Baltic J. Modern Computing, Vol. 14 (2025), No. 1, 50-69
https://doi.org/10.22364/bjmc.2026.14.1.03

# ADF$^2$S Feature Set of Anomaly Detection Framework Based on Knowledge Discovery

Imelda ZADEJA

Vidzeme University of Applied Sciences, Terbatas Street 10, Valmiera, Latvia

imelda.zadeja@va.lv

ORCID 0009-0000-7305-8332

**Abstract**. In the modern knowledge era, the exponential growth of digital solutions has led to the generation of vast amounts of data. This necessitates the development of data and knowledge-driven advanced techniques to extract insights and support informed decision-making. Within this context, the detection of anomalies, data points that deviate significantly from expected patterns, becomes crucial as these anomalies can arise due to diverse factors, including sensor errors, data corruption, and changes in underlying processes, all of which may impact system performance, accuracy, and overall efficiency. This paper thoroughly examines the diverse frameworks and architectures established for anomaly detection across multiple domains. It highlights the complexity associated with the nature of anomalies, which are often domain-specific and contextually bound, thereby presenting significant challenges in devising a universal framework capable of addressing anomalies regardless of the domain or context of the application. To address these challenges, the author proposes a comprehensive anomaly detection framework feature set (ADF$^2$S) that captures the functional, structural, and operational dimensions of anomaly detection frameworks. The proposed anomaly detection framework feature set (ADF$^2$S) and its cross-framework evaluation contribute a practical foundation for researchers and practitioners, supporting the development of anomaly detection frameworks capable of balancing scalability, interpretability, and resilience.

**Keywords**. anomaly detection, anomaly detection framework, knowledge discovery, machine learning, anomaly detection framework feature set (ADF$^2$S).

## 1. Introduction

In an era of exponential growth of data, the ability to detect anomalous behaviour within large and complex datasets has become more critical than ever. From cybersecurity threats and fraud detection to industrial system failures and health monitoring, anomaly detection serves as a foundational technique for ensuring reliability, security, and efficiency across domains. Anomalies, also known as outliers, novelties, or exceptions, represent patterns in data that deviate significantly from expected behaviour (Chandola et al., 2009). Their identification is often an early indication of critical events or hidden knowledge (Pimentel et al., 2014). Anomaly detection is naturally related to the broader field of knowledge discovery. As a central step in the knowledge discovery process,

anomaly detection enables the extraction of actionable and previously unknown insights from massive datasets. In this context, machine learning plays a crucial role by providing scalable, adaptable, and intelligent models capable of identifying complex and subtle deviations from norms (Aggarwal, 2013). Whether through supervised, unsupervised, or semi-supervised approaches, machine learning techniques are frequently at the core of modern anomaly detection frameworks. The importance of anomaly detection has grown rapidly with the increasing volume, velocity, and variety of data generated in real-time from various systems.

The presence of sensors, interconnected systems, and user-driven digital platforms has led to environments where millions of data points are produced every second. This data presents unique challenges, including the high dimensionality of data, concept drift, class imbalance, and lack of labelled data, particularly for rare or unknown anomalies (Goldstein and Uchida, 2016). These difficulties necessitate sophisticated frameworks capable of adaptive learning, contextual awareness, and explainability. Beyond technical complexity, anomaly detection also faces conceptual and operational challenges. The very definition of what constitutes an "anomaly" is often domain-specific and context-dependent (Chandola, et al., 2009). In some scenarios, anomalies may reflect genuine faults; in others, may reveal novelties or trends (Pimentel et al., 2014). Thus, frameworks must be precise in identifying irregularities and flexible enough to interpret their significance in a dynamic environment (Ahmed et al., 2015). New frontiers are being explored in areas such as explainable AI and graph-based anomaly detection, which integrate expert feedback into the learning process (Choi et al.,2022) (Eberle and Holder, 2015). As such, anomaly detection is an actual research area that adapts to emerging technological, societal, and methodological shifts.

## 2. From knowledge to knowledge discovery: foundations for anomaly detection

In the context of modern data-driven systems, the concept of knowledge extends far beyond raw information. Knowledge can be broadly defined as structured, contextualised, and actionable information, which enables informed decision-making, problem-solving, and innovation (Nonaka and Takeuchi, 1995). Unlike data, which consist of unprocessed facts, or information, which refers to organized data imbued with meaning, knowledge reflects an integration of experience, interpretation, and insight. With the digital transformation of industries and services, knowledge workers - individuals who engage in tasks involving critical thinking, analysis, and problem-solving - have become central to organizational value creation (Davenport, 2005). As data volume and complexity increase, manual or intuitive methods for deriving knowledge become insufficient, paving the way for computational support in uncovering patterns and relationships (Jansevskis and Osis, 2024). The paradox of today's digital ecosystem lies in the fact that while organisations are inundated with data, they often lack knowledge. Extracting useful, actionable insights from terabytes or petabytes of data demands systems that go beyond storage and computation, require systems that understand the semantics of information and can detect deviations from expected patterns (Thudumu et al., 2020). This is where knowledge discovery becomes indispensable.

Knowledge Discovery in Databases (KDD) refers to the process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996). This process focuses on transforming data into knowledge through a sequence of operations, including data selection, preprocessing, transformation, data mining, and interpretation/evaluation (Frawley et al., 1992). Within this process, data mining plays a crucial role by applying algorithms to extract patterns from data (Al-Faouri, 2023). In this context, anomaly detection is one of the core data mining tasks, alongside classification, clustering, and association rule learning (Klosgen and Zytkow, 1996). In contrast to traditional analytical approaches, knowledge discovery enables systems to identify unexpected behaviours, emergent trends, and hidden structures, which are often indicated by anomalies.

## 3. Overview of anomaly detection

Anomaly detection has been an intriguing field for researchers and practitioners that has been studied for centuries (Nassif et al., 2021). Numerous distinct methods and approaches have been developed over time to detect anomalies across different applications, domains, and contexts. Starting from statistical methods to machine learning and neural networks, anomaly detection methods have been developed and reshaped by employing sophisticated and complex methods that improve the performance and efficiency of such systems (Mishra and Kumar, 2022). The concept of anomalies is defined as patterns in data that do not conform to a well-defined notion of expected behaviour (Barnard and Stryker, 2023). As such, patterns or behaviours in data that deviate from normal behaviour are referred to as anomalies or outliers, depending on the domain or context in which anomaly detection techniques are applied (Steenwinckel et al., 2021) (Injadat et al., 2018). Examples of such applications include fraud detection, loan application processing, monitoring of medical conditions, cybersecurity intrusion detection, and fault detection for predictive maintenance (Nassif et al., 2021).

