JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Anomaly Detection in Industrial IoT Sensor Streams: A Systematic Review of Data Engineering Pipelines and Detection Mechanisms

[1] Sriharsha Gudi,
[1]Individual Researcher, Senior Data Engineer,11723 NE 117th CT, Kirkland, WA, USA 98034.
sgudi3966@gmail.com

Abstract:
Industrial Internet of Things (IIoT) have witnessed tremendous growth with mass deployments in energy, logistics, and manufacturing industries. As a result, huge range of sensory data has been introduced for real-time processing to ensure cost-effectiveness, safety, and consistency in operations. Traditional batch processing methods are not effective to handle failures, anomalies and cyber threats that require a quick response. This study uses a systematic literature review to identify a complete mechanism on how to create low latency and scalable pipelines for real time data optimization of the detection of anomalies in sensors from the IIoT. In this study, too, the problems of sensor mix, high data rates, edge to cloud bandwidth limitations and architectural limitations are discussed. Tools used for data ingestion (MQTT, Apache Kafka, etc.), edge computing, modular pipeline, and real-time analytics, are also investigated. Special emphasis is given on anomaly detection models that can manage high-dimensional data, such as, statistical models for process control, autoencoders, and hybrid methods related to machine learning. This study investigates various methodologies of optimizing a data pipeline for fault tolerance, load balancing and real-time alerts in the lifecycle by systematically reviewing the literature available to date. In this study, several other studies were synthesized and presented smart manufacturing scenario to analyse use case and obtained very low latency and reduced downtime. In IIoT applications, such as quality assurance, security, and predictive maintenance, it is noted that real-time and scalable pipelines are key and achievable.

Keywords: IIoT, data pipeline, machine learning, MQTT, Apache Kafka, cloud environment, edge environment

1. Introduction

The environment data such as vibration, temperature, humidity and pressure data are collected by the sensor nodes, and the industrial application is provided with an environment where sensor nodes are heavily used for attaching to collect environment data (Moallemi et al, 2022). Fault tolerance and consistency in operation is highly critical in the energy sector (Enemosah and Chukwunweike, 2022), manufacturing industry, oil and gas industry and logistics industry. The sensor network can provide the most precise and current data on assets that are often too far to reach, optimize process, predictive maintenance and quality assurance with highest level of precision and accuracy. In addition, compared to the traditional approach of monitoring, which is based on the Supervisory Control and Data Acquisition (SCADA) system, with a few rules and manually performed tasks, the complex IIoT networks are more effective (Kim and Mukhiddinov, 2023). But for those who are going to do IIoT, they'll need to use a new type of program to process the vast amount of data, detect anomalies before they become disasters and sift through the noise to find the signal. In Figure 1, the technology is illustrated and how the existing technology or systems have been replaced by more advanced technologies based on the IIoT systems' capability to detect anomalies. The good news is that there's lots of it around, but there are also many ways to fool the sensors, create a changing environment, cause the sensors to drift with time and even attack the

sensors. As a result of these potential risks, it is important to have a well-designed data pipeline that has several key components, which should be done correctly: The data flowing into the data pipeline should be cleaned and joined and outliers should be identified as far as down the data pipeline as possible (Wilbur et al., 2019). The majority of the contemporary solutions that have been developed in the IIoT area have been developed with the concept of real-time anomaly detection.

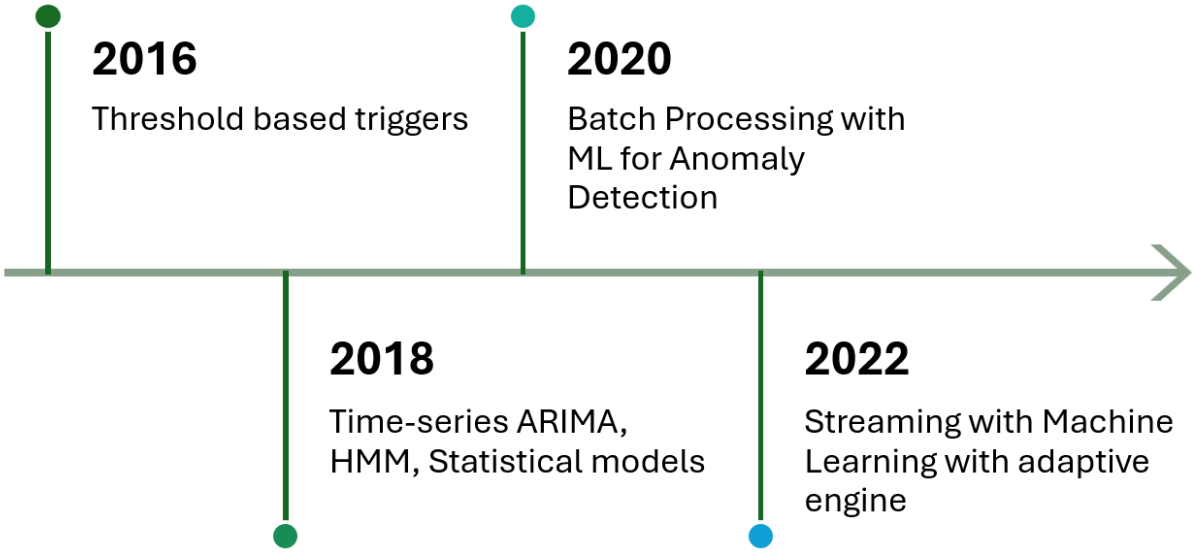

Figure 1 – Timeline of Evolution in IIoT Pipelines for Anomaly Detection

1.1 Problem Statement

This failure can have a disastrous impact in an industrial setting, whether it is due to equipment failure, a malicious user intruding on the system or sensors not measuring at the proper values. The traditional "batch" model, however, failed to meet the needs of the systems that focus on assessing the economic viability and efficiency of the system as well as safety while the system operates in real time (Liu et al, 2023). Moreover, the information from sensors in different dimensions is gathered and used to create the challenge. In general, these signals are "asynchronous", "non-stationary" and "limited" (Gao et al 2022), which makes it hard to model the events of these signals; it is likely to be "outliers" or "temporal correlation" and/or "pattern events" in the context. If the shaft is rotating, the anomaly is small (e.g., irregular vibration) and if left uncorrected in time causes halt production or causes a great asset damage (Shajihan et al, 2022). To tackle these challenges, scalable data pipelines need to continuously ingest data and draw inferences with low latency and constantly pre-process the data with real-time anomaly detection. It is also required in remote sensing (e.g., offshore platforms and/or mining) but less human involvement is required for smart factories (Tang et al., 2019). There are multiple traditional pipelines that don't support real time drift, time-window and model retraining due to lack of support for industrial automation and integration of the Machine Learning (ML) paradigm. The comparison/contrast between the legacy alerting model's vs the Adaptive and Modern Anomaly detection models/approaches in an IIOT environment is shown in the following table (Table 1 below): An anomaly detection misdetection could be a significant threat to employees, damage assets and result in monetary risks to operations. The concept of data Architecture needs to be addressed to overcome the ability gap from static and fragmented silos to smart Data pipelines (Bono et al, 2023).

