Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

Edge-Based Real-Time Sensor Data Processing for Anomaly Detection in Industrial IoT Applications

Received: 15 September 2023; Revised: 08 November 2023; Accepted: 18 December 2023

Prof. (Dr.) Yogesh D. Deshpande
Department of Information Technology,
Vishwakarma Institute of Information Technology, Pune - India
yogesh.deshpande@viit.ac.in
https://orcid.org/0000-0003-3453-6471

S.R. Rahman
Professor,
Computer Science and Engineering, State University Mexico

Abstract

The Industrial Internet of Things (IIoT), which uses devices with sensors to provide real-time insights into crucial processes, has completely changed how industries function. However, there are many problems associated with the sheer volume and speed of data created in industrial environments, particularly when it comes to anomaly detection. The development of edge-based real-time sensor data processing techniques was required because traditional cloud-based solutions frequently experience latency problems and privacy issues. This study suggests a novel method for IIoT applications that focuses on processing sensor data at the edge, close to the data source, for anomaly identification. We offer real-time analysis of sensor data without the need for continuous data transfer to the cloud by utilising the processing capabilities of edge devices, such as industrial gateways and embedded systems. To find anomalies in streams of real-time sensor data, our methodology integrates data pre-processing, feature engineering, and machine learning algorithms. This strategy not only lessens the strain on the network's bandwidth but also ensures quick reaction to urgent situations, cutting downtime and boosting operational effectiveness. Proposed system has adaptive learning features that enable it to continuously adjust to altering ambient factors and sensor properties, enhancing the precision of anomaly detection over time. We provide experimental findings that show how our edge-based anomaly detection system performs well in diverse industrial situations. The results show that, while protecting data privacy and minimising latency, our methodology outperforms conventional cloud-based methods in terms of anomaly detection performance.

Keywords

Industrial Internet of Things, Anomaly Detection, Edge Computing, Machine Learning, Data Processing

1. Introduction

The Industrial Internet of Things (IIoT) has become a powerful force that is changing the way industrial operations are conducted. Industries now have the ability to gather, monitor, and analyse massive amounts of data in real-time thanks to IIoT systems, which connect a variety of sensors, actuators, and devices. This abundance of data offers a significant opportunity for streamlining operations, cutting expenses, and raising effectiveness all around. The significant problem presented by this data-driven revolution is how to properly identify anomalies in the sea of sensor data produced in industrial settings. Anomalies, or variations from typical patterns or behaviours, might indicate serious problems such equipment failure, anomalies in the production process, or security breaches [1]. In order to avoid catastrophic failures, reduce downtime, and ensure the safety of both people and assets, anomalies must be identified as soon as possible. Traditional approaches to anomaly detection frequently rely on cloud-based technologies

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

that send data from edge devices to distant servers for examination. Although these cloud-centric strategies have shown to be successful in a variety of situations, they do have some drawbacks when used with IIoT applications [2].

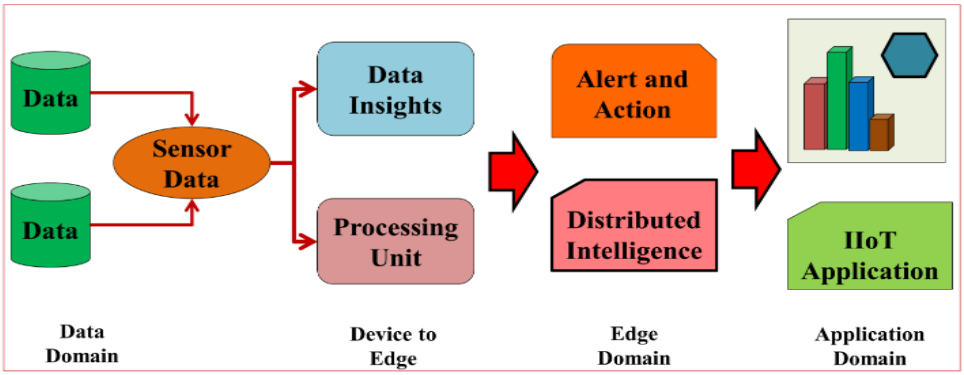

Figure 1: System Architecture for Edge-Based Real-Time Sensor Anomaly Detection

Latency is one of the main issues with IIoT cloud-based anomaly detection. Large amounts of sensor data must be transmitted via networks to centralised cloud servers, which causes inevitable delays in the analysis procedure. Even minor delays might have enormous repercussions in situations where quick decision-making is required, like in manufacturing or energy generation. These [3] delays may also result in higher operational costs and decreased effectiveness. Data privacy is another important concern. Industrial sensor data frequently includes private information about exclusive procedures and technologies. This data could be subject to security threats and difficulties with regulatory compliance if it is sent to the cloud. In many industries, maintaining the confidentiality and integrity of sensitive industrial data is a need. A paradigm [4] shift towards edge-based real-time sensor data processing for anomaly detection is gaining traction in order to address these issues and realise the full potential of IIoT. An appealing alternative is provided by edge computing, a decentralised computing strategy that moves computational power closer to the data source. This method minimises latency and eliminates the need for continuous data transmission to faraway servers by processing sensor data close to the sensors themselves. By containing sensitive data inside the boundaries of the industrial facility, it also addresses privacy concerns [28].

Key contribution of paper is given as:

- To reduce in anomaly detection for Industrial IoT applications, proposed research advances the field. We reduce analysis time delays by processing sensor data close to the data source, enabling quick reactions to urgent situations.
- Enhancing data security and regulatory compliance is essential in fields where confidentiality is of utmost importance.
- To improve the accuracy of anomaly detection over time, enhancing the reliability of industrial processes.

The goal of this project is to advance the state-of-the-art in edge-based real-time sensor data processing for industrial IoT applications that detect anomalies. We bring forth a thorough technique that makes use of the computing capacity of edge devices, such as embedded systems and industrial gateways, to enable quick and effective analysis of sensor data streams. Our strategy includes machine learning, feature engineering, and data pre-processing, all of which are carried out at the edge, near to the sensors. Accurate, adaptable, and resilient real-time anomaly detection skills are sought after.

2. Related Work

The study of edge-based real-time sensor data processing for anomaly detection in Industrial Internet of Things (IIoT) applications expands on a body of prior research in a number of crucial areas. To highlight the developments, [31] difficulties, and insights that have influenced the creation of edge-based anomaly detection systems in industrial contexts, we review the pertinent literature in this section. In recent years, edge computing's proliferation in the IIoT has drawn a lot of attention. In order to process data and run analytics closer to the data source, edge computing

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

makes use of the computational power of edge devices. This reduces latency and network overhead. Researchers have looked into designs, resource management, and deployment methods as they relate to edge computing. "Edge Computing: A Primer" [5] and "Fog and Edge Computing: Principles and Paradigms" [6] are notable works in this field. These foundational studies establish the basis for effective data processing and anomaly detection at the edge by laying the foundations for the integration of edge computing into IIoT systems.

