance was flagged as anomalous [1]. The learned representations exist in high-dimensional latent spaces that are difficult for humans to interpret, and the complex interactions between thousands or millions of parameters obscure the decision-making process. In industrial environments, operators and maintenance personnel require explanations to validate alerts, diagnose root causes, and determine appropriate interventions. A model that simply outputs an anomaly score without explaining which sensors, features, or temporal patterns contributed to the decision provides limited actionable value. This lack of transparency can undermine trust and adoption, particularly in safety-critical applications where decisions must be justified and documented. Overfitting and Generalization: Deep models with large parameter spaces are prone to overfitting, especially when training data is limited or does not comprehensively represent the range of normal operational conditions. Models may learn to reconstruct training data patterns perfectly but fail to generalize to new normal variations, resulting in excessive false alarms. Conversely,

they may learn overly general representations that fail to detect subtle anomalies. Proper regularization, validation strategies, and architecture design are essential but challenging in unsupervised settings where ground truth is unavailable [29]. Maintenance and Updating: Industrial processes evolve over time due to equipment changes, product modifications, operational adjustments, and seasonal variations. Deep learning models may require retraining to adapt to these changes, which is resource-intensive and may necessitate downtime or parallel operation. Classical methods are often easier to update incrementally or recalibrate based on new data [17]. Given these limitations, many industrial applications continue to rely on unsupervised or weakly supervised methods based on classical machine learning that require fewer assumptions, operate efficiently under resource constraints, and offer better transparency [5]. While deep learning methods remain an active area of research and may become more practical as computational resources become cheaper and interpretation techniques improve, the current industrial reality often favors simpler, more interpretable approaches that can be deployed, maintained, and trusted by domain experts.

2.4 Challenges in Industrial Applications

Deploying anomaly detection systems in operational industrial environments introduces several practical challenges that extend beyond the theoretical performance of modeling algorithms. These challenges arise from the characteristics of industrial data, the constraints of operational environments, and the requirements of human operators who must interact with and act upon system outputs.

Industrial anomalies are extremely rare, often representing less than 0.1% of observations, which creates extreme class imbalance [12]. This scarcity makes it difficult for supervised models to learn patterns associated with anomalies, and models may simply classify most instances as normal to achieve high accuracy. The limited availability of labeled anomalies also restricts the ability to detect novel or previously unseen failure modes. Evaluation is further complicated, as traditional performance metrics can be misleading in datasets dominated by normal instances.

Sensor measurements in industrial systems are subject to multiple sources of noise and variability. Measurement noise arises from sensor imperfections, electromagnetic interference, quantization errors, and environmental factors. Sen-

sor drift occurs when calibration gradually shifts over time, causing systematic changes that do not reflect actual process variations. Missing values result from sensor failures, communication errors, or maintenance activities, while inconsistent sampling occurs when sensors operate at different frequencies or experience irregular dropouts. These challenges necessitate robust preprocessing, filtering, and modeling techniques that do not overfit to noise while preserving meaningful signals [6].

Many industrial systems operate under strict latency requirements, necessitating real-time or near-real-time anomaly detection to enable timely interventions. Edge devices and programmable logic controllers often have limited memory, processing power, and energy, and cloud-based solutions may be constrained by latency or security concerns. Consequently, models must be lightweight, efficient, and interpretable, with mechanisms for online learning or periodic updating to adapt to evolving process conditions without requiring extensive retraining [21].

Industrial processes are highly interconnected, meaning that anomalies may be contextual rather than observable from individual sensors in isolation. A reading may appear normal during one phase of production but anomalous during another, and relationships between variables may change depending on product type, operational mode, or environmental conditions. Models must capture these multivariate dependencies and contextual relationships, either through sophisticated feature engineering or inherently multivariate modeling approaches [5].

Processes are also non-stationary, evolving over time due to equipment aging, seasonal variations, changes in raw material properties, product mix variations, operator interventions, and intentional process modifications [17]. Static models trained on historical data may become outdated, generating false alarms for legitimate operational patterns or failing to detect new anomalies. Addressing this requires retraining, online adaptation, ensemble approaches capable of handling multiple operational regimes, and robust MLOps practices to manage model monitoring, validation, versioning, deployment, and continuous lifecycle maintenance in production environments. In industrial settings, MLOps plays a critical role in ensuring that models remain reliable after deployment [31]. Continuous monitoring of data distributions and prediction behavior is necessary to detect data drift and concept drift. Automated pipelines for data ingestion, validation, retraining, and testing help maintain performance while reducing man-

ual intervention. Version control for datasets, models, and configurations supports traceability and regulatory compliance, while reproducible training workflows enhance maintainability. Deployment strategies such as staged rollouts, shadow testing, and rollback mechanisms are essential to minimize operational risk. Furthermore, integration with existing industrial IT and OT infrastructures requires secure communication protocols, strict access control, and alignment with cybersecurity policies.

Excessive false alarms directly impact system usability and operator trust. Even models with high specificity may produce many false positives due to extreme class imbalance, leading to alert fatigue and the risk that genuine anomalies are ignored [36]. Balancing sensitivity and specificity requires careful threshold selection, post-processing, and human-in-the-loop verification.

In alignment with Industry 5.0 principles [8], explainability is a core requirement for industrial anomaly detection. Operators must understand why a model flagged an anomaly, which sensors or features contributed to the decision, and how this relates to potential equipment failures or process deviations [1]. Explainable AI supports trust, informed decision-making, and effective human intervention. Transparent models reduce cognitive load, improve situational awareness, and allow maintenance teams to act confidently and efficiently, making anomaly detection both actionable and human-centric [7].

These challenges collectively motivate the use of flexible, unsupervised or weakly supervised techniques that can learn normal behavior directly from data, operate efficiently under resource constraints, adapt to evolving conditions, and provide transparent outputs that support rather than replace human decision-making [29]. The selection of appropriate modeling approaches must balance detection performance with computational efficiency, interpretability, maintainability, and alignment with human-centric operational requirements.

2.5 Evaluation Metrics and Validation Strategies

A recurring challenge in industrial anomaly detection is the absence of labeled anomaly data, which makes traditional evaluation methods, such as accuracy, precision, recall, F1-score, and ROC curves, impossible to compute directly [5]. These metrics require ground truth labels indicating which observations are truly anomalous, information that is typically unavailable in real-world industrial settings where anomalies are rare, diverse, and inconsistently documented.

Without ground truth labels, validation relies on indirect, qualitative, or semi-quantitative strategies that present their own challenges and limitations. An important exception arises in regression-based detection frameworks, where the prediction task itself is quantitatively evaluable even when anomaly labels are absent. The following subsubsections discuss these two complementary evaluation paradigms in turn.

2.5.1 Evaluation Strategies for Unsupervised Anomaly Detection

When anomaly detection is framed as an unsupervised problem, the absence of ground truth labels forces reliance on indirect or qualitative validation approaches, each with distinct strengths and limitations.

Synthetic Anomaly Injection: One approach to evaluation in the absence of real labels involves artificially injecting synthetic anomalies into datasets containing predominantly normal data. These synthetic anomalies can be created by adding noise, introducing sudden shifts, manipulating feature values, or generating outliers based on domain knowledge. The model's ability to detect these injected anomalies provides a quantitative assessment of performance. However, this approach has significant limitations: synthetic anomalies may not accurately represent the characteristics of real faults, which often manifest as subtle, multivariate, and context-dependent patterns rather than obvious outliers [25].

Proxy Labels and Alarm Signals: Some industrial systems maintain existing alarm signals, operator logs, or maintenance records that might serve as proxy indicators of anomalies. In this work, proxy indicators such as existing alarm signals were initially considered to approximate ground truth anomalies. However, these signals did not correlate reliably with meaningful deviations in sensor behavior and were therefore unsuitable as evaluation labels. This disconnect can occur for several reasons: existing alarms may be poorly calibrated, triggering based on simplistic threshold rules that generate many false positives; operators may not consistently log all incidents, particularly minor issues that are quickly resolved; maintenance records may document scheduled activities rather than reactive responses to detected anomalies; and the timing of logged events may not align precisely with the sensor-level deviations that precede observable failures.

Expert-Driven Validation: In the absence of reliable quantitative labels, the assessment of model performance often depends on expert knowledge from domain specialists familiar with the process and its operational dynamics [36]. Subject matter experts, such as experienced operators, process engineers, or maintenance technicians, review detected anomalies and provide qualitative assessments of their validity and significance. Experts can evaluate whether flagged instances correspond to realistic process deviations, whether the timing and context of alerts make operational sense, and whether the features contributing to anomaly scores align with known fault indicators. This evaluation approach, although qualitative, is common and pragmatic in industrial contexts where anomalies are extremely rare or inconsistently recorded. Expert validation provides critical context that quantitative metrics cannot capture, including the operational relevance of detected anomalies, the distinction between true faults and acceptable process variations, and the actionability of alerts from a maintenance perspective.

Interpretability as a Validation Requirement: The reliance on expert-driven validation strongly reinforces the importance of interpretability [1]. Models must provide transparent and explainable outputs that enable human experts to understand why particular instances were flagged as anomalous, which sensors or features contributed most to the anomaly score, and how the detected deviation relates to known operational patterns or failure modes. Without interpretability, experts cannot effectively validate model outputs, and the system becomes a black box that generates alerts without providing the context necessary to determine their validity. Interpretable outputs might include feature importance scores indicating which sensors contributed most to a particular anomaly detection [22], visualizations showing how an anomalous instance differs from typical normal patterns, temporal context showing the evolution of variables leading up to and following a detected anomaly, and comparisons to similar past instances that were or were not confirmed as genuine issues.

Unsupervised Validation Metrics: In fully unsupervised settings, researchers have proposed various metrics that do not require ground truth labels. These include measures of cluster cohesion and separation (e.g., silhouette scores), reconstruction errors in autoencoder-based methods, statistical consistency of anomaly scores across multiple model runs, and stability of detected anomalies across different model hyperparameters [29]. While these metrics provide some guidance, they are indirect indicators that do not directly measure detection per-

formance and may not correlate well with operational utility.

Progressive Deployment and Feedback Loops: A practical strategy for validation in industrial settings involves progressive deployment with continuous feedback collection. The system is initially deployed in monitoring mode, where anomaly alerts are generated but not acted upon automatically. Operators and maintenance personnel review alerts, provide feedback on their validity, and gradually build trust in the system. Over time, this feedback creates a partially labeled dataset that can be used to refine thresholds, adjust model parameters, or retrain models. Eventually, as confidence grows, the system may transition to more active roles, such as triggering automated diagnostic routines or influencing maintenance scheduling.

2.5.2 Regression-Based Anomaly Detection and Residual Analysis

An alternative paradigm for anomaly detection in industrial systems is based on predictive modeling and residual analysis, a strategy that is particularly common in predictive maintenance applications [3]. In this setting, a regression model is trained to predict a target variable, such as temperature, vibration amplitude, pressure, energy consumption, or remaining useful life, using historical sensor data. Unlike fully unsupervised anomaly detection approaches, regression-based methods operate in a supervised learning framework with clearly defined target values. Because the task is formulated as a prediction problem, standard regression performance metrics can be computed directly. Model accuracy can be evaluated using quantitative measures such as the coefficient of determination (R^2), mean squared error (MSE), and root mean squared error (RMSE). These metrics provide a direct assessment of how well the model captures the normal relationships between input features and the target variable, partially alleviating the label scarcity problem that characterizes unsupervised evaluation.

Anomalies are then identified indirectly through residual analysis. The residual is defined as the difference between the observed value and the model's predicted value. Under normal operating conditions, residuals are expected to remain within a predictable range that reflects model uncertainty and inherent process variability [36]. Significant deviations between predictions and actual observations may indicate abnormal system behavior, emerging faults, or pro-

cess disturbances. Thresholds for anomaly detection can be defined statistically (e.g., based on standard deviations of residuals), empirically (based on historical residual distributions), or adaptively through dynamic thresholding techniques. This distinction is important. A model with strong predictive accuracy (high R^2, low RMSE and MSE) establishes a reliable baseline of expected system behavior. Deviations from this baseline become meaningful signals of potential anomalies. Nevertheless, large residuals do not necessarily correspond to genuine faults; they may arise from unmodeled dynamics, regime changes, sensor noise, or shifts in operating conditions. Therefore, residual-based anomaly detection still requires contextual interpretation, expert validation, or supplementary diagnostic analysis. However, although regression performance can be rigorously quantified, the accuracy of anomaly detection itself cannot be directly computed in the absence of labeled anomaly data. Metrics such as precision, recall, F1-score, or ROC curves remain unavailable without ground truth labels indicating which residual excursions correspond to true faults. Consequently, evaluation focuses on two complementary aspects: how accurately the model predicts future behavior under normal conditions, and how significantly and consistently observed data deviate from those predictions. In practice, regression-based anomaly detection provides a structured and interpretable framework. It links anomaly identification directly to deviations from learned physical or statistical relationships, making explanations more intuitive for operators [22]. Residual trends over time can also reveal gradual degradation patterns, supporting condition-based maintenance strategies [18]. Despite its advantages, this approach remains subject to the broader validation challenges discussed in the preceding subsubsection, particularly the absence of labeled anomalies and the need for human-centered interpretation of detected deviations. The evaluation challenge in industrial anomaly detection thus extends beyond selecting appropriate metrics to encompass the broader question of how to validate and deploy systems in environments where ground truth is scarce, stakes are high, and human expertise remains essential [5]. This reality necessitates a human-centric design philosophy where anomaly detection systems are conceived not as autonomous decision-makers but as tools that augment human capabilities through transparent, interpretable, and actionable insights [8].

3 Data Description

3.1 Data Source and Collection Process

3.1.1 Historical Data Extraction (Phase I)

Initial model training and exploratory analysis were conducted using a dataset spanning a one-year period. This data was retrieved via structured queries (SQL) from the company's primary relational databases and exported into a standardized CSV format for offline processing. This historical snapshot provided the necessary volume to capture seasonal trends and diverse machine states within the industrial environment.

3.1.2 Validation and Testing Set (Phase II)

To evaluate the robustness and generalizability of the trained models, a secondary two-month "unseen" dataset was extracted. This recent batch reflects current operational conditions and serves as the primary benchmark for testing the anomaly detection system's performance against real-time drift or recent mechanical changes.

3.1.3 Deployment and Connectivity Roadmap

While current analysis relies on batch processing, the research framework is designed for eventual integration into the factory's live ecosystem. Two primary deployment paths are being considered:

1. Remote Integration: Establishing a secure Virtual Private Network (VPN) tunnel to the company's internal servers to allow for periodic automated data polling.
2. Edge/On-Premise Deployment: Porting the anomaly detection and monitoring systems directly onto the company's local server infrastructure. This approach aligns with Industry 5.0 principles by reducing latency and ensuring data sovereignty and security within the plant's own firewall.