## 4. Anomaly detection frameworks

An essential consideration in the development of anomaly detection frameworks is the acknowledgement of the domain-specific nature of anomalies (Alam et al., 2019). This characteristic introduces significant challenges in creating a generic framework capable of detecting all forms of anomalies, regardless of domain-specific conditions or data characteristics (Pang et al., 2021). The complexity associated with high-dimensional data and contextual anomalies cannot be understated, as they prevent the generalizability of detection approaches across different applications (Choi et al., 2021). Moreover, there is a need for a comprehensive taxonomy that systematically analyses the diverse anomaly detection frameworks' features and capabilities based on their characteristics and level of granularity (Feng et al., 2025) (Zhou et al., 2022).

Anomaly detection can signal system faults, cyberattacks, or inefficiencies, making their timely and accurate detection crucial for operational resilience and efficiency (Cauteruccio et al., 2021) (Fan and You, 2024). Over the past decade, anomaly detection frameworks have evolved from rule-based approaches to sophisticated machine learning

driven systems capable of processing complex, high-dimensional, and temporal data (Chai et al., 2022). Modern anomaly detection frameworks and architectures often integrate components, including feature selection, model ensemble, time-series analysis, and explainability. These components are crucial for identifying relevant patterns and attributing anomalous behaviour to specific systemic or operational contexts. The effectiveness of such frameworks largely depends on how they leverage selected features, whether statistical, contextual, or domain-specific, to inform the detection process (Alam et al., 2019).

In this research, three influential anomaly detection frameworks are analysed in depth to examine how each leverages selected features for anomaly identification. These frameworks were chosen following a systematic review of 54 anomaly detection frameworks identified through searches of scientific databases (Scopus, ScienceDirect, and IEEE Xplore) using the terms "anomaly detection framework" and "anomaly detection architecture". Framework selection was guided by the following criteria: [1] the framework was published within the past ten years; and [2] the corresponding publication achieved a Field-Weighted Citation Impact (FWCI) of at least five according to Scopus, indicating a substantial contribution to the domain. In addition to these criteria, the selected frameworks represent distinct architectural paradigms and methodological diversity, thereby providing a comprehensive cross-section of the anomaly detection landscape. Based on these criteria, the following state-of-the-art anomaly detection frameworks were selected: "Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning" (Liu et al., 2015), "An Ensemble Learning Framework for Anomaly Detection in Building Energy Consumption" (Araya et al., 2017), and "Time-Series Anomaly Detection Service at Microsoft" (Ren et al., 2019). Collectively, these frameworks serve as representative examples of the broader design space, encompassing supervised and unsupervised learning, ensemble-based modelling, time-series anomaly detection, domain-specific integration, and production-grade scalability. Moreover, they have demonstrated significant impact in both academia and industry, as reflected in their citation volume, adoption, and influence on subsequent research and industrial solutions.

## 4.1. Opprentice: automatic anomaly detection through machine learning

Among the foundational contributions to the field of anomaly detection, the framework depicted in Figure 1, proposed in the research article "Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning", stands out for its practical orientation and system-level applicability (Liu et al., 2015). Unlike traditional methods that often require expert intervention for algorithm selection and parameter tuning, Opprentice introduces an end-to-end, automated anomaly detection system that leverages a meta-learning strategy to dynamically select the most appropriate detection algorithm based on the characteristics of the input data (Liu et al., 2015). This feature enhances usability and lowers the technical barrier for deployment in real-world monitoring systems. The framework is structured around several key components: a data collector and repository, a feature extractor, a model training and selection engine, and a runtime detector. Together, these components enable the system to process large volumes of heterogeneous system metrics, train and select the most suitable detection model, and

then apply this model for real-time anomaly detection with minimal computational overhead.

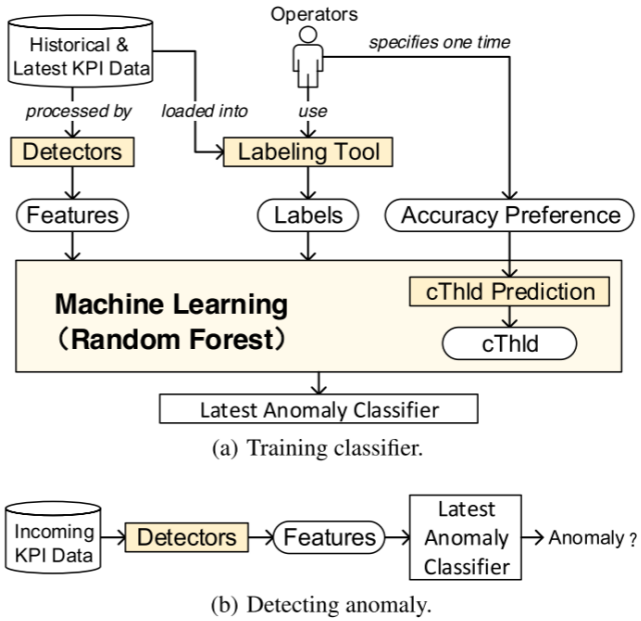


**Figure 1:** Opprentice anomaly detection architecture (Liu et al., 2015)

The framework is particularly relevant in large-scale IT operations and cloud environments, where telemetry data is generated continuously and must be processed in real time (Liu et al., 2015). By validating the approach on extensive production datasets, the study demonstrates high detection accuracy and low false-positive rates, thereby offering a scalable and efficient solution for anomaly detection in operational settings. The Opprentice anomaly detection framework contributes significant value to the domain by bridging the gap between theoretical models and practical, large-scale deployments. Moreover, it has influenced the design of subsequent industrial-grade detection systems, underscoring its enduring relevance and foundational role in the evolution of automated, machine learning–based anomaly detection frameworks.

## 4.2. Ensemble anomaly detection framework in energy consumption

The second selected framework in Figure 2, developed focuses on smart buildings, where energy consumption data presents a challenging time series prediction problem. In their work, the authors proposed an Ensemble Anomaly Detection (EAD) framework specifically designed to detect anomalous patterns in building energy consumption data (Araya et al., 2017). The EAD framework is based upon the core principles of Collective Contextual Anomaly Detection using the Sliding Window (CCAD-SW) framework,

adapted by the authors. The results indicated that the EAD framework improved the sensitivity of the CCAD-SW by 3.6% and reduced the false alarm rate by 2.7% (Araya et al., 2017). As a result, it demonstrates the effectiveness of the ensemble approach in enhancing anomaly detection capabilities.