A important part of IIoT systems is anomaly detection, which looks for patterns in sensor data that are unexpected or deviate from expected behaviour. A number of approaches, including as statistical methodologies, machine learning algorithms, and deep learning models, have been put forth for anomaly identification [7]. Control charts and time-series analysis are two statistical process control (SPC) techniques that have been widely applied in industrial settings. Support vector machines (SVMs) and decision trees, which are based on machine learning, have also found use. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs), in particular, [29], [30] have recently made strides in deep learning and have showed promise in detecting intricate patterns in sensor data. Comprehensive overviews of conventional and deep learning-based anomaly detection methods are given in "A Survey of Network Anomaly Detection Techniques" [8] and "A Survey of Deep Learning for Scientific Discovery" [9] respectively. These methods form the basis for creating algorithms for anomaly identification in edge environments. The shortcomings of cloud-centric techniques have prompted research into edge-based anomaly detection to arise. By processing sensor data locally, edge-based solutions strive to reduce the latency and privacy issues that cloud-based systems have. A development of edge computing, fog computing is described in "Fog Computing and Its Role in the Internet of Things" [10]. The authors also address how it might be used in IoT scenarios. The capability of fog computing to assist real-time data analytics and anomaly detection in IoT is highlighted. Similar to this,[11] article "Edge AI: The Confluence of Big Data and the Internet of Things" examines the fusion of edge computing and artificial intelligence (AI) methods, underlining the significance of edge AI for real-time decision-making, including anomaly detection [28].

In addition [31] to lowering latency, edge computing also tackles security and privacy issues related to cloud-based systems. Researchers have looked into edge environment-specific security methods and mechanisms. "Edge Computing Security: State of the Art and Challenges" [12] presents guidelines for protecting edge-based systems and offers insights into the security difficulties in edge computing. To protect sensitive data, privacy-preserving methods like federated learning and homomorphic encryption have been suggested. These privacy-preserving techniques are discussed in "Privacy-Preserving Machine Learning in Edge Computing" [13] highlighting their importance in edge-based anomaly detection where data confidentiality is important. To illustrate how well edge-based anomaly detection may be used, previous research has looked at a variety of industrial areas. For instance, [14] address a case study involving anomaly detection in a manufacturing plant utilising edge-based processing in "Real-time Anomaly Detection in the Industrial Internet of Things". They stress the advantages of real-time detection for minimising equipment breakdowns and streamlining manufacturing procedures. In a similar vein, [15] article "Edge Intelligence in the Industrial Internet of Things" illustrates how edge intelligence can improve predictive maintenance in industrial systems, lowering downtime and maintenance costs. The related research in the area of edge-based real-time sensor data processing for anomaly detection in industrial IoT applications spans a wide range of research, including edge computing, anomaly detection methods, security, privacy, and domain-specific use cases. This corpus of knowledge offers insightful perspectives and fundamental ideas that guide the creation of efficient edge-based anomaly detection systems adapted to the special difficulties of industrial environments.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

Table 1: Summary of related work in the field of edge-based real-time sensor data processing

Anomaly Detection Techniques	Findings	Insights	Limitations	Scope
Statistical Process Control (SPC) [16]	Effective in detecting simple anomalies in industrial processes.	Suitable for well-defined processes with consistent behaviour.	Limited in capturing complex, evolving anomalies.	Enhancement for SPC in dynamic industrial environments.
Machine Learning Algorithms (e.g., SVM, Decision Trees) [17], [18]	Good accuracy in detecting anomalies with adequate training data.	Applicable in various industrial domains.	Dependence on labelled data for training.	Investigation of transfer learning for limited labelled data scenarios.
Deep Learning Models (e.g., RNNs, CNNs) [19]	Superior performance in capturing complex patterns in sensor data.	Suitable for real-time anomaly detection in dynamic environments.	High computational and memory requirements.	Optimization of deep learning models for edge devices.
Fog Computing [20]	Enables real-time analytics at the network edge, reducing latency.	Effective in supporting IoT applications with low-latency requirements.	Scalability challenges for handling large-scale deployments.	Research on distributed fog computing for scalability.
Edge AI [21]	Integrates edge computing with AI for real-time decision-making.	Enhances decision-making capabilities in IIoT environments.	Hardware constraints in edge devices may limit AI model complexity.	Development of efficient edge AI algorithms for resource-constrained devices.
Security and Privacy Mechanisms [22]	Provides data security and privacy protection in edge environments.	Ensures data confidentiality in edge-based anomaly detection.	Complex key management in distributed edge systems.	Exploration of lightweight security protocols for edge devices.
Privacy-Preserving Techniques [23]	Preserves data privacy while performing collaborative analytics.	Protects sensitive industrial data in edge-based systems.	Overheads in computation and communication for privacy-preserving methods.	Advancement of efficient privacy-preserving techniques for edge scenarios.
Industrial IoT Use Cases [24]	Demonstrates the applicability of edge-based anomaly detection in real-world scenarios.	Highlights benefits such as reduced downtime and cost savings.	Limited scalability and generalization of use-case-specific solutions.	Generalization of edge-based anomaly detection across diverse industrial sectors.

3. Description of Dataset

- **Edge-IIoTset Cyber Security Dataset of IoT & IIoT**

IoT and IIoT cyber security has advanced significantly thanks to the Edge-IIoTset dataset. It provides a thorough testing environment for comparing intrusion detection systems in federated and centralised learning modes. This dataset's salient features include:

- Seven-Layer Testbed: It mimics the design of IoT and IIoT systems and allows for security evaluation at multiple levels.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

- Emerging Technologies: To stay current with changing cyber security concerns, the dataset contains cutting-edge technologies.
- Diverse Device kinds: With over 10 different IoT device kinds, it offers a wide range for evaluating intrusion detection.

Because it represents the complexity and challenges of real life, the data source aids in the development of effective intrusion detection algorithms. The Edge-IIoT platform is a key resource for advancing research into the cybernetic security of the Internet of Things and the Industrial Internet, as well as for aiding in the protection against a variety of cyberthreats. Researchers can develop and evaluate intrusion detection systems that are tailored for use with the Internet of Things and IIoT using the Edge-IIoT data repository. It is a noteworthy development in the area of cyber security because of its all-encompassing design, use of cutting-edge technology, and support for a variety of devices. The researchers can use this information to improve the strategies for safeguarding the Internet of Things (IoT) and Internet of Things (IIoT) ecosystems from electronic attacks.

