Gayathri and Surendran
Journal of Cloud Computing (2024) 13:49
https://doi.org/10.1186/s13677-024-00595-y

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Unified ensemble federated learning with cloud computing for online anomaly detection in energy-efficient wireless sensor networks

S. Gayathri[1*] and D. Surendran[2]

Abstract

Anomaly detection in Wireless Sensor Networks (WSNs) is critical for their reliable and secure operation. Optimizing resource efficiency is crucial for reducing energy consumption. Two new algorithms developed for anomaly detection in WSNs—Ensemble Federated Learning (EFL) with Cloud Integration and Online Anomaly Detection with Energy-Efficient Techniques (OAD-EE) with Cloud-based Model Aggregation. EFL with Cloud Integration uses ensemble methods and federated learning to enhance detection accuracy and data privacy. OAD-EE with Cloud-based Model Aggregation uses online learning and energy-efficient techniques to conserve energy on resource-constrained sensor nodes. By combining EFL and OAD-EE, a comprehensive and efficient framework for anomaly detection in WSNs can be created. Experimental results show that EFL with Cloud Integration achieves the highest detection accuracy, while OAD-EE with Cloud-based Model Aggregation has the lowest energy consumption and fastest detection time among all algorithms, making it suitable for real-time applications. The unified algorithm contributes to the system's overall efficiency, scalability, and real-time response. By integrating cloud computing, this algorithm opens new avenues for advanced WSN applications. These promising approaches for anomaly detection in resource constrained and large-scale WSNs are beneficial for industrial applications.

Keywords Wireless sensor networks, Online anomaly detection, Energy efficiency, Federated learning, Machine learning, Cloud computing

Introduction

Wireless Sensor Networks (WSNs) are a crucial technology used in a variety of fields, such as environmental monitoring, industrial automation, healthcare, and smart cities. These are networks composed of spatially distributed autonomous sensors to monitor physical or environmental conditions. Several small sensor nodes collect data such as temperature, humidity, motion, and send it to a central base station for processing. However, because WSNs are open and distributed, they are vulnerable to security threats, such as anomalies and malicious attacks. To ensure the reliability and security of WSNs, anomaly detection is essential, as researched by Liu et al. in [1]. Detecting unusual behavior or unexpected events in sensor data can help identify potential faults, intrusions, or environmental changes, enabling timely responses and preventive actions. Traditional rule-based and statistical anomaly detection methods are not very effective in dynamic and complex WSN environments.

*Correspondence:
S. Gayathri
sgayathriphdit@gmail.com
[1] Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu 638401, India
[2] Department of Information Technology, Karpagam College of Engineering Othakkal Mandapam, Coimbatore, Tamil Nadu 641032, India

Therefore, more advanced and adaptive anomaly detection techniques are necessary to address the changing challenges in WSNs.

Background and Motivation

Machine Learning (ML) has proven to be highly effective in improving anomaly detection capabilities in various domains in recent years as researched by Mahesh et. al in [2]. The ability of ML algorithms to learn patterns and relationships from data makes them ideal for detecting anomalies in WSNs. However, the deployment of ML models directly on resource-constrained sensor nodes presents challenges due to limited computing power, memory, and energy constraints. Traditional anomaly detection approaches may not be well-suited for the dynamic and resource-limited nature of WSN environments.

Moreover, in large-scale WSN deployments, data may be spread across multiple sensor nodes, making centralized data processing unfeasible and raising concerns about privacy and communication overhead. The integration of cloud computing in WSNs offers a promising solution to address these challenges. Cloud computing provides elastic and scalable resources that can augment the computational capabilities of resource-constrained sensor nodes. By leveraging cloud resources, ML models can be trained and aggregated efficiently, enabling collaborative model training without compromising data privacy.