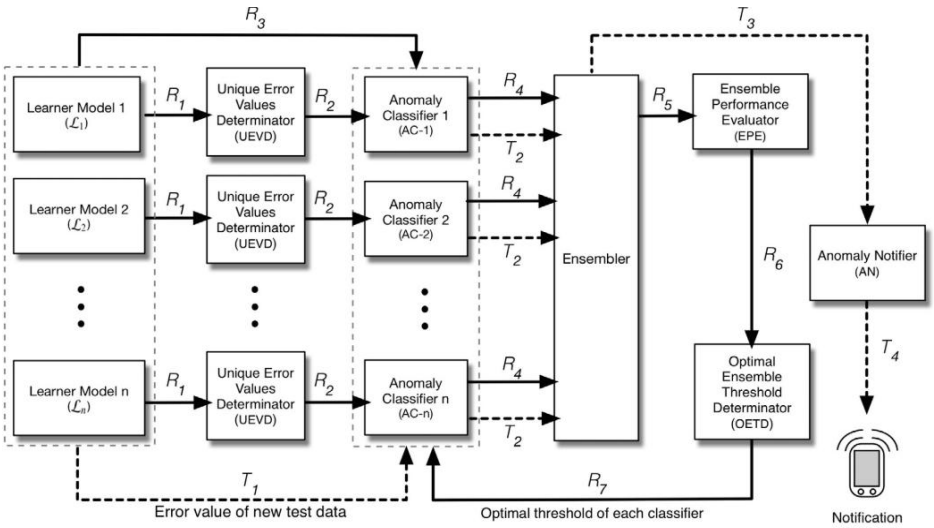


**Figure 2:** Ensemble anomaly detection (EAD) framework (Araya et al., 2017)

The primary aim of the framework is to enhance the accuracy and robustness of anomaly detection in the presence of complex, multivariate, and temporally dynamic energy usage profiles. The EAD framework employs a heterogeneous ensemble learning approach that integrates multiple base learners, including One-Class Support Vector Machines (OC-SVM), Isolation Forests, and Autoencoders, each capturing distinct characteristics of the data (Araya et al., 2017). By aggregating the outputs of these individual models, the ensemble produces a composite anomaly score that is more resilient to the limitations of any single method (Araya et al., 2017). These predictive components are trained to estimate expected energy consumption based on historical data and contextual variables. Anomalies are then inferred from deviations between predicted and observed values. The EAD framework enhances robustness and reliability by combining the outputs of pattern-based and prediction-based models using an ensemble approach. It is found that the optimal detection threshold for the ensemble is the result of an independent optimization process, emphasizing the importance of ensemble calibration.

## 4.3. Time series anomaly detection service at Microsoft

In this research, the authors present a comprehensive framework for time-series anomaly detection that represents a production-scale system deployed at Microsoft, as illustrated in Figure 3. The framework is designed for real-time monitoring of time-series data across a range of cloud-based applications, including Office and Azure, and emphasizes

scalability, automation, and low latency (Ren et al., 2019). To achieve these objectives, the authors propose a novel algorithm that combines Spectral Residual (SR) and Convolutional Neural Network (CNN) techniques. This work represents the first application of the SR model, originally developed for visual saliency detection, to the domain of time-series anomaly detection (Ren et al., 2019). The integration of SR and CNN significantly enhances detection performance, leading to substantial improvements in anomaly detection accuracy and robustness.

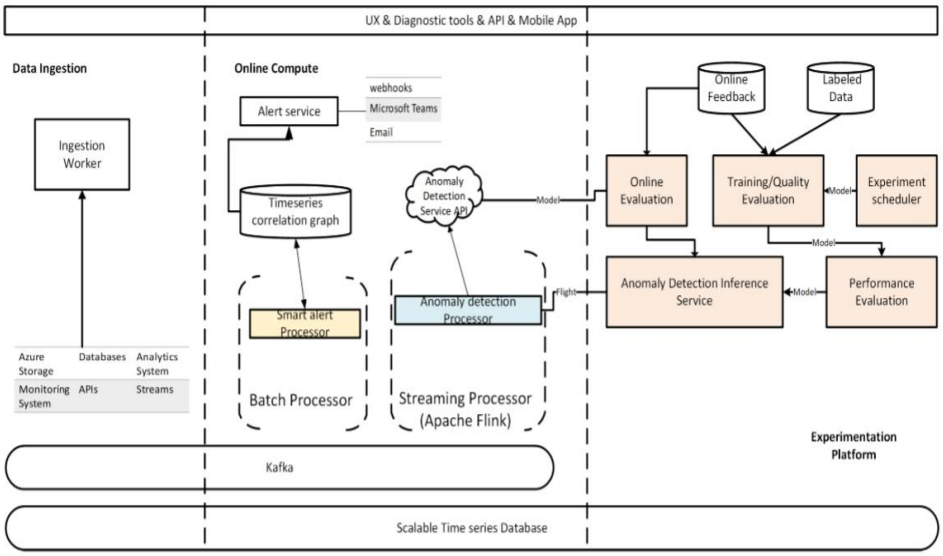


**Figure 3:** Time series anomaly detection system at Microsoft (Ren et al., 2019)

The proposed framework outperforms existing state-of-the-art models, achieving an F1-score improvement of over 20% on Microsoft production data (Ren et al., 2019), demonstrating the effectiveness of the combined approach in real-world applications. It is designed to be general and efficient, enabling seamless integration with online monitoring systems and providing rapid alerts for critical metrics, essential for large-scale, real-time operations. The framework maintains an unsupervised learning approach by generating synthetic anomalies to train the CNN. In practical deployments, it has allowed product teams to detect issues more quickly, reduce manual intervention, and accelerate diagnostic processes.

## 5. Anomaly detection framework feature set

The design and evaluation of anomaly detection frameworks require a structured understanding of the core capabilities these systems must possess. Based on a comprehensive review of the literature and the analysis of existing state-of-the-art frameworks mentioned in the previous section, the author proposes a feature set that

captures the most crucial dimensions along which anomaly detection frameworks can be characterised. As shown in Figure 4, the proposed Anomaly Detection Framework Feature Set (ADF²S) includes: Data Handling & Input Capabilities, Anomaly Detection Type Support, Detection Techniques Flexibility, Adaptability, Explainability and Interpretability, and Data Quality Aware Anomaly Detection. Each feature of the ADF²S is further divided into subcategories, which represent specific mechanisms, data properties, or learning paradigms relevant to the main feature.