Table 1 – Common Tools and Technologies across Pipeline Layers

Pipeline Stacks	**Tools/Technologies**	**Functions**
Ingestion broker	Kafka, AMQP, MQTT	Reliably transporting message
Sensor and Edge Computing	Raspberry Pi, Arduino, PLCs, STM32	Encoding and collection of data
Model inference	ONNX, TorchScript, TensorFlow Lite	Decision-making and detecting anomalies
Stream processing	Azure Stream, Apache Flink, and Spark streaming	Processing of events in real-time
Visualization	Power BI, Kibana, and Grafana	Alerting and dashboard
Storage	Amazon S3, HDFS, Influx DB, Prometheus	Scalable storage and time-series archival

1.2 Aim and Objectives

After conducting a systematic review of literature, this research is focused at extracting a scalable Data pipelines and Architecture for Anomaly detection in an IoT sensor network. It will demonstrate the distributed and modular data structure in a changing world of Industry. The following are some of the main objectives – To detect anomaly from the data from IIoT, using pipelines. The authors categorize the Anomaly Detection models, provide a brief description of each of the models and discuss the usage of these models with scalable data. Talk about Hybrid models, machine learning and deep learning models for anomaly detection. In this paper the research gaps i.e. in real-time detection of anomalies from the sensor data of IIoT will be discussed and research directions for future will be given.

2. Literature Review

2.1. How Anomaly Detection can be developed in an IIoT Environment

Anomaly detection has been evolved along with technological advances in IIoT systems. First, it's episodic analysis, logs are analyzed and stored in the rearview mirror. A small group of scientists believes that IIoT development model was not suitable for taking fast actions in dynamic environments (Kim and Mukhiddinov, 2023). The main problems with the batch model were latency of the data, anomalies not being found and storage costs. Real-time processing is a dramatic change, where on-going analysis of created data is possible (Wilbur et al, 2019). Industries can react to anomalies within milliseconds on stream analytics models like Apache Flink, Apache Kafka and Azure Stream Analytics. A comparative study of anomaly detection from threshold to ML based streaming was discussed in this paper and Fig 1 shows this. The foundation stone for the construction and implementation of light models around data sources is edge computing, which can effectively shorten the latency of cloud systems in the round trip (Liu et al, 2023). It was especially important in environments with having poor connectivity like rural manufacturing or offshore rigs. Machine Learning (ML) is used in streaming to off-the-shelf models such as Isolation Forest, k-means and LSTM which can be deployed in data pipelines. With these models, both failure patterns and novel behaviors are detected.

2.2. There are currently a number of data pipeline IIoT Architectures

High throughput pipelines of data need to be established to enable the potential of a flexible and scalable modern IIoT system for each of the modalities. The basic structure of a typical data pipeline includes layers including data ingestion, data acquisition, model inference, stream processing, visualization and storage as mentioned by Shajihan et al, 2022. A high data rate per layer with minimal data loss and integrity is required due to network jitter/latency – typically in the order of kilohertz per layer. They are pushed by Programmable Logic Controllers (PLCs) and/or sensor gateway to the data collection layer (e.g. AMQP, OPC-UA or MQTT protocols). To break and buffer incoming streams of sensors, ingestion layer such as RabbitMQ or Kafka are usually employed as message brokers. The dissociating of consumers and producers in multi-node systems can help to provide fault tolerance and reliability. Pipeline is based on the notion of stream processing. Aggregation and anomaly detection (Bono et al, 2023), windowed calculations and so on can be performed using tools, including Spark Streaming and Apache Flink. A different sensor traffic can be supported by these models which can be dynamically scaled to match the load, orchestrated and containerized using Kubernetes. Inference engines are combined with the layer of stream processing to conduct training of ML models on past data. Light models can be deployed to ONNX Runtime/TensorFlow Serving (Jiang et al, 2023) for low latency, e.g., making prediction. These can be created to detect outliers as they occur, and can start loops or alarms at the edge to react immediately. Processed data is stored in the archival layers and stored layers in the time-series database such as Prometheus or InfluxDB for efficient query and high compression and for future audits, model retraining, for root cause analysis, etc. (Santhosh et al, 2020). Visualization dashboards can be used to add health metrics, model outputs and anomalies to real-time visibility like Kibana or Grafana. No matter what architecture modules are used, there are a number of integration-related issues due to various protocol standards and the differences in hardware. The majority of anomaly detection mechanisms in the field of IIoT are based on these data pipelines that make up operational backbone.

2.3. Research Gap

The following are some key gaps that need to be addressed to reduce latency, leverage multi-modal data and scale sensor data; although pipelines are now relatively mature to benefit from the IIoT. When the number of sensors grows, for example, thousands of sensors are deployed in a smart factory, scalability has become an issue. There are partitioning and synchronization issues in stream and ingestion layers which make real-time

analytics complicated (Li et al, 2022). Latency is also a challenging affair, especially when sub-second detection is needed in environments for operations where safety is needed. While there are some advantages to using Edge when compared to cloud resources, propagation of updates, deployment of models (Liu et al, 2022), etc., may be an issue. In addition, overhead is needed for the data pre-processing tasks like deduplication, normalization and outlier removal, which are not easily parallelizable. Another significant challenge is fusion of multimodal data. The IIoT systems are continuously equipped with a variety of vibration, thermal, acoustic and pressure sensors having various noise characteristics and sampling (Sun et al, 2023). These data flows can be combined into one context, but will need to be context aware and feature aligned. But the existing dissimilar data synchronization approaches are limited in their effectiveness and are not capable to include correlations among sensors or consider the fidelity of the results based on anomaly detection. There are a number of commercial products available to allow adapting of the dynamic model while running, when the behavior of the sensors changes as a result of changes in the environment. Table 2 below compares the model types for IIoT pipeline with respect to sensor fusion, latency and scalability. Going forward, future research efforts should focus on proposing architectures to cope with hybrid nature of edge and cloud coordination, adaptive learning and interoperability. From the pilot testing stage (Pan et al., 2023) the developments will be scaled up to high throughput and complex environments.

Table 2 – Difference between Model types for detecting anomalies

Models	**Interpretability Level**	**Accuracy Level**	**Latency Level**	**Cost of Computing**	**Ideal use cases**
Machine Learning	Moderate	High	Mid	Low or sometimes medium	When anomaly is multivariate
Statistical model (like ARIMA)	High	Mid	Low	Too low	For tracking threshold
Deep Learning	Low	Too high	High	High	When data is non-linear and temporal

3. Methodology

3.1. Research Approach

The approach for this study is Systematic Literature Review (SLR) which is followed by Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA). The PRISMA 2020 flow diagram for identification of records to final inclusion of studies for organizing the relevant findings of the IIoT anomaly detection is shown in figure 2. For this study, only articles that were relevant to the study were selected from peer-reviewed and high impact journals in the years 2019-2025. A comprehensive search was carried out in five major databases to capture the recent advances and mechanisms for anomaly detection in IIoT sensor data. There are 1267 records identified in the initial phase – Scopus (n = 524), IEEE Xplore (n = 331), Science Direct (n = 231), SpringerLink (n = 125), and Google Scholar (n = 56). Of those, 247 duplicate records, 67 records were ineligible and 44 other records were excluded and 903 records were screened. There were 634 records found which were irrelevant to the abstract, and were therefore excluded. There were 275 reports that were wanted. Unfortunately, 53 reports lacked full text and 222 reports were evaluated for eligibility criteria. Of those reports, 187 were not included, either because they were not using an anomaly detection approach (134) or the article was not focused on anomaly detection (42) or was missing information (11). Lastly, thirty-five studies were selected for the study.

