

A Software-Defined IoT Analytics Platform for Multi-Method Anomaly Detection Using Docker and the ELK Stack

Khalid S. Al-Tahat, Faculty of Computer Studies, Arab Open University, Amman, Jordan, Email: k_tahat@aou.edu.jo

Hamzeh J. Aljawawdeh, Faculty of Information Technology, Zarqa University, Zarqa, Jordan, Email: hamzeh.aljawawdeh@zu.edu.jo

Mohammad Azmi Al-madi, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan, Email: m.almadi@zuj.edu.jo, ORCID: [0000-0001-9941-039X]

Shorouq Ata Elmanaseer, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan, Email: Almanaseershorouq@gmail.com, ORCID:

[0000-0002-6045-4958]

Abdallah AL Sabbagh, Faculty of Engineering, Beirut Arab University, Tripoli, Lebanon, Email: a.alsabbagh@bau.edu.lb, ORCID: [0000-0002-9015-8498]

Corresponding Author: Khalid S. Al-Tahat, Email: k_tahat@aou.edu.jo

JSSSEE | eISSN: 2979-143X | Vol. 1 No. 2 (2026) | OPEN ACCESS

<https://doi.org/10.66823/bsbjij38>

Abstract

The rapid growth of Internet of Things (IoT) deployments has resulted in massive streams of heterogeneous sensor data, making dependable anomaly detection indispensable for supporting the security, operational efficiency, and reliability of a system. Nonetheless, several current approaches rely on a single detection technique, limiting robustness when handling noisy and partially labelled IoT data. This paper presents a software-defined IoT analytics platform that incorporates several anomaly detection techniques through a containerised ELK (Elasticsearch-Logstash-Kibana) architecture deployed using Docker. The platform consolidates the use of supervised machine learning, fuzzy logic inference, unsupervised Isolation Forest approach, and rule-based detection to assess real-world sensor data and improve the robustness of anomaly detection performance. Experiments were carried out on a dataset that contains 9,606 IoT sensor records, where the most effective performance was achieved by the supervised learning model with 98.14 precision, 96.66% accuracy, an F1-score of 93.54%, and 89.35% recall. On the other hand, 87.70% accuracy was achieved by the fuzzy logic model with a 75.71% F1-score. Additionally, 426 high-confidence anomalies were determined by the cross-method intersection analysis, revealing the effectiveness of integrating heterogeneous detection approaches. The experimental results demonstrate that the produced platform strengthens the robustness of anomaly detection while preserving cost efficiency, scalability,

and low deployment complexity, and ensuring its appropriateness for industrial and research IoT monitoring applications.

Keywords: Internet of Things; Anomaly Detection; ELK Stack; Software-Defined IoT; Fuzzy Logic; Machine Learning

1. Introduction

The fast adoption of Internet of Things (IoT) technologies has facilitated the implementation of large-scale sensor networks in a wide range of operational contexts, which include smart cities, environmental monitoring, industrial automation, and healthcare systems. These environments incessantly create high-volume and high-velocity data streams, which should be processed and analysed in a

reachable real-time to enhance automation, monitoring, and data-driven decision-making (Hassan et al., 2020; Singh et al., 2020; Soori et al., 2023; Umar & Jordanov, 2026). Since IoT technologies pointedly enhance system intelligence and operational efficiency, the increasing scale, including the complexity of these systems, introduces new challenges that are related to anomaly detection and data reliability monitoring accuracy.

In practice, IOT sensor data streams are susceptible to numerous irregularities produced by unstable connectivity, device malfunctions, configuration errors, or environmental interference. If such irregularities remain unobserved, they can deteriorate the performance of a system, distort analytical insights, and potentially create safety and security risks (Ahmed et al., 2016). Accordingly, anomaly detection (AD) has become an essential requirement for upholding consistent and reliable IoT operations. Effective anomaly detection techniques allow early determination of abnormal behaviour by allowing systems to respond promptly and maintain operational stability.

Despite the growing significance of anomaly detection in IoT environments, several analytics and monitoring platforms depend on a single detection technique that bounds their efficiency across real-world deployments. Rule-based detection approaches support ease of interpretation and transparency, yet they often cannot capture complex behavioural patterns within dynamic environments. Furthermore, machine-learning methods may identify further complex patterns but can sacrifice interpretability or require further annotations that do not exist in practice (Ashraf et al., 2024; Krzysztoń et al., 2024; Shenify et al., 2024). Compared with traditional approaches, machine learning approaches can determine further sophisticated patterns. However, it requires labelled datasets and may lack interpretability (Ashraf et al., 2024; Haider et al., 2025). Furthermore, labelled anomaly data is characteristically scarce in IoT datasets since abnormal events occur intermittently and are infrequently documented. This complicates the evaluation and training of supervised models (Shenify et al., 2024). As a result, relying on a single anomaly detection approach can cause unpredictable performance and reduced robustness when handling noisy IoT data streams (Krzysztoń et al., 2024).

Another limitation of several up-to-date IoT monitoring systems is the absence of integrated analytical environments that are able to integrate several anomaly detection techniques. In several practical applications, anomaly detection outputs are assessed independently, and current platforms do not deliver mechanisms for methodically comparing results from different analytical methods through an integrated workflow (Ahir & Shaikh, 2024). This fragmentation makes it problematic for analysts to comprehend anomalies effectively and to determine the most dependable detection strategies across different contexts.

To overcome the aforementioned issues, this paper produces a software-defined IoT data analytics platform, which consolidates several anomaly detection techniques across a scalable and unified analytics approach. The produced platform makes use of the ELK stack (Elasticsearch, Logstash, and Kibana) by representing it as an essential data ingestion, storage, and visualisation pipeline, where containerization technologies such as Docker allow flexible system configuration and modular dissemination. Through this architecture, multiple anomaly detection approaches are integrated by involving fuzzy logic inference, rule-based checks, unsupervised machine learning using Isolation Forest, and supervised machine learning models. The integration of these heterogeneous methods aims to improve anomaly detection reliability while allowing systematic comparison and visualisation of detection outcomes (Aljawawdeh, Abuezhayeh, et al., 2023).

The motivation behind this study is that real-world IoT environments are integrally noisy, dynamic, and complex, making it difficult for conservative single-method anomaly detection techniques to extract the entire structures of abnormal performances. By merging several analytical methods in a software-defined architecture, the produced platform delivers a simplified environment for scalable deployment, experimentation, and improved interpretability of anomaly detection outcomes.

The key objective of this study is to design and implement a lightweight and scalable IoT analytics platform that can enhance reliable anomaly detection without the need for expensive commercial monitoring systems. Specifically, the aim of this study is based on four significant points, which comprise:

- (1) To investigate important concepts and issues that are associated with anomaly detection in IoT environments,
- (2) To design and implement an effective and reliable scalable software-defined analytics platform,
- (3) To assess several anomaly detection techniques in an integrated workflow, and
- (4) To assess the performance of detection based on the use of interactive visualisation tools and quantitative evaluation metrics.

The produced platform is enhanced based on the use of open-source technologies, alleviating the dissemination of costs and supporting accessibility for practitioners and researchers who need simplified experimentation environments. Additionally, the modular architecture permits the platform to continuously progress as IoT systems develop, while enhancing extensibility, scalability, and reuse through various IoT monitoring scenarios.

This paper highlights the limitations of the current IoT monitoring approaches by providing a software-defined IoT analytics platform that is improved from an extended MSc. thesis study. The proposed platform consolidates open-source analytics and containerization technologies to effectively enhance scalable anomaly detection workflows. Specifically, the system is built around Docker-based deployment and the ELK stack (Elasticsearch, Logstash, and Kibana) for data visualisation, data processing, and data ingestion. Unlike conventional monitoring solutions that are based on a single detection mechanism, the platform integrates several complementary methods, which involve Isolation Forest, rule-based detection, Random Forest classification, and fuzzy-logic inference in an integrated analytical pipeline. The underlying premise is that anomaly detection reliability is developed by comparing the results of heterogeneous methods and

enabling the interpretation of their intersection as a robust confidence signal instead of handling differences as analytical noise.