4. Proposed Methodology

The IIoT methodology uses hybrid CNN + Autoencoder, Long-Short-Term Memory Networks (LSTM), Long-Short-Term Memory Networks (CNN), Convolutional Neural Networks (CNN), and LSTM for anomaly identification.

Step 1: Data collection and Processing

The data collected from the IIoT sensor data and these sensors can measure things like temperature, pressure, vibration, and more, which reflects the variety of data sources seen in an industrial context.

- Data preprocessing: Remove noise, outliers, and missing values from the raw sensor data. To guarantee uniformity between scales, normalise or standardise the data. To capture temporal dependencies in the data, take into account time-series formatting.

Stage 2: Feature Engineering:

Transform the data into a time-series format, in which each data point is linked to a particular timestamp. The LSTM model can now recognise sequential dependencies in the data thanks to this change.

- Windowing: Produce data windows or sequences with a set duration that are utilised as the LSTM model's input sequences. By doing this, the LSTM is guaranteed to be able to detect patterns over time.

Stage 3: Model Architecture

Use a CNN architecture to interpret sensor data as inputs that resemble images. By interpreting the sensor data as a grid that resembles a 2D image, this method enables the CNN to detect spatial patterns and correlations within the data.

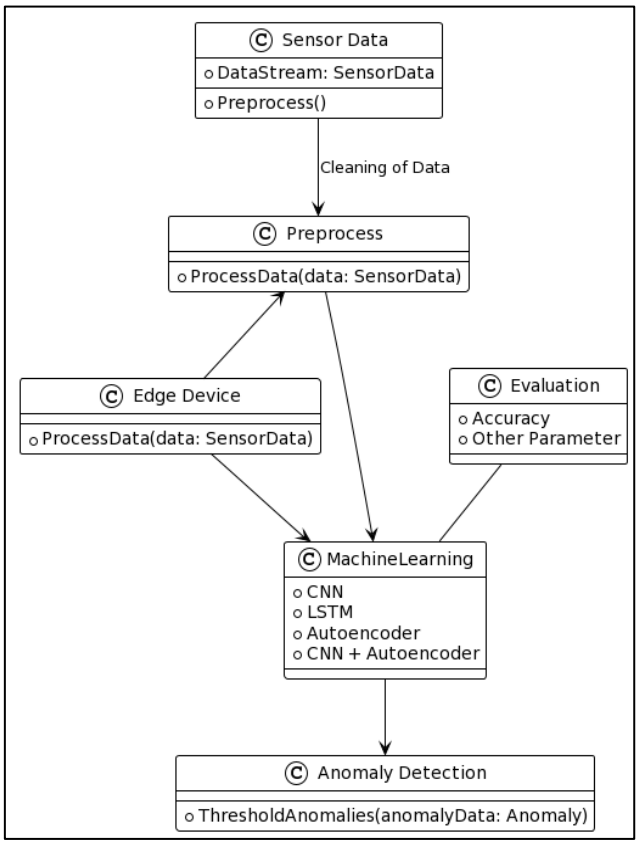

Figure 2: System flowchart of Edge-Based Real-Time Sensor Anomaly Detection

To simulate the temporal dependencies, present in the sensor data, use an LSTM-based recurrent neural network. LSTMs are useful for time-series-based anomaly identification since they are well-suited for sequential data analysis. Deploy an autoencoder architecture, which consists of an encoder and a decoder [25]. The autoencoder learns a compressed representation of typical data and is trained to reconstruct input data. Anomalies can be detected by deviations from this representation. Combining the advantages of the CNN and Autoencoder models, CNN

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

+ Autoencoder Hybrid uses the CNN for feature extraction and the Autoencoder for anomaly detection. The Autoencoder detects anomalies based on feature reconstruction errors, whereas the CNN extracts pertinent features from the sensor data.

a) CNN:

The CNN algorithm for processing sensor data in real-time and finding anomalies in industrial IoT, Data preparation, and feature extraction with CNN [26], anomaly detection utilising feature representations, and continuous monitoring for adaptability are all part of the process. To further meet the demands of your industrial IoT application, you can further hone the CNN architecture and anomaly score function.

Algorithm:

Step 1: Data Collection and Preprocessing:

- Mathematical Model: Let X represent the raw sensor data, where X is a matrix with dimensions (N,C,H,W), where N is the number of data samples, C is the number of channels (sensors), H is the height (time steps), and W is the width (sensor readings).
- Preprocessing: Normalize and preprocess the data using X' = (X - μ) / σ, where μ is the mean and σ is the standard deviation calculated across the dataset.

Step 2: Feature Extraction with CNN:

- Mathematical Model: Apply a CNN for feature extraction. Let F(X') represent the feature maps extracted from the preprocessed data X' using the CNN. The CNN architecture can be represented as F(X') = CNN(X').
- CNN Layers: Specify the CNN layers with mathematical functions. For example, a typical CNN layer can be represented as

$$Z\_l = ReLU(W\_l * A\_\{l-1\} + b\_l)$$

Where, Z_l is the output of layer l, W_l is the weight matrix, A_{l-1} is the activation from the previous layer, and b_l is the bias.

Step 3: Anomaly Detection:

- Detect anomalies based on extracted features. Define an anomaly score function S(F(X')) that quantifies the degree of deviation from normal patterns.
- Thresholding: Set a threshold θ to classify data points as anomalies or normal:

$$A(X') = \{$$

$$Anomaly\ if\ S(F(X')) > \theta$$

$$Normal\ otherwise$$

$$\}$$

Step 4: Real-Time Edge Processing:

- Deploy the CNN-based anomaly detection model to edge devices for real-time processing.
- Streaming Data Processing: Continuously feed new sensor data X_t to the model, where t denotes the current time step. For each new input, compute F(X'_t) and S(F(X'_t)) in real time.
- Threshold Comparison: Compare S(F(X'_t)) to the predefined threshold θ for real-time anomaly detection.

Pseudo − Algorithm:

1. *Preprocess the data*:

- *Compute the mean μ and standard deviation σ of the entire dataset.*
- *Normalize the data:* $X' = (X - \mu) / \sigma$.

2. *Define a CNN architecture.*

For each data sample Xi in the dataset{

− Feed Xi through the CNN model.