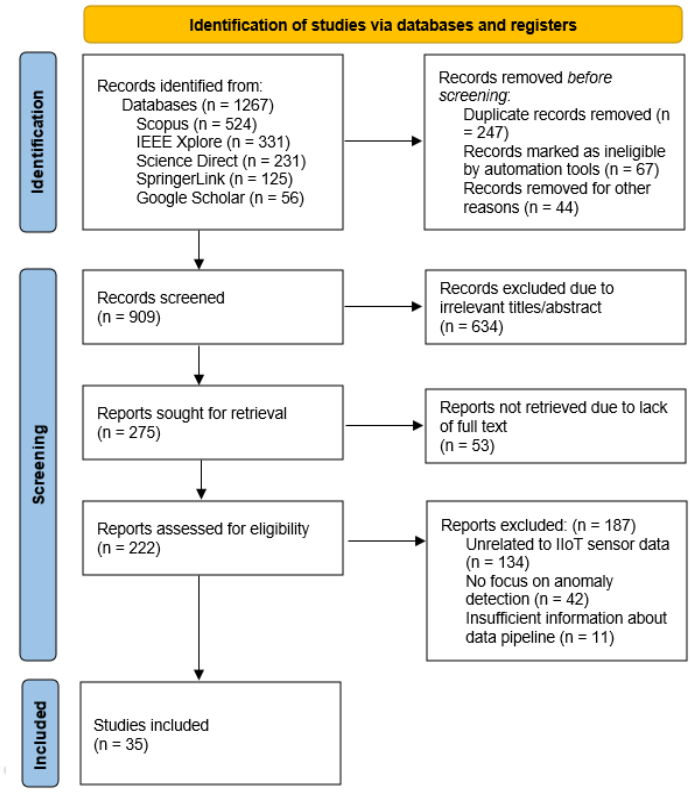

Figure 2 – PRISMA Flow Diagram

3.2. Search Strategy

The literature search in the key databases (IEEE Xplore, Scopus, Springer Link, Science Direct, Google Scholar and ACM) was conducted, and only relevant studies were chosen. Search using Boolean operator. The context of the search terms is summarized in Table 3. Table 4 below shows the inclusion and exclusion criteria was performed to filter out the irrelevant article and only the relevant articles were selected.

Table 3 -Search Terms and Combinations

Context	Search Combinations
Detection	"IIoT" AND "anomaly detection"
Data Architecture	"Industrial Internet of Things" AND "data pipeline"
Computing	"Machine Learning" AND "anomaly detection"
Models	"ML" OR "machine learning" OR "Deep Learning" OR "DL"
Sensing	"Industrial sensors" AND "stream processing"

Initially, a wide range of studies were collected during initial search. After the screening of full texts, titles and abstracts, only those studies that had relevance to the criteria were included in further analysis.

Table 4 Inclusion and Exclusion Criteria

Inclusion Criteria	
Publication year	Studies must be published between 2019 and 2025
Relevance	Studies must be related to IIoT sensor data, anomaly detection, and data pipeline
Language	Records must be published in English language
Full text	Studies must be available in full text
Exclusion Criteria	
Screening	Studies having irrelevant title/abstract
Inclusion	Studies not meeting above inclusion criteria
Information	Studies having insufficient information about data pipeline
Duplication	Studies having duplicate publications

Initially, a wide range of studies were collected during initial search. Once full texts, titles and abstracts were screened against the criteria, only relevant studies were selected for further analysis.

4. Architecture of Data Engineering Pipeline

4.1 The Scalable IIoT Data Pipeline enables you to attach layers to the pipeline

End-to-end IIoT data pipeline is composed of several components which take care of processing huge amounts of data from various and high frequency sources. The first layer in the pipeline starts at the layer of raw data encoded and collected from the distributed sensors for sending out. It follows ingestion layer, a reliable and light layer which is based on protocols such as AMQP and MQTT (Jiang et al, 2023). An engine of stream processing handles the real-time operations of computations such as event correlation, anomaly scoring and predictive modeling. In this tier, typical tools are Apache Spark Streaming, Flink (stateless and stateful) (Santhosh et al, 2020). These engines are usually used in isolated environments, and can also automatically scale the computations based on the workload, via Kubernetes. Then, the pipeline routes data which has been processed for modeling inference layers when it comes to execute machine learning models. The optimizations are done under the low latency mode and the models are converted to TensorFlow Lite or ONNX format (Li et al., 2022). Three models for decision making are used to initiate to automatic control and/or alert. The architecture for modular pipeline for cloud/edge computing is shown in Figure 3. Finally, complex queries and retrieval of data required some sort of data storage—either a specialized database or data lakes—were required. The storage to be used for retraining of ML models, compliance and auditing purposes. Operational knowledge of data analysts and technicians on top of the pipeline is supplied by visualization dashboards. There are some protocols and software tools which need scalable architecture and need to be coordinated. In this table, the technologies used in each of the pipeline stages and the interoperability needs are presented which should be addressed to enable low latency, smooth function.

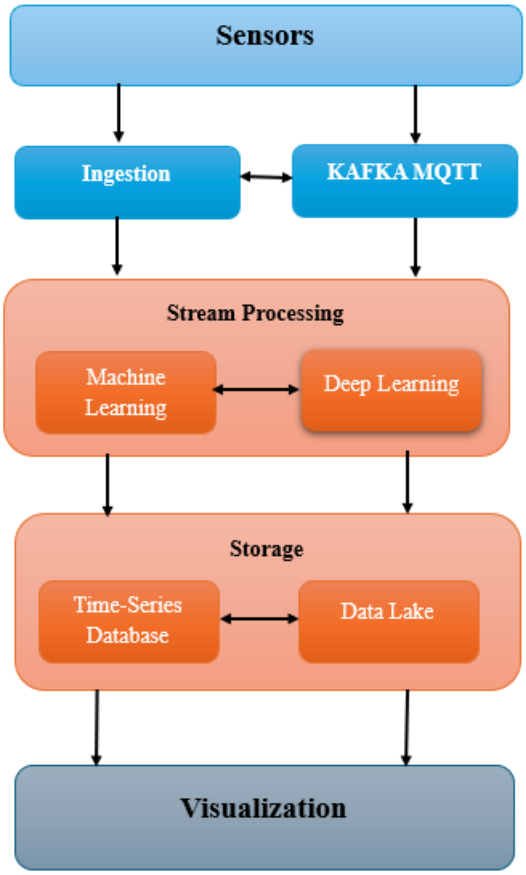

Figure 3 – Architecture for End-to-End IIoT Pipeline

The architecture layer in figure 3 illustrates the separation of the different elements of the device-level data, stream, ingestion brokers, storage layers, inference layers and visualization tools.