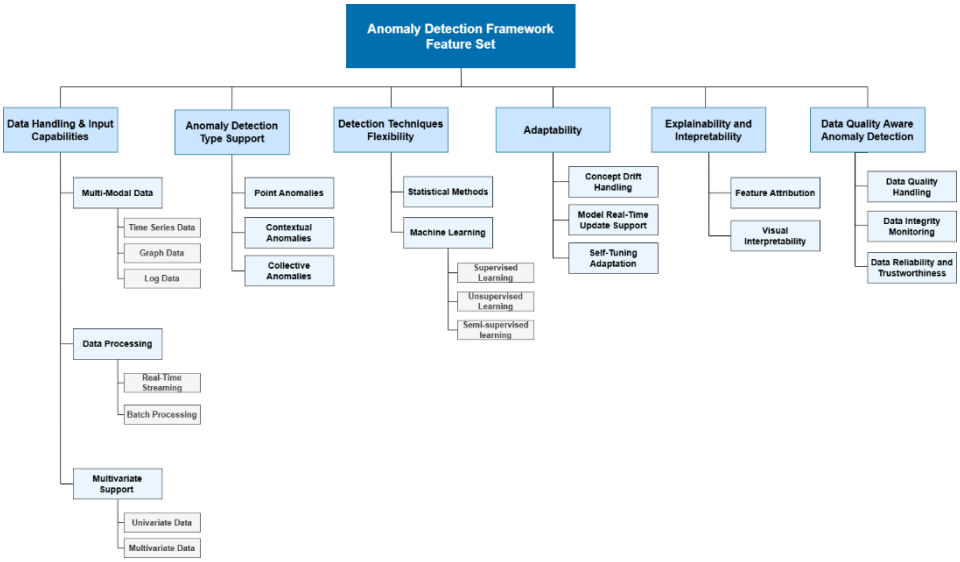


**Figure 4:** Anomaly detection framework feature set (ADF²S)

The primary categories in ADF²S capture high-level functional dimensions, such as data handling, detection techniques, and adaptability, because they reflect broad system-level requirements that recur across domains. However, these high-level capabilities are too coarse to meaningfully differentiate between frameworks. To address this, each main feature is decomposed into sub-features that represent the concrete technical mechanisms, algorithmic paradigms, or system design choices through which the high-level capability is operationalized. The hierarchical structure aligns with the way this research organizes anomaly detection methods and system capabilities. Grouping conceptually related mechanisms under a shared parent feature ensures that the taxonomy maintains conceptual coherence and follows established scientific conventions. Overall, the feature set taxonomy provides a conceptual platform for evaluating the completeness, flexibility, and domain suitability of anomaly detection frameworks, enabling a nuanced discussion of how design choices affect system performance and applicability.

The review followed an iterative synthesis approach: first, a broad corpus of anomaly detection studies, including machine learning–based approaches, time-series models, and system-level architectures, was examined to identify recurring functional requirements and modelling practices. These concepts were then cross-referenced with capabilities described in research articles and industry-grade platforms to ensure practical relevance and contemporary applicability. Features were grouped and refined through thematic clustering, allowing coherent high-level categories and their corresponding sub-features to emerge. This methodology ensured that the resulting ADF$^2$S taxonomy is grounded in scientific knowledge while reflecting operational considerations observed in real-world anomaly detection deployments. Consequently, the feature set represents an evidence-based synthesis of the key conceptual and technical elements consistently highlighted across both academic literature and industrial implementations.

## 5.1. Data handling and input capabilities anomaly detection feature

The capability of an anomaly detection framework largely depends on its ability to effectively handle and process the data on which it operates. The data handling and input capability feature captures a framework's flexibility in accommodating diverse data types, processing paradigms, and levels of structural complexity (Goldstein and Uchida, 2016). This feature is further subdivided into three components: multi-modal data support, data processing, and multivariate support, each of which plays a critical role in defining the framework's operational scope and applicability across domains.

Modern systems generate data from heterogeneous sources, including sensors, logs, network traffic, and structured time-series metrics. Anomaly detection frameworks that support multi-modal data can integrate insights across different data representations, enabling a more holistic understanding of system behaviour. In this taxonomy, multi-modal data is categorized into three subtypes: time-series data, graph data, and log data. Time-series data is the most prevalent input format for anomaly detection, particularly in domains such as industrial monitoring, finance, and healthcare (Choi et al., 2021) (Thudumu et al., 2020). Frameworks must process temporally ordered observations to capture trends, seasonality, and temporal anomalies (Ren et al., 2019).

Graph-structured data, increasingly important in cybersecurity, social networks, and network infrastructure, encodes relationships among entities. Frameworks supporting graph data can identify structural or topological anomalies, such as compromised nodes or anomalous information flows (Eberle and Holder, 2015). Log data, common in IT and software systems, consists of unstructured or semi-structured textual records. Effective anomaly detection frameworks parse, tokenize, and extract patterns from logs to detect deviations in event sequences or semantic content (Filzmoser et al., 2008).

Anomaly detection frameworks also differ in their data ingestion and analysis strategies, which are broadly categorized as batch processing and real-time streaming. Batch processing involves analysing data in fixed intervals or accumulated batches and is well-suited for retrospective analysis, scheduled system checks, or environments where immediate response is not critical (Xu et al., 2023). Due to its computational simplicity and ease of implementation, this approach is commonly adopted by traditional frameworks (Araya et al., 2017). In contrast, real-time streaming enables continuous processing of incoming data streams, allowing anomalies to be detected as they occur. Such capabilities are essential in domains requiring near real-time responses, including

financial fraud detection, cloud service monitoring, and fault detection in critical infrastructure (Ren et al., 2019).

In complex systems, anomalies often arise from interactions among multiple variables rather than from isolated data streams. The multivariate support sub-feature captures a framework's ability to model these interdependencies (Wang et al., 2025). Univariate approaches analyse each data stream independently and, while computationally efficient, may fail to detect correlated anomalies spanning multiple variables. Multivariate approaches, by contrast, analyse multiple data streams simultaneously, enabling the detection of more complex and interrelated anomaly patterns. Collectively, these data handling and input capabilities define how effectively an anomaly detection framework can operate across varying data types, processing requirements, and system complexities.

## 5.2. Anomaly detection type support anomaly detection feature

The anomaly detection type support feature characterizes the types of anomalies a framework is capable of identifying, which are closely related to the nature of the underlying data. Broadly, anomalies are commonly classified into three categories: point anomalies, contextual anomalies, and collective anomalies (Chandola et al., 2009). A point anomaly occurs when an individual data instance significantly deviates from the rest of the dataset and is therefore considered anomalous. This is the simplest and most commonly addressed form of anomaly. Examples include a sudden spike in a patient's blood pressure or an unusually large transaction amount in a banking system. A contextual anomaly arises when a data instance is anomalous only within a specific context, despite appearing normal in other circumstances (Chandola et al., 2009). In such cases, contextual information, such as time, location, or environmental conditions, is essential for correct interpretation. For example, a temperature of 10 °C may be normal during winter but anomalous during summer in a given region. Collective anomalies refer to groups of related data instances that are anomalous when considered together, even though individual instances within the group may not appear anomalous on their own (Araya et al., 2017). These anomalies often emerge from sequential or correlated patterns in the data. A typical example is a series of login attempts from different IP addresses within a short time window, which may indicate a coordinated attack (Choi et al., 2021). Because collective anomalies are not easily identifiable at the level of individual data points, long-term contextual and temporal analysis is particularly important for their detection. Consequently, support for different anomaly types constitutes a critical characteristic of anomaly detection frameworks. It influences the choice of detection techniques, contextual modelling strategies, and overall system design, thereby aligning framework capabilities with the specific anomaly patterns present in the target application domain.

## 5.3. Detection techniques flexibility anomaly detection feature

The detection technique flexibility feature captures the range of methods and algorithms employed by anomaly detection frameworks across different system architectures. Statistical anomaly detection techniques represent some of the earliest approaches in this domain (Chandola et al., 2012). These methods model normal data behaviour using

statistical assumptions and identify anomalies as deviations from the learned distribution. Common statistical approaches include parametric techniques such as Gaussian distribution modelling, hypothesis testing, and time-series models (Bhuyan et al., 2014). In recent decades, machine learning–based techniques have become increasingly prominent in anomaly detection. Machine learning aims to automate knowledge acquisition from data examples (Bose and Mahapatra, 2001) and is widely used to construct models that distinguish between normal and anomalous patterns (Mahapatra and Bose, 2001). Advanced machine learning algorithms are particularly effective in handling high-dimensional data and learning complex patterns, thereby improving detection accuracy and efficiency (Herr et al., 2021).