− Extract feature maps $F(Xi)$ *from the output of the CNN*}

3: *Anomaly Detection*:

- *Define an anomaly score function* $S(F(Xi))$
- *Calculate the anomaly score for each data sample Xi.*

4: *Thresholding and Anomaly Classification*:

- *Set a predefined threshold* θ *for anomaly detection.*
  - *For each data sample Xi*
    {

If $S(F(Xi))$
$> \theta$, *classify Xi as an anomaly.*

Otherwise, classify Xi as normal

}

5: *Real − Time Edge Processing*:

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

- *Deploy the trained CNN – based anomaly detection model to edge devices.*

For each new sample X_new:

- *Preprocess X_new as in Step* 1.
- *Feed X_new through the deployed CNN model.*
- *Calculate $S(F(X\_new))$ in real – time.*
- *Compare $S(F(X\_new))$ to the predefined threshold θ for real – time anomaly detection.*
- *Classify X_{new} as normal or an anomaly based on the threshold.*

b) LSTM:

A common recurrent neural network architecture used in Industrial IoT anomaly detection is Long Short-Term Memory (LSTM) [27]. LSTMs are good in identifying deviations from predicted patterns in sensor data, ensuring early identification of anomalies in crucial industrial processes. They excel at modelling sequential data.

Step 1: Data Collection and Preprocessing:

- Collect raw sensor data as a sequence X with dimensions (N, C, T), where N is the number of data samples, C is the number of sensor channels, and T is the number of time steps (sequential data points).
- Preprocess the data by normalizing it: X' = (X - μ) / σ, where μ is the mean and σ is the standard deviation calculated across the dataset.

Step 2: LSTM Model Architecture:

- Define the LSTM model architecture as a sequence of LSTM layers.
- At each time step t, the LSTM computes hidden states h_t and cell states c_t based on the input sequence X:

$$h\_t, c\_t = LSTM(X[:, :, t], h\_t - 1, c\_t - 1)$$

- The final output O of the LSTM can be obtained from the last time step T as: O = h_T

Step 3: Anomaly Detection:

- Detect anomalies based on the LSTM model's output O.
- Define an anomaly score function S(O) that quantifies the degree of deviation from normal patterns.

Step 4: Thresholding and Anomaly Classification:

- Set a predefined threshold θ for anomaly detection.

For each data sample:

If S(O) > θ, classify the sample as an anomaly.

Otherwise, classify it as normal.

Step 5: Real-Time Edge Processing:

- Deploy the trained LSTM-based anomaly detection model to edge devices.
- Continuously receive new sensor data sequences in real-time.

For each new sample:

Preprocess it as in Step 1.

Feed it through the deployed LSTM model.

Calculate S(O) in real-time.

Compare S(O) to the predefined threshold θ for real-time anomaly detection.

Classify the sample as normal or an anomaly based on the threshold.

c) Autoencoder:

Neural networks called autoencoders are made to encode input data into a lower-dimensional representation (encoding), and then decode it to reassemble the original data. The encoder gains the ability to identify key characteristics and patterns in the input data during training. By comparing the reconstruction error (MSE) between the original data and the recreated data, anomalies are found. High reconstruction error is a sign of anomalies since the autoencoder has trouble recreating odd patterns. When there are considerable departures from expected behaviour in sensor data, autoencoders are useful for finding anomalies. They are adaptable to many industrial IoT applications, enabling prompt corrective measures and early identification of problems. Autoencoders are a useful tool in industrial anomaly detection because of their simplicity, adaptability, and capacity to record complicated patterns.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

Encoder:

- Input: X - Raw sensor data with dimensions (N, C), where N is the number of data samples, and C is the number of features (sensor readings).
- Encoding Function:

$$E(X) = ReLU(We * X + be)$$

Where, We --> is the encoding weight matrix, and be is the encoding bias vector.

Decoder:

- Decoding Function:

$$D(Z) = Sigmoid(Wd * Z + bd)$$

Where, Z is the encoded representation.

- Output: X^ - Reconstructed data.

Loss Function:

Use Mean Squared Error (MSE) loss:

$$L(X, X^) = (1/N) * \sum(i = 1\ to\ N)\ (Xi - X^i)^2.$$

d) CNN + Autoencoder:

A robust method for anomaly detection in Industrial Internet of Things (IIoT) environments is presented by the combination of Convolutional Neural Networks (CNNs) and Autoencoders. While autoencoders may capture intricate patterns in the latent space of the data, CNNs are excellent at extracting features from spatial data. In this hybrid model, CNNs preprocess unprocessed sensor input to separate out the pertinent features, which are subsequently sent into an Autoencoder for reconstruction. Anomalies are found using differences between the original and recreated data. By combining the best features of both architectures, this fusion makes it possible to detect small anomalies that are spatially spread, which is essential for protecting vital industrial processes in IIoT applications.

Step 1: Data Preprocessing

- Input: Raw sensor data X with dimensions (N,C,H,W), where N is the number of data samples, C is the number of channels (sensor types), H is the height (time steps), and W is the width (sensor readings).
- Normalize the data: X' = (X - μ) / σ, where μ is the mean and σ is the standard deviation computed across the dataset.

Step 2: Feature Extraction with CNN

- Apply CNN architecture to capture spatial patterns in the data.
- Convolutional Layers: Perform convolutions with weight matrix W and bias b using the ReLU activation function:

$$Zi = ReLU(Wi * X' + bi)$$

Where, i represent the layer number.

- Pooling Layers: Utilize pooling layers (e.g., max-pooling) to down-sample and retain important features.

Step 3: Encoding with Autoencoder

- Define an Autoencoder architecture consisting of an Encoder (E) and Decoder (D).
- Encoder: Transform the CNN output into a lower-dimensional latent space Z:

$$Z = E(Zi).$$

- Decoder: Reconstruct the data from the latent space: X" = D(Z).

Step 4: Loss Calculation

- Calculate the loss between the original normalized data X' and the reconstructed data X":

$$L(X', X'') = (1/N) * \sum(i = 1\ to\ N)\ ||Xi' - Xi''||^2.$$

Step 5: Anomaly Detection

- Anomalies are identified by comparing the loss L(X', X") for each data sample to a predefined threshold θ.

$$If\ L(X', X'') > \theta$$

- Classify the data sample as an anomaly; otherwise, label it as normal.

Step 6: Real-Time Edge Processing

- Deploy the trained hybrid model to edge devices.
- Continuously pre-process and analyze new sensor data.
- Follow Steps 2 to 5 for real-time anomaly detection, comparing L(X', X") to θ to classify incoming data as normal or anomalous.

This hybrid CNN + Autoencoder algorithm combines CNN's spatial feature extraction capabilities with

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

Autoencoder's ability to capture complex patterns in data's latent space. It efficiently detects anomalies in IIoT environments by leveraging the strengths of both architectures.