4.2. Edge Computing

In the IIoT systems that use edge computing, data volumes, as well as bandwidth and latency, have all been transformed. However, in other industrial applications that are not typically centralized, like distributed mining, refinery, and isolated manufacturing in remote location, centralized computation is not feasible, as it is

high latency, or has the issue of connectivity reliability (Sun et al, 2023). It is treated with edge computing to do the computation locally around or at the source of the data. Typically, the edge nodes will be connected to a single edge board or micro-controller such as NVIDIA Jetson or Raspberry Pi etc. To compress, filter and/or regularize the data before sending them out, the nodes will be used to perform light routines for analytics of the raw data. (Pan et al, 2023) During this stage of pre-processing, only data anomalies or data with high value are passed on to cloud platforms and the amount of data that is sent is also minimized. In addition, using edge computing, the running of inference models can be made real-time to make fast decisions about safety, e.g. alarm when temperature is below/above a threshold, or stop the machine. Some of these decisions are critical decisions (Zheng et al, 2019) because safety of a worker or the quality of the outcomes of a worker's work may be compromised within the few milliseconds. On the edges, ML models are just a quantized version of the large-scale models that are converted to tools such as ONNX Runtime or TensorFlow Lite. Moreover, human machine interfaces (HMIs), also known as dashboards, can be created to carry out the functions linked to the edge computing architecture and information can be pushed to the operations without remote logon. These are visualization of active alerts, thresholds or sensor health. The core platform is normally developed further, and is synchronised on time with the systems being able to process the rules and model/firmware changes. Edge computing can help provide context-aware and rapid, smooth responses to anomalies in the context of IIoT (Avci et al, 2019).

4.3. Data Ingestion

In IIoT pipeline, the data ingestion layer transports huge amount of time-series data from the sensor to the processing environment. In the organizations like MQTT, Apache Kafka, etc. (Ye et al, 2023), real-time ingestion brokers are utilized. They can be used to connect the consumers (typically databases) and producers (typically gateways or sensors) in a low latency, scalable and fault tolerant manner. MQTT is a simple protocol especially suited for lossy and narrow networks. It offers three different “Quality of Service (QoS)” levels and low overhead that enable it to deliver reliably in edge computing even if the network is not reliable (Verma et al, 2020). Local gateway/edge networks connect to the MQTT instances (HiveMQ/Mosquitto), to share some of the load on the uplink. Kafka is designed to be used in enterprise-scale environments that have high throughput and durability. It can provide the partitioning and persistence of distributed message, and can be consumed by multiple applications that are related to the real-time processing (Zhao et al, 2022). Kafka's offset-tracking and replication features help to safeguard the availability of data when the system fails. It's not ideal, however, for a data center that is a combined node versus heavy duty architecture. Both the AMQP and the RabbitMQ systems feature a message queueing mechanism and have advanced routing mechanism. They're not very capable of handling large loads when it comes to ingesting them, but RabbitMQ has a very fine-grained system to acknowledge message that's excellent for mission critical applications (Chen et al, 2014). When selecting the ingestion model, it is important to take into account the requirements of throughput, protocol interoperability, resource constraints and temporal sensitivity. Indeed, they are cloudifying Kafka, and ingesting MQTT into a hybrid model. The combinations include some compromises on efficiency of resource and performance without compromising the integrity of the data. The ingestion layer for the IIoT pipeline creates the arterial system that allows for the real-time transfer of data for operational performance, automation and anomaly detection.

4.4. Which would be more suitable for your business requirements?

What if you are using information from a Data Lake or Time-Series? As a future longer term and real time analytics solution the analytics solution in IIoT pipeline is analysed with regards to pipeline storage. There are two formats that are quite popular—time-series databases (TSDBs) and data lakes. There are multiple different types of preferences with each having a distinct function, and most of the time they are associated to the other preferences (Du et al, 2022). Data can be gathered on an hourly or second-by-second basis (usually) and stored in a time series database (such as Prometheus, TimescaleDB or InfluxDB). These include features like thresholds, live dashboards and continuous queries, retention policies, and compression models. When it is crucial to access the latest data and detect anomalies in real time, their performance is good (Lu et al, 2023). Meanwhile, schema-based storage of processed and raw data in different formats including Avro, JSON, Parquet is offered in data lakes. The metadata and bulk telemetry could be stored somewhere else, like Hadoop HDFS, Amazon S3 or Azure Data Lake for analysis to understand the root cause, compliance or retrain the model (Li et al, 2020). A Data Lake is a place to store unstructured, structured and semi-structured data. These two are also used in a couple of pipelines in the IIoT.

5. Anomaly Detection Models

5.1. Time-Series Models

aIn the context of IIoT applications, a typical scenario is that of the monitoring of time-series data, which is typically done through the use of anomaly detection algorithms, pre-composed by sensors located around important data systems. The traditional methods are statistical methods such as moving average, control chart and auto-regressive method to find abrupt changes in the previous baselines (Sun et al, 2023). Easy to understand and inexpensive, but not strong enough in complicated industrial environments, where the results of the sensors vary due to changes in industrial activities. With the help of ML techniques revolution has happened in the field, models have learned the baseline approach from the previous data. Anomaly detection methods such as One-class SVM, K-Means clustering, Isolation Forest (Pan et al, 2023) and other methods are used to detect anomalies, which are considered outlier/rare events. These models can be used for multivariate relations around modalities of the sensor, and can be more comprehensive in terms of detection of anomalous patterns. The latest technologies, such as "Temporal Convolutional Networks (TCNs)", autoencoders and LSTM networks are moving deep learning another step further to yield even better results in the field of non-linear/subtle anomaly detection. These models are well suited to model with high-dimensionality (Zheng et al, 2019), and are able to model long range dependencies as well as time series input. But typically, the amount of data and resources required in DL models and lack of layers that explain the model, makes them difficult to interpret. Choices of approaches are based on solid hardware, application and risk tolerances. In case of real-time applications, it would be preferable to have light ML models and statistical methods as they are quick and in case of Data centers having high computing power, it would be preferable to have deep learning methods. The models selected (excluding the axes of accuracy, interpretability, computations requirements and latency) are compared independently of those stated above as shown in Table 5.