Based on the availability of labelled data, anomaly detection techniques are commonly categorized into supervised, unsupervised, and semi-supervised approaches. Supervised anomaly detection relies on labelled datasets containing both normal and anomalous instances. In this setting, a predictive model is trained to discriminate between the two classes. However, supervised approaches face several challenges, including the severe imbalance between normal and anomalous samples and the difficulty of obtaining accurate and representative labels for anomalies (Chatterjee and Ahmed, 2022). Additionally, discrepancies between training and testing anomaly distributions can degrade performance. In contrast, unsupervised anomaly detection does not require labelled training data and operates under the assumption that anomalies are rare compared to normal instances. These methods identify deviations based on inferred data structure or distribution. While widely applicable, unsupervised approaches may produce high false alarm rates when the underlying assumptions about anomaly rarity or data distribution do not hold (Nassif et al., 2021). Consequently, much recent research has focused on improving unsupervised techniques, particularly for large-scale and unlabelled datasets (Pang et al., 2021).

Semi-supervised anomaly detection occupies a middle ground, where training data consists only of labelled normal instances. Any data point that deviates from the learned normal behaviour is flagged as anomalous. Because anomaly labels are not required, semi-supervised methods are often more practical than fully supervised approaches. Many semi-supervised techniques can also be adapted to operate in an unsupervised manner by assuming that training data contains very few anomalies, which minimally affect model learning (Nassif et al., 2021). Across application domains such as finance, manufacturing, and the Internet of Things (IoT), supervised, unsupervised, and semi-supervised machine learning approaches have been extensively studied and applied (Torr and Murray, 1993) (Marsland, 2001). Recent studies suggest that semi-supervised methods often outperform supervised approaches in real-world settings, particularly when labelled anomaly data are scarce or incomplete (Sunny et al., 2022). These techniques are especially suitable for high-dimensional datasets with large proportions of unlabelled data, a common characteristic in many operational environments (Filzmoser et al., 2008).

## 5.4. Adaptability anomaly detection feature

The adaptability feature captures a framework’s ability to maintain accuracy and relevance in dynamic environments where data distributions evolve. In real-world systems, such as cloud infrastructure and financial markets, the assumption of

stationarity rarely holds (Chatterjee and Ahmed, 2022). Consequently, anomaly detection models must incorporate mechanisms to respond to shifting behavioural patterns, commonly referred to as concept drift, to prevent performance degradation over time (Chatterjee and Ahmed, 2022). This dimension comprises three interrelated sub-features that collectively define a framework's adaptability: concept drift handling, model real-time update support, and self-tuning adaptation.

Concept drift refers to gradual changes in the underlying data distribution that alter the definition of normal behaviour. If left unaddressed, such changes can lead to increased false positives or missed anomalies. Frameworks that support concept drift handling can detect these shifts and adjust their internal models accordingly, for example, by recalibrating decision boundaries, retraining models, or updating statistical parameters (Chatterjee and Ahmed, 2022).

Model real-time update support denotes a framework's ability to incrementally update its models in response to incoming data, rather than relying on full retraining. This capability is critical in real-time applications where batch retraining is computationally expensive or insufficiently responsive to rapid system changes (Zuo et al., 2019). In anomaly detection, real-time updates enable continuous model refinement as new patterns emerge, allowing for rapid adaptation. Frameworks with this capability are particularly well suited to streaming data environments, such as network monitoring and predictive maintenance in sensor networks (Choi et al., 2021).

Self-tuning adaptation refers to internal mechanisms that automatically adjust detection parameters, such as thresholds, sensitivity levels, or window sizes, without manual intervention (Alam et al., 2019). This capability reduces reliance on domain expert tuning, thereby improving scalability and deployment efficiency. By adapting to variations in noise levels, system load, and behavioural patterns, self-tuning mechanisms help anomaly detection frameworks maintain robust performance across changing operational conditions.

## 5.5. Explainability and interpretability anomaly detection feature

In modern anomaly detection frameworks, explainability and interpretability have emerged as essential features, particularly in safety-critical and regulated application domains (Zhong et al., 2023). These capabilities describe the extent to which a model's decision-making process, specifically its anomaly detection outcomes, can be understood, traced, and justified by human users. As anomaly detection systems are increasingly integrated into operational pipelines, the ability to explain why a particular data point is classified as anomalous is critical for establishing trust, ensuring accountability, and enabling informed responses (Choi et al., 2022). This feature dimension comprises two primary subcomponents: feature attribution and visual interpretability, each contributing in distinct ways to user understanding of the detection process.

Feature attribution refers to a framework's ability to identify and quantify the influence of individual input features on an anomaly detection decision. In machine learning–based systems, particularly those employing ensemble or deep learning methods, detection mechanisms are often opaque (Zhong et al., 2023). Feature attribution techniques mitigate this opacity by producing saliency scores or importance rankings that highlight which variables or input patterns contributed most strongly to an

anomaly decision. For example, when a spike in network latency is flagged as anomalous, feature attribution may indicate that the deviation is driven primarily by unusual memory usage rather than CPU load. Such insights are essential for root cause analysis, model validation, and debugging by domain experts (Choi et al., 2022). In addition, model-agnostic explanation methods can be integrated to provide interpretable outputs even for black-box models.

Visual interpretability concerns the presentation of anomaly detection results through intuitive and informative visual representations. Effective visualization enables system operators and analysts to assess the validity of detected anomalies, explore deviations across time and variables, and make informed decisions based on system feedback. By rendering complex statistical or algorithmic outputs accessible to non-technical stakeholders, visual interpretability significantly enhances the usability and operational integration of anomaly detection frameworks (Zhong et al., 2023).

## 5.6. Data quality-aware anomaly detection feature

Anomaly detection frameworks are rarely applied to fully reliable data sources; instead, they typically operate under conditions of uncertainty, noise, missing values, or partial system failure. To maintain performance and reliability under such conditions, frameworks must incorporate mechanisms that recognize and adapt to data quality issues. This motivates the inclusion of data quality–aware anomaly detection as an important feature dimension, emphasizing that effective anomaly detection must be sensitive to the integrity, completeness, and reliability of input data (Kittler et al., 2014). When properly integrated, data quality awareness improves detection precision and the trustworthiness of system outputs. This feature dimension is defined by three key subcategories.