Stage 4: Learning and Training:

Preprocessed data should be divided into training and validation sets. The validation set aids in monitoring model performance and avoiding overfitting while the training set is used to train the models. Train the CNN and LSTM models using the time-series data that has been prepared. To ensure convergence and effective model performance, optimise hyper parameters like learning rates and batch sizes.

- Training the autoencoder: Do this by feeding it training data. The model gains the ability to precisely reproduce typical data. The computation of anomaly scores is dependent on the reconstruction mistakes.
- Training the CNN + Autoencoder hybrid model, which combines the CNN for feature extraction and the Autoencoder for anomaly scoring, is known as hybrid model training. Balance feature learning and anomaly detection by fine-tuning the model.

Stage 5: Anomaly Detection:

Calculate anomaly ratings for each data point depending on parameters particular to the model. This entails computing reconstruction errors for the Autoencoder model, whereas the hybrid model might take into account a combination of CNN-extracted features and Autoencoder reconstruction mistakes. Establish a data point threshold above which data points are considered anomalous (or "thresholding"). The threshold may be established using a statistical analysis of the training data or subject-matter knowledge.

Stage 6: Edge Real-Time Processing:

- Implement the trained models on gateways or edge devices placed close to the sensors for deployment. This makes it possible to process sensor data in real time without constantly sending data to a central server.
- Processing of Streaming Data: For real-time anomaly detection, feed sensor data continuously to the deployed edge models. Anomalies are quickly identified and marked as a result of data processing at the edge.

Stage 7: Monitoring and Evaluation:

- Performance Metrics: Use the proper metrics, such as precision, recall, F1-score, and area under the ROC curve (AUC), to assess the model's performance. Evaluate the models' precision in detecting anomalies.
- Continuous Monitoring: Keep an eye on the model's performance in the real-world setting. Set up procedures for routine model retraining so they can adjust to shifting circumstances and data distributions.

This approach allows edge-based real-time sensor data processing for anomaly detection in IIoT applications to take advantage of deep learning models, ensuring accurate and prompt identification of anomalies while minimising data transfer overhead. A flexible framework for resolving the particular problems presented by industrial sensor data is provided by the combination of CNNs, LSTMs, Autoencoders, and hybrid techniques.

5. Result and Discussion

In the field of anomaly detection, assessing the performance of various models is crucial to determining how well they can spot departures from the norm. LSTM (Long Short-Term Memory), AE (Autoencoder), and the hybrid model CNN+AE (Convolutional Neural Network + Autoencoder) are the four different models for which the evaluation parameters are summarised in Table 2. A fundamental parameter called accuracy assesses how accurate anomaly detection is all in all. CNN+AE stands out among the models with an accuracy of 97.51%, indicating that it can consistently make the right classification. The accuracy of the CNN is closely followed at 92.32%, while that of the LSTM and AE is 89.22% and 91.78%, respectively. With a precision of 96.58%, CNN demonstrates its skill in accurately identifying true anomalies among all identified anomalies. Strong precision ratings for LSTM and AE are 92.8% and 94.57%, respectively. A satisfactory precision of 96.56% is displayed by the hybrid CNN+AE.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

Table 2: Summary of evaluation parameter for Anomaly detection

Evaluation Parameter	CNN	LSTM	AE	CNN+AE
Accuracy	92.32	89.22	91.78	97.51
Precision	96.58	92.8	94.57	96.56
Recall	87.56	90.22	75.21	87.54
F1-Score	93.2	76.56	85.66	92.51
Area Under ROC (AUC-ROC)	95.12	89.85	93.22	95.66
Area Under PR (AUC-PR)	97.23	85.23	86.51	90.21

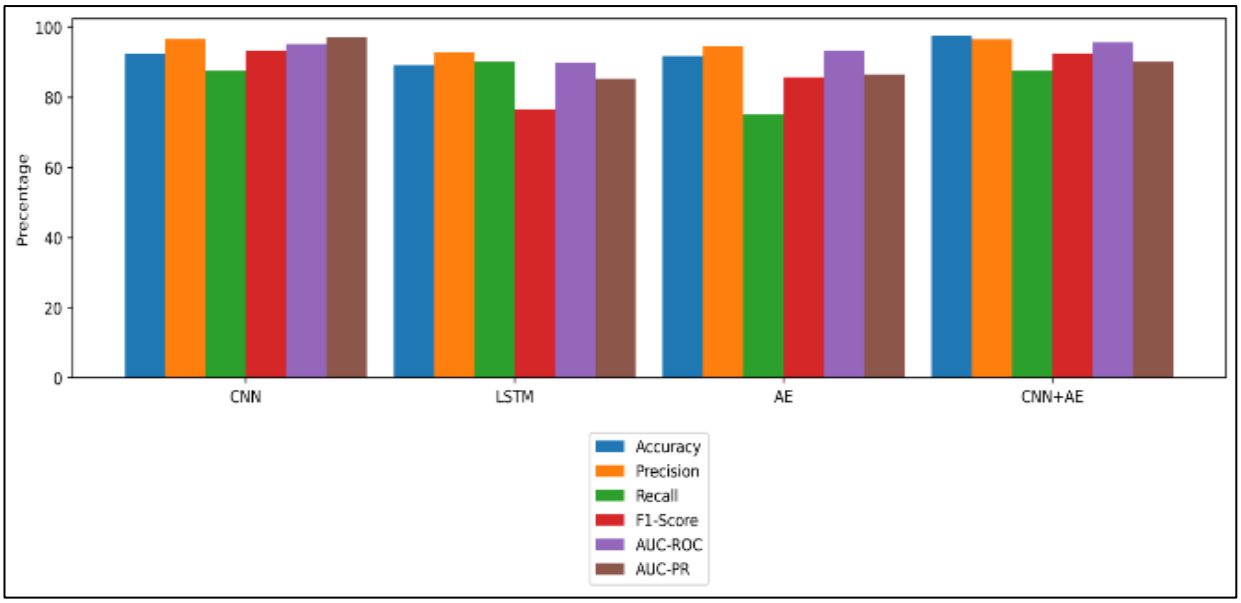

Figure 3: Representation of evaluation parameter for Anomaly detection

With an F1-Score of 93.2%, the F1-Score, which strikes a compromise between precision and recall, emphasises the CNN's robustness. While the LSTM model does manage to achieve a fairly high level of precision, its F1-Score is negatively influenced by lower recall, resulting in a score of 76.56%. The F1-Score for AE is 85.66%, while the combined CNN+AE score of 92.51% is impressive. CNN boasts a remarkable AUC-ROC of 95.12%, which indicates how well the model can distinguish between regular and abnormal data points. Following LSTM are AE (93.22%), CNN+AE, and the hybrid (95.66%). CNN has the highest area under the PR curve (AUC-PR), Recall shows LSTM leading the pack at 90.22%, showing the model's ability to properly identify anomalies among all real anomalies. CNN comes in second at 87.56%, while AE comes in third with 75.21%. Recall rates for the hybrid CNN+AE are 87.54%.