Table 5 – Comparing Types of Models for Detecting Anomaly in IIoT

Model	Analysis	Accuracy	Latency	Computation Needs	Where to use?
ML (SVM or Isolation Forest)	Moderate	High	Moderate	Medium	Detecting anomalies
Statistics (Z-score, ARIMA)	High	Moderate	Low	Low	Threshold tracking
Deep Learning	Low	Too high	High	High	Non-linear and temporal data

5.2. Hybrid Models

The systems in IIoT are a popular solution to the 100 % rule or data-based systems – anomaly detection is a popular technique that has been applied in hybrid systems. In these models the physics-based models are coupled with adaptive AI models (Avci et al, 2019). Physics based models: models based on first principles which simulate behaviour based on thermodynamic, mechanical and electrical properties. Lastly, AI models could be added to these simulations, models that would be able to detect anomalies that are not related to assumptions, data or sensor failure and so on (Ye et al., 2023). It simplifies the ability to become more believable and solid in the regulated environment when it's needed for the anomaly detection pipeline. In predictive maintenance, for example, expected ranges can be predicted based on physics-based models and compression load and RPM. To detect abnormal residents, the encoder can be applied to a neural network to match the control error, leakage or wear (Verma et al, 2020) – if the data observed is outside of the expected range. Moreover, false positives are addressed by the hybrid models with the alerts being cross-checked between the ML and simulation models. It is particularly valuable if there are some anomalies, which may occur during the calibration process because of their environment (e.g. temperature change) but not because of fault. The other one is digital twin setting where digital twin parameters are continuously updated based on the digital flow of the sensor data and digital AI is utilized (Zhao et al, 2022). While combined models are more conservative, they also have a more robust design of the produced system but are flexible and adaptable to dynamic environments. They have a balance of the empirical and deterministic learning as in modern best practices with reliability testing.

5.3. Weak Data Drift, Model Training in streaming

When the behaviour of a sensor changes over time due to changes of their environments and/or their operations (concept drift), correct results are key to successful anomaly detection. In real-time pipelines, it is crucial that an anomaly detection of the data, data ingestion and models adaptation happen without any manual interaction (Chen et al, 2014). Methods that can be applied to incrementally update the model include online learning techniques like K-means, Adaptive random forest and Hoeffding Trees. The models are also modified to adjust to the slow variations of the pattern in real-time and unusual variations are considered as exceptional variations (Bao et al., 2019). A way to cope with the drift can be to have models for the "sliding window" method where models are retrained with the latest samples in a given prospect. This technique can be used for systems that will be used for a period of time, such as a conveyor system or the bottling line in the industries. Sudden changes can be identified by using statistical characteristics like autocorrelation, variance and means (Du et al, 2022). The other approach is to have a set of weights to be applied to the past predictors that are correctly predicted, and then use ensemble learning. This Ensemble method could be used for updating of a poor performing model(s) and/or pruning in different contexts (Lu et al, 2023). A concept drift detection is carried out by Adaptive Windowing (ADWIN) to detect the requirement for trainings. Labeling of the data has been challenging. Use semi-supervised and active learning to only ask for unwanted predictions in order to reduce labelling costs (Li et al, 2020). The operator confirmations and the confirming the model with the operator (annotation) and feedback loops can be used to retrain the model and further improve the accuracy of the model. Finally, the successful deployment of streaming is all about models' ability to keep stability and responsiveness to ensure accuracy without untouchable operators with false positives.

5.4. Lightweight Models

In the scope of the IIoT network, it's crucial to deploy the anomaly detection model to an edge device, due to the need to save the bandwidth and real-time response. Moving to edge and reducing transmission dependency to cloud can help to reduce the latency (Mohandas, 2023). It does cause some restrictions on memory, energy and computation, though. Models can be optimized for these limitations by pruning models, quantization and distillation of models and their architectures. The most popular method for modelling size reduction is quantization (or coding) of the weights used in the floating-point model that are stored in integer types, this could save up to 4 times the memory compared to the floating-point model with little loss of precision (Difallah et al, 2013). This will ensure that the knowledge will be passed down from the complex models and neurons will have less impact which will be reduced by distillation. Many frameworks exist that enable such light models, like Edge Impulse, PyTorch Mobile and TensorFlow Lite. For instance, they optimize runtimes with hardware and operator of Google Coral or NVIDIA Jetson Nano, edge hardware (Boccagna et al, 2023). The reliability of the edge device, and fault recovery, could be increased by including other protective functions, e.g., watchdog functions and local buffers. To ensure that the model is not tampered with, certain layers of security can be added to the model, such as encrypted binaries of the model and secure bootloaders. The models are likewise hugely user friendly, if there is not an excessive amount of Internet. Synchronization of dashboard (Cloud & Periodic), operation parameters, motor on/off is locally synchronized and orchestrated. Indeed, the edge computing technology can offload computations, encourage local operators and provide increased failover capability in an autonomous decision-making process. Such benefits are very critical in industries where safety is a critical factor, for example, power substations, oil rigs and processing units.

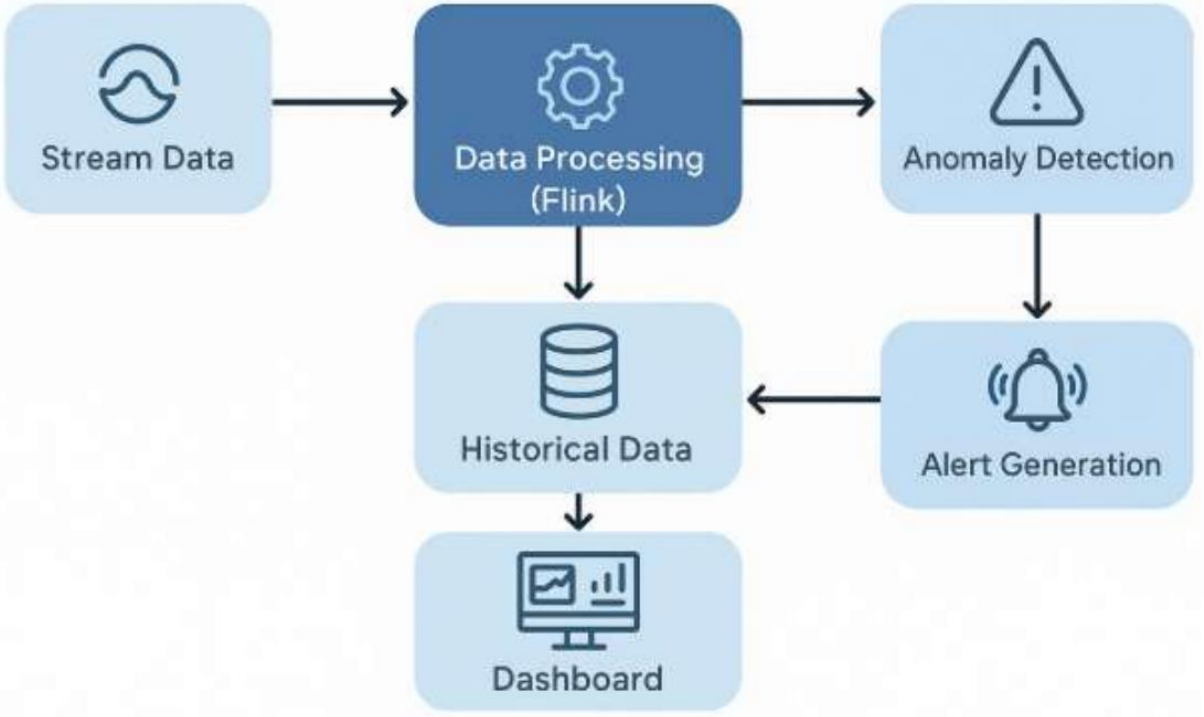

Figure 4 - Alert Generation Pipeline using Flink for Notification and Anomaly Detection