The key contributions of this study are classified into four significant points. First, it produces a low-cost and modular IoT analytics architecture that incorporates Python-based anomaly detection with ELK visualisation dashboards within a reproducible containerised environment. Second, the study operationalises four complementary anomaly detection paradigms, which comprise supervised classification, fuzzy inference, unsupervised learning, and rule-based logic in a single analytical workflow. Third, it presents an agreement-based analysis technique that provides an interpretable determination of high-confidence anomalies by evaluating intersections among detection approaches. Fourth, the produced platform is assessed based on the use of a real IoT sensor dataset, which delivers comparative insights into the detector behaviour, practical deployment trade-offs, and performance characteristics.

Notwithstanding these contributions, multiple limitations must be acknowledged. The experimental evaluation depends on a historical IoT dataset that does not completely introduce real-time streaming conditions, which occur in operational environments. Moreover, anomaly labels that are applied for evaluation are created according to rule-based heuristics instead of expert verification that can represent potential bias in the ground truth. Furthermore, the study concentrates on a limited set of sensor features and detection techniques, prioritising interpretability and computational efficiency over further resource-intensive deep learning methods. Future research can spread the produced platform by integrating real-time data streaming, advanced deep learning models, and additional sensor modalities together to improve anomaly detection performance and scalability.

2. Related Work and Research Gap

This study introduces the theoretical foundations and applicable literature that develop the design and implementation of the produced platform. It reviews important terms that are relevant to Internet of Things (IoT) architectures, IoT data analytics, and containerization technologies, with specific emphasis on their functionalities in enabling effective and reliable anomaly detection. Additionally, the literature of this study investigates current research efforts and technological solutions for identifying and analysing anomalous patterns in IoT-generated data. The reviewed literature provides the conceptual background, which is essential in comprehending the issues related to IoT data processing and represents the research gaps that are highlighted within this study. Section 2.1 highlights the theoretical framework, which aims to design and assess the structure of the produced platform in this paper. Section 2.2 represents the literature of the previous studies related to this research.

2.1 Theoretical Framework

This section summarises the theoretical terms that reinforce the design and evaluation of the produced platform. It introduces essential values that are in relation to IoT systems, the nature of the data they create, and the role of analytics in identifying meaningful insights from the prospective data. These terms formulate the research methodology and deliver an approach for positioning the produced platform within the wider context of IoT data processing and anomaly detection research.

2.1.1 IoT Background

The Internet of Things (IoT) refers to a network of unified physical objects that are fortified with communication, software, and sensor capabilities that allow them to collect, transmit, and

exchange data across the Internet. These interrelated devices incorporate a broad range of applications, which involve consumer-oriented technologies such as wearable devices and smart home appliances, including large-scale infrastructure systems and industrial equipment (Singh et al., 2020).

The concept of IoT emerged progressively from earlier ideas that may connect everyday objects to computer networks. Early advancements in the late 20th century involved experiments in home automation and demonstrations of network-connected devices, such as vending machines that can provide reports of their operational status remotely (Hassan et al., 2020). Although these early prototypes cannot constitute the current IoT ecosystem, they produced the foundational term of merging physical environments with digital communication networks, which later became central to IoT advancement.

The Internet of Things (IoT) term appeared in the early 2000s, when Kevin Ashton coined the term when working at Procter & Gamble. Ashton proposed a system in which physical objects could be exclusively recognized, connected to networks, and can exchange data with other systems and devices (Hassan et al., 2020). This vision laid the foundation for improving IoT from isolated connected devices to a wider range of data-driven infrastructure that is centred on communication, identification, and sensing technologies.

In recent years, the Internet of Things (IoT) has grown rapidly due to improvements in wireless connectivity, scalable cloud computing platforms, and low-cost sensors, which enhance large-scale data processing and storage. These technological enhancements have disseminated the adoption of IoT within several domains, which involve transportation, agriculture, healthcare, and smart cities, mainly in environments that are in demand of real-time monitoring and incessant data collection (Hassan et al., 2020; Soori et al., 2023). Moreover, developing technologies such as edge computing, artificial intelligence-driven analytics, and evolving cybersecurity frameworks have significantly facilitated the rapid development of IoT dissemination and operational feasibility (Haider et al., 2025). As IoT ecosystems remain improving, effective data processing and analytics are critical for integrating real-time monitoring and informed decision-making through several industries (Umar & Jordanov, 2026). In general, IoT systems can be abstracted as layered architectures that comprise data processing components, sensing devices, application services, and communication networks, which support the provision of end-user functionality (Babun et al., 2021).

2.1.2 Software-Defined Architecture Concept

Software-Defined Networking (SDN) has essentially transmuted network architecture by isolating the control plane from the data plane. This architectural separation handles the centralised network by permitting network behaviour to be implemented in software, and simplifies the modification of control policies without making changes to the underlying forwarding hardware (Biswal & Patel, 2022). Although SDN was originally represented as a networking paradigm, its main design principle is to isolate decision-making processes from execution techniques that have affected the improvement of wider software-defined architectures beyond the conventional network layer (Biswal & Patel, 2022).

In IoT analytics platforms, software-defined (SD) principles are implemented at the application and data-processing layers to enhance system scalability and flexibility. In this method, significant platform components such as visualisation, analytics processing, and data ingestion are designed as independent modules, which together represent a consolidated analytical system. This modular architecture permits separate components to be improved, disseminated, and scaled independently, thereby improving system flexibility while evading tight

coupling with particular infrastructure environments or data sources (Aljawawdeh, Sabri, & Maghrabi, 2023; Muzumdar et al., 2024).

In the proposed platform, software-defined (SD) principles are implemented conceptually and not via an immediate implementation of software-defined networking technologies (e.g., SDN or OpenFlow controllers). The system is structured as a layered architecture where data generation and collection occur at the sensor layer, while decision logic and detection are implemented across Python-based analytical modules, and visualisation and data storage are managed based on the use of the ELK stack. This separation of responsibilities reproduces a software-defined method to IoT analytics, by enabling modular system design and flexible integration of analytical components. Additionally, the architecture supports containerised dissemination and modular analytics pipelines, by simplifying scalable improvement and future system delays.

2.1.3 Docker and Containerization

Containerization has become an important technique for improving reproducible, portable, and scalable software systems. Amongst containerization platforms, Docker is broadly identified since it delivers lightweight virtualisation, which allows applications with their dependencies to be packed into isolated containers and executed reliably through various computing environments (Muzumdar et al., 2024).

Docker containers are different from conventional virtual machines (VMs) in terms of running applications in isolated user spaces, which share the host system's kernel and do not demand a comprehensive guest operating system. This lightweight architecture alleviates resource consumption and startup time by supporting Docker's widespread adoption within cloud-native applications, microservices architectures, and software-defined system design (Muzumdar et al., 2024). In IoT data analytics environments, containerization is predominantly valuable as platforms can frequently incorporate heterogeneous data streams from varied devices and should coordinate several services, which involve analytics processing, data ingestion, visualisation dashboards, and storage and indexing. Docker allows such components to function independently by supporting interoperable services and enhancing system modularity, maintainability, and fault isolation (Muzumdar et al., 2024).

Furthermore, containerization supports the software-defined (SD) architecture that has been broadly implemented and utilised in this paper by enabling system behaviour to be identified via software configuration and excluding hardware-specific setups. This method allows different related services to be simply modified, replaced, or scaled through various environments without reconstructing the host system when developing flexibility dissemination (Muzumdar et al., 2024). Moreover, containerization improves experimental reproducibility by packaging the entire software stack that is applied in experiments, and ensuring reliable execution conditions for anomaly-detection evaluations and analytics workflows (Muzumdar et al., 2024). In this paper, Docker delivers the dissemination foundation for running the ELK stack and related analytical modules as coordinated containerised services.

2.1.4 ELK Stack for IoT Data Analytics

The ELK stack is a broadly used open-source platform for collecting, storing, analysing, and visualising data across several interactive dashboards (Ahir & Shaikh, 2024). It is mainly highly appropriate for IoT environments, where data streams are stereotypically time-sensitive, heterogeneous, and high-volume. In IoT analytics, ELK supports a scalable framework for handling sensor behaviour, identifying anomalies, and enabling data-driven decision-making (Ogbuefi et al., 2021).