Data quality handling refers to a framework's ability to process and remain robust to common data inconsistencies, including noise, missing values, and outliers. In many domains, such as sensor networks, log telemetry, and business analytics, data degradation is unavoidable (Kittler et al., 2014). To mitigate its impact, frameworks employ preprocessing or model-level strategies, such as imputation, smoothing, or robust statistical techniques, ensuring that detected anomalies are not artefacts of poor data quality (Alam et al., 2019). Data quality handling focuses on input resilience, and data integrity monitoring concerns a framework's ability to detect failures in the data collection, transmission, or storage pipeline. These failures may include corrupted files, communication breakdowns, timestamp misalignment, or format inconsistencies, all of which can compromise anomaly detection results. Frameworks equipped with integrity monitoring mechanisms may enforce schema validation, track metadata consistency, or generate alerts when data streams deviate from expected structural norms (Sunny et al., 2022). This capability is particularly important in automated pipelines, where detection reliability depends on the integrity of upstream data sources. Beyond technical integrity, data reliability and trustworthiness capture a framework's ability to assess and communicate confidence in both the input data and its anomaly detection outputs. This may involve assigning reliability scores to data sources, quantifying uncertainty in model predictions, or flagging results derived from potentially compromised inputs.

## 6. Anomaly detection frameworks' alignment with feature sSet

This research introduces the Anomaly Detection Framework Feature Set (ADF²S) to analyze the selected anomaly detection frameworks. The objective is to assess how ADF²S aligns with the architectural and functional characteristics of these frameworks. Accordingly, a systematic evaluation of Opprentice (Liu et al., 2015), the Ensemble Anomaly Detection (EAD) framework (Araya et al., 2017), and the Microsoft Time-Series Anomaly Detection service (Ren et al., 2019) is conducted using the proposed ADF²S. Table 1 summarizes the presence or absence of each feature, while the subsequent discussion provides interpretative insights into the architectural focus, strengths, and limitations of each framework. The evaluation follows a structured feature-mapping methodology, in which each ADF²S feature and sub-feature is examined against the technical descriptions, architectural details, and methodological explanations reported in the corresponding publications. For each feature, the degree of support is assessed using a three-level classification scheme: (✓) Supported, indicating that the framework explicitly implements or relies on the feature as part of its core design; (✗) Not Supported / Not Mentioned, indicating that no evidence of the feature is provided or that the capability is absent; and (!) Partially Supported, denoting cases in which a feature is indirectly present, incompletely supported, applied only in a limited context, or implied without full integration into the framework's operational pipeline. This interpretive classification enables a consistent and reproducible evaluation across heterogeneous frameworks, particularly given the diversity of algorithmic approaches, data modalities, and architectural assumptions represented in the selected works.

The Opprentice framework places strong emphasis on data preprocessing and the use of supervised machine learning for anomaly classification. It supports both univariate and multivariate time-series data and employs an ensemble of statistical detectors for feature extraction prior to classification using a Random Forest model (Liu et al., 2015). This design enables effective detection of point anomalies in well-labelled batch datasets. Opprentice also incorporates a basic self-tuning adaptation mechanism through threshold smoothing, providing limited adaptive behaviour. However, the framework does not support real-time streaming and instead relies on periodic batch retraining. In addition, Opprentice lacks model real-time update capabilities, support for contextual or collective anomaly types, and feature attribution mechanisms for interpretability. The framework also does not explicitly address data integrity monitoring or provide explainable outputs, which limits its applicability in regulated or safety-critical environments.

The Ensemble Anomaly Detection (EAD) framework is specifically tailored to the energy consumption domain and employs supervised learning through a combination of pattern-based collective contextual anomaly detection using sliding windows (CCAD-SW) and prediction-based detectors, including Support Vector Regression (SVR) and Random Forest models (Araya et al., 2017).

**Table 1:** Anomaly Detection Frameworks Feature Set (ADF$^2$S) Alignment

| Anomaly Detection Framework Feature Set (ADF$^2$S) | | | Opprentice anomaly detection architecture (Liu, 2015) | Ensemble anomaly detection (EAD) framework (Araya, 2017) | Time series anomaly detection at Microsoft service (Ren, 2019) |
|---|---|---|---|---|---|
| **Data Handling & Input Capabilities** | Multi-Modal Data | Time-series Data | ✓ | ✓ | ✓ |
| | | Graph Data | X | X | X |
| | | Log Data | X | X | X |
| | Data Processing | Real-time streaming | X | X | ✓ |
| | | Batch processing | ✓ | ✓ | ✓ |
| | Multivariate Support | Univariate Data | ✓ | ✓ | ✓ |
| | | Multivariate Data | ✓ | ✓ | ✓ |
| **Anomaly Detection Type Support** | Point Anomalies | | ✓ | ✓ | ✓ |
| | Contextual Anomalies | | X | !(partial) | ✓ |
| | Collective Anomalies | | X | ✓ | ✓ |
| **Anomaly Detection Technique Flexibility** | Statistical Methods | | ✓ | ✓ | ✓ |
| | Machine Learning | Supervised Learning | ✓ | ✓ | X |
| | | Unsupervised Learning | X | X | ✓ |
| | | Semi-supervised Learning | X | X | ✓ |
| **Adaptability** | Concept Drift Handling | | !(partial) | X | ✓ |
| | Model Real-Time Update Support | | X | X | ✓ |
| | Self-Tuning Adaptation | | ✓ | X | ✓ |
| **Explainability and Intepretability** | Feature Attribution | | X | X | !(partial) |
| | Visual Interpretability | | ✓ | X | ✓ |
| **Data Quality Aware** | Data Quality Handling | | ✓ | X | ✓ |

✓ **Supported** - The framework fully implements this feature as part of its design or methodology.
X **Not Supported / Not Mentioned** - The framework does not implement or indicate the presence of the feature.
! (Partial) - The framework provides limited or indirect support for this feature.

It supports multivariate data by integrating contextual features such as weather conditions and occupancy information, and is among the few frameworks that explicitly address collective anomalies. EAD's primary strengths lie in its high detection accuracy within its target domain and its modular ensemble-based architecture. Nevertheless, the framework is fundamentally batch-oriented and lacks support for real-time streaming, incremental model updates, and adaptive threshold tuning. Moreover, EAD does not incorporate data quality assessment mechanisms or provide explainability and interpretability features, such as analytics or feature analysis.

The Microsoft Time-Series Anomaly Detection service represents a production-grade system designed for scalability and operational efficiency (Ren et al., 2019). It is engineered to process millions of time-series metrics daily and supports unsupervised and semi-supervised detection techniques based on the integration of Spectral Residual (SR) and Convolutional Neural Networks (CNNs). The framework offers real-time streaming capabilities, adaptive learning, and concept drift handling through synthetic anomaly generation and feedback mechanisms. It also includes interpretability support in the form of dashboards, reports, and diagnostic outputs, making it well-suited for cloud service monitoring and enterprise telemetry analysis. Additionally, it does not explicitly address data integrity monitoring or data reliability and trustworthiness assessment. While visual diagnostics enhance operational transparency, the framework lacks fine-grained feature-level attribution methods characteristic of more recent explainable machine learning approaches.