3.2 Description of Sensors, Variables, and Operational Context

3.2.1 F1

Name	Description	Type
allarme_dati_memoria_errati	Allarme dati in memoria errati	BOOL
allarme_fiamma_bruciatore	Allarme fiamma bruciatore	BOOL
allarme_inter_comunicazione	Allarme inter-comunicazione controllore	BOOL
allarme_mancanza_tensione	Allarme mancanza di tensione	BOOL
allarme_movimento_carrello	Allarme movimento carrello	BOOL
allarme_porta_aperta_riscaldamento	Allarme porta aperta in riscaldamento	BOOL
allarme_porta_aperta_start	Allarme porta aperta allo START	BOOL
allarme_problemi_riscaldamento	Allarme problemi di riscaldamento	BOOL
allarme_protezione_motori	Allarme protezione motori	BOOL
allarme_riserva 1	Primo slot per un allarme inseribile in un secondo momento	BOOL
allarme_riserva 2	Secondo slot per un allarme inseribile in un secondo momento	BOOL
allarme_sonda_temperatura 1	Allarme sonda di temperatura 1 guasta	BOOL
allarme_sonda_temperatura 2	Allarme sonda di temperatura 2 guasta	BOOL
allarme_sovratemperatura_forno	Allarme sovratemperatura forno	BOOL
allarme_temperatura_bassa	Allarme temperatura bassa per generazione vapore	BOOL
allarme_termostato_sicurezza	Allarme termostato di sicurezza	BOOL
fase_attiva	La fase del ciclo di cottura del forno	INT
ricetta	Numero Programma (Ricetta)	INT
seq_avvio_ciclo	Stato di esecuzione del comando	INT
set_attivo_temperatura	Set attivo temperatura °C	FLOAT
stato_esecuzione_comando	Stato di esecuzione del comando (attivo o non attivo)	BOOL
stato_funzionamento	Stato di funzionamento macchina	INT
stato_macchina 1	Stato macchina	INT
stato_macchina 2	Stato Macchina	INT
temperatura_visualizzata	Temperatura visualizzata sullo schermo del pannello °C	FLOAT
sonda_1_f1	Lettura Sonda 1 Forno °C	FLOAT
sonda_2_f1	Lettura Sonda 2 (Forno o Vaporiera) °C	FLOAT
temperatura_visualizzata_f1	Temperatura visualizzata sullo schermo del pannello °C	FLOAT

Table 3.1: F1_131

3.2.2 F2

Name	Description	Type
allarme_dati_memoria_errati	Allarme dati in memoria errati	BOOL
allarme_fiamma_bruciatore	Allarme fiamma bruciatore	BOOL
allarme_inter_comunicazione	Allarme inter-comunicazione controllore	BOOL
allarme_mancanza_tensione	Allarme mancanza di tensione	BOOL
allarme_movimento_carrello	Allarme movimento carrello	BOOL
allarme_porta_aperta_riscaldamento	Allarme porta aperta in riscaldamento	BOOL
allarme_porta_aperta_start	Allarme porta aperta allo START	BOOL
allarme_problemi_riscaldamento	Allarme problemi di riscaldamento	BOOL
allarme_protezione_motori	Allarme protezione motori	BOOL
allarme_riserva 1	Primo slot per un allarme inseribile in un secondo momento	BOOL
allarme_riserva 2	Secondo slot per un allarme inseribile in un secondo momento	BOOL
allarme_sonda_temperatura 1	Allarme sonda di temperatura 1 guasta	BOOL
allarme_sonda_temperatura 2	Allarme sonda di temperatura 2 guasta	BOOL
allarme_sovratemperatura_forno	Allarme sovratemperatura forno	BOOL
allarme_temperatura_bassa	Allarme temperatura bassa per generazione vapore	BOOL
allarme_termostato_sicurezza	Allarme termostato di sicurezza	BOOL
fase_attiva	La fase del ciclo di cottura del forno	INT
ricetta	Numero Programma (Ricetta)	INT
seq_avvio_ciclo	Stato di esecuzione del comando	INT
set_attivo_temperatura	Set attivo temperatura °C	Float
stato_esecuzione_comando	Stato di esecuzione del comando (attivo o non attivo)	BOOL
stato_funzionamento	Stato di funzionamento macchina	INT
stato_macchina 1	Stato macchina	INT
stato_macchina 2	Stato Macchina	INT
temperatura_visualizzata	Temperatura visualizzata sullo schermo del pannello °C	Float
sonda_1_f2	Lettura Sonda 1 Forno °C	Float
sonda_2_f2	Lettura Sonda 2 (Forno o Vaporiera) °C	Float
temperatura_visualizzata_f2	Temperatura visualizzata sullo schermo del pannello °C	Float

Table 3.2: F2_132

3.2.3 F3

Name	Description	Type
allarme_dati_memoria_errati	Allarme dati in memoria errati	BOOL
allarme_fiamma_bruciatore	Allarme fiamma bruciatore	BOOL
allarme_inter_comunicazione	Allarme inter-comunicazione controllore	BOOL
allarme_mancanza_tensione	Allarme mancanza di tensione	BOOL
allarme_movimento_carrello	Allarme movimento carrello	BOOL
allarme_porta_aperta_riscaldamento	Allarme porta aperta in riscaldamento	BOOL
allarme_porta_aperta_start	Allarme porta aperta allo START	BOOL
allarme_problemi_riscaldamento	Allarme problemi di riscaldamento	BOOL
allarme_protezione_motori	Allarme protezione motori	BOOL
allarme_riserva 1	Primo slot per un allarme inseribile in un secondo momento	BOOL
allarme_riserva 2	Secondo slot per un allarme inseribile in un secondo momento	BOOL
allarme_sonda_temperatura 1	Allarme sonda di temperatura 1 guasta	BOOL
allarme_sonda_temperatura 2	Allarme sonda di temperatura 2 guasta	BOOL
allarme_sovratemperatura_forno	Allarme sovratemperatura forno	BOOL
allarme_temperatura_bassa	Allarme temperatura bassa per generazione vapore	BOOL
allarme_termostato_sicurezza	Allarme termostato di sicurezza	BOOL
fase_attiva	La fase del ciclo di cottura del forno	INT
ricetta	Numero Programma (Ricetta)	INT
seq_avvio_ciclo	Stato di esecuzione del comando	INT
set_attivo_temperatura	Set attivo temperatura °C	Float
stato_esecuzione_comando	Stato di esecuzione del comando (attivo o non attivo)	BOOL
stato_funzionamento	Stato di funzionamento macchina	INT
stato_macchina 1	Stato macchina	INT
stato_macchina 2	Stato Macchina	INT
temperatura_visualizzata	Temperatura visualizzata sullo schermo del pannello °C	Float
sonda_1_f3	Lettura Sonda 1 Forno °C	Float
sonda_2_f3	Lettura Sonda 2 (Forno o Vaporiera) °C	Float
temperatura_visualizzata_f3	Temperatura visualizzata sullo schermo del pannello °C	Float

Table 3.3: F3_133

3.2.4 BL

Name	Description	Type
IDRicettaAttuale	Numero di programma memorizzato per impastare il pane	INT
IDRicetta_Write	Numero di programma passato per la prossima ricetta	INT
NomeRicettaAttuale	Nome della ricetta o commessa attualmente in uso	STRING
NomeRicetta_Write	Nome della ricetta o commessa da fare come successiva	STRING
g_xAlarm00	Allarme generico	BOOL
g_xAlarmFungo	Allarme che indica se il fungo di emergenza è premuto	BOOL
g_xAlarmGenerale	Allarme generale	BOOL
g_xAlarmInterventoEmergenze	Allarme intervento emergenze	BOOL
g_xAlarmInverter	Allarme inverter	BOOL
g_xAlarmMagnetoTermico	Allarme magnetotermico	BOOL
g_xAlarmMancaAria	Allarme mancanza aria	BOOL
g_xAlarmRiparo 1	Allarme riparo	BOOL
g_xInterventoMagnetotermico	Allarme che indica un intervento sul magnetotermico	BOOL
g_xLetturaVa	Non specificato nei documenti	BOOL
iOrdineAnnullato	Ordine annullato	BOOL
iStatoMacchina	Stato della macchina	INT
iVelSpira	Velocità della spirale	INT
iVelSpiraManuale1	Velocità manuale della spirale	INT

Table 3.4: BL_140

3.2.5 C1

Name	Description	Type
B_Allarme_Pres_Min_Max	Allarme pressostato pressioni	BOOL
B_Allarme_Rele_Termico	Allarme termico compressore	BOOL
B_Allarme_Sonda_Ambiente	Allarme sonda ambiente	BOOL
B_Allarme_Sonda_Sbrinamento	Allarme sonda sbrinamento	BOOL
B_Allarme_Sonda_Ur	Allarme sonda umidità	BOOL
B_Allarme_Termico_Res_Caldo	Allarme termico resistenze caldo	BOOL
B_Allarme_Termico_Res_Sbrina	Allarme termico sbrinamento	BOOL
B_Allarme_Termico_Res_Ur	Allarme termico umidità	BOOL
B_Allarme_Termico_Ventole_Evap	Allarme termico ventilatori interni	BOOL
B_Allarme_Termostato_Ur	Allarme sicurezza umidità	BOOL
B_OUT_Aspiratore	Attuazione estrattore umidità	BOOL
B_OUT_Vent_Condensatore	Attuazione ventilatore condensatore	BOOL
B_Visual_Allarmi	Allarme generale	BOOL
Out_Compressore	Attuazione compressore freddo	BOOL
Out_Lievitazione	Attuazione resistenze caldo	BOOL
Out_Luce_Cella	Attuazione luce interna	BOOL
Out_Sbrinamento	Attuazione resistenze sbrinamento	BOOL
Out_Umidita	Attuazione generatore umidità	BOOL
Out_Ventilatori	Attuazione ventilatori interni	BOOL
Pulsante_Avvio_Ciclo	Stato del pulsante di avvio ciclo della cella	BOOL
Richiesta_cambio_ricetta	Stato della richiesta di cambio ricetta	BOOL
Richiesta_cambio_setpoint_temperatura	Stato della richiesta cambio setpoint temperatura	BOOL
Richiesta_cambio_setpoint_umidita	Stato della richiesta cambio setpoint umidità	BOOL
Numero_ricetta_richiesto	Numero ricetta richiesto	INT
Setpoint_temperatura_richiesto	Setpoint temperatura richiesto	FLOAT
Setpoint_umidita_richiesto	Setpoint umidità richiesto	FLOAT
W_Selezione_Prg	Selezione programma	INT
W_Set_Temp	Temperatura °C	FLOAT
W_Set_Ur	Umidità RH%	FLOAT
W_Sonda_Ambiente_Cal	Temperatura cella °C	FLOAT
W_Sonda_Sbrinamento_Cal	Temperatura evaporatore	FLOAT
W_Sonda_Ur_Cal	Umidità RH%	FLOAT

Table 3.5: C1_138

3.2.6 C2

Name	Description	Type
B_Allarme_Pres_Min_Max	Allarme pressostato pressioni	BOOL
B_Allarme_Rele_Termico	Allarme termico compressore	BOOL
B_Allarme_Sonda_Ambiente	Allarme sonda ambiente	BOOL
B_Allarme_Sonda_Sbrinamento	Allarme sonda sbrinamento	BOOL
B_Allarme_Sonda_Ur	Allarme sonda umidità	BOOL
B_Allarme_Termico_Res_Caldo	Allarme termico resistenze caldo	BOOL
B_Allarme_Termico_Res_Sbrina	Allarme termico sbrinamento	BOOL
B_Allarme_Termico_Res_Ur	Allarme termico umidità	BOOL
B_Allarme_Termico_Ventole_Evap	Allarme termico ventilatori interni	BOOL
B_Allarme_Termostato_Ur	Allarme sicurezza umidità	BOOL
B_OUT_Aspiratore	Attuazione estrattore umidità	BOOL
B_OUT_Vent_Condensatore	Attuazione ventilatore condensatore	BOOL
B_Visual_Allarmi	Allarme generale	BOOL
Out_Compressore	Attuazione compressore freddo	BOOL
Out_Lievitazione	Attuazione resistenze caldo	BOOL
Out_Luce_Cella	Attuazione luce interna	BOOL
Out_Sbrinamento	Attuazione resistenze sbrinamento	BOOL
Out_Umidita	Attuazione generatore umidità	BOOL
Out_Ventilatori	Attuazione ventilatori interni	BOOL
Pulsante_Avvio_Ciclo	Stato del pulsante di avvio ciclo della cella	BOOL
Richiesta_cambio_ricetta	Stato della richiesta di cambio ricetta	BOOL
Richiesta_cambio_setpoint_temperatura	Stato della richiesta cambio setpoint temperatura	BOOL
Richiesta_cambio_setpoint_umidita	Stato della richiesta cambio setpoint umidità	BOOL
Numero_ricetta_richiesto	Numero ricetta richiesto	INT
Setpoint_temperatura_richiesto	Setpoint temperatura richiesto	FLOAT
Setpoint_umidita_richiesto	Setpoint umidità richiesto	FLOAT
W_Selezione_Prg	Selezione programma	INT
W_Set_Temp	Temperatura °C	FLOAT
W_Set_Ur	Umidità RH%	FLOAT
W_Sonda_Ambiente_Cal	Temperatura cella °C	FLOAT
W_Sonda_Sbrinamento_Cal	Temperatura evaporatore	FLOAT
W_Sonda_Ur_Cal	Umidità RH%	FLOAT

Table 3.6: C2_139

3.3 Data Preprocessing

3.3.1 Overview: Concept and Motivation

Data preprocessing constitutes a fundamental stage in the data analysis and machine learning pipeline, encompassing a set of techniques and procedures applied to raw data to transform it into a clean, consistent, and structured format suitable for analysis and model development. In real-world applications, data are rarely collected in a form directly amenable to modeling, as they frequently contain noise, inconsistencies, missing values, and irrelevant information.

The primary objective of data preprocessing is to enhance data quality. High-quality data are essential for constructing reliable analytical models, as the performance and validity of data-driven methods depend critically on the char-

acteristics of the input data. Inadequately prepared data can result in biased outcomes, diminished predictive performance, and erroneous conclusions, irrespective of the sophistication of the modeling techniques employed.

Data preprocessing also ensures compatibility between the data and the underlying assumptions of the selected analytical or machine learning methods. Many algorithms require data to conform to specific formats or statistical properties, such as normalized scaling, absence of missing values, or consistent feature distributions. By addressing these requirements during preprocessing, the risks of model instability and convergence issues are mitigated.

Furthermore, preprocessing enhances both the efficiency and interpretability of data analysis. By eliminating redundant features, rectifying inconsistencies, and standardizing representations, preprocessing reduces computational complexity and facilitates clearer interpretation of model outputs. Consequently, data preprocessing transcends its role as a mere preparatory step and emerges as a critical component directly influencing the success of the entire data science workflow.

Following the general principles outlined above, a systematic series of preprocessing operations was applied to the dataset to ensure consistency, reliability, and suitability for subsequent time series analysis. These operations are described in the sequence in which they were executed.

3.3.2 Removal of Irrelevant Features

Columns containing non-informative or meaningless values were systematically removed from the dataset. These included signals exhibiting constant values, empty entries, random text strings, or features lacking relevance to the analytical objectives. The elimination of such features reduces noise, decreases computational complexity, and improves model interpretability.

Additionally, the dataset exhibited occasional spikes exceeding the physical constraints of the machinery. These anomalies were classified as point outliers attributable to sensor malfunctions rather than genuine physical phenomena. Following consultation with domain experts from Sottoriva and Colip and analysis of warning and alarm sensors in both fermentation chambers (*celle di lievitazione*), several malfunctioning sensors were identified and their data streams excluded.

A significant challenge identified during the data audit was the presence of

non-numeric artifacts within otherwise numeric telemetry streams, necessitating data type validation and cleansing procedures.

3.3.3 Removal of Duplicate Timestamps

Duplicate records corresponding to identical timestamps were identified and removed. Duplicate temporal entries can distort temporal patterns and introduce bias into time series models; therefore, ensuring a one-to-one mapping between timestamps and observations is essential for accurate analysis.

A specific anomaly was observed wherein multiple data points were recorded for a single timestamp. This "double-entry" phenomenon typically stems from database synchronization conflicts or sensor instability. Table **??** illustrates an example of this duplication issue.

_field	_value	name	timestamp	day
temperatura_visualizzata	214.9	F1_131	2025-01-17 09:22:00	2025-01-17
temperatura_visualizzata	217.4	F1_131	2025-01-17 09:22:00	2025-01-17

Table 3.7: Example of Duplicated Samples

Given that the variance between duplicated values was statistically negligible, a deduplication strategy was implemented whereby the system retained the first entry and discarded subsequent duplicates, thereby maintaining a strictly monotonic time index.