5.5. Take into account Flink and Spark Streaming

Discuss about Flink and Spark Streaming. With minimal delay, Industrial IoT can detect anomaly when the data of the sensors is processed. Real time pipelines are a required component in many real-world applications, where there are various technologies, such as Apache Flink and Apache Spark Streaming that can be used to implement a fault-tolerant system and processing systems (Verma et al, 2020). The idea behind micro-batch model is to break up each data into smaller batches, then Apache Spark streaming to process each micro-batch in a short time interval. This can introduce some delay, but can give some protection in the event of spikes and can easily be used in conjunction with query models (Zhao et al, 2022). In such systems, legacy Hadoop needs to be interoperable and in general, Spark is utilized. Apache Flink is intended to process every event in real time; one at a time. It allows the low latency windowing, advanced event processing (CEP) and back pressure management for the IIoT to prevent the negative impacts and failures within milliseconds (Chen et al, 2014). The semantics and checkpointing of Flink offers good industrial data processing. Both platforms are able to connect to other messaging systems such as MQTT and Apache Kafka for seamless ingestion of streams of IIoT sensors. In addition, Flink provides watermarking and event processing functionalities that are vital to a distributed system with intermittent or jittery connections and/or message delivery (Bao et al, 2019). For low latency scenarios, sometimes the desire of the application and the system are key to deciding on using Flink or Spark. Maybe it would be preferable to leverage the stream-oriented system in Flink instead of Spark. But Spark has improved when it comes to integration, with the current big data tools.

5.6. This will result in a rethinking of the Confidence Scores, Thresholds and Alerts

The rethinking of the Confidence Scores/Thresholds and Alerts. After the data is an anomaly, it is crucial to have alert logic to know what to do with the data. It offers confidence scores, thresholds and diagrams logic tier levels of converting probabilistic responses to alarms (Du et al, 2022). Threshold can be related to a static or an operational context, depending on historical baseline or dynamic performance, via weighted functions and/or moving average. Dynamic thresholds are flexible thresholds, for example, for warm-up cycles, or seasonal changes in the IIoT systems (Lu et al., 2023). For instance, if oscillations get larger when the motor is starting up, then a vibration sensor can be used to control the oscillations, but it can alarm if oscillations follow a pattern within a pre-specified window prior to motor start up. Based on the anomaly confidence score, the anomalies that can be derived from models can be sorted by the anomaly confidence and/or the severity of the anomaly. These scores can be used to run alerts across operations, filter alerts and prioritize alerts to curb alert fatigue and eliminate false alerts. For instance, LSTM detector can be utilized to monitor a reconstruction error, analyze and normalize the reconstruction error to get the confidence levels of anomalies (Li et al 2020). Many different signals are sometimes combined in composite alert logic, such as rule-based thresholds, time-series predictions and expectations based on mechanics, to set off an alarm only when it is time to do so to meet various criteria. The light decision trees in this stage control the specificity and sensitivity, and in this stage fuzzy inference and

Boolean logic (Mohandas, 2023) is applied. A typical approach is to combine the logic for generating an alert together at the edge or in a layer of orchestration in the stream processor. To reduce network traffic, it is possible to alert in highly available systems over AMQP or MQTT. For critical alerts automated actions are provided: local visual alarms, stopping machinery, can be appealed. It is a logic that can be translated into triggering the alarm, that is the time when data is being ingested (as can be seen in Figure 4).

5.7. Operational Dashboards & Maintenance

If these anomalies and triggering of alarms can be seamlessly incorporated into the utility's dashboards and maintenance, it could be beneficial. Dashboards are used as the human to machine interface which assist plant managers and maintenance personnel to detect anomalies, monitor system status and coordinate actions. Real-time APIs and data sources are often connected to the platforms or web interfaces to build contemporary dashboards.Modern dashboards can be created on platforms or on web interfaces, that talk to real-time APIs and data sources. It has existing equipment as well as dashboard displays of historical information related to anomaly, drill-downs to sensor data and alert logs. Furthermore, it could be possible to enrich the analysis of the root cause with context information (maintenance history, technician serial or notes from the technician) (Boccagna et al, 2023). Dashboards have in-built visualizations in order to make it easier to read, such as using sparklines to identify trends, traffic lights to indicate health status and geographic overlays. However, the response rate and accountability can be enhanced and alerts can be set high, if the “Computerized Maintenance Management System (CMMS)” can be used to generate a work order automatically (Becker et al, 2020). Data should be pushed to data layers on dashboards from a system perspective without impacting performance or providing feedback loop with pipelines. Limit information sent/received to Summary/Filtered info to RESTful endpoint and/or pub-sub. The alerts generated by the Spark or Flink processors for analytics and display are saved in NoSQL databases. With operational dashboards, and alert logic, organizations get end-to-end visibility of asset health, enabling them to allocate resources proactively, give their assets proactive maintenance, and ensure all automation goals are met.

6. Use Case Example from Smart Manufacturing

A portion that will make the real-time anomaly detection process available to the market, and will use this process in the context of smart manufacturing environment (which has been studied in recent years). The findings in this section are based on the findings by Jamiu and Chukwunweike (2023) and other recent studies.

6.1. Plant set-up and sensor data

After the above, Jamiu and Chukwunweike (2023) chose a medium scale manufacturing plant working with smart manufacturing and then implemented the authentication and conduct anomaly detection real time with the help of pipeline. Over 200 edge linked sensors are used and integrated around the CNC machines, the production line, electricity meters, cooling vents etc. The streams of time series data are generated by these sensors like power usage, temperature, vibration, currents, and acoustic noise (Du et al, 2022). MQTT broker was designed with the vision of sending a small message, and simplicity of deployment and integration from the edge to the core for sensor network. Normalizing signal and noise reduction from the streams were applied on the streams before they are sent to the core cluster of Kafka, which are accomplished by using a local fog node in which the data can be aggregated. Per day the plant produces about 2.4 million records from sensors and sensors have varying frequencies (from 1Hz to 50Hz) depending on the nature of the sensor. (Lu et al, 2023) The most promising application in the plant is for the purposes of sensing the failure of the exact cutting process with the fast-rotating spindle. Micro-level changes in the vibration (oscillation) were measured by installing vibration sensors around the spindle bearings, this was an indication of wear out or misalignment of spindle bearings. This anomaly problem was caused by older anomalies which was occurring approximately 10 days prior to the huge cost down time. Detection of events in real-time for inference and stream ingestion in the pipeline were made using Flink as mentioned. In addition, past and offline faults are used to train LSTM-based model and confidence weighting was used to generate anomaly score. A Hybrid between multivariate drift and threshold deviation is used as the trigger to activate dashboards when pre-failure situations arise (Li et al, 2020). This experimental application will enable IIoT pipelines to go into a production environment without any restrictions on operational up time or on control workflows.