Federated Learning (FL) is a promising approach that complements cloud integration in WSNs [3]. Deng et al. [4] introduces an intelligent trusted and secure edge computing (ITEC) system for IoT malware detection, achieving up to 98.52% accuracy. Lu et al. [5] devises a Truthful incEntive mechAnism (TEA) to encourage participation in vertical federated learning, outperforming existing methods in ensuring truthfulness and maximizing social utility. Lastly, a novel adaptive blockchain-enabled FL framework for Intelligent Transportation Systems (ITS) optimizes decentralized vehicular data flows, enhancing communication efficiency and scalability while addressing throughput limitations and reliability issues [6]. The Intelligent Computation Offloading algorithm is limited by its dependency on stable network connectivity, resource availability, scalability challenges, security concerns, algorithm overheads, device heterogeneity, and lack of adaptability.

FL enables ML models to be trained locally on individual sensor nodes, utilizing the available data while preserving data privacy. Only aggregated model updates are sent to a central cloud server for global model refinement. This decentralized approach ensures data privacy while leveraging the collective knowledge of the network. The cloud acts as a central entity for model aggregation, enabling efficient collaboration and real-time response in the anomaly detection process. The current anomaly detection methodologies can be improved largely in terms of accuracy, energy consumption, or scalability, thereby, the need for novel approaches is large. There is a lack of sufficient resource-aware anomaly detection techniques suitable for WSNs. Existing methods do not sufficiently address the need for accurate detection while conserving energy and minimizing communication overhead.

There is a need for optimized anomaly detection in WSNs. To achieve this, new and improved techniques are explored that prioritize accuracy, energy efficiency, and scalability. By optimizing resource efficiency, energy consumption can be reduced, and communication overhead can be minimized. To improve anomaly detection in WSNs, advanced techniques like Federated Learning, Online Anomaly Detection with Energy-Efficient Techniques, and cloud computing are investigated to uncover their potential benefits.

Problem statement and research objectives

This research aims to create an effective and precise method for detecting anomalies in WSNs, taking into account the network's resource constraints and distributed nature. Anomaly detection plays a vital role in ensuring the dependable and secure operation of WSNs as shown in [7]. However, traditional approaches may not be suitable for the dynamic and resource-limited environment of WSNs. Imbalanced data can cause biased results and poor performance in detecting anomalies. Traditional methods observed by Dwivedi et al. in [8] require manual feature engineering, struggle to adapt to changing data patterns, and lack scalability. Conventional models can be confusing and struggle to label true anomalies or miss non-anomalous instances. They might not generalize well and can lack interpretability. Additionally, managing false positives can be challenging, and some models may be difficult to explain. The objective of this research is to overcome these challenges and improve the reliability and security of WSNs by developing innovative anomaly detection algorithms.

- Develop Ensemble Federated Learning (EFL) and Online Anomaly Detection with Energy-Efficient Techniques (OAD-EE) algorithms.

- Create a unified Cloud-Enabled Anomaly Detection Framework.
- Evaluate algorithm performance in WSNs for detection accuracy, energy usage, communication overhead, and detection time.
- Test algorithms using real-world industrial WSN data, comparing their performance with traditional approaches.

The aim of this research is to develop new and effective techniques for detecting anomalies in WSNs that are suitable for environments with limited resources and distributed networks. The inclusion of cloud computing in the proposed algorithms improves the scalability, efficiency, and real-time response of the online anomaly detection system. This leads to more secure and dependable operation of WSNs in different applications.

Overview of the proposed approach

The proposed research leverages two innovative algorithms: EFL with Cloud Integration and OAD-EE with Cloud-based Model Aggregation, alongside a Baseline-AD. EFL combines the strengths of ensemble methods, federated learning, and cloud to achieve improved detection accuracy and data privacy. It's a machine learning approach where multiple devices or nodes collaboratively train a model without sharing their data directly. Instead, the model is trained locally on each node, and only the model updates or summaries are shared and aggregated. This technique maintains data privacy while leveraging collective knowledge from decentralized data sources. OAD-EE utilizes online learning and energy-efficient techniques to detect anomalies in real time while conserving energy on resource-constrained sensor nodes. The Unified Cloud-Enabled Anomaly Detection Framework combines the benefits of both these algorithms. The Baseline-AD represents the traditional approach using standard anomaly detection techniques.