3.3.4 Target Signal Selection

Primary signals of interest were identified and designated as target variables for subsequent analysis, modeling, and monitoring. This step clearly delineates the prediction or monitoring objectives and distinguishes target signals from auxiliary or explanatory features.

3.3.5 Temporal Indexing and Resampling

The timestamp column was designated as the dataframe index, thereby transforming the dataset into a time series-oriented structure. Employing a timestamp index enables chronological ordering of observations and facilitates

the application of time-based operations such as resampling, interpolation, and temporal aggregation, which are essential for time series analysis.

The complete dataset spans an observation period of 14 months, partitioned into a training and development period comprising 12 months of historical data employed to capture annual seasonality and baseline operational behavior, and a validation and testing period consisting of 2 consecutive months of recent data used to simulate a live deployment scenario.

The raw data were originally captured at a high-resolution frequency of one sample per minute, with sampling period $T_s = 1$ min. While high-frequency data are valuable for detecting instantaneous anomalies, they present challenges for long-term trend analysis, including elevated noise levels and substantial storage requirements.

To optimize the dataset for anomaly detection and long-term monitoring, a downsampling procedure was implemented. Data were aggregated using an hourly average, reducing the dataset size by a factor of 60 while attenuating high-frequency stochastic noise often irrelevant to mechanical degradation processes. Preliminary analysis compared different aggregation intervals. As illustrated in Figures 3.1 and 3.2, a 10-minute interval provides higher granularity but retains greater noise levels, whereas the hourly interval more clearly reveals underlying operational cycles and trends.

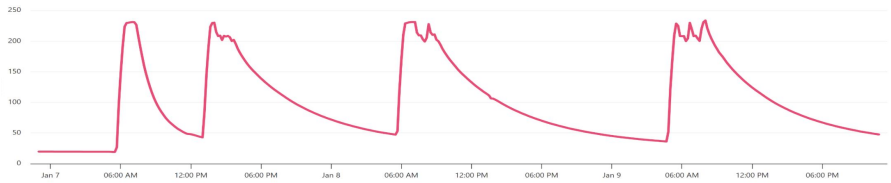

Figure 3.1: 10-Minute Sampling Intervals

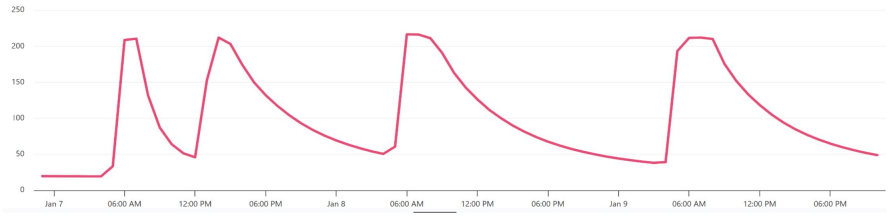

Figure 3.2: Hourly Sampling Intervals

The system architecture maintains flexibility for future adaptations. While hourly intervals prove sufficient for current monitoring objectives, the pipeline

is designed to accommodate shorter intervals, such as 5-minute or 10-minute aggregations, should specific use cases, such as detecting rapid thermal fluctuations, require enhanced temporal sensitivity.

3.3.6 Handling Missing Values

Following the resampling procedure, missing values and temporal gaps in the time series were identified and addressed using context-appropriate techniques. Two distinct types of missing data patterns were detected: point-wise missingness, consisting of isolated single missing values, and block missingness, comprising extended gaps spanning several hours.

Depending on the nature and extent of missingness, values were either interpolated to preserve temporal continuity or removed via row-wise deletion. This approach ensures data integrity while minimizing the introduction of bias or artificial patterns into the dataset.

3.4 Clean Data Summary

3.4.1 F1_131

F1_131 consists of 28 signals, of which 20 do not convey meaningful information. These signals are either predominantly constant, redundant, or duplicates of other measurements, and therefore do not provide independent value to the system.

For the remaining signals, an initial exploratory data analysis (EDA) was conducted, followed by a preliminary application of the Isolation Forest algorithm combined with AcME explainable AI techniques. Based on these analytical results, as well as discussions and consultations with SMACT engineers and technical staff incorporating their domain expertise, a subset of signals was selected and retained as informative and relevant features for subsequent analysis.

- stato_funzionamento
- ricetta
- set_attivo_temperatura
- temperatura_visualizzata
- fase_attiva

- seq_avvio_ciclo

Categorical signals are 'stato_funzionamento','ricetta','fase_attiva','seq_avvio_ciclo' and 'set_attivo_temperatura','temperatura_visualizzata' are numerical signals.

stato_funzionamento		ricetta		fase_attiva		seq_avvio_ciclo	
Value	dist.	Value	dist.	Value	dist.	Value	dist.
0.0	0.911005	0.0	0.606850	1.0	0.872329	0.0	0.929303
1.0	0.056828	2.0	0.392768	14.0	0.100885	1.0	0.037534
2.0	0.029644	1.0	0.000352	2.0	0.015519	2.0	0.031841
4.0	0.001859	3.0	0.000029	3.0	0.010933	4.0	0.000644
5.0	0.000340			13.0	0.000292	5.0	0.000340
6.0	0.000292			4.0	0.000027	6.0	0.000292
3.0	0.000017			15.0	0.000015	7.0	0.000046
7.0	0.000015						

Table 3.8: Categorical EDA - F1

	set_attivo_temperatura	temperatura_visualizzata
count	7201.000000	7201.000000
mean	212.229162	60.317964
std	12.028047	56.884002
min	0.000000	15.300000
25%	212.000000	24.200000
50%	212.000000	31.190000
75%	212.000000	72.275000
max	250.000000	251.400000

Table 3.9: Numerical EDA - F1

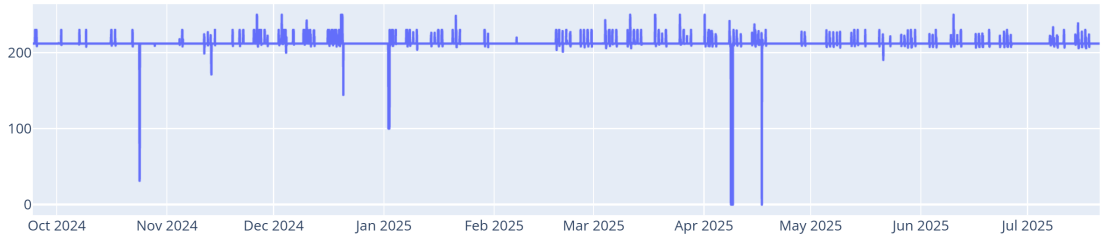

Figure 3.3: set_attivo_temperatura - F1

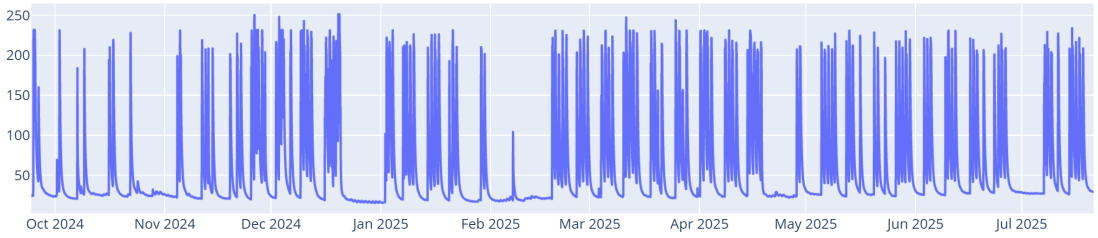

Figure 3.4: temperatura_visualizzata - F1

3.4.2 F2_132

F2_132 exhibits a high degree of similarity to F1_131. Accordingly, the same methodological procedure was applied, including exploratory data analysis, anomaly detection modeling, and consultation with domain experts. This process led to the selection of an equivalent subset of informative signals for this module.

- stato_funzionamento
- ricetta
- set_attivo_temperatura
- temperatura_visualizzata
- fase_attiva
- seq_avvio_ciclo

Categorical signals are 'stato_funzionamento','ricetta','fase_attiva','seq_avvio_ciclo' and 'set_attivo_temperatura','temperatura_visualizzata' are numerical signals.

stato_funzionamento		ricetta		fase_attiva		seq_avvio_ciclo	
Value	dist.	Value	dist.	Value	dist.	Value	dist.
0.0	0.896529	0.0	0.619949	1.0	0.932693	0.0	0.917490
1.0	0.066837	4.0	0.375847	14.0	0.035214	1.0	0.043440
2.0	0.035540	2.0	0.004204	2.0	0.018042	2.0	0.037471
4.0	0.000511			3.0	0.013579	6.0	0.000701
5.0	0.000360			4.0	0.000290	4.0	0.000477
6.0	0.000183			13.0	0.000183	5.0	0.000370
3.0	0.000039					7.0	0.000051

Table 3.10: Categorical EDA - F2

	set_attivo_temperatura	temperatura_visualizzata
count	7201.000000	7201.000000
mean	212.556581	65.043008
std	11.160248	58.535476
min	0.000000	15.400000
25%	212.000000	25.093333
50%	212.000000	36.290000
75%	212.000000	82.945000
max	250.000000	251.118333

Table 3.11: Numerical EDA - F2

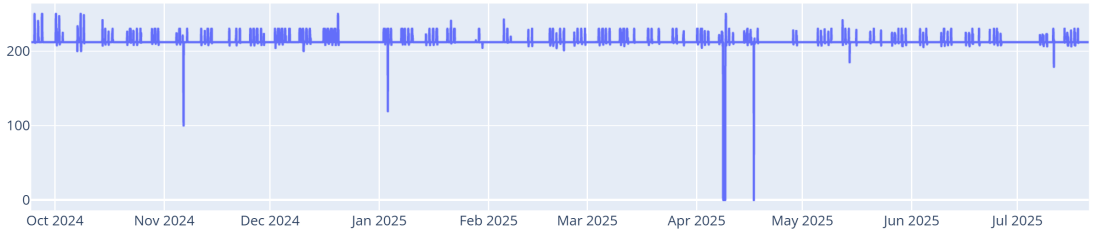

Figure 3.5: set_attivo_temperatura - F2

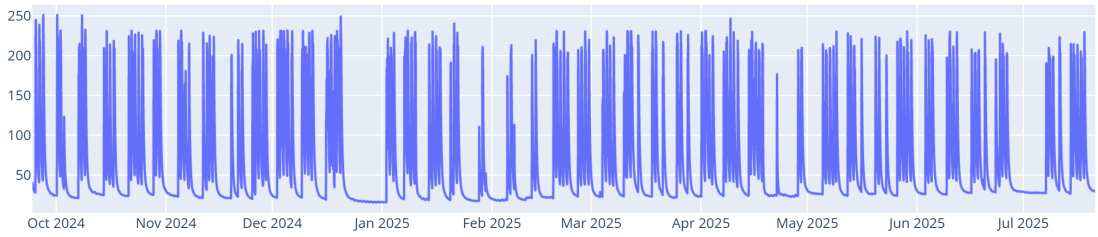

Figure 3.6: temperatura_visualizzata - F2

3.4.3 F3_133

F3_133 demonstrates similar characteristics to both F1_131 and F2_132. Its data distribution and signal behavior closely resemble those of the previously analyzed modules, as illustrated in the tables and figures presented below.

- stato_funzionamento
- ricetta
- set_attivo_temperatura

- temperatura_visualizzata
- fase_attiva
- seq_avvio_ciclo

Categorical signals are 'stato_funzionamento','ricetta','fase_attiva','seq_avvio_ciclo' and 'set_attivo_temperatura','temperatura_visualizzata' are numerical signals.

stato_funzionamento		ricetta		fase_attiva		seq_avvio_ciclo	
Value	dist.	Value	dist.	Value	dist.	Value	dist.
0.0	0.893389	0.0	0.683482	1.0	0.900364	0.0	0.913839
1.0	0.065018	2.0	0.302966	14.0	0.066491	1.0	0.043300
2.0	0.036805	1.0	0.006722	2.0	0.018918	2.0	0.039167
4.0	0.002852	7.0	0.006310	3.0	0.013879	4.0	0.001788
5.0	0.001320	6.0	0.000382	13.0	0.000303	5.0	0.001337
3.0	0.000313	5.0	0.000138	5.0	0.000042	6.0	0.000569
6.0	0.000303			4.0	0.000002		

Table 3.12: Categorical EDA - F3

	set_attivo_temperatura	temperatura_visualizzata
count	7201.000000	7201.000000
mean	211.815636	68.181824
std	12.784208	59.207768
min	0.000000	15.700000
25%	212.000000	25.800000
50%	212.000000	41.540000
75%	212.000000	90.236667
max	250.000000	234.821667

Table 3.13: Numerical EDA - F3

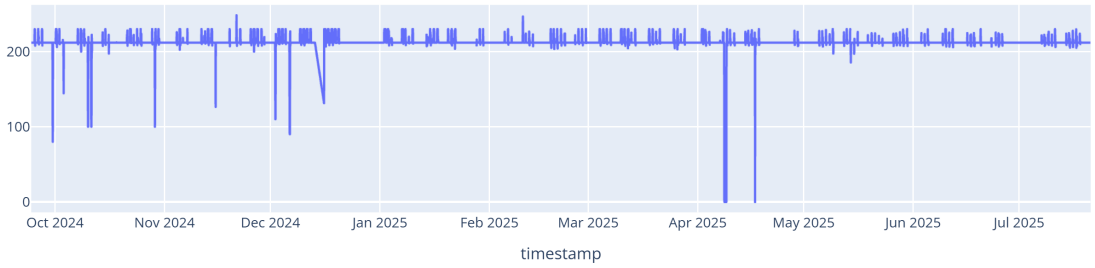

Figure 3.7: set_attivo_temperatura - F3

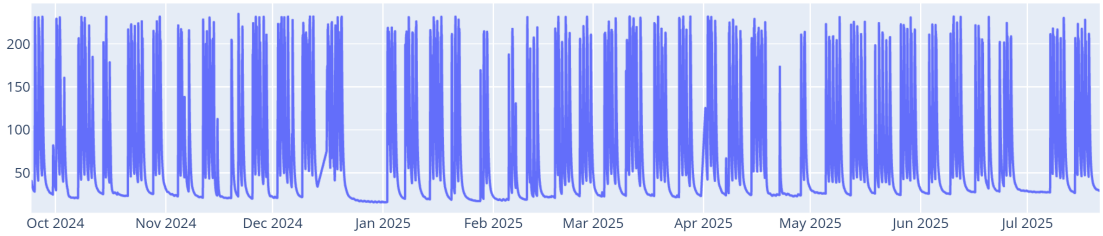

Figure 3.8: temperatura_visualizzata - F3

3.4.4 BL_140

BL_140 consists of 18 signals, of which 9 do not convey meaningful information. These signals are largely constant, redundant, or duplicates of other measurements, and therefore do not provide independent value to the system.

For the remaining signals, an exploratory data analysis (EDA) was conducted, followed by an initial application of the Isolation Forest algorithm combined with AcME explainable AI techniques. Based on the analytical results, as well as discussions and consultations with SMACT engineers and technical staff incorporating their domain expertise, a subset of signals was selected and retained as informative and relevant features for subsequent analysis.