6.2. Latency, Acc and Throughput

A pipeline for throughput, anomaly detection metrics and latency proved useful in the past to build a good real time anomaly detection system at large scale. The actionable alert and generation of sensor data were measured

with processing latency of 112 milliseconds which is better than the average processing time of 3 seconds for conventional SCADA (Mohandas, 2023). Being a legacy system proved to be beneficial more in the batch sterilization or in the event of laser cutting at a high speed, when the decision making needed to be made within a millisecond. With a cluster with 2 parallel jobs and four task managers, Jamiu and Chukwunweike (2023) reported throughput level of 13000 events per second. It has been validated for scalability and a maximum rate of 11000 events per second (Difallah et al, 2013) was able to be processed by the system. Jamiu and Chukwunweike (2023) reported the F1 score as being consistent at 0.87, the accuracy as measured by F1 score, recall and precision around annotated failure as 0.87, 0.91 and 0.88 respectively. However, interestingly, the contextual elements are coupled with false positives like, equipment usage cycles, ambient temperatures in the anomaly model. The z-score pruning was used with edge-based filtering to suppress triggers induced by noise without having to lose sensitivity (Boccagna et al, 2023). The metrics used for baseline values of SCADA (AeP) is summarized in Table 6. The results have demonstrated the differences in performances of the conventional and modern monitoring systems. The more specific the quicker the more likely it will be up and running, and the less expensive it will be to keep it that way (Becker et al, 2020).

Table 6 – Baseline vs Optimized Pipeline Metrics

Metrics	Traditional SCADA	IIoT Pipeline
Peak throughput (event per second)	2,500	13,000
Average latency (in milliseconds)	3,000	112
False positives (weekly)	9.3	2.1
F1 Score (detecting anomalies)	0.61	0.87
Monthly downtime (in hours)	7.2	2.6

Source - Jamiu and Chukwunweike (2023)

With multimodal input and model-based logic having an important role, real-time system can theorize anomalies and help to increase the accuracy of diagnosis. Also, no human involvement in the feedback loop was required and was there from the anomaly to the decision, it was retroactive feedback.

7. Discussion

7.1. Modern Data Pipeline vs Conventional SCADA Systems

The traditional structure of the SCADA system that has been implemented in the existing manufacturing systems, and that of the modern manufacturing system (IIoT pipeline) needs to be compared to gain the advantage of technology. Generally, the SCADA system consists of “centralized human-machine interfaces (HMIs)” and “programmable logic controllers (PLCs)” to survey and uncover concealed factors which cause events. These systems have good capacity in threshold alerts, but are not good regarding context-based/ adaptive anomaly detection (Islam et al, 2022). The traditional SCADA systems are based on a set of rules. For instance, if it is set, an alarm will sound constantly if there is no fault, and during a machine's start up. Those false positives result in mistrust, and desensitize the alarm. Also, the polling architecture of a SCADA leads to low bandwidth and latencies of high frequency streams (Nong et al, 2023). The pipeline she was concerned about is quite dynamic with regard to IIoT pipeline," Alcantara said. With LSTMs, one also makes the assumption that there is a correlation between the signals and the context - the signals to failure might be different from the benign ones, for instance. These enhancements in addition to the decreased false positives assist to intervene before mechanical failure (Altabey and Noori, 2022).

7.2. Future Research Directions

Due to bandwidth and privacy constraints, future research can be carried out on detecting anomalies using the federated learning (FL) framework. In case local data is available a preliminary FL model can be trained with local data, in case there is no data, the same can be harmonized with the core aggregator, that aggregates the updates. This approach minimizes latency and data privacy with collaborative learning around sensors which have been geographical dispersed (Altabey and Noori, 2022). FL detection is supported for IIoT in cases where the information from different sites may be prohibited to be aggregated in one policy, or it might not be feasible at all. For instance, offshore platforms and oil rigs can automatically observe the changes in vibration parameters, temperature/pressure parameters of manufacturing process at a remote location and parameters can also be important in shared model (Haque et al, 2021). In typical settings, dealing with distributed non-independent and non-identity (non-IID) databases, dealing with heterogeneity and handling imbalance are also

challenges. Furthermore, there are also differences in sampling rates and sensor drift and mis-labelling which also cause local gradients. Also, the number of communications required for exchanging the models parameters should be minimized and low bandwidth networks (Cherdo et al, 2023) should be used. A few recently developed techniques have been applied to these problems such as safe aggregation, federated averaging, adaptive weighting etc.

7.3 Limitations of the Study

The study is limited by a number of factors –

1. In this study, there is no primary evidence/experiments being used, therefore, it is a secondary study. The findings of this study will only be relevant to the English literature, there may be differences in other languages.
2. Some of the more recent developed applications were not submitted to the review process, due to the fast pace of the evolution of IIoT technologies.
3. Comparisons between the performance of the various models are difficult because the industry applications, data sets and criterion used to assess the performance of the models is different.

8. Conclusion

In this study, the scalability of data pipelines for the purpose of detecting anomalies in real-time in IIoT networks has been studied. Recent research has pointed out the developing trends of advanced industrial data fusion, real-time monitoring and low latency decision-making. The recent research has pointed out the emerging trend of advanced industrial data fusion, real-time monitoring and low latency decision making. With a critical review of recent studies, there are significant challenges observed in data loss, scalability, security and latency. This study presented an end-to-end data pipeline to fill these gaps, using real-time ingestion models, edge computing, hybrid models for anomaly detection, and time-series databases. Responsiveness and resilience are analyzed with key techniques like reinforcement learning, federated learning, and adoption of digital twins. Case scenarios are compared to conventional systems and measurable outcomes measuring real-time alerts, throughput, and accuracy are measured. The study also points to the future scope of constructing IIoT models that are resilience, ethical and smart. The findings in this research will help organizations move towards real-time predictive optimization.

Future Directions

There are several research gaps identified in this study which should be covered in future research. The results of this study have shown some of the anomaly detection models but the majority of these models are based on the central processing of data. To develop further advanced DL models with strong capability to detect anomalies in IIoT data pipeline, future studies will be conducted. There were also some recent studies dealing with decentralized frameworks, but these are not extensively covered and should be the subject of future research. This study suggests the use of light models for anomaly detection, designing data pipeline modular, and using a secure framework. The focus of this study is on fault tolerance, observability and provenance of data to ensure compliance and performance. Finally, digital twins can be coupled with ongoing policy tuning and simulation for enhancing operations decision support and efficiency.

References

1. Altabey, W. A., & Noori, M. (2022). Artificial-intelligence-based methods for structural health monitoring. *Applied Sciences*, *12*(24), 12726.
2. Avci, O., Abdeljaber, O., Kiranyaz, S., & Inman, D. (2019, May). Convolutional neural networks for real-time and wireless damage detection. In *Dynamics of Civil Structures, Volume 2: Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics 2019* (pp. 129-136). Cham: Springer International Publishing.
3. Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, *18*(2), 401-421.
4. Becker, S., Schmidt, F., Gulenko, A., Acker, A., & Kao, O. (2020, December). Towards aiops in edge computing environments. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 3470-3475). IEEE.
5. Boccagna, R., Bottini, M., Petracca, M., Amelio, A., & Camata, G. (2023). Unsupervised deep learning for structural health monitoring. *Big Data and Cognitive Computing*, *7*(2), 99.