Although the selected anomaly detection frameworks demonstrate advanced capabilities and have a significant impact in the field, none fully support the complete Anomaly Detection Framework Feature Set (ADF2S). Opprentice is optimized for supervised batch learning, offering moderate robustness but limited scalability and real-time adaptability. The Ensemble Anomaly Detection (EAD) framework provides high domain-specific accuracy but remains static and lacks transparency in interpretability. In contrast, Microsoft's Time-Series Anomaly Detection is the most comprehensive and scalable, though it sacrifices deep model explainability for high-throughput, unsupervised deployment.

These findings highlight the trade-offs between adaptability, interpretability, and scalability that current frameworks must navigate. They also emphasize the need for modular and extensible designs, where features such as data quality awareness, explainability, and adaptability are integral components of anomaly detection systems. Comparative analysis of Opprentice (Liu et al., 2015), The Ensemble Anomaly Detection (EAD) (Araya et al., 2017), and Microsoft's Time-Series service (Ren et al., 2019) reveals distinct design methodologies and feature priorities aligned with their respective application domains and operational contexts. This feature-based comparison also identifies gaps that future research and framework development can address.

## 7. Conclusions and discussion

This study proposes a structured set of features for evaluating and designing anomaly detection frameworks, offering a comprehensive taxonomy of functional and architectural dimensions that define the capabilities of such systems. The anomaly detection framework feature set (ADF$^2$S) encompasses six major dimensions: Data

Handling and Input Capabilities, Anomaly Detection Type Support, Detection Technique Flexibility, Adaptability, Explainability and Interpretability, and Data Quality-Aware Anomaly Detection, each subdivided to capture more granular operational characteristics. Through a systematic analysis of three influential frameworks, the proposed ADF$^2$S proved effective in identifying the architectural strengths and limitations of each system. Opprentice framework demonstrated strong support for multivariate, supervised learning with efficient data preprocessing and limited self-tuning capabilities, but lacked real-time adaptability, interpretability, and data integrity and reliability capabilities. The Ensemble Anomaly Detection (EAD) framework, while rich in domain-specific integration and collective anomaly support, remained static, with no support for adaptive learning and data quality-aware anomaly detection. In contrast, Microsoft Time Series Anomaly Detection emerged as the production-ready solution, offering scalability, streaming data support, unsupervised adaptability, and visual interpretability, though it lacked broader data quality support and deep feature-level explainability. These findings reveal that no single framework fully satisfies all feature sets, highlighting gaps in scalability, adaptability, and interpretability. Moreover, the absence of data quality-aware anomaly detection considerations across the framework's points to a gap in current state-of-the-art developments. The proposed anomaly detection framework feature set (ADF$^2$S), thus offers practical guidance for designing anomaly detection frameworks that are modular, adaptive, interpretable, and robust to data fluctuations.

Future research may further extend the ADF$^2$S taxonomy toward the development of a unified anomaly detection framework that combines architectural abstraction with domain-specific adaptability. Although ADF$^2$S already provides a structured and comprehensive lens for the qualitative analysis of anomaly detection frameworks, this study primarily focuses on taxonomic synthesis rather than empirical validation. Consequently, quantitative metrics demonstrating the utility or performance of ADF$^2$S are not included. Future studies could operationalise the ADF$^2$S into measurable constructs, such as feature-coverage indices, framework-completeness scores, or decision-support utility measures to empirically assess how effectively the feature set supports system design, comparison, and selection. Such quantitative extensions, aligned with approaches adopted in related domain-specific anomaly detection evaluations, are introduced in the research study (Zadeja and Osis, 2025), thereby strengthening the evidence base of ADF$^2$S and broadening its applicability across diverse anomaly detection contexts.

## Acknowledgements

The author has used Grammarly (Grammarly, 2025) writing assistant platform for grammatical and paraphrasing needs. The research paper is written by the author exclusively.

## References


Aggarwal, C. C. (2013). *An introduction to outlier analysis*, in *Outlier Analysis*, Springer, New York. DOI: 10.1007/978-1-4614-6396-2_1.

Ahmed, M., Mahmood, A. N., Hu, J. (2015). A survey of network anomaly detection techniques, *J. Netw. Comput. Appl.* DOI: 10.1016/j.jnca.2015.11.016.

Alam, M. R., Gerostathopoulos, I., Prehofer, C., Attanasi, A., Bures, T. (2019). A framework for tunable anomaly detection, in *Proc. IEEE Int. Conf. Software Architecture (ICSA 2019)*, Hamburg, Germany, pp. 201–210. DOI: 10.1109/ICSA.2019.00029.

Al-Faouri, A. H. (2023). Adopting data mining as a knowledge discovery tool: The influential factors from the perspectives of information systems managers, *Inf. Sci. Lett.* **12**(5), 1851–1861. DOI: 10.18576/isl/120529.

Araya, D. B., Grolinger, K., ElYamany, H. F., Capretz, M. A., Bitsuamlak, G. (2017). An ensemble learning framework for anomaly detection in building energy consumption, *Energy Build.* **151**, 191–206. DOI: 10.1016/j.enbuild.2017.02.05.

Barnard, J., Stryker, C. (2023). *What is anomaly detection?* Available at https://www.ibm.com/think/topics/anomaly-detection.

Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools, *IEEE Commun. Surv. Tutor.* **16**(1), 303–336. DOI: 10.1109/SURV.2013.052213.00046.

Bose, I., Mahapatra, R. K. (2001). Business data mining—a machine learning perspective, *Inf. Manage.* **39**(3), 211–225. DOI: 10.1016/S0378-7206(01)00091-X.

Cauteruccio, F., Cinelli, L., Corradini, E., Terracina, G., Ursino, D., Virgili, L., Fortino, G. (2021). A framework for anomaly detection and classification in multiple IoT scenarios, *Future Gener. Comput. Syst.* DOI: 10.1016/j.future.2020.08.010.

Chai, Z., Liu, Y., Wang, M., Guo, Y., Shi, F., Li, Z., Du, J. (2022). Quantum anomaly detection of audio samples with a spin processor in diamond, preprint, arXiv:2201.10263. DOI: 10.48550/arXiv.2201.10263.

Chandola, V., Banerjee, A., Kumar, V. (2009). Anomaly detection: A survey, *ACM Comput. Surv.* **41**(3), 1–58. DOI: 10.1145/1541880.1541882.

Chandola, V., Banerjee, A., Kumar, V. (2012). Anomaly detection for discrete sequences: A survey, *IEEE Trans. Knowl. Data Eng.* **24**, 823–839. DOI: 10.1109/TKDE.2010.235.

Chatterjee, A., Ahmed, B. S. (2022). IoT anomaly detection methods and applications: A survey, *Internet Things*. DOI: 10.1016/j.iot.2022.100568.

Choi, H., Kim, D., Kim, J., Kim, J., Kang, P. (2022). Explainable anomaly detection framework for predictive maintenance in manufacturing systems, *Appl. Soft Comput.* DOI: 10.1016/j.asoc.2022.109147.