- g_xAlarmMancaAria
- iStatoMacchina
- iVelSpira
- iVelSpiraManuale1

The numerical signals are 'iVelSpiraManuale1','iVelSpira' and 'g_xAlarmMancaAria','iStatoMacchina' are categorical.

g_xAlarmMancaAria		iStatoMacchina	
Value	dist.	Value	dist.
0.0	0.999656	2.0	0.497490
1.0	0.000344	0.0	0.455255
		1.0	0.047255

Table 3.14: Categorical EDA - BL

	iVelSpiraManuale1	iVelSpira
count	7084.000000	7084.000000
mean	79.866867	38.537032
std	20.873572	5.141634
min	19.233333	13.000000
25%	65.000000	40.000000
50%	70.000000	40.000000
75%	100.000000	40.000000
max	100.000000	50.000000

Table 3.15: Numerical EDA - BL

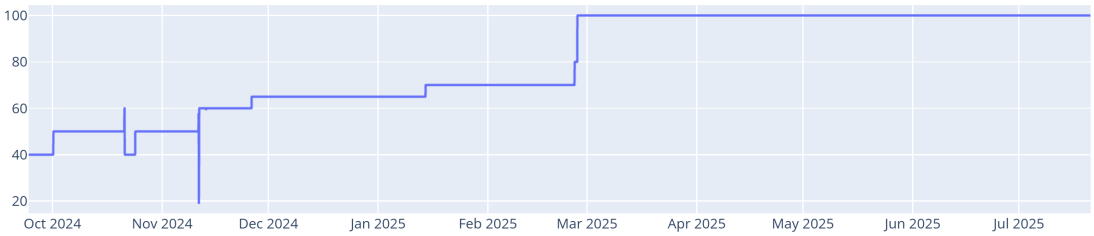

Figure 3.9: iVelSpiraManuale1 - BL

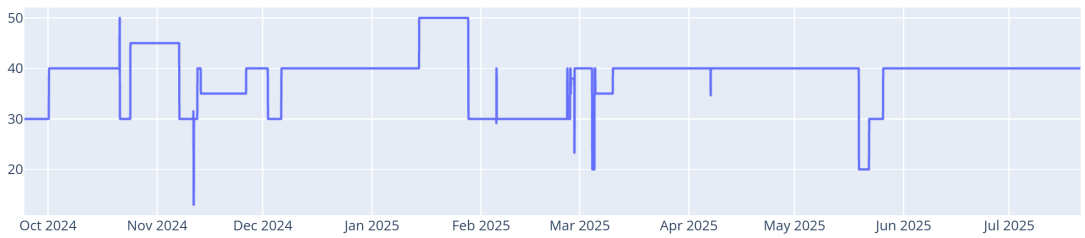

Figure 3.10: iVelSpira - BL

3.4.5 IS_141, C1_138, AND C2_139

IS was not extensively utilized during the data collection period, as it is reserved primarily for special recipes and occasional production runs. Consequently, the available dataset was limited, and only minimal domain-specific feedback could be obtained regarding its operational behavior.
Furthermore, C1 and C2 experienced sensor malfunction issues during the observation period, as confirmed by the company technician. Due to these hardware irregularities, it was not possible to reliably distinguish between valid measurements and deviations caused by sensor faults. As a result, the collected data from these sensors could not be considered trustworthy for analytical purposes.

Therefore, further analysis was postponed until the hardware and internal system issues are fully resolved. Future work will focus on evaluating the cleaned and validated dataset once reliable measurements are ensured.

4 Methodology

4.1 Feature Engineering

We performs comprehensive feature engineering on a multivariate time series dataset containing both categorical and numerical variables. The objective is to transform raw temporal signals into a rich feature set suitable for machine learning models by incorporating temporal dynamics, interactions, frequency-domain characteristics, and calendar effects.

4.1.1 Overview

The engineered features can be grouped into the following categories:

- Categorical feature encoding and aggregation
- Alarm-based features
- Numerical time-domain features
- Numerical cross-signal interaction features
- Frequency-domain features
- Time-based and cyclic features

4.1.2 Categorical Feature Engineering

Categorical variables in industrial time series data commonly represent machine states, operational modes, or discrete events rather than continuous physical quantities. As a result, these variables cannot be directly processed by most machine learning algorithms and must be transformed into numerical representations [4]. This transformation must preserve both the semantic meaning of the categories and their temporal behavior in order to remain informative for downstream modeling tasks.

In this work, each categorical signal is first isolated from the raw dataset prior to any temporal resampling. Treating each categorical variable as an independent time series ensures that its original temporal ordering is preserved and that subsequent aggregation operations do not introduce misalignment or unintended interactions between signals. This step is particularly important in industrial settings, where categorical state changes may occur asynchronously and at irregular intervals.

Once isolated, categorical values are transformed using one-hot encoding. For each unique category within a given signal, a binary indicator variable is created, with column names prefixed by the original signal name to avoid ambiguity when multiple categorical features are present. For example, a signal describing machine state with possible values `ON`, `OFF`, and `IDLE` is represented by three binary variables, each indicating the presence or absence of the corresponding state at a given timestamp. This encoding allows categorical information to be expressed in a numerical format compatible with machine learning models while preserving the full categorical semantics.

One-hot encoding is explicitly chosen over label encoding due to the temporal aggregation applied later in the pipeline [4]. Label encoding introduces artificial ordinal relationships between categories, which are not meaningful for nominal machine states. When such encoded values are averaged during resampling, the resulting numerical values lose their interpretability and no longer correspond to valid system states. In contrast, averaging one-hot encoded binary indicators produces meaningful quantities that represent the proportion of time spent in each category within a given time window, thereby preserving the original semantic interpretation of the data.

After encoding each categorical signal, all one-hot encoded features are concatenated into a single unified categorical feature matrix indexed by time. This

unified representation enables joint modeling of multiple machine modes and discrete events while maintaining a consistent temporal structure across all categorical variables.

To align categorical features with the hourly numerical features used in the model, the unified categorical matrix is resampled to an hourly frequency. Mean aggregation is applied to each one-hot encoded variable. For binary indicators, the resulting hourly mean corresponds to the fraction of time during which a particular category was active within the hour. This representation captures not only the presence of a given state but also its duration, allowing partial occupancy of states within an aggregation window to be explicitly modeled.

Following resampling, some hourly timestamps may contain missing values due to sparse event logging or gaps in categorical state reporting. These missing values are handled using the same strategy described in the preprocessing section, ensuring temporal continuity and consistency across both categorical and numerical features.

As a result of this feature engineering process, categorical variables are transformed into a numerical, time-indexed feature matrix that is fully aligned with the hourly numerical dataset. The engineered features capture both state occurrence and state duration, enabling machine learning models to learn operating modes, discrete process configurations, and regime-dependent behavior over time in a principled and interpretable manner.

4.1.3 Alarm-Based Features

Alarm signals play a crucial role in industrial time series analysis, as they explicitly encode abnormal operating conditions, safety warnings, or deviations from nominal process behavior [36]. Unlike continuous sensor measurements, alarm variables typically represent discrete events or binary indicators whose presence often correlates with process instability, faults, or transitional operating regimes. For this reason, alarms are treated as a distinct feature group and are engineered to provide both localized and global information about system health.

In this feature engineering pipeline, alarm-related information is derived from the hourly dataset and incorporated in two complementary forms: individual alarm indicators and an aggregated alarm intensity measure. This dual

representation allows the model to capture both the specific impact of individual alarms and the overall level of alarm activity within a given time interval.

Each alarm signal listed in the Alarms set is directly included as a separate feature. These signals are assumed to be already aligned at an hourly resolution and typically take binary or count-based values, indicating whether a particular alarm was active during the corresponding time interval. By preserving individual alarm channels, the model is able to learn alarm-specific effects, such as the influence of certain fault conditions or warnings on the target variable. This is particularly important in industrial settings where different alarms may correspond to distinct failure modes or process constraints.

In addition to individual alarm indicators, an aggregated alarm feature is constructed by summing all alarm signals across each hourly time step. This aggregated variable represents the total number of active alarms within a given hour and serves as a proxy for the overall severity or intensity of abnormal system behavior. High values of this feature indicate periods characterized by multiple simultaneous alarms, which are often associated with critical operating conditions or cascading failures [36].

The inclusion of this aggregated alarm measure provides a compact, high-level summary of system stress that complements the more granular individual alarm features. While individual alarms capture localized and specific fault information, the aggregated alarm count enables the model to detect broader patterns related to alarm congestion, systemic instability, or operational overload.

By integrating both individual and aggregated alarm-based features into the categorical feature matrix, this step enriches the dataset with explicit indicators of abnormal events and system health. These features enhance the model's ability to distinguish between normal and anomalous operating regimes and to learn complex relationships between alarm activity, process dynamics, and downstream performance or failure outcomes.

4.1.4 Rolling Window Statistics

Rolling window statistics are employed to capture the short- and medium-term temporal dynamics of numerical signals by summarizing their recent historical behavior [9]. In time series originating from industrial processes, instantaneous measurements often provide limited information in isolation, as system behavior is typically influenced by accumulated effects, gradual drifts, and transient fluctuations over time. Rolling statistics address this limitation by transforming raw signals into features that encode local trends, variability, and cumulative behavior within predefined temporal windows.

In this feature engineering pipeline, rolling statistics are computed for each numerical variable at multiple window lengths, specifically 3, 6, 12, and 24 hours. The use of multiple window sizes enables the representation of process dynamics across different temporal scales, ranging from short-term fluctuations to daily patterns. This multi-scale approach is particularly valuable in industrial contexts, where some process responses manifest rapidly, while others evolve more slowly due to thermal inertia, material accumulation, or control system dynamics.

For each rolling window, three statistical measures are extracted: the mean, the standard deviation, and the sum. The rolling mean provides a smoothed representation of the signal, reducing the impact of high-frequency noise and highlighting local trends or shifts in operating levels. This allows the model to infer whether a variable has been persistently increasing, decreasing, or stabilizing over the recent past, which is often more informative than the instantaneous value alone.

The rolling standard deviation captures local variability within the window and serves as an indicator of process stability. Elevated variability may reflect unstable operating conditions, control issues, or the onset of abnormal behavior, even when average levels remain within nominal ranges. Conversely, low variability can indicate steady-state operation or controlled regimes. By explicitly encoding variability, the model gains sensitivity to fluctuations that may precede faults or performance degradation.

The rolling sum aggregates the signal over the window and is particularly useful for variables that represent rates, counts, or accumulated quantities. In such cases, the sum reflects the total contribution of the signal over the recent time horizon, enabling the model to reason about cumulative effects, such as total energy usage, material throughput, or exposure to stress conditions.

To ensure continuity at the beginning of the time series, rolling computations are performed with a minimum number of observations equal to one. This design choice avoids the introduction of excessive missing values in early time steps while still allowing rolling features to evolve naturally as more historical data becomes available.

Overall, rolling window statistics enrich the numerical feature space by embedding contextual information about recent system behavior directly into the feature set. By providing localized summaries of level, variability, and accumulation across multiple time scales, these features allow downstream models to capture temporal dependencies and dynamic patterns that are not accessible through raw measurements alone.

4.1.5 Cross-Signal Interaction Features

Cross-signal interaction features are introduced to explicitly model the relationships and dependencies between different numerical variables in the system [5]. In complex industrial processes, individual signals rarely operate in isolation; instead, system behavior emerges from the interaction between multiple process variables, such as pressures, temperatures, flows, and control signals. Capturing these interdependencies is essential for enabling predictive models to learn coordinated behavior, balance constraints, and coupled dynamics that cannot be inferred from univariate features alone.

In this feature engineering pipeline, pairwise interaction features are constructed for all combinations of numerical signals. For each pair of variables, two complementary forms of interaction are computed: the difference between the signals and their ratio. These transformations are designed to encode both absolute and relative relationships between variables, providing the model with interpretable measures of imbalance, dominance, and proportionality.

The difference between two signals represents their absolute deviation at a given time step. This feature is particularly informative when the variables are expected to follow similar trajectories under normal operating conditions, such as redundant sensors, upstream and downstream measurements, or controlled input–output pairs. Deviations in the difference can indicate process inefficiencies, sensor drift, or abnormal operating regimes. By including difference-based interaction features, the model gains the ability to detect misalignments or divergences between related signals that may precede faults or performance degradation.

In addition to absolute differences, ratio-based interaction features are computed to capture relative scaling relationships between variables. Ratios are especially useful when the magnitude of a signal is meaningful only in relation to another, for example when assessing efficiency, normalization effects, or load-dependent behavior. To ensure numerical stability, a small constant is added to the denominator, preventing division by zero and preserving continuity in the feature space.

These interaction features effectively embed domain-agnostic relational knowledge into the dataset, allowing the model to learn constraints and dependencies that would otherwise need to be inferred implicitly. By transforming pairs of raw signals into difference and ratio representations, the feature engineering process exposes higher-order structure in the data, enabling more robust modeling of multivariate dynamics.

Overall, the inclusion of cross-signal interaction features enhances the expressiveness of the feature set by incorporating information about how variables relate to one another at each point in time. This is particularly valuable in industrial time series analysis, where system behavior is often governed by coordinated interactions rather than isolated signal trends.

4.1.6 Frequency-Domain Features

Frequency-domain features are introduced to capture periodic and oscillatory patterns in numerical time series that may not be readily observable in

the time domain [34]. Industrial processes often exhibit cyclic behavior driven by mechanical rotations, control loops, production schedules, or external influences. While time-domain features such as lags and rolling statistics describe local temporal dynamics, they are limited in their ability to explicitly characterize repetitive structures and dominant cycles. Frequency-domain analysis addresses this limitation by decomposing signals into their spectral components.

In this feature engineering pipeline, frequency-domain information is extracted by applying a periodogram to each numerical signal [34]. The periodogram estimates the power spectral density of a time series, quantifying how the signal's variance is distributed across different frequencies. Prior to this computation, missing values are removed to ensure numerical stability and to avoid distortion of the spectral estimate. The resulting spectrum provides a compact representation of the signal's periodic content over the observed time horizon.

From the estimated power spectrum, a single dominant frequency is identified for each numerical variable. This dominant frequency corresponds to the frequency at which the spectral power reaches its maximum, indicating the most energetically significant periodic component in the signal. The extracted value is then assigned as a constant feature across all time steps for the corresponding variable.

The inclusion of the dominant frequency feature allows the model to distinguish between signals characterized by different intrinsic rhythms, such as fast oscillations induced by control mechanisms versus slower cycles driven by operational schedules or environmental factors. Even though the dominant frequency does not vary over time within the dataset, it encodes a global structural property of each signal that can influence how temporal patterns are interpreted in conjunction with other features.

By augmenting the feature set with frequency-domain information, the model gains access to complementary insights that are orthogonal to standard time-domain representations. This integration of spectral characteristics enhances the model's ability to recognize periodic behavior, discriminate between signals with similar time-domain statistics but different cyclic structures, and

improve overall interpretability in systems where oscillatory dynamics play a significant role.

4.1.7 Time-Based and Calendar Features

Time-based and calendar features are incorporated to explicitly encode the temporal context in which observations occur [14]. In many industrial and operational systems, process behavior is not only driven by internal dynamics but also influenced by exogenous temporal factors such as shift schedules, maintenance routines, production planning, and daily or seasonal demand patterns. Without explicit temporal encoding, a learning algorithm must infer these recurring structures indirectly from the data, which can be inefficient and unreliable. Time-based feature engineering addresses this limitation by making temporal regularities directly observable to the model.

In this pipeline, several calendar attributes are extracted directly from the timestamp index of the numerical dataset. These include the hour of the day, the day of the week, the day of the month, and the month of the year. Together, these variables provide a coarse but informative description of when each observation occurs within commonly relevant temporal cycles. Such features allow the model to capture systematic variations in process behavior associated with operational schedules, such as differences between working and non-working hours, weekday versus weekend effects, or monthly seasonality.

However, raw calendar variables suffer from an important limitation: they impose an artificial ordinal structure and discontinuities at cycle boundaries. For example, the transition from hour 23 to hour 0 or from month 12 to month 1 represents a small temporal change in reality but appears as a large numerical jump in raw form. To overcome this issue, cyclic encodings based on sine and cosine transformations are applied to each calendar variable [14]. These transformations map temporal positions onto a continuous circular representation, preserving the inherent periodicity of time.