Elasticsearch is an essential component of the ELK stack, which supports the delivery of scalable storage and rapid search capabilities for enormous datasets. It can store records as JSON documents, making it highly appropriate for IoT data, where each sensor reading can be shown as an individual event containing attributes such as timestamp, device ID, and measurement values. Furthermore, Elasticsearch enhances significant analytical operations, which involve aggregations and time-based filtering (e.g. averages and counts). Its disseminated architecture allows for horizontal scaling while the number of interconnected devices and data volumes increase (Ahir & Shaikh, 2024).

Logstash represents another significant component of the ELK stack, which functions as the data ingestion and preprocessing layer. It aims to collect data from many different sources, parses and converts it into a reliable structure, and shifts it to Elasticsearch for analysis and storage purposes. In IoT environments, where sensor data is likely to arrive in heterogeneous formats, Logstash pipelines highlight data quality problems by standardising timestamps, handling missing values, normalising field names, and enriching records with extra available metadata (e.g. sensor location or type).

Kibana represents the visualisation layer of the ELK stack, which allows interactive investigation and management of stored data. It provides dashboards, visualisations, and query tools that enable users to analyse time-series data, monitor device activity, and detect system behaviour in a closer real-time environment. In fact, such capabilities make Kibana mainly beneficial for operational management and research analysis across several IoT environments (Ahir & Shaikh, 2024; Shaheen et al., 2017).

2.2 Literature Review

Current research on IoT anomaly detection (AD) is based on two key directions: deployment-oriented monitoring platforms that can manage to handle high-velocity telemetry data, and algorithmic approaches that enhance the performance of detection based on different encountered issues such as limited labelled data and class imbalance. This study sits at the intersection of such directions by improving a software-defined and containerised ELK-based analytics platform, which consolidates several anomaly detection paradigms. Additionally, the platform implements consensus (intersection) analysis to raise confidence in explored anomalies by providing comparative results among different complementary detection approaches.

From a systems perspective, the Elastic ecosystem (ELK) delivers a practical basis for operational analytics by providing production-grade data ingestion, visualisation capabilities, and storage. Previous studies have investigated and assessed ELK's machine-learning features by utilising real-world monitoring data and finding that multivariate anomaly detection may efficiently detect anomalies more than single-metric approaches can detect. This can fail to determine abnormal behaviour when individual signals are still in the normal required ranges (Ahir & Shaikh, 2024). These outcomes represent the significance of designing correlated signals within anomaly detection workflows. The current study implements an ELK-based pipeline on IoT sensor datasets and incorporates several heterogeneous detection approaches such as isolation forest, fuzzy inference, supervised learning, and rule-based logic in a unified analytics framework instead of only depending on particular built-in Elastic detection tools.

Current research frames of IoT anomaly detection (AD) represent a security or intrusion detection (IDS) issue, which depends on deep learning pipelines. For instance, Ayad et al. (2025) produced a real-time IoT IDS by utilising a one-class learning framework, which integrates an asymmetric stacked autoencoder for dimensionality alleviation with a single-class deep neural

network to highlight different issues such as resource-constrained devices, class imbalance, and scarce labels (Ayad et al., 2025). Knowing that these methods are efficient and reliable for high-dimensional network traffic analysis, they prioritise ML-driven detection and deliver limited support for interpretability and integrated monitoring workflows. In comparison with prior studies, the produced platform in this paper merges machine-learning models, fuzzy inference, and rule-based reasoning with a software-defined analytics environment, accenting interpretability and system-level monitoring and excluding a stand-alone detection model.

Hybrid, “multi-technique” security detection became more common when cutting false alarms and enhancing response time. Ashraf et al. (2024) described a hybrid IoT-security approach, which merges ML and complementary techniques to increase resilience against emerging threats while alleviating false positives and latency, and arguing that no single approach is still tolerable under shifting noisy traffic and attack patterns. This paper shares the same core premise, robustness through combination, but is different in two essential ways: first, the hybrid design represents a platform (software-defined, containerised ELK) instead of a detector-only pipeline; second, robustness is operationalised on intersection analysis based on detector families, which provides a higher-confidence anomaly set. A platform-level mechanism is infrequently formed in security-oriented hybrid IDS studies (Ashraf et al., 2024).

One operational strand takes into account disseminated architectures that scale anomaly detection (AD) in heterogeneous devices and domains within mixed real-world distributions. Muntean (2024) produced a real-time approach for IoT and smart-city environments by utilising a distributed multi-agent design for enhancing scalability and responsiveness. Since this approach highlights a real deployment issue, which is coordinating signals from varied IoT sources, the platform within this study implements a further straightforward strategy. It prioritises ease of reproducibility and distribution via a Dockerized ELK stack, while combining several analytical methods and involving unsupervised learning, fuzzy inference, rule-based checks, and supervised models into a unified analytical layer. This approach is mainly appropriate for organisations that are in demand of rapid adoption without redesigning their systems into a completely automated distributed architecture.

Fuzzy and clustering-oriented methods remain broadly investigated since they can manage and process uncertainty, inconsistent and incomplete labels. Shenify et al. (2024) presented fuzzy subspace clustering for IoT AD by implementing nano-topology rough set theory to determine a CORE subspace and then operate fuzzy clustering within that alleviated feature space. Furthermore, Shenify et al. (2024) highlighted the issue when high-dimensional IoT data can deteriorate conventional clustering performance. Their contribution is primarily an unsupervised framework built around subspace selection and fuzzy clustering. On the contrary, the platform in this paper implements fuzzy-logic inference as an interpretable, rule-driven detector that is audited and compared with supervised ML and IF in the same operational workflow. Alabbadi and Bajaber produced an understandable IoT intrusion detection method, which can consolidate deep learning with interpretability techniques, by attaining effective and reliable performance on current intrusion datasets (Alabbadi & Bajaber, 2025). Consequently, the study implements a system-level interpretability strategy by integrating machine-learning outputs with fuzzy inference and rule-based checks, while relying on ELK dashboards as a functional framework for examining and assessing explored anomalies.

The current research has identified detector fusion and robustness to labelling noise for anomaly detection in inadequately labelled IoT environments. For instance, Zahoor et al. developed a fusion method that merges the Isolation Forest and one-class SVM (CSAD) method with LIME-based explanations to enhance the robustness of anomaly detection (Zahoor et al.,

2025). In the same way, this research implements a multi-detector strategy based on the combination of Isolation Forest, fuzzy inference, supervised learning, and rule-based checks through an incorporated platform, and implements intersection-based analysis to detect high-confidence anomalies in settings where labelled data are limited.

Several previous studies support algorithmic performance or security-specific applications of anomaly detection (Aly & Behiry, 2025). However, they provide inadequate focus on how heterogeneous detection approaches are integrated, compared, and interpreted through an incorporated analytics platform. The nonappearance of combined cross-method comparison, visualisation, and operational support bounds the practical dissemination of anomaly detection systems in real-world IoT environments. Subsequently, an essential research gap is to improve IoT analytical platforms which integrates rule-based, machine-learning, and clustering-based detection methods in a consistent framework to enhance decision-making and effective analysis.

Although these studies are valuable, several limitations are still encountered. First, many security-oriented papers emphasise network intrusion traffic rather than low-dimensional environmental sensor data. Second, comparative studies often stop at the algorithm level and do not provide a unified operational platform for ingestion, indexing, visualisation, and investigation. Third, explainability is commonly discussed within a single detector family rather than across a portfolio of complementary methods. Finally, only a limited number of studies explicitly use cross-method agreement as an operational confidence mechanism, even though recent work on detector fusion suggests that combination strategies can improve robustness in weak-label settings (Karakaya, 2025; Zahoor et al., 2025).

The research gap addressed here is not only algorithmic, but operational and architectural. This implies that there is a high demand for IoT analytical environments that combine heterogeneous detectors, by exposing their results through a similar searchable dashboard layer, and producing disagreements analytically that are beneficial rather than being merely incidental. The platform in this research responds to such a demand by coupling a containerised ELK dissemination with several anomaly detectors, and involving an agreement-based interpretation layer.