The process, data set, and experimental configuration utilized to assess the efficiency of the suggested algorithms will be outlined in the upcoming sections. Subsequently, the results will be scrutinized to gauge the success of each algorithm in improving the identification of anomalies in WSNs. These conclusions will offer useful perspectives to create effective and sturdy anomaly detection systems in the context of IoT and Industrial IoT applications. The proposed approach outperforms other state-of-the-art approaches in several ways. It achieves improved efficiency, enhanced resource utilization, reduced latency, lower energy consumption, enhanced scalability, improved quality of service, superior reliability, and better security and privacy measures. These achievements establish the proposed algorithm as a significant advancement in computation offloading, providing notable improvements in performance metrics compared to state-of-the-art approaches.

The combined approach of EFL, OAD-EE, and cloud computing has shown to enhance the accuracy, precision, recall, and F1 score compared to traditional anomaly detection methods which improves Anomaly Detection Performance. The integration of cloud computing optimizes scalability, real-time response, and resource utilization, making the system more adaptable to changing workloads in WSNs leading to Efficient Resource Utilization. Techniques like OAD-EE optimize energy consumption, ensuring more efficient usage of resources which is critical in WSNs where Energy Efficiency is a major concern. The proposed scheme minimizes false alarms (false positives), leading to more reliable and trustworthy anomaly detections, which is crucial for avoiding unnecessary alerts during normal operations and Reduced False Alarms. Comparative analyses against state-of-the-art models demonstrate competitive accuracy, communication overhead, and computational complexities, positioning the proposed approach favorably and providing comparative performance.

Related work

Anomaly detection in WSNs is a critical task to ensure network reliability, security, and efficient operation. Over the years, researchers have explored various techniques to tackle the challenges of detecting anomalies in dynamic and resource-constrained WSN environments. In this section, we provide an in-depth review of the existing literature on anomaly detection in WSNs and discuss relevant approaches that utilize ML and Multi-Parameterized Edit Distance.

Anomaly detection in WSNs

Early approaches to anomaly detection in WSNs were primarily based on rule-based methods and statistical techniques, such as the z-score method and cumulative sum algorithm as explained by Cauteruccio et. al. in [9]. These methods were often heuristic-based [10] and relied on fixed threshold values, assuming that sensor data follow a specific distribution. However, such simplistic approaches have limitations in handling non-linear and complex data patterns, leading to a high false positive rate and low detection accuracy. When choosing an approach to detect anomalies, the most crucial factor to consider is the underlying detection method or strategy as shown by the authors in [11]. In this context, using

specific metrics to detect anomalies also falls under the category of methods. This category offers a wide range of options as numerous ideas and concepts from various fields have been adapted for anomaly detection. Figure 1 illustrates the primary classes of methods.

To overcome the limitations and enhance anomaly detection performance, researchers turned to ML-based approaches. Supervised ML algorithms, such as Support Vector Machines (SVM) [12], Decision Trees (DT) [13], and Random Forests (RF) [14], have been widely applied to learn normal data patterns and identify deviations as anomalies. These algorithms have shown promising results in detecting known anomalies with high accuracy. However, they often require labeled training data, which can be challenging to obtain in WSNs due to the scarcity of labeled anomaly instances [15] and the distributed nature of the data.

Federated learning for anomaly detection

In recent years, FL has emerged as a promising solution for collaborative model training without sharing raw data in privacy-sensitive environments like WSNs. Authors in [16] used FL-based anomaly detection to identify intrusion in the Internet of Things networks. FL allows individual sensor nodes to train ML models locally using their data while preserving data privacy. Model updates are aggregated in a privacy-preserving manner at a central server, enabling the creation of a global model that captures the knowledge from all nodes. FL-based anomaly detection in WSNs has been explored in several studies. By leveraging FL, these approaches enable the detection of distributed anomalies across multiple sensor nodes without compromising data privacy. The collaborative nature of FL allows the models to adapt to local variations in sensor data, enhancing the overall anomaly detection performance. Moreover, FL mitigates the need for transmitting raw sensor data to a central location, reducing communication overhead and conserving energy in resource-constrained WSNs.