Specifically, sine and cosine pairs are computed for the hour of the day, the day of the week, the day of the month, and the month of the year, each normalized by its respective cycle length. This representation allows the

model to learn smooth, periodic relationships and ensures that temporally adjacent points remain close in feature space, even at cycle boundaries. As a result, temporal proximity and cyclic continuity are accurately reflected in the engineered features.

By combining raw calendar attributes with their cyclic counterparts, the feature set provides both interpretability and mathematical consistency. The raw variables offer intuitive temporal references, while the cyclic encodings enable the model to exploit periodic patterns effectively. Overall, the inclusion of time-based and calendar features enriches the dataset with contextual information that is essential for modeling recurring temporal behaviors and improves the model's ability to generalize across different phases of operational and seasonal cycles.

4.2 Simple Statistical Approach for Time-Series Anomaly Detection

From the EDA results we saw that some huge drops or peaks have happened in the process so we thought the first and the easiest way to capture them is to design a statistical anomaly detection model based on Zscore or MAD values! Simple statistical approaches represent some of the earliest and most widely used techniques for anomaly detection in time-series data [5]. These methods are particularly attractive in industrial environments due to their low computational cost, ease of implementation, and high interpretability. By relying on fundamental statistical assumptions about normal system behavior, such approaches provide transparent decision rules that can be easily understood and validated by domain experts [24].

4.2.1 Z-Score-Based Anomaly Detection

The Z-score method is a parametric statistical approach that measures how far a data point deviates from the mean of a distribution in terms of standard deviations [11]. For a time-series x_t, the Z-score is defined as:

$$Z_t = \frac{x_t - \mu}{\sigma} \tag{4.1}$$

where μ and σ denote the mean and standard deviation of the time series, respectively. Data points with an absolute Z-score exceeding a predefined threshold (e.g., $|Z_t| > 3$) are considered anomalous. Z-score–based detection is most effective when the data follow a roughly Gaussian distribution and exhibit stationary behavior. In industrial time-series data, this method is often applied using sliding windows to adapt to gradual changes in system behavior [6]. Its primary advantages include simplicity, fast computation, and straightforward interpretability. However, the method is sensitive to outliers, as extreme values can distort the mean and standard deviation, potentially reducing detection accuracy.

4.2.2 Median Absolute Deviation (MAD)

The Median Absolute Deviation (MAD) is a robust statistical measure of variability that is less sensitive to outliers than the standard deviation [19]. It is defined as:

$$\text{MAD} = \text{median}(|x_t - \text{median}(x)|) \tag{4.2}$$

An anomaly score can be computed using a robust Z-score:

$$Z_t^{\text{MAD}} = \frac{0.6745(x_t - \text{median}(x))}{\text{MAD}} \tag{4.3}$$

where the constant 0.6745 ensures consistency with the standard deviation for normally distributed data [19]. MAD-based methods are particularly well suited for industrial time-series data that contain noise, non-Gaussian distributions, or occasional extreme values. Compared to the standard Z-score, MAD provides greater robustness and stability, making it more reliable for detecting genuine anomalies without being overly influenced by spurious spikes [28].

4.2.3 Applicability and Limitations

Simple statistical methods such as Z-score and MAD are commonly used as baseline approaches or as components within more complex anomaly detection frameworks [5]. Their key strengths include transparency, ease of deployment, and minimal data requirements. However, they also have limitations: they struggle with non-stationary data, multivariate dependencies, seasonal patterns, and complex temporal dynamics [36]. As a result, while these

methods remain valuable for real-time monitoring and initial screening, they are often complemented by machine learning or hybrid approaches in advanced industrial systems [25].

4.3 Residual-Based Anomaly Detection Using CatBoost Regression

Residual-based anomaly detection is a model-driven approach in which a predictive model is trained to learn normal system behavior, and anomalies are identified as significant deviations between predicted and observed values [36]. In this work, a CatBoost regression model is employed to estimate expected values of a target variable from historical operational data, after which the prediction residuals are analyzed to detect abnormal behavior.

4.3.1 CatBoost for Modeling Normal System Behavior

CatBoost is a gradient boosting algorithm based on decision trees, specifically designed to handle heterogeneous data, non-linear relationships, and complex feature interactions [26]. Unlike traditional gradient boosting methods, CatBoost incorporates several innovations that make it particularly effective for industrial applications. First, it natively handles categorical features without requiring extensive preprocessing such as one-hot encoding, which preserves information and reduces dimensionality. Second, it mitigates prediction shift through ordered boosting, which reduces overfitting by ensuring that each tree is built using an unbiased estimate of the residuals [26]. Third, it incorporates symmetric trees and efficient handling of missing values, allowing for robust modeling in real-world datasets where sensor readings may occasionally be missing or noisy.

Industrial datasets often include a mix of continuous sensor measurements, categorical process variables, and temporal features, and the relationships between these features can be highly non-linear and context-dependent. By training the CatBoost model primarily on data representing normal operating conditions, it learns a data-driven representation of expected system behavior without requiring explicit fault labels [5]. This makes it naturally suited for unsupervised or weakly supervised anomaly detection, where labeled examples of faults are rare or unavailable.

Let y_t denote the observed value of the target variable at time t, and $\hat{y}_t$ the corresponding prediction produced by the CatBoost model. The residual is computed as:

$$r_t = y_t - \hat{y}_t \tag{4.4}$$

Under normal operating conditions, residuals are expected to remain small and follow a stable distribution. Deviations from this pattern may indicate anomalies caused by equipment degradation, sensor faults, or process disturbances [17]. Compared to simpler regression methods such as linear regression or decision trees without boosting, CatBoost provides superior accuracy and robustness due to its ability to model non-linear interactions and capture higher-order dependencies between features, which are common in multivariate industrial processes [26].

4.3.2 Residual Thresholding for Anomaly Detection

Anomalies are detected by applying a statistical threshold to the residuals. Common approaches include fixed thresholds based on standard deviation, robust statistics such as Median Absolute Deviation (MAD) [28], or percentile-based thresholds. For example, a residual may be flagged as anomalous if:

$$|r_t| > \tau \tag{4.5}$$

where τ is a predefined threshold derived from the distribution of residuals observed during normal operation. This residual-based approach offers several advantages: it transforms anomaly detection into a problem that is interpretable and directly linked to prediction errors, thresholds can be adapted dynamically using rolling windows to account for process drift, and the method remains computationally efficient, enabling real-time monitoring of high-dimensional sensor streams [3].

4.3.3 Interpretability and Industrial Relevance

CatBoost also enhances the interpretability of the anomaly detection system. Feature importance metrics, along with SHAP (SHapley Additive exPlanations) values [22] or AcME-AD (Accelerated Model Explanations for Anomaly Detection), can be used to explain individual predictions and identify which variables

contributed most to a residual spike. This supports human-centric, transparent design principles aligned with Industry 5.0 [8], allowing operators and engineers to validate alerts, investigate potential faults, and make informed maintenance decisions [1].

Overall, residual-based anomaly detection using CatBoost regression provides a flexible, scalable, and interpretable framework for monitoring key signals in complex industrial time-series data. It strikes a balance between simple statistical approaches, which may miss subtle multivariate dependencies, and black-box deep learning models, which often lack transparency [25]. CatBoost's ability to model non-linear relationships, handle heterogeneous data, and provide actionable explanations makes it particularly suitable for real-world industrial applications such as predictive maintenance, process monitoring, and early fault detection [18].

4.4 Isolation Forest for Unsupervised Anomaly Detection

Isolation Forest (iForest) is a widely used unsupervised anomaly detection algorithm that identifies anomalies based on the principle of isolation rather than density estimation or distance measurement [20]. The core idea is that anomalous observations are fewer in number and exhibit attribute values that differ significantly from normal instances, making them easier to isolate through random partitioning of the data space.

4.4.1 Principle of Operation

Isolation Forest constructs an ensemble of binary decision trees, called isolation trees, by recursively partitioning the data using randomly selected features and split values [20]. Unlike traditional decision trees, isolation trees do not aim to optimize a predictive objective; instead, they focus solely on isolating individual observations. Points that require fewer splits to be separated from the rest of the data, resulting in shorter path lengths across the ensemble, are considered more likely to be anomalous.

The algorithm begins by randomly selecting a feature and a split value within the range of that feature, and recursively applies this procedure until each observation is isolated in a leaf node. This process is repeated to build a large ensem-

ble of isolation trees, which ensures that anomalies, being rare and different, are consistently isolated early. The anomaly score for a data point is derived from the average path length across all trees, with shorter paths indicating a higher likelihood of being anomalous [20].

Isolation Forest is particularly effective for industrial datasets because it captures multivariate relationships between signals, handles high-dimensional data efficiently, and makes no assumptions about the underlying distribution of the data [5]. Its reliance on random partitioning allows it to detect rare or subtle anomalies that may not be apparent through traditional statistical or distance-based methods. The ensemble approach also provides robustness, making it suitable for real-time monitoring of complex industrial processes where early detection of abnormal behavior is critical.

4.4.2 Applicability to Industrial Data

Isolation Forest is particularly well suited for industrial anomaly detection due to several key properties. First, it does not require labeled data, making it applicable in environments where fault annotations are scarce or unavailable [29]. Second, it scales efficiently to large datasets and high-dimensional feature spaces, which are common in Industry 4.0 systems with extensive sensor networks [16]. Third, it is effective at detecting global anomalies, such as abrupt sensor failures or significant process deviations.

In industrial time-series applications, Isolation Forest is typically applied to engineered features extracted from sliding windows, such as statistical summaries, frequency-domain characteristics, or residuals from predictive models. This allows the method to capture temporal behavior indirectly while maintaining computational efficiency [6].

4.4.3 Strengths and Limitations

The main strengths of Isolation Forest include its simplicity, scalability, and independence from distance or distributional assumptions [20]. Additionally, the method is less affected by the curse of dimensionality compared to distance-based techniques [2].

However, Isolation Forest also has limitations. It may struggle with contextual or collective anomalies, where abnormal behavior is only apparent in relation to temporal context or sequence patterns [5]. Furthermore, interpretabil-

ity is limited compared to simpler statistical approaches, although feature-level contributions can be approximated using post-hoc explanation methods [22]. The choice of contamination rate (i.e., the expected proportion of anomalies) can also influence detection performance and requires careful tuning in practice.

4.4.4 Role in Human-Centric Industrial Monitoring

Despite these limitations, Isolation Forest remains a valuable component in industrial anomaly detection pipelines. When combined with explainability techniques or used alongside residual-based and statistical methods, it contributes to robust and complementary detection capabilities [1]. Its unsupervised nature and scalability make it especially suitable for early-stage monitoring systems and exploratory analysis in complex industrial environments [25].

4.5 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) refers to a set of methods and principles that aim to make the behavior, decisions, and outputs of artificial intelligence systems understandable to human users [1]. Unlike traditional black-box models, which often provide accurate predictions without revealing the reasoning behind them, XAI seeks to expose the underlying logic, contributing factors, and decision pathways of AI models in a transparent and interpretable manner. This transparency is particularly important in complex machine learning systems where model decisions directly influence operational, economic, or safety-critical outcomes.

4.5.1 Goals of XAI

The primary goal of XAI is to bridge the gap between model performance and human understanding [1]. In practical terms, XAI enables users to answer fundamental questions such as why a particular prediction was made, which features influenced the outcome, and how changes in input variables would affect the model's behavior. By providing meaningful explanations, XAI supports model validation, debugging, and improvement, allowing practitioners to assess whether a model is behaving consistently with domain knowledge and operational expectations.

Another key goal of XAI is to facilitate human-in-the-loop decision-making. Rather than replacing human expertise, XAI is designed to complement it by offering interpretable insights that help users assess risk, prioritize actions, and intervene when necessary [7]. This aligns closely with the human-centric vision of modern industrial systems.

4.5.2 Importance of XAI in Industrial and Unsupervised Settings

The importance of XAI is particularly pronounced in industrial environments and unsupervised or weakly supervised anomaly detection systems [1]. In these contexts, models often operate without labeled ground truth, making it difficult to objectively verify the correctness of detected anomalies. Explanations allow engineers and operators to evaluate whether detected deviations correspond to meaningful faults, sensor errors, or benign variations.

Moreover, industrial applications frequently involve safety-critical processes, regulatory constraints, and high operational costs. In such settings, blind trust in black-box predictions is insufficient. XAI increases trust and accountability by enabling stakeholders to understand and justify model outputs [27]. This is especially relevant in Industry 5.0, where transparency, resilience, and human-centered design are core principles [8].

XAI also enhances diagnostic and prognostic capabilities by linking anomaly detection results to specific system components or process variables [18]. For example, feature-level explanations can reveal which sensors or operational parameters contributed most to an anomaly, enabling targeted maintenance actions and reducing downtime. Additionally, explainability supports continuous learning and system improvement by helping practitioners identify biases, concept drift, or model limitations over time [31].

4.6 SHapley Additive exPlanations (SHAP)

SHAP is a widely adopted explainability framework within Explainable Artificial Intelligence (XAI) that aims to interpret the predictions of machine learning models in a principled and consistent manner [22]. SHAP is grounded in Shapley values from cooperative game theory [33], where each input feature is considered a participant in a game whose outcome is the model's prediction.

The contribution of each feature is quantified by fairly distributing the prediction among all features based on their individual and joint effects.

One of the key strengths of SHAP is that it provides explanations with strong theoretical guarantees, including local accuracy, consistency, and missingness [22]. These properties ensure that SHAP explanations faithfully reflect the behavior of the underlying model and remain comparable across features. As a result, SHAP is particularly well suited for interpreting complex, non-linear models commonly used in industrial monitoring and anomaly detection applications.

4.6.1 Principle of SHAP

The core idea behind SHAP is to explain a model prediction by attributing it to individual features according to their marginal contributions across all possible feature combinations [22]. For a given input instance, the SHAP value of a feature represents the average change in the model output when that feature is added to a subset of other features, averaged over all possible subsets. This formulation avoids bias introduced by feature ordering and ensures a fair allocation of importance.

Formally, a model prediction $f(x)$ is decomposed as

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i, \tag{4.6}$$

where ϕ_0 denotes the expected value of the model output over a reference dataset and ϕ_i represents the SHAP value associated with feature i. Each SHAP value quantifies how much a feature increases or decreases the prediction relative to the baseline. The additive nature of this decomposition guarantees that the sum of all feature contributions exactly reconstructs the model output.

For tree-based models such as gradient boosting algorithms, SHAP can be computed efficiently using TreeSHAP [23], which exploits the structure of decision trees to calculate exact feature contributions. This makes SHAP computationally feasible for large-scale industrial datasets and enables its application in near-real-time monitoring scenarios.

4.6.2 Global SHAP Explanations

Global SHAP explanations aim to characterize the overall behavior of a model across the entire dataset rather than focusing on individual predictions [22]. Global importance is typically obtained by aggregating SHAP values over many observations, most commonly by computing the mean absolute SHAP value for each feature. Features with higher aggregated SHAP values are interpreted as having a greater overall influence on model predictions.

In industrial anomaly detection, global SHAP explanations provide insight into which process variables the model relies on most under normal operating conditions. They support model validation by allowing engineers to compare learned feature importance with domain knowledge and physical intuition [1]. Additionally, global SHAP analysis can reveal changes in feature relevance over time, thereby assisting in the detection of process drift and supporting long-term system monitoring and optimization.

4.6.3 Local SHAP Explanations

Local SHAP explanations focus on individual predictions and seek to answer why a model produced a specific output for a particular observation [22]. For a given input instance, the model prediction is decomposed into feature-level contributions relative to the expected baseline. Positive SHAP values indicate that a feature increases the prediction, while negative values indicate a decreasing effect.