3. Platform Design and System Architecture

In this section, the produced platform was designed through four practical requirements, which comprise: modularity, reproducibility, interpretability, and accessibility. Rather than integrating the analytical workflow into a single host-specific installation, the system isolates data ingestion, storage, visualisation, and analytics into loosely coupled services, which are amended or replaced in an independent manner.

The disseminated environment uses Ubuntu Linux, Docker, and Docker Compose to allow Elasticsearch, Logstash, and Kibana to function as integrated services. Python 3.12 is applied for feature engineering, preprocessing, result aggregation, and detector execution. This design reproduces software-defined thinking at the analytics layer: system behaviour is represented across configuration, scripts, and containers instead of being represented on fixed hardware arrangements (Muzumdar et al., 2024). The primary services are organised as Docker containers that makes the environment reliable and averts dependency problems. Docker Compose (DC) is afterwards utilised to combine these services together and handle the links among them. For the data pipeline, the platform is based on the ELK stack. Logstash receives the IoT readings and organises them. Elasticsearch stores the data and delivers rapid search, and Kibana delivers the dashboards for identifying the data and reviewing detection results. Additional Python modules

were improved to enhance the analysis by integrating machine-learning, fuzzy-logic modelling, and rule-based detection approaches.

The platform in Figure 1 functions as follows. Prior to analysis, the sensor data are preprocessed in Python and written as organised outputs, which contain derived anomaly attributes and raw measurements. Logstash controls the output directory, normalises timestamps and field types, and moves the necessary documents to Elasticsearch. After that, Kibana delivers dashboards for search, aggregation, filtering, and visual review. The outcome represents a combined environment where anomaly decisions are inspected at record level, over time, or by sensor node without exiting the platform.

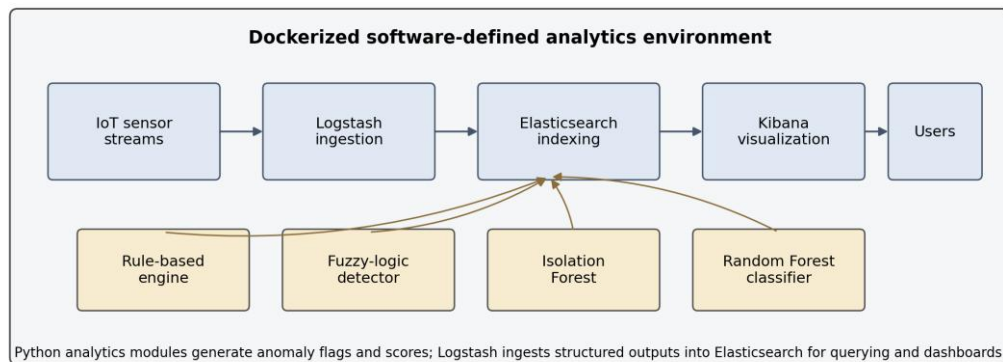


Figure 1: Software-defined platform architecture.

The architecture is taken into consideration as dissemination and interpretation are inherently intertwined in an implemented anomaly detection. If detector outputs are still separated in separate files or scripts, systematic comparison becomes operationally intensive. In contrast, indexing the entire detector results within a similar searchable store allows it to evaluate individual records, detector overlaps, and temporal patterns through an integrated interface.

The architecture is deliberately designed to be cost-effective. The entire set of essential components is open source and can be reproduced without reliance on proprietary monitoring platforms. That makes the platform beneficial not only for enterprise experimentation, but also for students and researchers who are in demand of a rigorous yet accessible analytical environment. Figure 2 illustrates the flowchart setup diagram of the produced platform, which underpins the architecture illustrated in Figure 1 above.

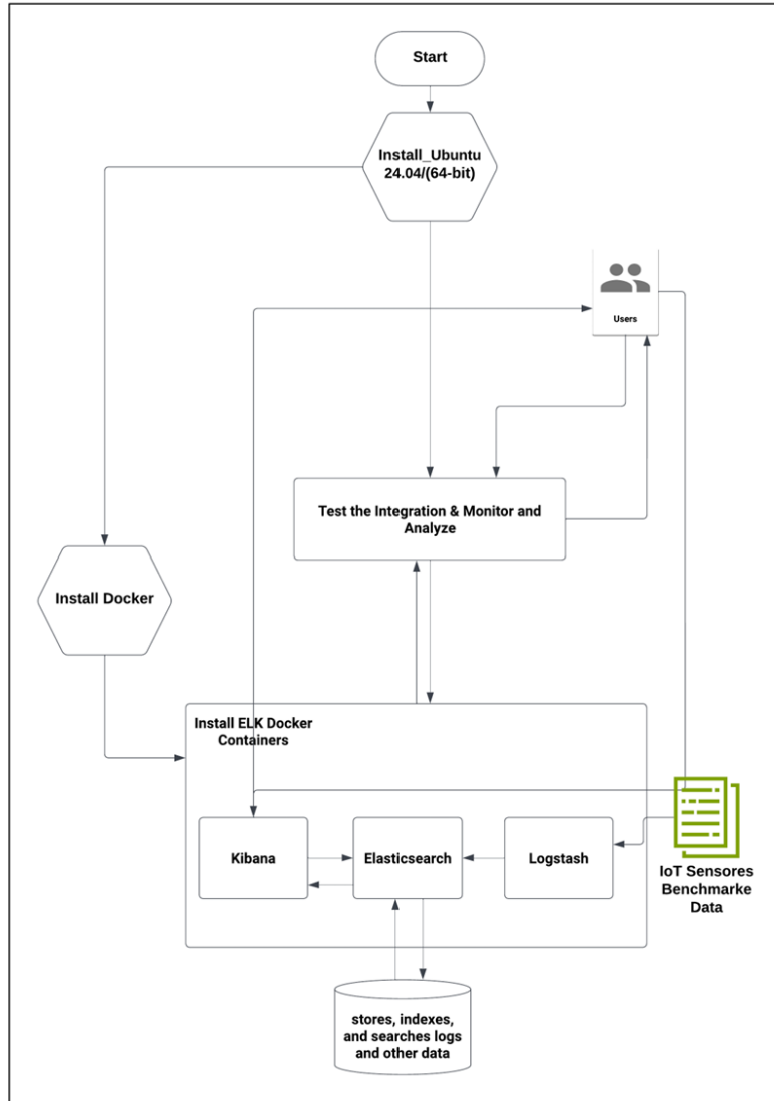


Figure 2: Flowchart setup diagram of the produced platform.

In this paper, the implementation of the IoT data analytics platform was carried out in a Linux-based environment, with Ubuntu as the main operating system. Ubuntu was adopted for several reasons. This refers to its robust community support, compatibility with improved tools, stability, and native integration with container technologies such as Docker. This setting, combined with the selected tools, provided the flexibility and control that are required to create an SD platform that can monitor and process complex data processing and different analytical functions.

The system configuration started with installing Ubuntu 22.04.3 LTS, followed by updating system packages and installing significant improved dependencies. The researcher used Python 3 as the primary programming language since it has a massive ecosystem related to data processing and machine-learning libraries, and generated a dedicated virtual environment to ensure the isolated execution of the entire analytical scripts within this system described in this paper.

In the following step, Docker and Docker Compose (DC) were installed and configured at the same time to integrate scalable SD deployment and modularity. Docker was identified as it permits operating every component related to the ELK stack in its own separate container, enhancing reproducibility and making configuration. Conversely, DC is exploited here to establish various containers, which are generated in this system, manage several variables within this environment, and define persistent storage volumes for Elasticsearch. Therefore, enabling the execution of varied operations across different levels and the cooperation among several components.

Docker and DC are utilised and installed, and are configured on the Ubuntu environment to allow the dissemination of the SD that is related to the IoT analytical platform. Docker functions as a lightweight containerization tool, leaving them as separate services within the same system. Furthermore, DC was installed to support the orchestration of multi-container workflows. This methodology is adopted to facilitate greater repeatability and further effective fault isolation while making it not complex to handle the system by generating a highly appropriate environment for SD architectures.