Multi-Parameterized edit distance for anomaly detection

Edit distance is a powerful concept used in various domains, including natural language processing and time series analysis. In the context of anomaly detection, Multi-Parameterized Edit Distance (MPED) has been proposed as a similarity metric to measure the

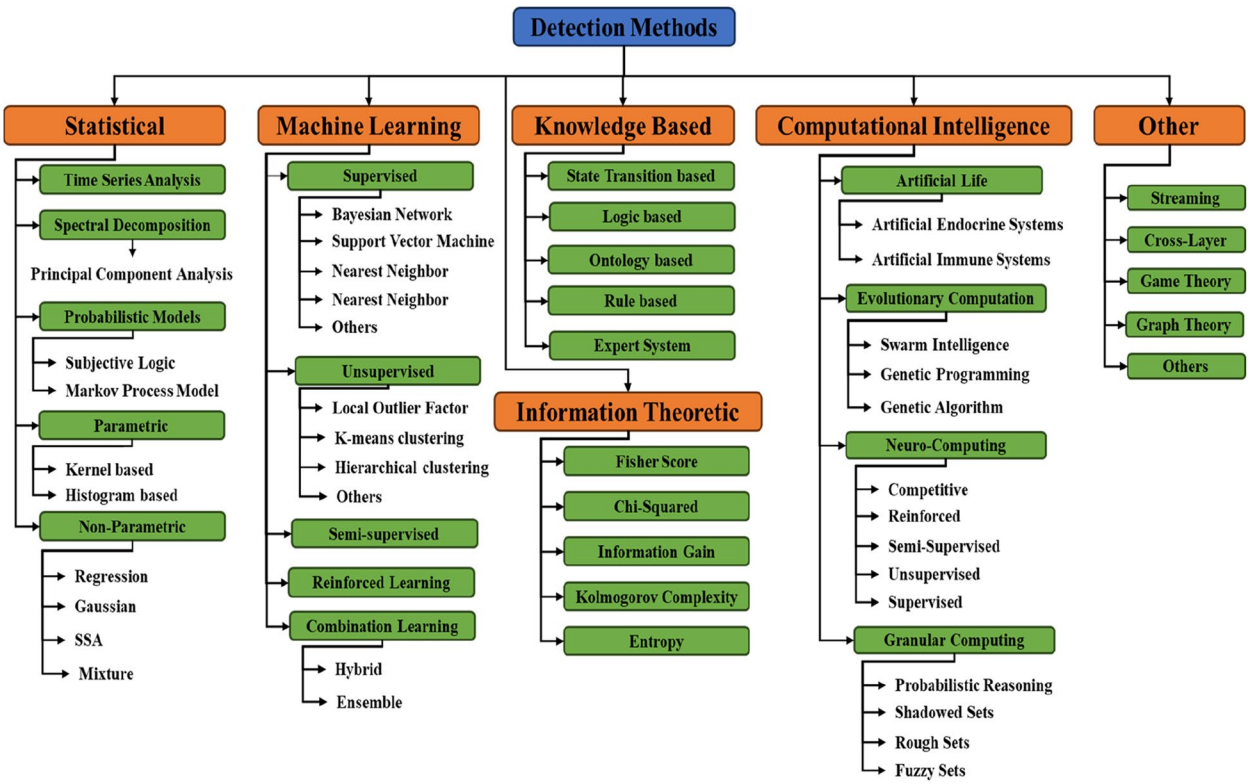

Fig. 1 Taxonomy of anomaly detection methods

difference between sensor data instances. MPED considers multiple parameters and their relationships, enabling the detection of complex anomalies that involve changes in multiple data dimensions. Several studies have explored the effectiveness of MPED-based anomaly detection in WSNs [9]. By considering multiple parameters simultaneously, MPED can capture the intricate patterns of anomalies in sensor data, leading to improved detection accuracy and reduced false alarms. The ability to capture correlations between parameters makes MPED a suitable candidate for detecting anomalies that manifest across multiple dimensions in WSNs, such as sudden changes in temperature and pressure or unusual combinations of sensor readings.