In the context of industrial anomaly detection, local SHAP explanations are particularly valuable for interpreting anomalous events. When a large residual or anomaly score is observed, local SHAP values identify which sensors or process variables contributed most strongly to the deviation. This enables root-cause analysis, supports operator decision-making, and increases trust in AI-assisted monitoring systems by linking detected anomalies to meaningful physical and operational factors [18].

4.7 AcME-AD

4.7.1 Overview of AcME-AD

AcME-AD (Accelerated Model Explanations for Anomaly Detection) is an explainable artificial intelligence (XAI) framework designed to provide fast, local, and model-agnostic explanations for anomaly detection systems [38]. The method addresses a key limitation of many anomaly detection approaches: while they are effective at identifying abnormal behavior, they often operate as black boxes, offering little insight into why a given data point is considered anomalous.

AcME-AD focuses on explaining individual anomaly scores produced by an anomaly detection model, making it particularly suitable for high-dimensional industrial and time-series data where interpretability is essential for trust, diagnosis, and corrective action.

4.7.2 What AcME-AD Is

AcME-AD is a local explanation method tailored specifically for anomaly detection models [38]. Rather than explaining global model behavior, it explains single anomalous observations by quantifying how each feature contributes to the anomaly score assigned by the model.

The method is model-agnostic, meaning it can be applied to a wide range of anomaly detection algorithms, including distance-based, density-based, and reconstruction-based models. Importantly, AcME-AD is designed to be computationally efficient, enabling near-real-time explanations, which is critical in industrial monitoring and operational environments.

4.7.3 How AcME-AD Works

The core idea behind AcME-AD is to explain an anomaly by comparing it to a reference distribution of normal data and measuring how deviations along individual features contribute to the final anomaly score [38].

The explanation process begins by identifying a set of reference points

that represent normal system behavior. These reference points are typically selected from the training data or from data points with low anomaly scores. For a given anomalous observation, AcME-AD evaluates how the anomaly score changes when individual features are progressively replaced or approximated using values drawn from the reference distribution.

Rather than exhaustively perturbing all feature combinations, which would be computationally prohibitive, AcME-AD employs an accelerated approximation strategy. This strategy estimates feature contributions by leveraging the structure of the anomaly score function and the statistical properties of the reference data. As a result, it avoids the high computational cost associated with traditional perturbation-based XAI methods such as SHAP or LIME [27].

The output of the method is a feature attribution vector, where each value represents the contribution of a specific feature to the anomaly score. Positive contributions indicate features that increase the likelihood of the observation being anomalous, while negative or near-zero contributions correspond to features aligned with normal behavior.

4.7.4 Goal of AcME-AD

The primary goal of AcME-AD is to make anomaly detection transparent, actionable, and trustworthy [38]. In practical terms, the method aims to answer the question:
"Which features caused this observation to be detected as anomalous, and to what extent?"

By providing clear, quantitative explanations, AcME-AD enables domain experts to:

- Understand whether detected anomalies correspond to meaningful system issues
- Distinguish true anomalies from false positives
- Identify root causes of abnormal behavior
- Support decision-making in maintenance, safety, and quality control tasks

A secondary goal of the method is to achieve these explanations without sacrificing performance or scalability, making it suitable for real-world deployment in large-scale monitoring systems.

4.7.5 Benefits of AcME-AD

One of the main benefits of AcME-AD is its computational efficiency [38]. Compared to traditional explanation methods, which rely on repeated model evaluations under numerous feature perturbations, AcME-AD significantly reduces runtime while maintaining high explanation fidelity. This makes it feasible to generate explanations for streaming data or large datasets.

Another key advantage is its model-agnostic nature. Because AcME-AD does not rely on internal model parameters or gradients, it can be applied uniformly across different anomaly detection techniques [1]. This flexibility is particularly valuable in industrial environments where heterogeneous models may coexist.

AcME-AD also provides highly interpretable local explanations. By attributing anomaly scores directly to individual features, the method produces explanations that align naturally with human reasoning and domain knowledge. This interpretability facilitates collaboration between data scientists and domain experts, improving trust and adoption of anomaly detection systems [1].

Finally, the method supports operational reliability and accountability. In safety-critical or regulated settings, the ability to justify why an anomaly was detected is essential [8]. AcME-AD contributes to this requirement by making anomaly detection decisions auditable and explainable.

4.7.6 Summary

AcME-AD is an efficient, model-agnostic XAI framework for anomaly detection that provides fast and interpretable local explanations of anomaly scores [38]. By quantifying feature-level contributions relative to normal behavior, it bridges the gap between high-performance anomaly detection and human interpretability. Its combination of scalability, flexibility, and explanatory power

makes it particularly well suited for industrial time-series monitoring and other real-world anomaly detection applications [25].

5 Experiments and Results

5.1 Isolation Forest

The multivariate Isolation Forest model was employed to identify anomalous samples in the dataset. The focus was placed on detecting the top five anomalies based on the calculated anomaly scores.

5.1.1 F1

Following the initial analysis, Global AcME was applied in conjunction with consultations with domain experts to evaluate feature importance and identify the most informative variables. This process enabled the reduction of redundant or less informative signals. For instance, features such as **sonda_2_f1** and **stato_funzionamento** were excluded from subsequent analysis. The anomaly detection procedure was then repeated on the refined feature set.

The top five detected anomalies corresponded to the following timestamps:

- 2025-01-02 15:00:00
- 2025-04-09 05:00:00
- 2024-12-20 09:00:00
- 2024-12-20 11:00:00
- 2024-12-20 10:00:00

It is noteworthy that the anomalies identified on 2024-12-20 are associated with the same cooking process. The distribution of these anomalies across the relevant signals is illustrated in Figures 5.1 and 5.2.

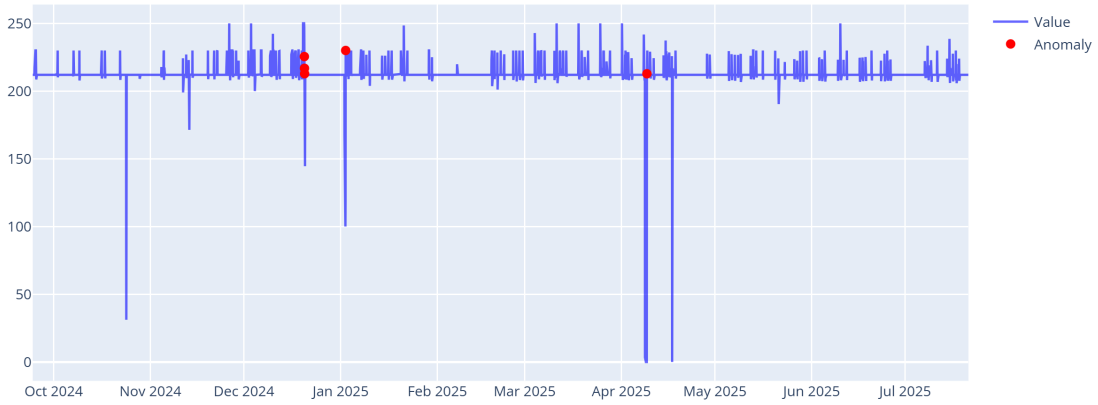

Figure 5.1: set_attivo_temperatura - Anomaly

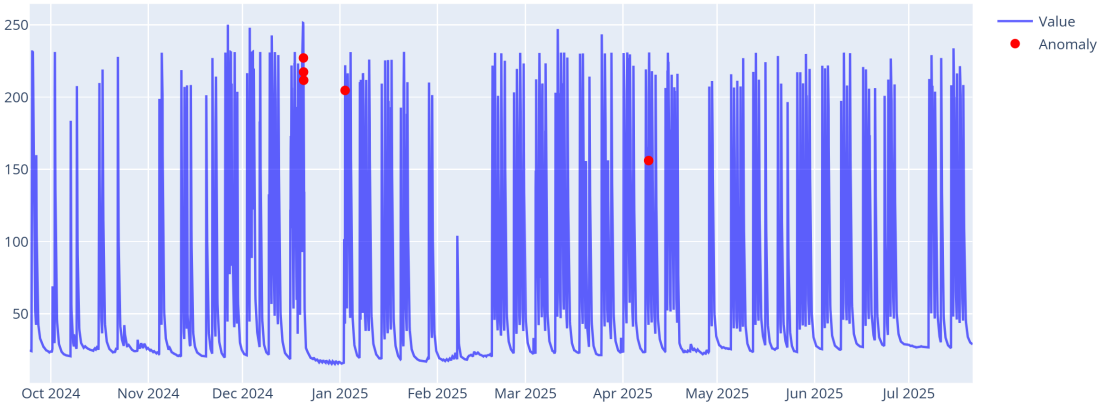

Figure 5.2: temperatura_visualizzata - Anomaly

Given that the primary objective of anomaly detection in industrial contexts is to inform operators about the causes of abnormal behavior, Local AcME was subsequently applied to anomalies to identify the specific signals contributing to these anomalies. One of the analysis as an example revealed the following signals as most indicative of deviation from standard operating conditions:

- *set_attivo_temperatura_lag_1*
- *seq_avvio_ciclo_1.0*

These results demonstrate the capability of the Isolation Forest, combined with explainable AI techniques, to not only detect anomalous events but also provide actionable insights that can guide operator interventions in the production process.

Since F1_131, F2_132, and F3_133 serve the same functional role within the plant and exhibit highly similar data characteristics, value ranges, and statistical distributions, they can be considered operationally equivalent for analytical purposes. Therefore, only the results corresponding to F1_131 are presented in detail, as the outcomes for F2_132 and F3_133 are highly comparable and do not provide additional distinct insights.

5.1.2 BL

After the initial exploratory analysis, Global AcME was applied alongside consultations with domain experts to better understand which features actually carried useful information. This step made it possible to clean up the feature set by removing variables that were redundant or simply not informative enough. For instance, features such as **g_xInterventoMagnetotermico**, **g_xAlarmInterventoEmergenze**, and **g_xAlarm00**, among others, were dropped before moving forward. With this refined feature set in hand, the Isolation Forest anomaly detection procedure was run again. The top five anomaly timestamps identified by the model are listed below:

- 2024-11-11 10:00:00
- 2024-11-11 11:00:00
- 2024-11-11 09:00:00
- 2025-05-22 06:00:00
- 2025-03-05 06:00:00

It is worth noting that the three anomalies detected on 2024-11-11 all belong to the same cooking process, which suggests that something went consistently wrong during that particular production cycle. The distribution of these anomalies across the relevant signals can be seen in Figures 5.3 and 5.4.

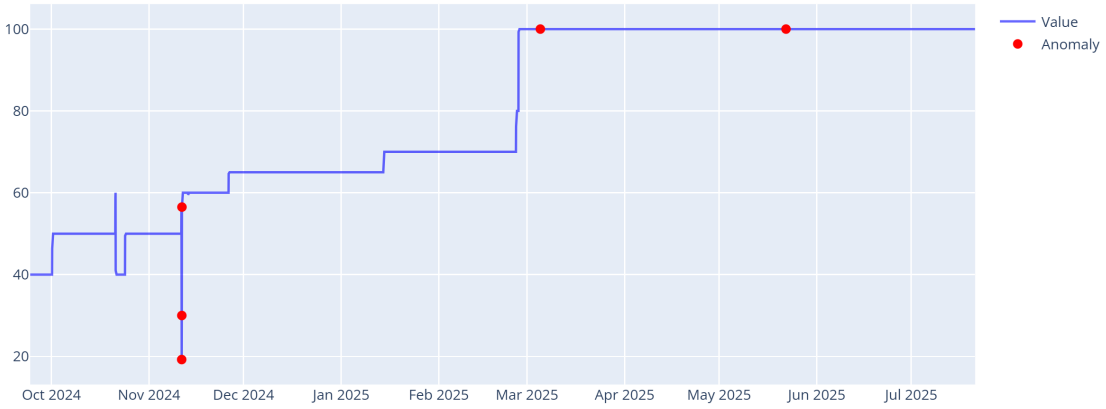

Figure 5.3: iVelSpiraManuale1 - Anomaly

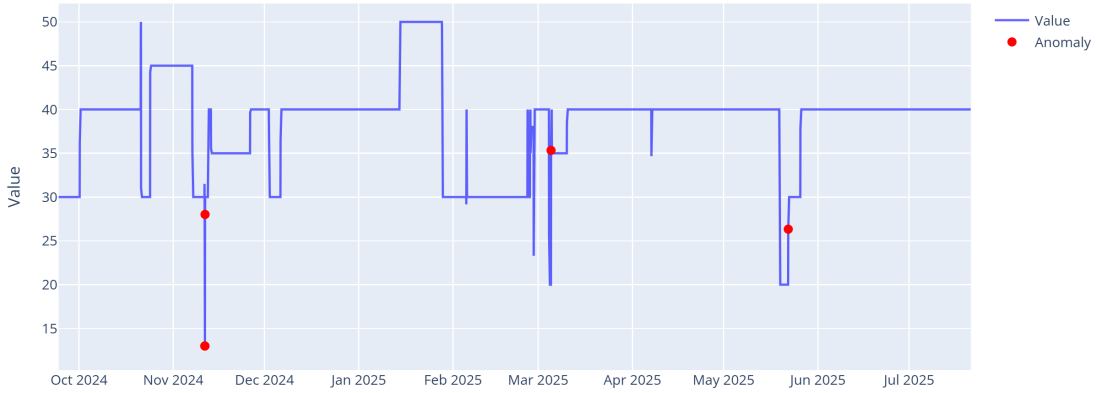

Figure 5.4: iVelSpira - Anomaly

Since the whole point of anomaly detection in an industrial setting is to help operators understand why something went wrong, not just that it did, Local AcME was then applied to the detected anomalies to pinpoint which signals were driving the deviations the most. One representative analysis pointed to the following features as the main contributors to the observed departures from normal operating conditions:

- *iVelSpira_roll_std_3*
- *iVelSpiraManuale1_roll_std_24*

These results show that pairing Isolation Forest with explainable AI techniques does more than just flag unusual events ,it also gives operators concrete, interpretable information they can actually act on during the production process.

That said, the newly provided dataset could not be reliably used for further validation or extended explainability analysis. The reason is straightforward: the primary monitored signals in that dataset are almost entirely constant, which means it simply does not capture the kind of variability needed to meaningfully support additional XAI investigations.

5.2 CatBoost Regression

5.2.1 F1

temperatura_visualizzata: The monitoring and prediction process was initially focused on the signal *temperatura_visualizzata*, which was designated as the primary target variable for modeling. The dataset was partitioned such that 15% of the initial data was used for testing. The CatBoost model was trained on the remaining portion, and the residuals were analyzed to detect anomalies. A residual threshold of 30 was chosen as a practical criterion for flagging deviations, although this threshold can be adjusted according to specific operational requirements. Using this approach, five anomalous points were identified. Some parts of results are illustrated in Figures 5.5 and 5.6.