6. Bono, F. M., Radicioni, L., Cinquemani, S., Benedetti, L., Cazzulani, G., Somaschini, C., & Belloli, M. (2023). A deep learning approach to detect failures in bridges based on the coherence of signals. *Future Internet*, *15*(4), 119.
7. Chen, P. Y., Yang, S., & McCann, J. A. (2014). Distributed real-time anomaly detection in networked industrial sensing systems. *IEEE Transactions on Industrial Electronics*, *62*(6), 3832-3842.
8. Cherdo, Y., Miramond, B., Pegatoquet, A., & Vallauri, A. (2023). Unsupervised anomaly detection for cars CAN sensors time series using small recurrent and convolutional neural networks. *Sensors*, *23*(11), 5013.
9. Difallah, D. E., Cudre-Mauroux, P., & McKenna, S. A. (2013). Scalable anomaly detection for smart city infrastructure networks. *IEEE Internet Computing*, *17*(6), 39-47.
10. Du, Y., Li, L. F., Hou, R. R., Wang, X. Y., Tian, W., & Xia, Y. (2022). Convolutional neural network-based data anomaly detection considering class imbalance with limited data. *Smart Struct Syst*, *29*(1), 63-75.
11. Enemosah, A., & Chukwunweike, J. (2022). Next-generation SCADA architectures for enhanced field automation and real-time remote control in oil and gas fields. *Int J Comput Appl Technol Res*, *11*(12), 514-29.
12. Gao, K., Chen, Z. D., Weng, S., Zhu, H. P., & Wu, L. Y. (2022). Detection of multi-type data anomaly for structural health monitoring using pattern recognition neural network. *Smart Struct. Syst*, *29*(1), 129-140.
13. Haque, N. I., Rahman, M. A., & Shahriar, H. (2021, July). Ensemble-based efficient anomaly detection for smart building control systems. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)* (pp. 504-513). IEEE.
14. Islam, J., Talusan, J. P., Bhattacharjee, S., Tiausas, F., Vazirizade, S. M., Dubey, A., ... & Das, S. K. (2022, May). Anomaly based incident detection in large scale smart transportation systems. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)* (pp. 215-224). IEEE.
15. Jamiu, O. A., & Chukwunweike, J. (2023). Developing scalable data pipelines for real-time anomaly detection in industrial IoT sensor networks. *International Journal Of Engineering Technology Research & Management (IJETRM). 2023Dec21*, *7*(12), 497-513.
16. Jiang, H., Ge, E., Wan, C., Li, S., Quek, S. T., Yang, K., ... & Xue, S. (2023, November). Data anomaly detection with automatic feature selection and deep learning. In *Structures* (Vol. 57, p. 105082). Elsevier.
17. Kim, S. Y., & Mukhiddinov, M. (2023). Data anomaly detection for structural health monitoring based on a convolutional neural network. *Sensors*, *23*(20), 8525.
18. Li, J., Chen, W., & Fan, G. (2022). Structural health monitoring data anomaly detection by transformer enhanced densely connected neural networks. *Smart Struct Syst*, *30*(6), 613-626.
19. Li, J., He, H., He, H., Li, L., & Xiang, Y. (2020). An end-to-end framework with multisource monitoring data for bridge health anomaly identification. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1-9.
20. Liu, G., Niu, Y., Zhao, W., Duan, Y., & Shu, J. (2022). Data anomaly detection for structural health monitoring using a combination network of GANomaly and CNN. *Smart Struct Syst*, *29*(1), 53-62.
21. Liu, Y., Wang, H., Zheng, X., & Tian, L. (2023). An efficient framework for unsupervised anomaly detection over edge-assisted internet of things. *ACM Transactions on Sensor Networks*.
22. Lu, Y., Lin, Q., Chi, H., & Chen, J. Y. (2023). Automatic incident detection using edge-cloud collaboration based deep learning scheme for intelligent transportation systems: Lu et al. *Applied Intelligence*, *53*(21), 24864-24875.
23. Moallemi, A., Burrello, A., Brunelli, D., & Benini, L. (2022). Exploring scalable, distributed real-time anomaly detection for bridge health monitoring. *IEEE Internet of Things Journal*, *9*(18), 17660-17674.
24. Mohandas, P. (2023). SAD: Sensor-based anomaly detection system for smart junctions. *IEEE Sensors Journal*, *23*(17), 20368-20378.
25. Nong, X., Luo, X., Lin, S., Ruan, Y., & Ye, X. (2023). Multimodal deep neural network-based sensor data anomaly diagnosis method for structural health monitoring. *Buildings*, *13*(8), 1976.
26. Pan, Q., Bao, Y., & Li, H. (2023). Transfer learning-based data anomaly detection for structural health monitoring. *Structural Health Monitoring*, *22*(5), 3077-3091.
27. Santhosh, K. K., Dogra, D. P., & Roy, P. P. (2020). Anomaly detection in road traffic using visual surveillance: A survey. *Acm Computing Surveys (CSUR)*, *53*(6), 1-26.
28. Shajihan, S. A. V., Wang, S., Zhai, G., & Spencer Jr, B. F. (2022). CNN based data anomaly detection using multi-channel imagery for structural health monitoring. *Smart Struct Syst*, *29*(1), 181-193.
29. Sun, R., Luo, Q., & Chen, Y. (2023). Online transportation network cyber-attack detection based on stationary sensor data. *Transportation research part C: emerging technologies*, *149*, 104058.

30. Tang, Z., Chen, Z., Bao, Y., & Li, H. (2019). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Structural Control and Health Monitoring*, *26*(1), e2296.
31. Verma, R. K., Pattanaik, K. K., Dissanayake, P. B. R., Dammika, A. J., Buddika, H. A. D., & Kaloop, M. R. (2020). Damage detection in bridge structures: An edge computing approach. *arXiv preprint arXiv:2008.06724*.
32. Wilbur, M., Dubey, A., Leão, B., & Bhattacharjee, S. (2019, June). A decentralized approach for real time anomaly detection in transportation networks. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 274-282). IEEE.
33. Ye, X., Wu, P., Liu, A., Zhan, X., Wang, Z., & Zhao, Y. (2023). A deep learning-based method for automatic abnormal data detection: Case study for bridge structural health monitoring. *International Journal of Structural Stability and Dynamics*, *23*(11), 2350131.
34. Zhao, M., Sadhu, A., & Capretz, M. (2022). Multiclass anomaly detection in imbalanced structural health monitoring data using convolutional neural network. *Journal of Infrastructure Preservation and Resilience*, *3*(1), 10.
35. Zheng, Z., Zhou, M., Chen, Y., Huo, M., & Chen, D. (2019). Enabling real-time road anomaly detection via mobile edge computing. *International Journal of Distributed Sensor Networks*, *15*(11), 1550147719891319.