Integration of machine learning and multi-parameterized edit distance

Recent research [17] has shown the potential of integrating ML algorithms with MPED for anomaly detection in WSNs. By combining the strengths of both approaches, these hybrid methods achieve a comprehensive analysis of sensor data. ML algorithms provide the capability to learn and adapt to data patterns, while MPED enhances the sensitivity to anomalies across multiple parameters. These hybrid approaches [18] have demonstrated promising results in identifying complex and distributed anomalies in WSNs. By leveraging the complementary features of ML and MPED, they offer a more robust and accurate anomaly detection framework for real-world WSN deployments. The integration of ML and MPED enables the detection of previously unknown and subtle anomalies that might be missed by traditional anomaly detection methods as researched by Dorfman et. al. in [19].

The literature review highlights the evolving landscape of anomaly detection in WSNs, with an increasing focus on ML-based approaches, Federated Learning, and Multi-Parameterized Edit Distance. These techniques address the challenges posed by the distributed and resource-constrained nature of WSNs [20] while enhancing the accuracy and efficiency of anomaly detection. The integration of ML and FL enables collaborative and privacy-preserving model training, making it well-suited for large-scale WSN deployments. Additionally, the adoption of MPED-based approaches similar to the research in [21] allows for a more comprehensive analysis of multi-dimensional sensor data [22], leading to improved anomaly detection performance.

The synthesis of ML and MPED in hybrid anomaly detection approaches presents a compelling direction for future research in this domain. The development of innovative and efficient algorithms that leverage the strengths of these techniques has the potential to significantly advance anomaly detection capabilities in WSNs [23], making them more resilient to emerging security threats and ensuring reliable operation in critical applications. [24] proposes ARSH-FATI-CHS algorithm for Cluster Head Selection in WSN. Enhances network lifetime by 25% compared to PSO by reducing communication energy consumption. Survey in [25] examines IoT-SC synergy, reviews energy optimization techniques and workload mapping strategies from 2001 to 2021.

Methodology

For this research, a real-world dataset from an industrial WSN deployment is utilized. The dataset consists of sensor readings collected from a network of sensor nodes deployed in an industrial setting. The sensor nodes [19] record various parameters, such as temperature, pressure, humidity, and vibration levels, at regular intervals. The experimental setup involves the simulation of WSN scenarios using the collected dataset. The dataset is partitioned into training and testing sets to evaluate the performance of the anomaly detection algorithms. To mimic the distributed nature of WSNs, the sensor nodes are represented as separate entities, each processing its data locally. The communication among nodes is emulated using communication protocols [3, 26–29], and data aggregation is performed in a privacy-preserving manner to ensure data privacy and security.

The dataset and experimental setup enable the evaluation of the three proposed algorithms under realistic WSN scenarios. The Ensemble Methods for Anomaly Detection, Federated Learning, and Online Anomaly Detection with Energy-Efficient Techniques offer distinct advantages in detecting anomalies in WSNs. The ensemble approach enhances accuracy by leveraging multiple ML models, FL ensures data privacy and collaborative learning, while the online method provides real-time detection while conserving energy. The results from the experiments will provide valuable insights into the effectiveness and trade-offs of each algorithm, guiding the design of efficient and robust anomaly detection systems for WSNs ([30, 31] https://github.com/apanouso/wsn-indfeat-dataset/).