Choi, K., Yi, J., Park, C., Yoon, S. (2021). Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines, *IEEE Access* **9**, 120043–120065. DOI: 10.1109/ACCESS.2021.3107975.

Davenport, T. H. (2005). *Thinking for a Living: How to Get Better Performances and Results from Knowledge Workers*, Harvard Business Review Press, Boston.

Eberle, W., Holder, L. B. (2015). Scalable anomaly detection in graphs, *Intell. Data Anal.* **19**, 57–74. DOI: 10.3233/IDA-140696.

Fan, Z., You, Z. (2024). Research on network intrusion detection based on XGBoost algorithm and multiple machine learning algorithms, in *Proc. Int. Conf.*, pp. 161–166. DOI: 10.54254/2753-8818/31/20241171.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework, in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, pp. 82–88.

Feng, W., Cao, Y., Chen, Y., Wang, Y., Hu, N., Jia, Y., Gu, Z. (2025). Multi-granularity user anomalous behavior detection, *Appl. Sci.* **15**(1). DOI: 10.3390/app15010128.

Filzmoser, P., Maronna, R., Werner, M. (2008). Outlier identification in high dimensions, *Comput. Stat. Data Anal.* **52**(3), 1694–1711. DOI: 10.1016/j.csda.2007.05.018.

Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C. J. (1992). Knowledge discovery in databases: An overview, *AI Mag.* **13**(3). DOI: 10.1609/aimag.v13i3.1011.

Grammarly (2025). *Grammarly*. Available at https://www.grammarly.com.

Goldstein, M., Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, *PLoS ONE* **11**(4). DOI: 10.1371/journal.pone.0152173.

Herr, D., Obert, B., Rosenkranz, M. (2021). Anomaly detection with variational quantum generative adversarial networks, *Quantum Sci. Technol.* DOI: 10.1088/2058-9565/ac0d4d.

Injadat, M., Salo, F., Nassif, A. B., Essex, A., Shami, A. (2018). Bayesian optimization with machine learning algorithms towards anomaly detection, in *Proc. IEEE GLOBECOM 2018*, Abu Dhabi, United Arab Emirates, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647714.

Jansevskis, M., Osis, K. (2024). Securing the future: The role of knowledge discovery frameworks, in Sipola, T., Alatalo, J., Wolfmayr, M., Kokkonen, T. (eds), *Artificial Intelligence for Security*, Springer, Cham. DOI: 10.1007/978-3-031-57452-8_5.

Kittler, J., Christmas, W., de Campos, T., Windridge, D., Yan, F., Illingworth, J., Osman, M. (2014). Domain anomaly detection in machine perception: A system architecture and taxonomy, *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(5), 845–859. DOI: 10.1109/TPAMI.2013.209.

Klosgen, W., Zytkow, J. (1996). Knowledge discovery in databases terminology, in Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge, MA, pp. 573–592.

Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., Feng, M. (2015). Opprentice: Towards practical and automatic anomaly detection through machine learning, in *Proc. ACM SIGCOMM Internet Measurement Conference (IMC 2015)*, Tokyo, Japan. DOI: 10.1145/2815675.2815679.

Mahapatra, I., Bose, R. K. (2001). Business data mining—a machine learning perspective, *Inf. Manage.*, 211–225. DOI: 10.1016/S0378-7206(01)00091-X.

Marsland, S. (2001). *On-line novelty detection through self-organisation, with application to inspection robotics*, PhD thesis, University of Manchester.

Mishra, S., Kumar, M. (2022). CNN-based anomaly detection in complex systems, *Computology: J. Appl. Comput. Sci. Intell. Technol.*, 54–57. DOI: 10.17492/computology.v2i2.2206.

Nassif, A., Talib, M., Nasir, Q., Dakalbab, F. (2021). Machine learning for anomaly detection: A systematic review, *IEEE Access* **9**, 78658–78700. DOI: 10.1109/ACCESS.2021.3083060.

Nonaka, I., Takeuchi, H. (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, Oxford. DOI: 10.1093/oso/9780195092691.001.0001.

Pang, G., Shen, C., Cao, L., van den Hengel, A. (2021). Deep learning for anomaly detection: A review, *ACM Comput. Surv.* **54**(2). DOI: 10.1145/3439950.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. (2014). A review of novelty detection, *Signal Process.* **99**, 215–249. DOI: 10.1016/j.sigpro.2013.12.026.

Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Zhang, Q. (2019). Time-series anomaly detection service at Microsoft, in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. DOI: 10.48550/arXiv.1906.03821.

Steenwinckel, B., Paepe, D., Hautte, S., Heyvaert, P., Bentefrit, M., Moens, P., Van Herwegen, S. (2021). FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing learning, *Future Gener. Comput. Syst.*, 30–48. DOI: 10.1016/j.future.2020.10.015.

Sunny, J. S., Patro, C. P., Karnani, K., Pingle, S. C., Lin, F., Anekoji, M., Ashili, S. (2022). Anomaly detection framework for wearables data: A perspective review on data concepts, data analysis algorithms and prospects, *Sensors* **22**(3). DOI: 10.3390/s22030756.

Thudumu, S., Branch, P., Jin, J., Singh, J. (2020). A comprehensive survey of anomaly detection techniques for high-dimensional big data, *J. Big Data*. DOI: 10.1186/s40537-020-00320-x.

Torr, P., Murray, D. (1993). Outlier detection and motion segmentation, in *Proc. SPIE*, pp. 432–443.

Wang, F., Jiang, Y., Zhang, R., Wei, A., Xie, J., Pang, X. (2025). A survey of deep anomaly detection in multivariate time series: Taxonomy, applications, and directions, *Sensors*. DOI: 10.3390/s25010190.

Xu, H., Pang, G., Wang, Y., Wang, Y. (2023). Deep isolation forest for anomaly detection, *IEEE Trans. Knowl. Data Eng.* **35**(12), 12591–12604. DOI: 10.1109/TKDE.2023.3270293.

Zadeja, I., Osis, K. (2025). Anomaly detection framework in cybersecurity: Features and design considerations, in Wolfmayr, M., Sipola, T., Kokkonen, T. (eds), *Emerging Technologies in Cybersecurity*, Springer.

Zhong, L., Zhang, Y., van Leeuwen, M. (2023). A survey on explainable anomaly detection, *ACM Trans. Knowl. Discov. Data* **18**(1). DOI: 10.1145/3609333.

Zhou, Y., Ren, H., Li, Z., Pedrycz, W. (2022). Anomaly detection based on a granular Markov model, *Expert Syst. Appl.* **187**. DOI: 10.1016/j.eswa.2021.115744.

Zuo, X., Yang, X., Dou, Z., Wen, J. (2019). Experience report: Deep learning-based system log analysis, *TREC Proc.* DOI: 10.48550/arXiv.2107.05908.

Received May 24, 2025, revised December 8, 2025, accepted January 16, 2026