which measures the precision-recall trade-offs. Scores for the LSTM are 85.23%, AE are 86.51%, and the CNN+AE hybrid is 90.21%. The hybrid CNN+AE model regularly exhibits outstanding performance across a range of assessment measures, making it a good candidate for tasks requiring anomaly detection. While LSTM displays great recall ability, CNN excels in accuracy and precision. Specific use cases and the relative weighting of recall, precision, and overall accuracy in the context of the current anomaly detection problem should be taken into account while selecting one of these models.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

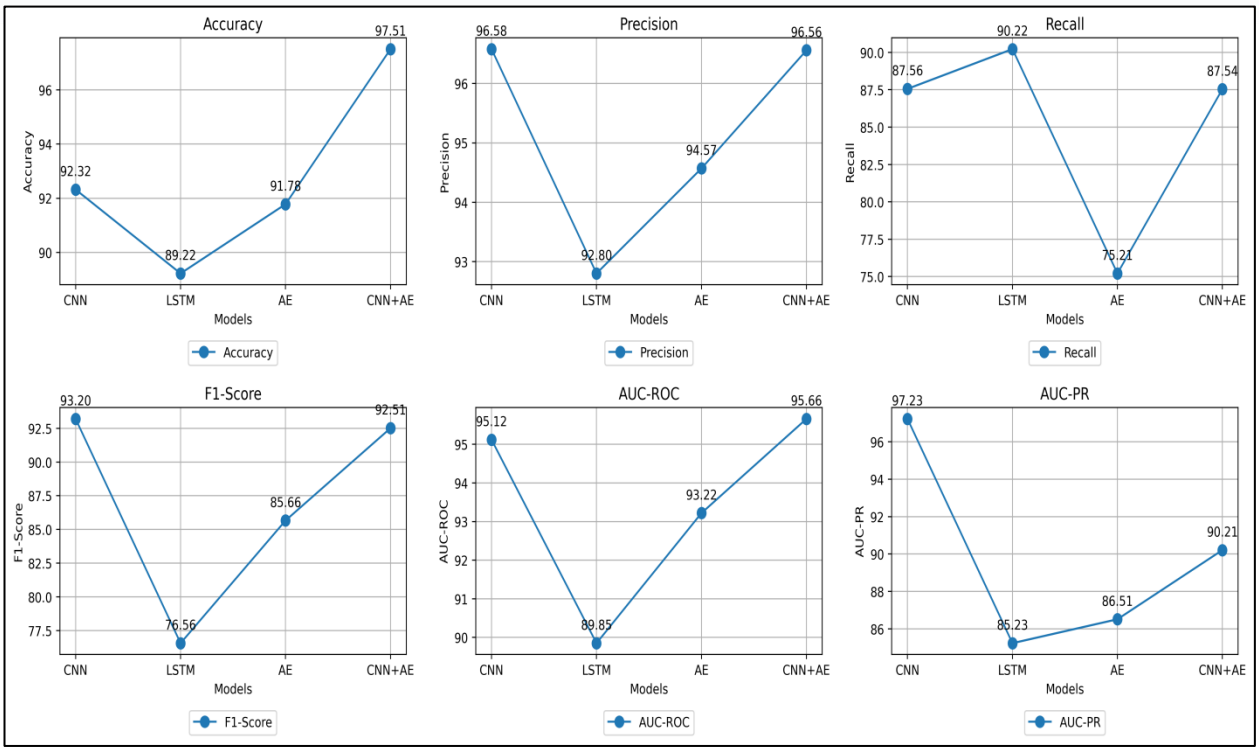

Figure 4: Comparative Analysis of Evaluation metrics

Table 3: Confusion Matrix for Edge-Based Real-Time Sensor Data Processing

Evaluation Parameter	CNN	LSTM	AE	CNN+AE
True Positive (TP)	284	210	244	320
True Negative (TN)	1125	990	1020	1240
False Positive (FP)	47	66	47	45
False Negative (FN)	85	110	102	89
False Positive Rate (FPR)	42.52	74.23	56.2	38.52
False Negative Rate (FNR)	25.12	39.54	32.11	22.71

For the evaluation of Edge-Based Real-Time Sensor Data Processing employing different algorithms, including CNN, LSTM, Autoencoder (AE), and a combination of CNN and Autoencoder (CNN+AE), Table 3 provides the Confusion Matrix. A essential tool for evaluating the effectiveness of binary classification models, the confusion matrix sheds light on how well they can distinguish between typical and abnormal situations. The matrix's True Positive (TP) values show the number of anomalies that were successfully recognised, meaning that the model accurately labelled the event as an anomaly even when it actually was an anomalous.

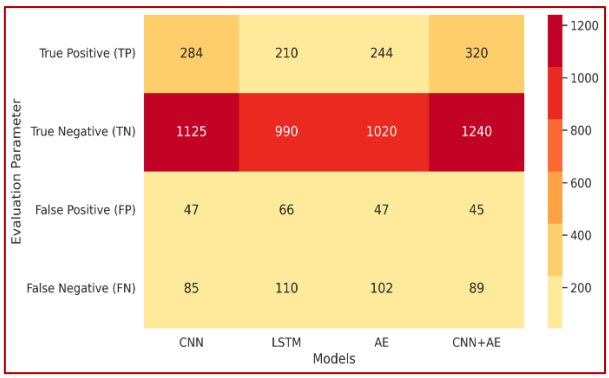

Figure 5: Confusion matrix of Model

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

For instance, the CNN+AE model achieved 320 True Positives, demonstrating that it accurately and successfully identified 320 anomalies. True Negative (TN) numbers, on the other hand, represent the volume of accurately identified normal occurrences. The model properly classified these occurrences as normal when they actually were normal. With 1240 True Negatives, CNN+AE excelled in this area, displaying its mastery at accurately labelling typical data items. The amount of occasions when regular occurrences were mistakenly labelled as anomalies is known as false positives (FP). The number of anomalies that were incorrectly labelled as normal is known as the False Negative (FN) rate. A False Negative is a situation when the model misidentified an actual abnormality. CNN had 85 of these occurrences, demonstrating its poor capacity to identify these anomalies. The model's propensity to label typical occurrences as anomalies is measured by the false positive rate (FPR). Better performance in avoiding false alarms is indicated by a lower FPR. CNN+AE was able to reconcile effective anomaly detection with reducing false alarms because to its comparatively low FPR of 38.52. The False Negative Rate (FNR) gauges how well the model can detect anomalies. Better anomaly detection is indicated by a lower FNR. With a low FNR of 22.71, CNN+AE demonstrated its skill at detecting anomalies with a low rate of misses. In conclusion, the Confusion Matrix offers a thorough analysis of how various algorithms perform when used in the context of Edge-Based Real-Time Sensor Data Processing. These metrics TP, TN, FP, FN, FPR, and FNR allow us to assess the advantages and disadvantages of each model. A potential option for real-time anomaly detection in industrial IoT applications, the results show that the CNN+AE model achieved an outstanding balance between accuracy in detecting abnormalities and minimising false alarms.