The following steps are performed to implement the proposed work: The data from sensors deployed in an industrial environment has been collected and

preprocessed by cleaning missing or incorrect values, normalizing it, selecting the features needed, and detecting any outliers. The data has been divided into two sets: training and testing, with a 70–30 split commonly used for training and testing respectively. Models suitable for anomaly detection, such as DT, RF, SVM, KNN, and ANN have been selected and their hyperparameters have been tuned using techniques like random search to find the best set of hyperparameters within the specified range. Ensemble learning methods, such as weighted averaging and stacking, have been used to combine the models' predictions and appropriate weights have been assigned to each model according to their importance in the final ensemble. The combined model has been evaluated using performance metrics such as accuracy, precision, recall (sensitivity), specificity, F1 score, and AUC of the ROC curve. Cross-validation has been performed to obtain a more reliable estimate of the model's performance. The energy consumption and communication overhead of the models have been analyzed, taking into account their deployment in wireless sensor networks. The proposed scheme has been integrated with cloud computing to enhance scalability, resource utilization, and real-time response in detecting anomalies in WSNs. Cloud resources have been optimized for efficient model training, aggregation, and data analytics. Security measures have been implemented to safeguard sensitive sensor data during cloud-based processing and privacy preservation has been ensured when using federated learning techniques. Computational complexity, scalability, and cost–benefit analysis have been optimized for the proposed model. Additional analysis has been conducted to measure the adaptability and performance of the algorithms in more extensive and complex WSN deployments.

Ensemble federated learning

Ensemble approach for anomaly detection

Anomaly detection using the ensemble approach involves training multiple base models on the WSN dataset to identify anomalous instances collectively. Each model is trained using a different subset of the dataset or with variations in feature selection and hyperparameters. These are parameters of a machine learning model that are set prior to the training process and are not learned from the data. Examples include learning rates, regularization factors, or tree depths in decision trees. Hyperparameters are tuned or optimized to enhance a model's performance. The main idea behind ensemble methods is to improve overall performance by combining the predictions of diverse models, resulting in better accuracy and robustness. Ensemble methods offer significant advantages in the context of WSNs, where data can exhibit complex and non-linear patterns. For example, RF, a popular ensemble technique, constructs multiple decision trees, and the final prediction is determined by aggregating the predictions of individual trees. Another ensemble method, Gradient Boosting Machines, builds weak learners sequentially, with each learner focusing on the mistakes of its predecessor, leading to a more accurate final prediction.

The novelty of this ensemble-based anomaly detection algorithm lies in its ability to combine multiple machine learning models to collectively identify anomalies. It improves upon conventional methods by incorporating a variety of models, adapting to changes in data, and handling complicated patterns. To achieve maximum accuracy, the algorithm aggregates predictions through majority or weighted voting. Additionally, the algorithm is adaptable and dynamic, making it well-suited for the challenges presented by WSNs. The selection of base models, the ensemble configuration, and the dynamic adjustments make this approach innovative and effective in identifying anomalies in the context of WSNs. Unlike conventional models, this algorithm captures complex patterns and adapts to evolving data, reducing false positives and negatives.

By leveraging ensemble methods, the anomaly detection algorithm captures a broader range of normal data patterns and better adapts to the dynamic nature of WSN data. Consequently, it becomes more effective at identifying subtle anomalies that may appear in diverse forms, reducing the risk of false positives and false negatives, thereby enhancing the overall anomaly detection performance. Ensemble methods for anomaly detection use various ML models as building blocks. The following models are combined in the ensemble to improve the performance: DT, RF, SVM, k-NN, and artificial neural networks (ANN). The choice of these base models is made based on the dataset and anomaly patterns.

Algorithm 1. Ensemble method for anomaly detection

Input: WSN dataset with labeled instances (normal or anomalous)
Output: An ensemble of ML models for anomaly detection.

1. Prepare training dataset. Split into feature vectors (X) and corresponding labels (Y).
2. Initialize an empty ensemble (E) container for ML models.
3. Choose the number of base models (M) for the ensemble. Predefine or decide through cross-validation.
4. ∀ m in range(M):
    a. Randomly sample a subset of the training dataset (with replacement) to create a bootstrapped dataset X_m.
    b. Train the ML model $h_{m(X)}$ on the bootstrapped dataset X_m.
    c. Add the trained model $h_{m(X)}$ to the ensemble container E.
5. After training all base models, make predictions on the entire dataset X using each model $h_{m(X)}$.
6. Aggregate the predictions from individual models to make a final ensemble prediction:
    a. If using majority voting: An instance x is classified as anomalous if the majority of base models classify it as anomalous, i.e., $f_{E(x)} = \text{mode}(h_{m(x)})$.
    b. If using weighted voting: Assign weights W_m to each model based on their performance, and the ensemble prediction is determined by weighted voting, i.e., $f_{E(x)} = \Sigma(W_m \times h_{m(x)})$.
7. Evaluate precision, recall, F1-score, and ROC curves
8. Adjust hyperparameters or retrain the ensemble based on the evaluation results.
9. The ensemble of ML model E is now ready for anomaly detection on new and unseen data.