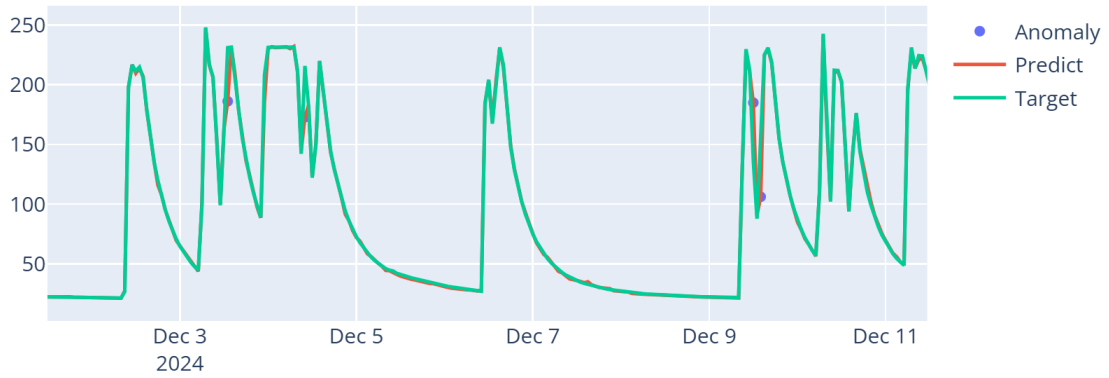

Figure 5.5: temperatura_visualizzata - Main Data

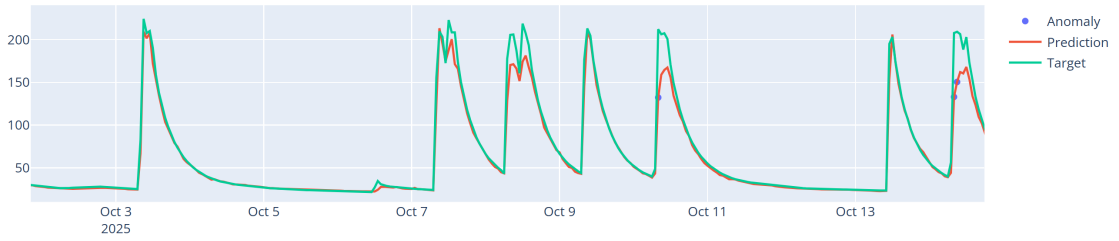

Figure 5.6: temperatura_visualizzata – Test Data

The model predictive performance metrics for both the main and test datasets are reported in Table 5.1. The results indicate strong predictive performance, with high R^2 values and low error measures.

temperatura_visualizzata	**Main Data**	**Test Data**
MSE	20.105	130.966
RMSE	4.484	11.444
R^2	0.993	0.961

Table 5.1: Performance metrics - temperatura_visualizzata

To support interpretability and provide actionable insights to operators, XAI techniques were applied. Both SHAP and AcME were used to assess feature importance at local level. The results consistently identified the most influential variables contributing to the residual anomalies. Figure 5.7 shows an example of local SHAP values for a single anomaly.

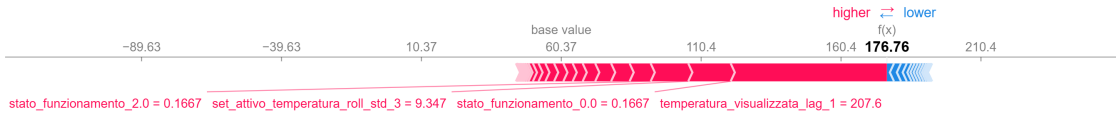

Figure 5.7: Local SHAP analysis

AcME was subsequently chosen as the primary XAI tool due to its computational efficiency and faster execution, enabling real-time interpretability. Using this approach, a set of signals contributing to the anomalies was provided to the operator for decision-making. For example, for one detected anomaly, the relevant features included:

- *temperatura_visualizzata_lag_1*
- *stato_funzionamento_0.0*

These results demonstrate that CatBoost, combined with interpretable residual analysis and XAI methods, provides a robust framework for detecting anomalies in industrial time series data while offering actionable insights that support human-centric monitoring and decision-making.

set_attivo_temperatura: The monitoring and prediction procedure was subsequently applied to the numerical signal *set_attivo_temperatura*, using the same test data proportion as in the previous experiments. The analysis was conducted on both the main dataset and a newly provided test dataset supplied by the company. Figures 5.8 and 5.9 illustrate a subset of the results, highlighting the detected anomalies.

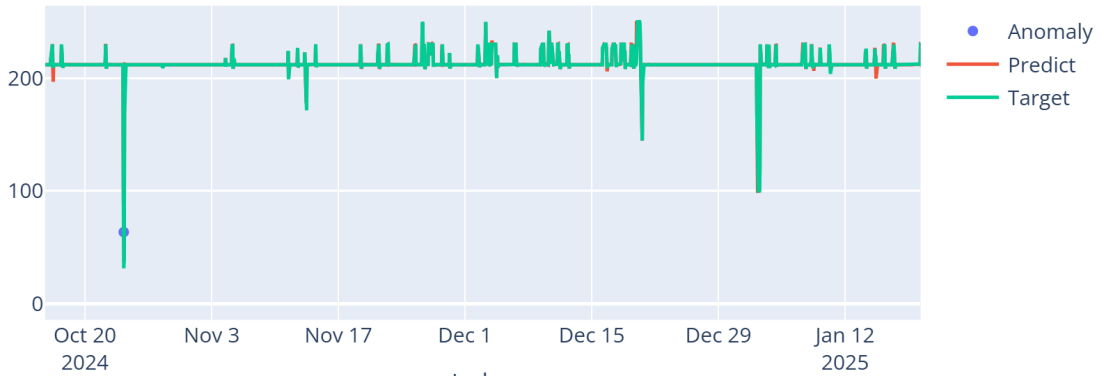

Figure 5.8: set_attivo_temperatura – Main Data

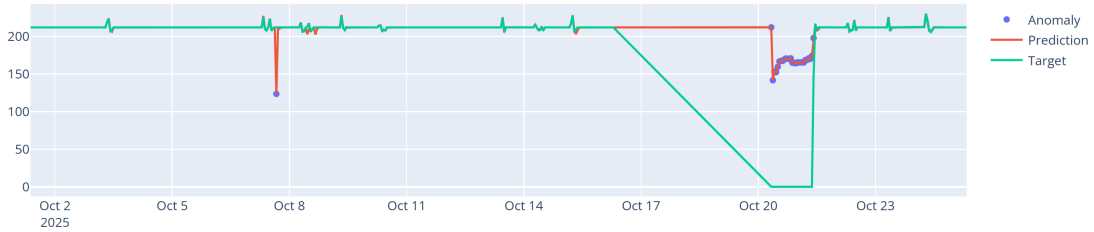

Figure 5.9: set_attivo_temperatura – Test Data

The predictive performance metrics for *set_attivo_temperatura* are summarized in Table 5.2. While the model demonstrates high accuracy on the main dataset, a noticeable decrease in performance is observed on the new test data, as reflected by a lower R^2 and higher error values. This drop is attributable to the presence of anomalies (as can be seen in the figure 5.9), which underscores

the primary objective of deploying such monitoring systems in industrial environments.

set_attivo_temperatura	Main Data	Test Data
MSE	1.894	1542.413
RMSE	1.376	39.273
R^2	0.978	0.334

Table 5.2: Performance metrics - set_attivo_temperatura

To provide actionable insights for operators, local XAI techniques were applied to interpret the detected anomalies and identify the specific signals contributing to deviations. This approach ensures that operators are informed about the underlying causes of anomalies and can take appropriate interventions to maintain process stability and quality.

Since F1_131, F2_132, and F3_133 serve the same functional role within the plant and exhibit highly similar data characteristics, value ranges, and statistical distributions, they can be considered operationally equivalent for analytical purposes. Therefore, only the results corresponding to F1_131 are presented in detail, as the outcomes for F2_132 and F3_133 are highly comparable and do not provide additional distinct insights.

5.2.2 BL

iVelSpiraManuale1: A CatBoost regression model was trained to monitor the *iVelSpiraManuale1* signal, using 85% of the available data for training and the remaining 15% for testing. A residual threshold of 30 was chosen as a reasonable criterion for flagging deviations from expected behavior, though this value can be freely adjusted depending on the specific operational context and requirements. Overall, the model performed very well, showing a strong agreement between the predicted and actual signal values.

iVelSpiraManuale1	Main Data	Test Data
MSE	0.1035	—
RMSE	0.3217	—
R^2	0.999	—

Table 5.3: Performance metrics - iVelSpiraManuale1

The predictive performance on the main dataset is illustrated in Figure 5.10, where the predicted values closely follow the actual measurements.

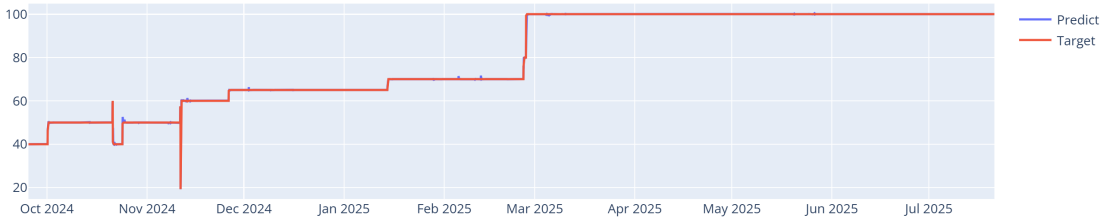

Figure 5.10: iVelSpiraManuale1 - Main Data

An anomaly detection threshold of 30 was applied, and no anomalies were identified. This outcome is consistent with the high R^2 score, confirming that the predicted values remain very close to the observed signal values.

The model was further evaluated on the designated test portion of the dataset, as shown in Figure 5.11.

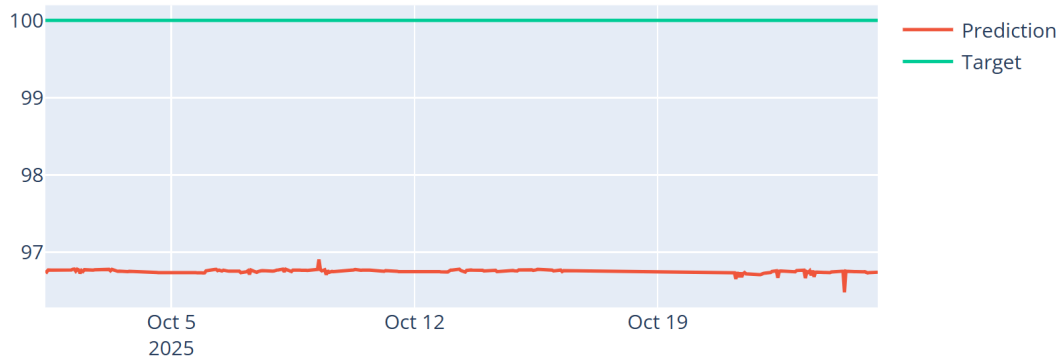

Figure 5.11: iVelSpiraManuale1 - Test Data

However, it is important to note that the test data segment consists predominantly of constant values and does not exhibit the same distributional characteristics as the training data. Consequently, this portion of the dataset cannot be considered fully representative for assessing generalization performance.

Overall, no anomalies were detected during the monitoring of the *iVelSpiraManuale1* signal, as the predicted values remained consistently aligned with the actual measurements.

iVelSpira: To monitor the *iVelSpira* signal, a CatBoost regression model was trained following the same procedure as before, maintaining the 85%–15% train–test split. A residual threshold of 30 was chosen as a reasonable criterion for flagging deviations from expected behavior, though this value can be freely adjusted depending on the specific operational context and requirements. The model performed very well, showing a strong agreement between predicted and actual values, as reflected by the metrics reported in Table 5.4.

iVelSpira	Main Data	Test Data
MSE	0.2803	—
RMSE	0.5294	—
R^2	0.988	—

Table 5.4: Performance metrics - iVelSpira

The trained CatBoost model was subsequently evaluated on a second dataset provided by SMACT for validation purposes. As shown in Figure 5.12, this validation dataset consists largely of constant values over time and therefore does not reflect realistic system variability. Consequently, it cannot be considered fully representative of typical operating conditions.

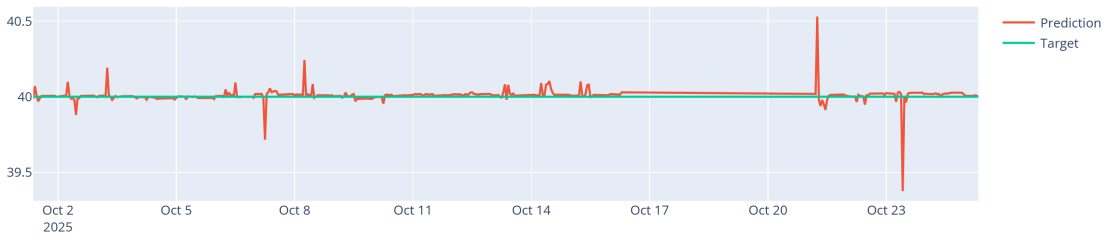

Figure 5.12: iVelSpira - Test Data

Despite the limited variability in the validation dataset, the trained model maintained high predictive accuracy and did not exhibit significant deviation from the desired target values of the *iVelSpira* signal.

5.3 Simple Statistical Approach

Exploratory Data Analysis revealed that certain signals exhibit abrupt drops or peaks that can be effectively identified using simple statistical techniques. For such cases, lightweight statistical models provide a fast and computationally efficient baseline for anomaly detection, particularly suitable for identifying obvious deviations from normal operating ranges.

5.3.1 F1

As observed during the exploratory analysis, the signal *set_attivo_temperatura* is characterized by relatively stable behavior under normal operating conditions, with limited variability but occasional pronounced drops. These characteristics make it well suited for threshold-based statistical anomaly detection.

The results of applying the statistical model to the main dataset are illustrated in Figure 5.13. Anomalies are identified by comparing observed values against bounds derived from the mean and standard deviation estimated during the training phase.

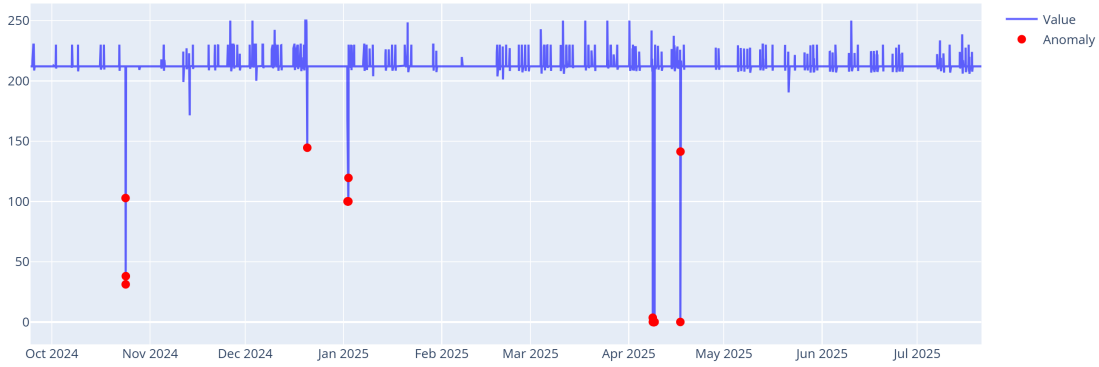

Figure 5.13: set_attivo_temperatura – simple approach - main Data

The same statistical thresholds, computed exclusively from the training data, were subsequently applied to the newly provided test dataset in order to assess the generalization capability of the approach. The corresponding results are shown in Figure 5.14.