A Python 3.12 virtual environment was generated in the project directory to accomplish data preprocessing, machine learning, and anomaly detection experiments, while also ensuring reproducibility and environment isolation within it.

The proposed platform combines Python-based analytical platforms with the ELK stack to allow end-to-end IoT data analytics. Such a combination enables the results created by Python scripts to be indexed in Elasticsearch automatically and detected by communicating with Kibana dashboards. The analytical workflow starts with Python-based processing, which achieves data preprocessing, result aggregation, anomaly detection, and feature extraction across the IoT dataset. The scripts yield organised CSV outputs that contain the original sensor data with derived anomaly detection indicators, involving Isolation Forest predictions, rule-based flags, supervised learning labels, and fuzzy logic scores.

In the proposed workflow, Logstash joins the Python analytics layer to Elasticsearch based on a dedicated ingestion pipeline. This pipeline controls the directory, which contains the created CSV outputs, standardises timestamps, parses the records, and maps attributes to further appropriate Elasticsearch data types. The processed data, which involves derived anomaly detection features and sensor measurements, is afterwards indexed in time-based Elasticsearch indices.

This process assures that the two types of data (derived analytical features and raw sensor measurements) are reliably linked and indexed. When Logstash completes processing the data, it is delivered to Elasticsearch and stored in time-based indices. Every record is indexed with the system so that it represents a structured document, and every document involves the following: sensor identifiers, temporal information, environmental measurements, and anomaly indicators that are created by the entire detection approach. Producing such documents causes to provide an integrated data representation and effective querying with robust support for filtering and aggregation processes throughout the system.

At the next stage of this system, Kibana handles the platform's exploration layer and visualisation tasks. After this stage, the obtained patterns are configured to determine and expose raw sensor fields, including the key target (anomaly-related attributes). Kibana dashboards were needed to present these outcomes so that the researcher can generate them for data visualisation purposes. This dashboard represents the following: anomaly timelines, sensor-level anomaly distributions, and comparative results.

In this paper, incorporating Python analytics with the ELK stack allows the platform to achieve a completely automated and reproducible analytical pipeline. Analytical outputs flow straightforwardly from offline Python processing through to a searchable, scalable, and visual analytics environment. This incorporation serves as the basis for the experimental assessment and analysis of the results presented in the following sections.

4. Materials and Methods

4.1 Dataset and preprocessing

This paper surveys an experimental and analytical method, which involves two parts: system implementation and data-driven evaluation. The following subsections present the central construct with detailed explanations.

4.1.1 Platform-Oriented Approach

As illustrated in Figure 3, the produced platform structures a software-defined, containerised architecture. Docker and Docker Compose structure the ELK stack as isolated services, guaranteeing efficient indexing, scalable data ingestion, and flexible visualisation. This layered design isolates the analytical pipeline from the fundamental infrastructure by allowing reliable execution and underpinning the collection, storage, and visualisation of anomaly detection outcomes, which are obtained through several analytical methods.

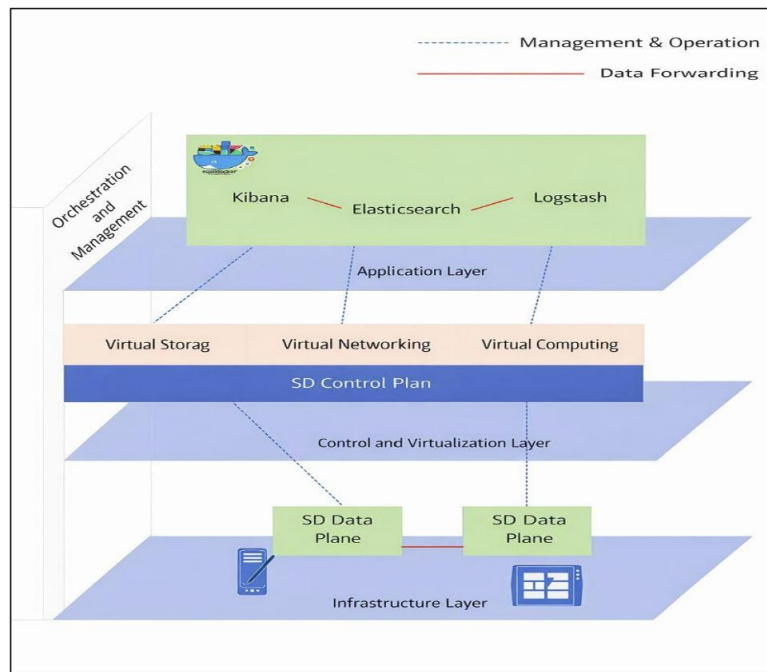


Figure 3: Software-defined and containerised architecture.

4.1.2 Data-Driven Analytical Approach (Anomaly Detection Methods)

To highlight the limitations of a single detection paradigm, the produced platform enhances a multi-method anomaly-detection strategy. Every method aims to support a separate facet of anomalous behaviour within the IoT sensor data, and deliver cross-validation of reported anomalies, interpretability, and resilience.

4.1.3 Knowledge Discovery Process

The entire analytical workflow follows the Knowledge Discovery in Databases (KDD) process, as exemplified in the second conceptual diagram (see Figure 4). The process is comprised of the following stages: data selection and sampling, data preprocessing and cleaning, feature transformation, AD and classification, and assessment using statistical performance metrics. In particular, every stage was obviously programmed using the Python programming language and consolidated into the ELK-based analytical pipeline.

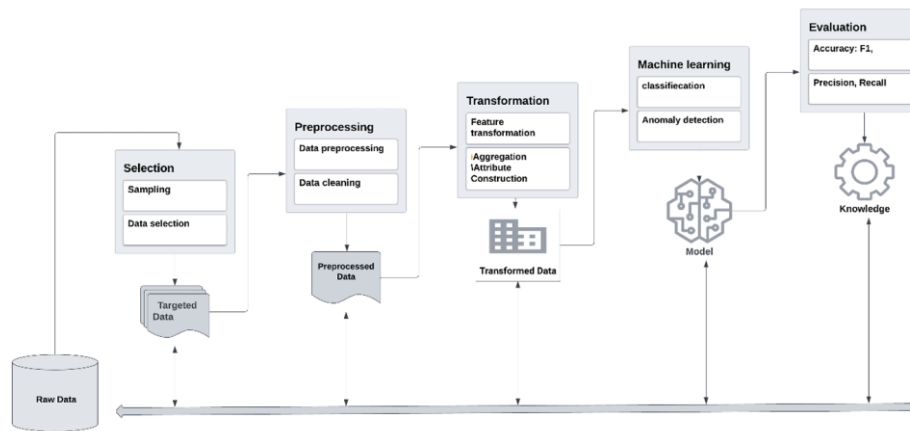


Figure 4: Knowledge Discovery in Databases (KDD) process.

4.2 Data Collection and Sampling

The reason behind the massive size of the original Intel Lab dataset, containing millions of time-series sensor readings, the researchers selected a random subset of the data to enable efficient experimentation and analysis for this study. Experiments were carried out on the Intel Lab sensor dataset, where a widely used benchmark that consists of environmental readings is collected from multiple wireless sensor nodes in an indoor laboratory setting. This procedure was conducted based on the use of a dedicated Python script, `read and clean.py` that reads the original dataset (`intel.csv`), which in turn implements initial data-cleaning operations.

The raw dataset contains millions of time-series records; therefore, a random subset of 10,000 observations was extracted for tractable experimentation. After cleaning and removal of incomplete rows, 9,606 records remained for analysis.

Since the original dataset was large, a subset was explored, and random sampling was utilised to preserve different statistical features and variability related to the original data. In particular, random sampling is applied to uphold diversity through temporal behaviour and sensor nodes, and to avert overload within the extensive system. The subset that is explored from a random sample was stored in an isolated file, namely, `intel sample.csv`, and served as the primary dataset for the entire following analytical stages.

The retained attributes involve date, time, epoch, mote identifier, temperature, humidity, light intensity, and voltage (see Table 1. This schema captures sensor-level variability and temporal context, whereas the retained measurements expose the platform to realistic problems (e.g. cross-feature correlation, variable ranges across nodes, missing readings, and natural noise).