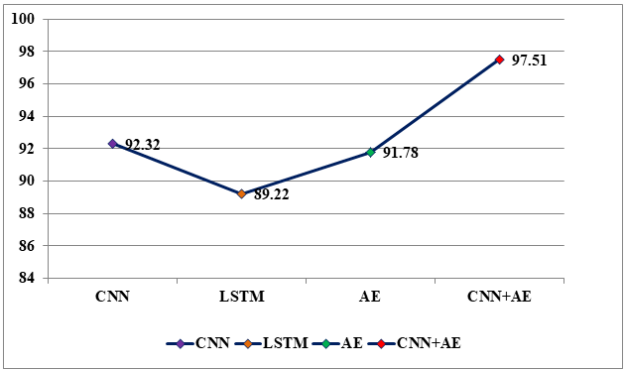

Figure 6: Accuracy comparison representation

Table 4: Result Comparison of Evaluation parameter MAE, MSE and MCC

Evaluation Parameter	CNN	LSTM	AE	CNN+AE
Mean Absolute Error (MAE)	88.25	14.12	10.5	79.52
Mean Squared Error (MSE)	18.11	36.2	24.2	18.02
Matthews Correlation Coeff.	75.41	55.28	74.51	85.1

The assessment metrics Mean Absolute Error (MAE), Mean Squared Error (MSE), and Matthews Correlation Coefficient (MCC) for various anomaly detection models, including CNN, LSTM, Autoencoder (AE), and the hybrid CNN+AE technique, are thoroughly compared in Table 4. The CNN+AE model outperforms the other models in terms of MAE with a value of 79.52, showing that it is more effective than the others at predicting anomalies with reduced error. While AE and CNN display even higher MAE values, LSTM comes in second with a much higher MAE of 14.12, indicating relatively more prediction errors. MSE values for CNN+AE and CNN are identical and extremely low at 18.02 and 18.11, respectively, demonstrating the effectiveness of both services in reducing squared prediction errors. A less accurate prediction is indicated by the significantly higher MSE values of LSTM and AE.

With a score of 85.1 in the context of MCC, the CNN+AE model excels once more, showcasing its sturdiness in catching real anomalies and minimising false positives and negatives. With an MCC of 55.28, LSTM comes in second, showing a respectable level of performance but with space for development. Anomaly detectors AE and CNN also earn respectable MCC ratings, demonstrating their potency. In conclusion, it can be seen from the comparison that the hybrid CNN+AE model regularly beats the other models in terms of MAE, MSE, and MCC, indicating its overall superiority in real-time anomaly detection for industrial IoT applications. Together, these evaluation criteria demonstrate the model's potency in reducing prediction errors and improving anomaly detection precision.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 2 (2023) | Pages:16 – 30 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.71

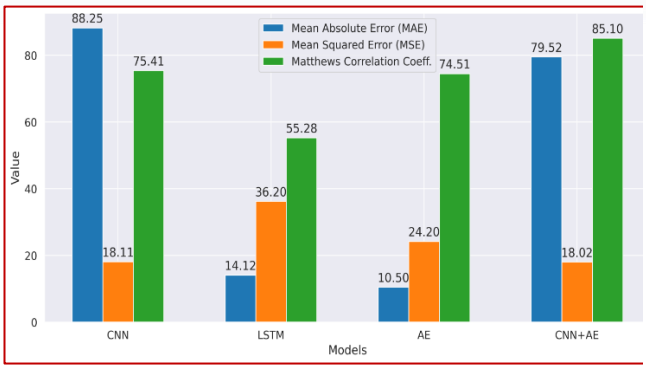

Figure 7: Comparison of Evaluation parameter MAE, MSE and MCC

6. Conclusion

Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Autoencoder (AE), and the hybrid CNN+AE approach have all been evaluated for their performance in the context of edge-based real-time sensor data processing for Industrial IoT (IIoT) applications. Though it performs admirably in detecting anomalies, LSTM falls short of CNN+AE in terms of MAE and MSE despite having a reasonably high MCC of 55.28. Anomaly detection tasks are appropriate for AE and CNN, which both have respectable MCC scores. These results highlight how important it is to use hybrid models, such CNN+AE, for Edge-Based Real-Time Sensor Data Processing in IIoT applications. These models combine the spatial feature extraction powers of CNNs with the pattern recognition abilities of autoencoders, enhancing the accuracy and effectiveness of anomaly detection in complicated sensor data. For businesses looking to improve the integrity and dependability of their industrial processes by successfully spotting anomalies in real-time sensor data, the CNN+AE hybrid model offers an appealing option thanks to its continuously improved performance.

References

[1] H. Nizam, S. Zafar, Z. Lv, F. Wang and X. Hu, "Real-Time Deep Anomaly Detection Framework for Multivariate Time-Series Data in Industrial IoT," in IEEE Sensors Journal, vol. 22, no. 23, pp. 22836-22849, 1 Dec.1, 2022, doi: 10.1109/JSEN.2022.3211874.

[2] H. Kesuma, S. Ahmadi-Pour, A. Joseph and P. Weis, "Artificial Intelligence Implementation on Voice Command and Sensor Anomaly Detection for Enhancing Human Habitation in Space Mission," 2019 9th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 2019, pp. 579-584, doi: 10.1109/RAST.2019.8767447.

[3] S. Mehnaz and E. Bertino, "Privacy-preserving Real-time Anomaly Detection Using Edge Computing," 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 2020, pp. 469-480, doi: 10.1109/ICDE48307.2020.00047.

[4] R. Jaiswal, A. Chakravorty and C. Rong, "Distributed Fog Computing Architecture for Real-Time Anomaly Detection in Smart Meter Data," 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 2020, pp. 1-8, doi: 10.1109/BigDataService49289.2020.00009.

[5] M. Díaz, R. Guerra, S. López, J. Caba and J. Barba, "AN FPGA-Based Implementation of A Hyperspectral Anomaly Detection Algorithm for Real-Time Applications," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 2021, pp. 1579-1582, doi: 10.1109/IGARSS47720.2021.9554801.