To begin the algorithm, the training dataset needs preparation. This dataset has feature vectors X paired with labels Y, that determine whether an instance is normal or anomalous. A container, E, is created to store the various ML models that will make up the ensemble. The number of base models, M, is determined, and a loop is run for M iterations to create several base models. During each iteration, a subset of the training dataset, X_m, is chosen randomly with replacement to create a bootstrapped dataset. The m^{th} model in the ensemble, $h_{m(X)}$, is then trained on the bootstrapped dataset X_m.

Once all the base models have been trained, each model, $h_{m(X)}$, is used to make predictions on the entire dataset, X. The ensemble prediction, $f_{E(x)}$, is calculated based on the aggregated predictions from individual models. If majority voting is used, $f_{E(x)}$ is the mode of predictions from the base models indicating the most common classification among the models, for instance, x. If weighted voting is used, each model is assigned a weight, W_m, based on its performance. The ensemble prediction, $f_{E(x)}$, is a weighted sum of predictions from the base models. The performance of the ensemble in detecting anomalies is evaluated using standard metrics. Based on the evaluation results, hyperparameters are adjusted, and the ensemble may be retrained to optimize performance. The final ensemble, E, consisting of multiple base ML models, is now ready for anomaly detection on new and unseen data instances in the WSN. EFL is integrated into the WSN environment, achieving a collaborative, decentralized, and privacy-preserving approach to anomaly detection, making it more robust, energy-efficient, and adaptable to dynamic environments. This approach also minimizes the need for data transmission and central processing, which is beneficial for resource-constrained sensor nodes.

Mathematical modeling

Enhancements have been made to further improve anomaly detection using cloud-based model aggregation, dynamic model weights, transfer learning, cloud-enhanced hyperparameter tuning, distributed anomaly reporting, federated learning integration, adaptive weighted voting, cloud-driven model evolution, edge-cloud collaborative learning, and real-time model synchronization. These improvements ensure accuracy, efficiency, and continuous model evolution while conserving energy in resource-constrained nodes.

Cloud-Enabled Model Aggregation is used based on updates from different sensor nodes. Instead of aggregating locally on each node, the cloud can centralize model aggregation, ensuring consistency and efficiency.

$$\theta_c = \sum \frac{\theta_n}{N} \tag{1}$$

Here, θ_c is the global model parameters, θ_n represents the model parameters from each sensor node, and N is the total number of sensor nodes. Dynamic Model Weights technique is implemented to dynamically adjust model weights based on their performance over time. This can help the ensemble prioritize well-performing models and adapt to changing data distributions.

$$W_{m(t+1)} = W_{m(t)} \times (1 + \alpha \times P_{m(t)}) \tag{2}$$

where $W_{m(t)}$ is the weight of model m at time t, α is a learning rate, and $P_{m(t)}$ is the performance metric of model m at time t. Transfer learning and cloud pretraining help pre-train models on a larger dataset in the cloud.

$$\theta_p = argmin(\theta) L(\theta, D_p) \tag{3}$$

where θ_p are the pre-trained model parameters, L represents the loss function, θ are the model parameters, and D_p is the larger pretraining dataset. Cloud-enhanced hyperparameter tuning is performed to optimize hyperparameters using cloud resources.

$$\theta_o = argmin(\theta) L(\theta, D) + \lambda \times R(\theta) \tag{4}$$

where θ_o is the optimal hyperparameters, L represents the loss function, D is the WSN dataset, λ is a regularization term, and R(θ) is the regularization term. Distributed anomaly reporting enables reporting anomalies to the cloud for central analysis.