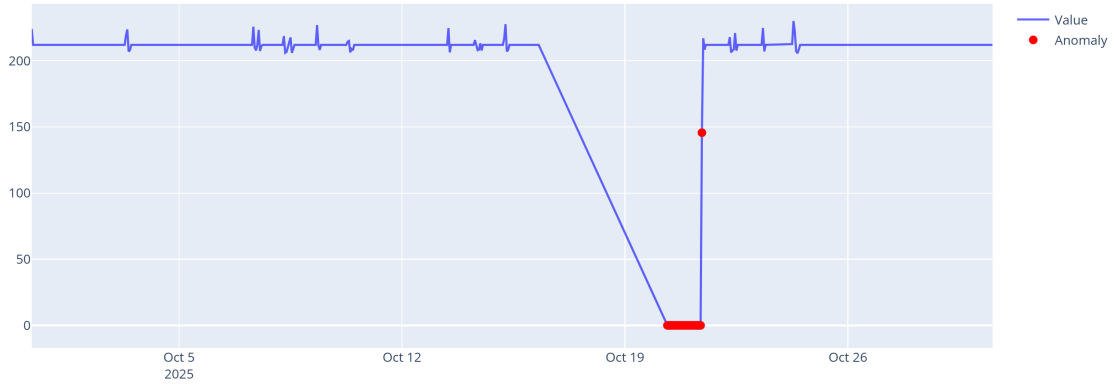

Figure 5.14: set_attivo_temperatura – simple approach - Test Data

5.4 Comparison of Models

CatBoost regression combined with residual-based analysis proved highly effective for monitoring selected key signals, providing accurate predictions and interpretable deviation patterns under normal operating conditions. However, when the objective is to capture multivariate relationships across all signals within a process, Isolation Forest demonstrated superior suitability. Its ability to model complex interdependencies in an unsupervised manner, combined with low computational overhead and fast execution, makes it particularly well suited for large-scale industrial monitoring scenarios where timely detection is critical.

6

Discussion

This chapter discusses the outcomes of the proposed anomaly detection framework, reflecting on what proved effective, what limitations were encountered, and the insights gained regarding both the monitored industrial process and the applied methodologies. The discussion also highlights the practical value of the developed approach for the industrial partner and its potential generalization to other use cases, while acknowledging technical and organizational constraints.

Overall, the combination of model-based anomaly detection and explainable artificial intelligence (XAI) techniques demonstrated strong practical effectiveness [1]. In particular, both global and local explainability methods played a central role in validating model outputs and supporting human interpretation. Among the evaluated approaches, the AcME method proved especially well suited for industrial deployment due to its computational efficiency and its ability to provide both global and instance-level explanations [38]. Compared to more computationally demanding XAI methods, AcME enabled rapid feedback while preserving sufficient explanatory depth, making it suitable for near-real-time analysis and operator-facing applications.

Signal-level monitoring using CatBoost regression combined with residual-based anomaly detection yielded highly accurate predictive performance for key process variables [26]. The model successfully learned normal operating behavior and produced stable residual distributions under nominal conditions. As a result, sudden drops or peaks in the monitored signals were detected reliably through residual thresholding [36]. A major strength of this approach lies in

its interpretability: detected anomalies could be directly linked to prediction errors, and local XAI explanations allowed operators to understand which features contributed most to each anomalous event [22]. This significantly enhances trust in the system and supports informed decision-making, which is essential in industrial environments aligned with Industry 5.0 principles [8].

The multivariate Isolation Forest model proved effective in capturing complex interactions among multiple signals in a high-dimensional feature space [20]. It was able to identify unusual samples that differed from typical system behavior based on joint feature relationships rather than individual signal deviations. Its unsupervised nature and low computational cost make it particularly attractive for large-scale monitoring scenarios [25]. However, the absence of reliable ground truth labels prevented the computation of conventional accuracy metrics [5]. Attempts to use existing alarm signals as proxy labels were unsuccessful, as several devices were affected by malfunctions and no consistent relationship was observed between alarm activation and meaningful process anomalies. Consequently, validation relied primarily on expert-driven analysis, where domain specialists assessed the plausibility and operational relevance of detected anomalies [36].

Several insights about the monitored system emerged from this work. First, certain anomalies manifested as clear deviations in individual signals and were well captured by regression-based monitoring, while others were only detectable when considering multivariate relationships across the process [5]. Second, explainability was essential not only for operator trust but also for validating model behavior in the absence of labels [1]. Without transparent explanations, it would have been difficult to assess whether detected anomalies corresponded to realistic process deviations or spurious patterns in the data.

From a practical perspective, the proposed framework provides tangible value to the industrial partner. It enables early detection of abnormal behavior, supports root-cause analysis through interpretable explanations, and operates under realistic computational constraints [3]. The use of lightweight models and fast XAI methods facilitates integration into existing monitoring workflows without requiring extensive infrastructure changes [31]. Moreover, the framework supports human-in-the-loop operation, where operators remain central to decision-making rather than being replaced by automated systems [7].

The approach is also highly transferable to other industrial use cases. The combination of signal-level predictive monitoring, multivariate unsupervised

detection, and explainable outputs is applicable to a wide range of processes involving time-series sensor data, including manufacturing, energy systems, and process industries [18]. The methodology does not depend on process-specific fault labels, making it suitable for domains where anomalies are rare, poorly documented, or evolving over time [29].

Despite these strengths, several limitations must be acknowledged. From a technical perspective, the lack of labeled anomaly data limits quantitative evaluation and complicates threshold selection [5]. Unsupervised methods inherently rely on assumptions about normal behavior, which may shift as processes evolve [17]. Additionally, feature engineering choices can influence model sensitivity and require careful tuning. From an organizational standpoint, effective deployment depends on the availability of domain experts to interpret results and provide feedback, as well as on operator training and acceptance [1]. Integrating anomaly detection outputs into existing operational practices remains a non-trivial challenge [31].

In summary, the results demonstrate that combining efficient machine learning models with fast and interpretable XAI techniques offers a robust and practical solution for industrial anomaly detection [25]. While limitations remain, particularly in validation and organizational integration, the proposed human-centric framework represents a meaningful step toward transparent, trustworthy, and deployable industrial monitoring systems [8].

7 Future Work

While this research demonstrates the effectiveness of unsupervised and weakly supervised anomaly detection methods for industrial monitoring, several directions remain open for future work. These directions aim to improve data quality, enable real-time deployment, enhance model performance, and strengthen the robustness and usability of the system in practical industrial environments.

7.1 Improved and More Accurate Data Collection

High-quality data is a fundamental requirement for reliable anomaly detection [5]. In industrial environments, data is frequently affected by sensor noise, missing values, asynchronous sampling, calibration drift, and inconsistent logging practices [6]. Future work should focus on improving the accuracy, reliability, and consistency of data acquisition pipelines through higher-precision sensors, regular calibration procedures, and improved synchronization across heterogeneous data sources. Beyond technical improvements, enriching raw sensor readings with structured metadata and contextual annotations, such as descriptions of operating conditions, production phases, or recent maintenance activities, would provide models with the additional context needed to correctly distinguish between normal operational variation and genuine anomalies. Such semantic enrichment of the data collection process represents a straightforward yet impactful step toward more reliable and interpretable detection.

An equally important limitation of the current work concerns the near-

complete absence of labeled data. While fully labeled datasets are often unrealistic in industrial settings, even a small number of reliable labels associated with known faults, confirmed abnormal episodes, or maintenance events could significantly improve model evaluation, threshold calibration, and overall confidence in deployed systems [29]. Sparse annotations of this kind would allow partial quantitative assessment of detection performance and bridge the gap between purely unsupervised monitoring and supervised learning. Acquiring even a modest set of verified anomaly examples should therefore be considered a high-priority investment for any future deployment of this framework.

In this context, Active Learning combined with continuous operator feedback represents a particularly promising and practical strategy [32]. By iteratively querying subject matter experts for labels on the most informative, uncertain, or borderline samples, Active Learning can efficiently incorporate expert knowledge while minimizing labeling effort. Beyond initial labeling campaigns, future systems should provide operators with structured mechanisms to validate, correct, or reject model outputs in real time. This operator-in-the-loop paradigm would enable the model to be continuously tuned and refined through Active Learning cycles driven by field experience, improving alignment between algorithmic outputs and expert expectations over time [7]. Such a feedback-driven approach would ultimately strengthen both the practical relevance and the operational trust of the system, ensuring that the model adapts to the specificities of the monitored environment rather than remaining static after initial training.

Future work may also explore automated data quality monitoring mechanisms that detect sensor malfunctions, data corruption, or distribution shifts prior to anomaly detection. Early detection of data integrity issues would further enhance system reliability and operator trust [31].

7.2 Real-Time Deployment and Online Monitoring

Although offline analysis is valuable for model development and validation, real-world industrial applications require real-time or near-real-time anomaly detection [21]. Future research should therefore focus on deploying the proposed framework within live production environments, where data streams continuously and decisions must be made under strict latency constraints.

Key challenges include efficient data streaming, low-latency inference, and

seamless integration with existing industrial control systems. Edge computing architectures represent a promising solution, enabling anomaly detection and explanation to be performed closer to the data source while reducing communication delays and dependence on centralized infrastructure [16].

Real-time deployment also necessitates robust mechanisms for handling concept drift, as industrial processes evolve due to equipment aging, maintenance actions, and changing operational conditions [17]. Online learning or periodic model adaptation strategies could help maintain detection accuracy over time. In addition, future systems should incorporate real-time explainability, allowing operators to immediately understand why an anomaly was detected [38]. Lightweight and computationally efficient XAI methods will be essential to support rapid decision-making in operational contexts [1].

For productive rollout and long-term sustainability, future work should also consider the adoption of MLOps best practices to manage the full lifecycle of machine learning solutions [31]. This includes versioning of data and models, automated testing and validation pipelines, continuous integration and deployment (CI/CD), monitoring of model performance in production, and systematic retraining procedures. Integrating anomaly detection systems within an MLOps framework would ensure reproducibility, traceability, scalability, and reliable operation in industrial environments.

7.3 Model Improvements and Methodological Enhancements

Future work can further enhance the anomaly detection models through methodological refinements. One promising direction is the development of hybrid or ensemble frameworks that combine simple statistical detectors, residual-based regression models, and multivariate unsupervised techniques such as Isolation Forest [25]. Such ensembles could improve robustness by capturing complementary types of anomalies and reducing sensitivity to individual model limitations.

Additional improvements may involve more advanced feature engineering and representation learning, including methods to capture long-term temporal dependencies, evolving process dynamics, and automatically adaptive thresholds [13]. Introducing self-learning or adaptive mechanisms would allow mod-

els to evolve alongside the monitored system, improving long-term performance and reducing manual tuning requirements.

Finally, further emphasis should be placed on explainability and human-centric system design [1]. Deeper integration of XAI techniques with anomaly detection outputs could provide operators with clearer diagnostic insights and higher-level system understanding. Evaluating how explanations are interpreted, trusted, and acted upon by human operators represents an important research direction, particularly in the context of Industry 5.0, where AI systems are expected to augment human expertise rather than replace it [8].

7.4 Extension to Additional Plant Modules

Future work will also extend the proposed framework to other plant modules, including C1 and C2. However, before applying the anomaly detection pipeline to these modules, the data quality issues highlighted throughout this work, such as sensor malfunctions and inconsistent data collection, must first be addressed. Once these problems are resolved and the data acquisition process is sufficiently reliable, the results for C1 and C2 will be collected and reported in future publications.

References

[1] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115.

[2] Markus M. Breunig et al. "LOF: Identifying Density-Based Local Outliers". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 2000, pp. 93–104.

[3] Thyago P. Carvalho et al. "A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance". In: *Computers & Industrial Engineering* 137 (2019), p. 106024.

[4] Patricio Cerda, Gaël Varoquaux, and Balázs Kégl. "Similarity Encoding for Learning with Dirty Categorical Variables". In: *Machine Learning* 107.8–10 (2018), pp. 1477–1494.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Computing Surveys* 41.3 (2009), pp. 1–58.

[6] Andrew A. Cook, Gözde Misirli, and Zhong Fan. "Anomaly Detection for IoT Time-Series Data: A Survey". In: *IEEE Internet of Things Journal* 7.7 (2019), pp. 6481–6494.

[7] Mica R. Endsley. "Toward a Theory of Situation Awareness in Dynamic Systems". In: *Human Factors* 37.1 (1995), pp. 32–64.

[8] European Commission. *Industry 5.0: Towards a Sustainable, Human-Centric and Resilient European Industry*. Tech. rep. Luxembourg: Publications Office of the European Union, 2021.

[9] Ben D. Fulcher, Max A. Little, and Nick S. Jones. "Highly Comparative Time-Series Analysis: The Empirical Structure of Time Series and Their Methods". In: *Journal of the Royal Society Interface* 10.83 (2013), p. 20130048.

[10] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014, pp. 2672–2680.

[11] Frank E. Grubbs. “Procedures for Detecting Outlying Observations in Samples”. In: *Technometrics* 11.1 (1969), pp. 1–21.

[12] Guo Haixiang et al. “Learning from Class-Imbalanced Data: Review of Methods and Applications”. In: *Expert Systems with Applications* 73 (2017), pp. 220–239.

[13] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[14] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 2nd. Melbourne, Australia: OTexts, 2018.

[15] Ian T. Jolliffe. *Principal Component Analysis*. 2nd. New York: Springer, 2002.

[16] Heiner Lasi et al. “Industry 4.0”. In: *Business & Information Systems Engineering* 6.4 (2014), pp. 239–242.

[17] Jay Lee et al. “Prognostics and Health Management Design for Rotary Machinery Systems—Reviews, Methodology and Applications”. In: *Mechanical Systems and Signal Processing* 42.1–2 (2014), pp. 314–334.

[18] Yaguo Lei et al. “Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap”. In: *Mechanical Systems and Signal Processing* 138 (2020), p. 106587.

[19] Christophe Leys et al. “Detecting Outliers: Do Not Use Standard Deviation Around the Mean, Use Absolute Deviation Around the Median”. In: *Journal of Experimental Social Psychology* 49.4 (2013), pp. 764–766.

[20] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation Forest”. In: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*. IEEE, 2008, pp. 413–422.

[21] Yang Lu. “Industry 4.0: A Survey on Technologies, Applications and Open Research Issues”. In: *Journal of Industrial Information Integration* 6 (2017), pp. 1–10.

[22] Scott M. Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 4765–4774.

[23] Scott M. Lundberg et al. “From Local Explanations to Global Understanding with Explainable AI for Trees”. In: *Nature Machine Intelligence* 2.1 (2020), pp. 56–67.

[24] Douglas C. Montgomery. *Introduction to Statistical Quality Control*. 6th. New York: John Wiley & Sons, 2009.

[25] Guansong Pang et al. “Deep Learning for Anomaly Detection: A Review”. In: *ACM Computing Surveys* 54.2 (2021), pp. 1–38.

[26] Liudmila Prokhorenkova et al. “CatBoost: Unbiased Boosting with Categorical Features”. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018, pp. 6638–6648.

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[28] Peter J. Rousseeuw and Christophe Croux. “Alternatives to the Median Absolute Deviation”. In: *Journal of the American Statistical Association* 88.424 (1993), pp. 1273–1283.

[29] Lukas Ruff et al. “A Unifying Review of Deep and Shallow Anomaly Detection”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 756–795.

[30] Bernhard Schölkopf et al. “Estimating the Support of a High-Dimensional Distribution”. In: *Neural Computation* 13.7 (2001), pp. 1443–1471.

[31] David Sculley et al. “Hidden Technical Debt in Machine Learning Systems”. In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015, pp. 2503–2511.

[32] Burr Settles. “Active Learning Literature Survey”. In: *Computer Sciences Technical Report* 1648 (2009).

[33] Lloyd S. Shapley. “A Value for n-Person Games”. In: *Contributions to the Theory of Games* 2 (1953), pp. 307–317.

[34] Petre Stoica and Randolph L. Moses. *Spectral Analysis of Signals*. Upper Saddle River, NJ: Prentice Hall, 2005.

[35] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 5998–6008.

[36] Venkat Venkatasubramanian et al. “A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods”. In: *Computers & Chemical Engineering* 27.3 (2003), pp. 293–311.

[37] Xun Xu et al. “Industry 4.0 and Industry 5.0—Inception, Conception and Perception”. In: *Journal of Manufacturing Systems* 61 (2021), pp. 530–535.

[38] Valentina Zaccaria et al. “AcME-AD: Accelerated Model Explanations for Anomaly Detection”. In: *CoRR* abs/2403.01245 (2024). arXiv:2403.01245. arXiv: 2403.01245 [cs.LG].

Acknowledgments