Table 1: Intel Lab dataset attributes used in the experiments.

Attribute	Description
date	Calendar date of the reading (YYYY-MM-DD)
time	Timestamp with millisecond precision
epoch	Monotonically increasing sequence number per sensor
moteid	Unique identifier of the sensor node
temperature	Temperature in degrees Celsius
humidity	Temperature-corrected relative humidity
light	Light intensity in lux
voltage	Battery voltage in volts

The dataset reflects realistic IoT conditions, which involve: missing readings for certain epochs, variations across different sensor nodes, natural noise and measurement uncertainty, and correlation between variables. In fact, such characteristics made the dataset further appropriate for assessing AD mechanisms within the IoT environments.

4.3 Feature engineering and proxy labels

Since expert-annotated anomaly labels were not obtainable, the study created explainable proxy labels by utilising a rule engine. Eight rule-based conditions were applied from descriptive statistics and domain-informed plausibility ranges. Binary indicators identified abnormal temperature, missing-value, humidity, voltage, and light conditions. A record was labelled anomalous when at least one predefined condition was violated.

This choice does not imply that the rule-based labels are completely efficient. Nonetheless, they deliver an auditable and reliable reference when comparing detectors within a realistic weak-label setting. Such proxy labelling is a commonly observed trade-off in anomaly-detection research when manual annotation is expensive or infrequent (Ahmed et al., 2016; Pang et al., 2021).

4.4 Detection Modules

Four detectors were applied within the same workflow (see Figure 5). The rule-based module served both as an interpretable detector and as the source of proxy labels. The fuzzy-logic detector modelled gradual transitions between normal and abnormal states using three linguistic input variables with three membership functions, each creating a complete rule base of 27 inference rules. An incessant anomaly score was afterwards defuzzied, and a threshold was applied for binary classification.

Isolation Forest was utilised as the unsupervised component. The model was trained immediately on the benchmark data and was predicted to indicate statistically rare, high-isolation observations without depending on labels. The supervised module applied random forest classification and learned from the rule-based proxy labels. The cleaned dataset was fragmented into 70% training data (6,724 records) and 30% testing data (2,882 records).

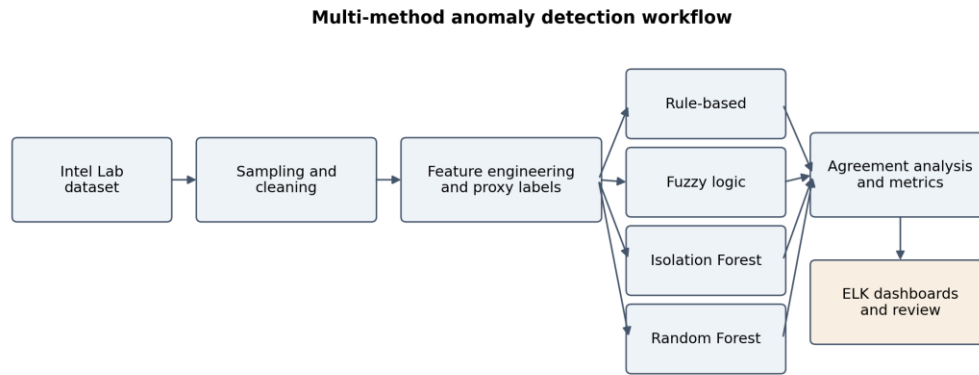


Figure 5: Multi-method anomaly-detection workflow.

4.5 Integration and Evaluation

The outputs of all detectors were collected into a single consolidated dataset consisting of four binary flags, which comprise: rule based flag, fuzzy flag, iforest flag, and supervised flag. This combined representation was indexed into Elasticsearch and utilised for Kibana-based review and quantitative evaluation.

Performance was assessed and seen to be relative to the rule-based proxy labels when utilising accuracy, precision, recall, and F1-score. Additionally, agreement analysis treated every detector as a set of anomalous records and computed higher-order and pairwise intersections. Records that are explored via many different approaches were interpreted as higher-confidence anomalies, while detector-specific flags were preserved as context-dependent cases or potentially borderline.

5. Results

5.1 Statistical Methods

The mathematical formulation related to the statistical metrics is applied to assess the AD methods, which are implemented in this study. The entire detection outputs were treated as binary classification results where each observation was labelled as anomalous or normal. For assessment purposes, AD outcomes were mapped into a binary confusion matrix and are defined as follows:

- True Positive (TP): Anomalous records correctly identified as anomalies
- True Negative (TN): Normal records correctly identified as normal
- False Positive (FP): Normal records incorrectly classified as anomalies
- False Negative (FN): Anomalous records incorrectly classified as normal.

These quantities form the basis for all subsequent evaluation metrics. In terms of accuracy, the proportion of correctly classified observations is measured by the accuracy over the total number of observations, where it is defined as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

Accuracy delivers a complete indication of classification correctness. Nevertheless, in AD tasks with imbalanced data distributions, it may not completely identify the detection effectiveness.

From the precision side, the reliability of anomaly predictions is assessed by measuring the proportion of explored anomalies which are truly anomalous. For this purpose, precision is defined as follows:

$$Precision = TP / (TP + FP) \quad (2)$$

A high-precision value denotes a low false-alarm rate that is predominantly significant in IoT monitoring systems when averting unrequired alerts. As for the recall, it denotes the actual positive rate or sensitivity, where a detection method is measured to determine actual anomalies efficiently as follows:

$$Recall = TP / (TP + FN) \quad (3)$$

High recall is critical in safety-critical IoT environments, where missing anomalous events can cause deteriorated performance or system failures. Furthermore, F1-score delivers a balanced assessment metric by integrating precision and recall within a single harmonic mean as follows:

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (4)$$

Such a metric is mainly appropriate for AD issues, which are characterised by class imbalance since it penalises extreme trade-offs between recall and precision.

In addition to conventional classification metrics, an overlap-based statistical analysis was carried out to assess agreement between different detection approaches. Let: A denotes the set of anomalies detected by the Rb method, B denotes the set detected by FL, C denotes the set detected by IF, and D denotes the set detected by the (SM) Pairwise and multi-set intersections were computed using set operations:

$$|A \cap B|, |A \cap C|, |A \cap D|, |B \cap C|, |B \cap D|, |C \cap D|, |A \cap B \cap C|.$$

Records belonging to several intersections were considered higher-confidence anomalies, whereas records explored by a single approach were treated as borderline cases or uncertain.

5.2 Overall anomaly statistics

All four detectors were applied to the 9,606-record dataset. The rule-based, fuzzy-logic, and supervised approaches obtained the same anomaly shares, while the isolation forest was evidently more conservative. This implies that the detectors can respond to various notions of abnormality: threshold violations, gradual uncertain deviations, statistically rare outliers, and learned decision boundaries (see Table 2 and Figure 6).

Table 2: Detected anomaly counts and rates across methods.

Method	Detected anomalies	Rate (%)
Rule-based	2,601	27.08
Fuzzy logic	2,265	23.58
Isolation Forest	481	5.01
Supervised	2,368	24.65

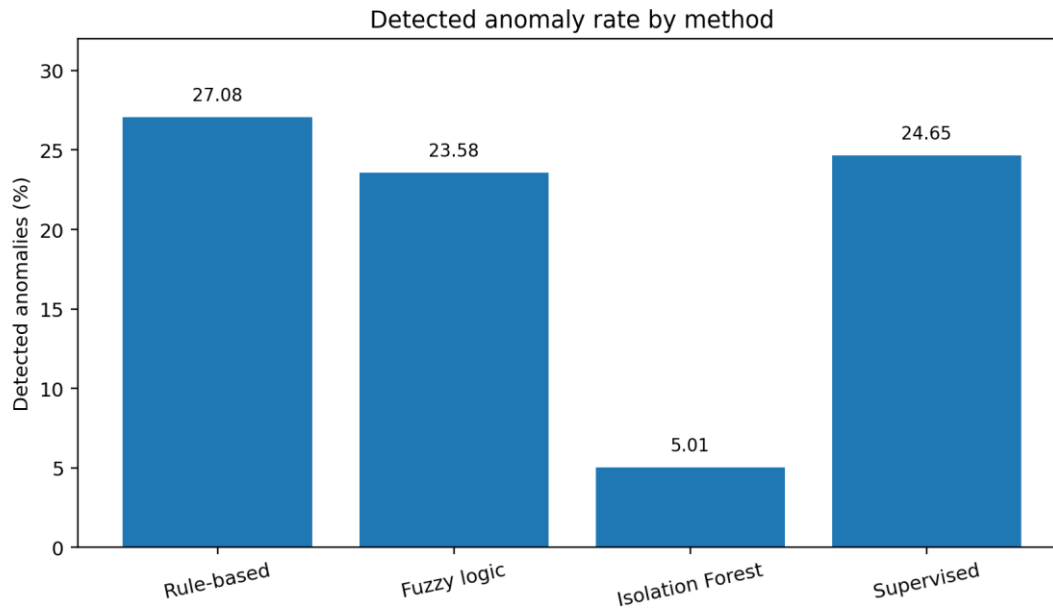


Figure 6: Detected anomaly rate by method.