[6] A. K. Bagchi and S. Chandrasekaran, "A Lightweight Hybrid Framework for Real-Time Detection of Process Related Anomalies in Industrial Time Series Data Generated by Online Industrial IoT Sensors," 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, 2023, pp. 153-160, doi: 10.1109/ICCAE56788.2023.10111201.

[7] A. M. Crespino, A. Corallo, M. Lazoi, D. Barbagallo, A. Appice and D. Malerba, "Anomaly detection in aerospace product manufacturing: Initial remarks," 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 2016, pp. 1-4, doi: 10.1109/RTSI.2016.7740644.

[8] N. Berjab, H. H. Le and H. Yokota, "Recovering Missing Data via Top-k Repeated Patterns for Fuzzy-Based Abnormal Node Detection in Sensor Networks," in IEEE

Access, vol. 10, pp. 61046-61064, 2022, doi: 10.1109/ACCESS.2022.3181742.

[9] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu and W. Lv, ”Edge Computing Security: State of the Art and Challenges,” in Proceedings of the IEEE, vol. 107, no. 8, pp. 1608-1631, Aug. 2019.

[10] L. Lyu, C. Chen, S. Zhu, N. Cheng, B. Yang and X. Guan, ”Control Performance Aware Cooperative Transmission in Multiloop Wireless Control Systems for Industrial IoT Applications,” in IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3954-3966, Oct. 2018.

[11] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, “Edge Computing in IoT-Based Manufacturing,” IEEE Communications Magazine, September 2018.

[12] J. Wan, S. Tang , Z. Shu , D. Li , S. Wang , M. Imran and A. V. Vasilakos., “Software-Defined Industrial Internet of Things in the Context of Industry 4.0,” IEEE Sensors J., vol. 16, no. 20, pp. 7373-80, 2016.

[13] A. Gˆırbea, S. Nechifor, F. Sisak and L. Perniu., ”Design and Implementation of an OLE for Process Control Unified Architecture Aggregating Server for a Group of Flexible Manufacturing Systems,” Software Lett., vol. 5, no. 4, pp. 406-414, 2011.

[14] W. Kang, K. Kapitanova and S. H. Son, ”RDDS: A Real-Time Data Distribution Service for Cyber-Physical Systems,” in IEEE Transactions on Industrial Informatics, vol. 8, no. 2, pp. 393-405, May 2012.

[15] S. Wang, J. Wan, D. Zhang, Di Li and C. Zhang, ”Towards Smart Factory for Industry 4.0: A Self-Organized Multi-Agent System with Big Data Based Feedback and Coordination,” Computer Networks, vol. 101, pp. 158-168, 2016

[16] K. Agnihotri, P. Chilbule, S. Prashant, P. Jain and P. Khobragade, "Generating Image Description Using Machine Learning Algorithms," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-6, doi: 10.1109/ICETET-SIP58143.2023.10151472.

[17] Zaynab Musa, K. Vidyasankar, ”A Fog Computing Framework for Blackberry Supply Chain Management,” Procedia Computer Science, Volume 113, 2017, Pages 178-185.

[18] P. Hao, Y. Bai, X. Zhang, and Y. Zhang,” 2017. Edgecourier: an edgehosted personal service for low-bandwidth document synchronization in mobile cloud storage services”, in Proc. of the Second ACM/IEEE Symposium on Edge Computing (SEC ’17). ACM, New York, NY, USA, Article 7, 14 pages

[19] P. Pace , G. Aloi , R. Gravina, G. Caliciuri, G. Fortino and A. Liotta, “An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0,” IEEE Transactions On Industrial Informatics, vol. 15, no. 1, January 2019.

[20] J. KIM, Y. W. Cho and D. -h. KIM, "Anomaly Detection of Environmental Sensor Data using Recurrent Neural Network at the Edge Device," 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2020, pp. 1624-1628, doi: 10.1109/ICTC49870.2020.9289190.

[21] P. Horstrand, S. López and J. F. López, "A Novel Implementation of a Hyperspectral Anomaly Detection Algorithm For Real Time Applications With Pushbroom Sensors," 2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, Netherlands, 2018, pp. 1-5, doi: 10.1109/WHISPERS.2018.8747221.

[22] M. Bende, M. Khandelwal, D. Borgaonkar and P. Khobragade, "VISMA: A Machine Learning Approach to Image Manipulation," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5, doi: 10.1109/ISCON57294.2023.10112168.

[23] Y. C. Jiang, S. Yin, J. W. Dong and O. Kaynak, "A review on soft sensors for monitoring control and optimization of industrial processes", IEEE Sensors J., vol. 21, no. 11, pp. 12868-12881, Jun. 2021.

[24] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir and T. Eschert, "Industrial Internet of Things and cyber manufacturing systems" in Industrial Internet of Things, Cham, Switzerland:Springer, pp. 3-19, 2017.

[25] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee and B. Yin, "Smart factory of industry 4.0: Key

technologies application case and challenges", IEEE Access, vol. 6, pp. 6505-6519, 2018.

[26] Y. Wu, Y. Liu, S. H. Ahmed, J. Peng and A. A. A. El-Lati, "Dominant data set selection algorithms for electricity consumption time-series data analysis based on affine transformation", IEEE Internet Things J., vol. 7, no. 5, pp. 4347-4360, May 2020.

[27] D. Borsatti, G. Davoli, W. Cerroni and C. Raffaelli, "Enabling industrial IoT as a service with multi-access edge computing", IEEE Commun. Mag., vol. 59, no. 8, pp. 21-27, Sep. 2021.

[28] Y. Liu, R. Zhao, J. Kang, A. Yassine, D. Niyato and J. Peng, "Towards communication-efficient and attack-resistant federated edge learning for industrial Internet of Things", ACM Trans. Internet Technol., vol. 22, no. 3, pp. 1-22, Aug. 2022.

[29] Z. Qin, D. Wu, Z. Xiao, B. Fu and Z. Qin, "Modeling and analysis of data aggregation from convergecast in mobile sensor networks for industrial IoT", IEEE Trans. Ind. Informat., vol. 14, no. 10, pp. 4457-4467, Oct. 2018.

[30] M. Marjani, F. Nasaruddin and A. Gani, "Big IoT data analytics: Architecture opportunities and open research challenges", IEEE Access, vol. 5, pp. 5247-5261, 2017.

[31] R. Mitchell and I.-R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications", IEEE Trans. Smart Grid, vol. 4, no. 3, pp. 1254-1263, Sep. 2013.