It can be inferred from the obtained results that the Rule-based, Fuzzy logic, and supervised approaches explored a comparable proportion of anomalies. Meanwhile, the Isolation Forest approach was considerably more conservative, identifying a much smaller subset of anomalous records.

5.3 Agreement Analysis

Agreement analysis strengthens the multi-method detection's value (see Table 3). The strongest pairwise overlap occurred between the rule-based and supervised detectors (2,324 shared anomalies) since the supervised model was trained on rule-derived labels. Moreover, the Fuzzy logic approach overlapped considerably with the results of the supervised and rule-based approaches, assuming that it identified the same, but softer notion of abnormality.

Most significantly, 426 records were flagged via all four detectors. These records form the most conservative high-confidence anomaly set through this study. Furthermore, 2,425 records were flagged by at least two approaches and 1,800 by at least three approaches, providing a graded way to differentiate robust operational alerts from records which deserve contextual review instead of performing a prompt action.

Table 3: Cross-method agreement counts.

Agreement set	Count
Rule-based $\cap$ Fuzzy logic	1,842
Rule-based $\cap$ Isolation Forest	434
Rule-based $\cap$ Supervised	2,324
Fuzzy logic $\cap$ Isolation Forest	476
Fuzzy logic $\cap$ Supervised	1,799
Isolation Forest $\cap$ Supervised	428
Rule-based $\cap$ Fuzzy logic $\cap$ Isolation Forest	430
All four methods	426
At least two methods	2,425
At least three methods	1,800

5.4 Statistical Performance

When evaluated compared to the rule-based proxy labels, the supervised Random Forest attained the most effective overall performance (see Table 4 and Figure 7). Its 96.66% accuracy and 93.54% F1-score point out that the learned model reproduced rule-derived limits with high consistency, while considerably alleviating classification errors in the held-out test set.

Fuzzy logic formed a balanced profile with moderate recall and precision that is reliable with its design target of modelling uncertain transitions and not only crisp threshold violations. The Isolation Forest approach demonstrated dissimilar behaviour where its precision was high (90.28%). However, its recall was low (16.69%). From a practical perspective, this renders it highly appropriate for determining rare, high-confidence outliers instead of broad anomaly coverage.

Table 4: Performance relative to rule-based proxy labels.

Method	Accuracy	Precision	Recall	F1-score
Fuzzy logic	0.8770	0.8132	0.7082	0.7571
Isolation Forest	0.7695	0.9028	0.1669	0.2816
Supervised	0.9666	0.9814	0.8935	0.9354

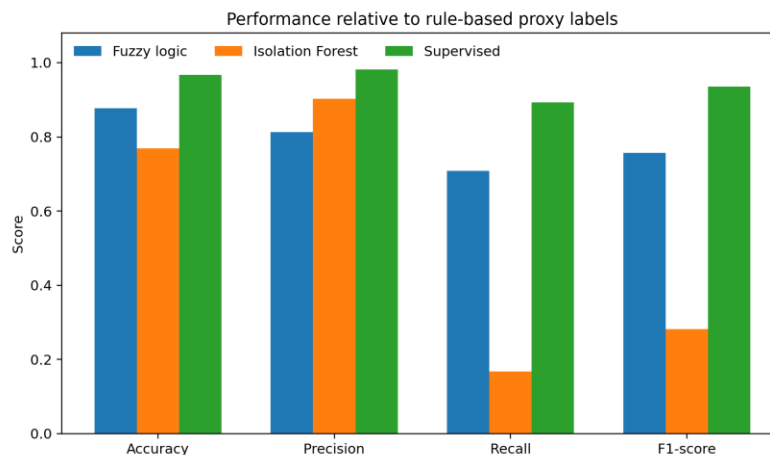


Figure 7: Comparative performance profile of the assessed approaches.

5.5 Confusion-matrix view and operational patterns

The confusion matrices facilitate greater clarity regarding the trade-offs within the model performance. Relative to the proxy labels, fuzzy logic produced 6,582 true negatives, 423 false positives, 759 false negatives, and 1,842 true positives. The Isolation Forest approach obtained 6,958 true negatives and only 47 false positives. Nonetheless, it missed several anomalies (2,167 false negatives) which illustrates its low recall. The Supervised approach combined 6,961 true negatives with 2,324 true positives and only 44 false positives, resulting in the most preferable overall balance.

Kibana dashboards delivered further operational perspective. Repeated anomaly concentration around sensor nodes 24 and 34 throughout multiple detectors suggests that the platform determines node-specific behaviour patterns and not only collects anomaly counts. In fact, this is valuable for practical investigation since it supports operators in prioritising sites, time windows, or devices that warrant a closer inspection.

6. Discussion

It can be inferred from the obtained results that anomaly behaviour in IoT data is multi-shaped instead of singular. Every detector responds to a different operational definition of abnormality. The rule-based module was seen to be the most explicit and auditable: every alarm must be traced back to a threshold or missing-value condition. That transparency is useful in applied environments. However, it also clarifies why the rule-based detector is relatively sensitive to borderline cases.

The Fuzzy Logic approach occupied a meaningful middle ground since it models gradual state transitions instead of sharp cutoffs. It delivered a more balanced precision-recall profile and appears more efficient for noisy environmental data where strict boundaries are not frequently realistic. It overlaps with both rule-based and supervised outputs where it serves as an interpretability-preserving complement to harder threshold logic.

Isolation Forest supports a deliberately different value proposition. Its small anomaly set and high precision imply that it is best used as a high-confidence outlier detector, particularly when the objective is to surface rare and structurally unusual records. Nonetheless, it would lose several anomalies in this dataset. This finding is consistent with the broader lesson that unsupervised outlier detectors are extremely beneficial; however, they are insufficient as sole decision engines when recall matters.

The supervised model achieved the most effective numerical performance, but its strength should be interpreted cautiously. Because it learned from rule-based labels, it mostly reproduced the logic embedded in those labels while smoothing inconsistencies and enhancing generalisation on held-out data. Its excellent performance therefore validates the consistency of the proxy-labelled learning setup instead of demonstrating independence from the rule-based detector.

The most practically significant outcome of the research is the agreement layer. The 426 records flagged by all four methods are robust candidates for triage, auditing, or downstream alerting. Meantime, records flagged by only a single method must not be dismissed. They can reveal context-dependent or method-specific signals, which deserve targeted review. This graded interpretation makes the platform more useful compared to any single-method deployment, as it converts disagreement into actionable analytical structure.

7. Limitations and Future Research

Many limitations must be considered before generalising the obtained results. First, the assessment used a historical benchmark dataset instead of a live streaming dissemination. The platform architecture can support real-time ingestion. On the other hand, streaming behaviour, latency, and drift adaptation were not examined directly in the current study.

Second, the rule-based labels used as proxy ground truth enhance interpretability, but also represent label dependence. The outstanding performance of the supervised model must be read in light of the assumptions embedded in the rule engine. External validation with expert annotations would provide more robust evidence of generalisation.

Third, the experimental feature space was limited to a small set of environmental measurements. Broader multimodal sensor sets and diverse IoT contexts may reveal further strengths or weaknesses in every detector. Consequently, future research must extend the architecture to richer contextual features, streaming data, expert-verified, adaptive rule learning, and semi-supervised labelling strategies.

8. Conclusion

This paper introduced a journal-style formulation of a software-defined IoT analytics platform that consolidates many different anomaly-detection approaches within a Dockerized ELK environment. The paper shows that the rule-based logic, fuzzy inference, Isolation Forest, and Random Forest classification approaches are orchestrated within a single reproducible workflow for dashboarding, ingestion, comparison, and indexing.

Across 9,606 cleaned records from the Intel Lab benchmark, the supervised model performed the highest overall accuracy and F1-score, the Fuzzy Logic approach provided a balanced response to uncertain sensor behaviour, and the Isolation Forest approach isolated a small but high-confidence set of outliers. Most significantly, the agreement layer caused a practical confidence hierarchy headed by 426 anomalies that are explored by all four approaches.

Based on the obtained results, it can be suggested that the most robust contribution of the platform is not a single best detector, but the combination of complementary detectors within an operational environment that makes their outputs interpretable, comparable, and searchable. That combination is, particularly, valuable in IoT settings where labels are infrequent, data quality changes over time, and practitioners require transparent reasoning and statistical performance.

Declarations

Data availability

The experiments used the publicly available Intel Lab dataset. Access details were reported in the source as Intel Berkeley Research Lab (2004).

Conflict of interest

The authors declare no conflict of interest.

References

- Ahir, D. D., & Shaikh, N. F. (2024). Evaluation of Elasticsearch ecosystem including machine learning capabilities. *International Journal of Safety and Security Engineering*, 14(4), 1293–1303.
<https://doi.org/10.18280/ijssse.140426>

- Ahmed, M., Naser Mahmood, A., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19–31. <https://doi.org/10.1016/j.jnca.2015.11.016>
- Alabbadi, A., & Bajaber, F. (2025). An intrusion detection system over the IoT data streams using explainable artificial intelligence (XAI). *Sensors*, 25(3), Article 847. <https://doi.org/10.3390/s25030847>
- Aljawawdeh, H., Abuezhayeh, S., Qaddoumi, E., & Maghrabi, L. (2023). Simplifying microservices migration with advanced genetic algorithms. In A. Hannoon & A. Mahmood (Eds.), *Artificial intelligence, internet of things, and society 5.0* (pp. 441–452). Springer. https://doi.org/10.1007/978-3-031-43300-9_36
- Aljawawdeh, H., Sabri, M., & Maghrabi, L. (2023). Toward serverless and microservices architecture: Literature, methods, and best practices. In A. Hannoon & A. Mahmood (Eds.), *Artificial intelligence, internet of things, and society 5.0* (pp. 573–584). Springer. https://doi.org/10.1007/978-3-031-43300-9_47
- Aly, M., & Behiry, M. H. (2025). Enhancing anomaly detection in IoT-driven factories using Logistic Boosting, Random Forest, and SVM: A comparative machine learning approach. *Scientific Reports*, 15, Article 23694. <https://doi.org/10.1038/s41598-025-08436-x>
- Ashraf, M. W. A., Singh, A. R., Pandian, A., Rathore, R. S., Bajaj, M., & Zaitsev, I. (2024). A hybrid approach using support vector machine rule-based system: Detecting cyber threats in Internet of Things. *Scientific Reports*, 14, Article 27058. <https://doi.org/10.1038/s41598-024-78976-1>
- Ayad, A. G., Sakr, N. A., & Hikil, N. A. (2025). Fog-empowered anomaly detection in IoT networks using one-class asymmetric stacked autoencoder. *Cluster Computing*, 28, Article 550. <https://doi.org/10.1007/s10586-025-05234-y>
- Babun, L., Denney, K., Celik, Z. B., McDaniel, P., & Uluagac, A. S. (2021). A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks*, 192, Article 108040. <https://doi.org/10.1016/j.comnet.2021.108040>
- Biswal, S. P., & Patel, S. (2022). Introduction to software defined networking. In A. Nayyar, B. Singla, & P. Nagrath (Eds.), *Software defined networks: Architecture and applications* (pp. 1–27). Wiley. <https://doi.org/10.1002/9781119857921.ch1>
- Haider, Z. A., Zeb, A., Rahman, T., Singh, S. K., Akram, R., Arishi, A., & Ullah, I. (2025). A survey on anomaly detection in IoT: Techniques, challenges, and opportunities with the integration of 6G. *Computer Networks*, 270, Article 111484. <https://doi.org/10.1016/j.comnet.2025.111484>
- Hassan, R., Qamar, F., Hasan, M. K., Aman, A. H. M., & Ahmed, A. S. (2020). Internet of Things and its applications: A comprehensive survey. *Symmetry*, 12(10), Article 1674. <https://doi.org/10.3390/sym12101674>
- Intel Berkeley Research Lab. (2004). *Intel Berkeley Research Lab sensor data repository*. Retrieved August 17, 2026, from <http://db.csail.mit.edu/labdata/labdata.html>
- Karakaya, A. (2025). A hybrid approach for IoT security: Combining ensemble learning with fuzzy logic. *Sensors*, 25(18), Article 5668. <https://doi.org/10.3390/s25185668>
- Krzysztoń, E., Rojek, I., & Mikołajewski, D. (2024). A comparative analysis of anomaly detection methods in IoT networks: An experimental study. *Applied Sciences*, 14(24), Article 11545. <https://doi.org/10.3390/app142411545>
- Muntean, M. V. (2024). Real-time detection of IoT anomalies and intrusion data in smart cities using multi-agent system. *Sensors*, 24(24), Article 7886. <https://doi.org/10.3390/s24247886>
- Muzumdar, P., Bhosale, A., Basyal, G. P., & Kurian, G. (2024). Navigating the Docker ecosystem: A comprehensive taxonomy and survey. *Asian Journal of Research in Computer Science*, 17(1), 42–61. <https://doi.org/10.9734/ajrcos/2024/v17i1411>
- Ogbuefi, E., Odofin, O. T., Abayomi, A. A., Adekunle, B. I., Agboola, O. A., & Owoade, S. (2021). A review of system monitoring architectures using Prometheus, ELK Stack, and custom dashboards. *Iconic Research and Engineering Journals*, 4(12), 408–419.
- Pang, G., Shen, C., Cao, L., & Van Den Hengel, A. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2), Article 38, 1–38. <https://doi.org/10.1145/3439950>
- Shaheen, A., Al-Sayyed, R., & Sleit, A. (2017). Improving visual analyses and communications of ontology by dynamic tree (case study: Computer system). *International Journal of Advanced and Applied Sciences*, 4(5), 62–66. <https://doi.org/10.21833/ijaas.2017.05.011>
- Shenify, M., Mazarbhuiya, F. A., & Wungreiphi, A. S. (2024). Detecting IoT anomalies using fuzzy subspace clustering algorithms. *Applied Sciences*, 14(3), Article 1264. <https://doi.org/10.3390/app14031264>
- Singh, S., Chana, I., & Buyya, R. (2020). Agri-info: Cloud based autonomic system for delivering agriculture as a service. *Internet of Things*, 9, Article 100131. <https://doi.org/10.1016/j.iot.2019.100131>

- Soori, M., Arezoo, B., & Dastres, R. (2023). Internet of Things for smart factories in Industry 4.0: A review. *Internet of Things and Cyber-Physical Systems*, 3, 192–204. <https://doi.org/10.1016/j.iotcps.2023.04.006>
- Umar, A., & Jordanov, I. (2026). Anomaly detection in IoT environment using machine learning: A survey. *IEEE Internet of Things Journal*, 13(10), 20477–20497. <https://doi.org/10.1109/IIOT.2026.3671102>
- Zahoor, A., Abbasi, W., Babar, M. Z., & Aljohani, A. (2025). Robust IoT security using isolation forest and one-class SVM algorithms. *Scientific Reports*, 15, Article 36586. <https://doi.org/10.1038/s41598-025-20445-4>