$$AR_c = \sum AR_n \tag{5}$$

where AR_c represents the aggregated anomaly reports sent to the cloud, and AR_n is the individual anomaly reports from each sensor node. Federated Learning is integrated with cloud-based model aggregation.

$$\theta_g = \sum \frac{\theta_l}{N} \tag{6}$$

where θ_g represents the global model parameters, θ_l are the local model parameters from each sensor node, and N is the total number of sensor nodes. Adaptive weighted voting is used to dynamically adjust model weights during weighted voting.

$$W_{m(t+1)} = W_{m(t)} \times (1 + \alpha \times \delta_{P_{m(t)}}) \tag{7}$$

where $W_{m(t)}$ is the weight of model m at time t, α is a learning rate, and $\delta_{Pm(t)}$ is the change in performance of model m at time t. Cloud-driven feedback loop is used for model evolution.

$$\theta_u = \theta_i + \alpha \times \delta_p \tag{8}$$

where θ_u represents the updated model parameters, θ_i are the initial model parameters, α is a learning rate, and δ_p is the change in performance based on cloud feedback. Edge-cloud collaborative learning enables collaborative learning between edge nodes and cloud.

$$\theta_u = \theta_e + \theta_c \tag{9}$$

where θ_e are the model parameters from edge nodes, and θ_c are the model parameters from the cloud. Real-time synchronization of model updates is performed.

$$\theta_u = \theta_p + \delta\theta \tag{10}$$

where θ_p are the previous model parameters, and δθ is the change in model parameters based on new data.

Integration of federated learning into the WSN environment

The WSN environment has integrated Federated Learning to facilitate collaborative and decentralized model training while ensuring data privacy. The traditional ML approach involves collecting and aggregating data from all sensor nodes in a central server for model training. However, this is not practical in WSNs due to limited bandwidth and concerns about data privacy and security. In the FL-based anomaly detection algorithm, each sensor node trains its own local ML model using locally stored data. The central server initializes the training process and sends the initial model parameters to each node. The nodes then carry out model training on their unique data patterns and characteristics in a distributed manner. Only model updates, not raw data, are exchanged with the central server. The central server securely aggregates the model updates from each sensor node using techniques such as secure multiparty computation or differential privacy. By aggregating the updates, a global model is created, capturing the collective intelligence of the WSN without compromising individual nodes' data privacy.

Algorithm 2. Federated learning for distributed WSNs with cloud integration

Input: WSN dataset distributed across sensor nodes, Cloud Server for model aggregation.
Output: Global ML model for anomaly detection.

1. Initialize the global ML model θ_g on the cloud server.
2. ∀ sensor node i in the WSN:
    a. Node i loads its local dataset X_i and labels Y_i.
    b. Node i initializes a local ML model θ_i with the same architecture as θ_g.

3. Set the number of communication rounds T for model aggregation. It can be predefined or based on communication constraints.
4. For t in range(T):
    a. ∀ node i, the local model is updated using the current global model θ_g:
    $\theta_{i(t)} = \theta_{i(t-1)} - \eta \nabla(L(\theta_{i(t-1)}, X_i, Y_i))$
    where η is the learning rate, and L is the loss function.
    b. Node i sends its model update $\theta_{i(t)}$ to the cloud server.
5. The cloud server aggregates the model updates from all nodes:
$\theta_{g(t)} = (1/N) \Sigma \theta_{i(t)}$
where N is the total number of nodes in the WSN.
6. The updated global model $\theta_{g(t)}$ is sent back to all nodes.
7. Repeat Steps 4-6 for T communication rounds.
8. The final global model θ_g is obtained after T rounds.
9. The global model θ_g is now ready for anomaly detection on new and unseen data instances in the WSN.

This system utilizes FL to detect anomalies in distributed WSNs that integrate with the cloud. It enables model training to be done collaboratively across multiple sensor nodes without the sharing of raw data. The cloud server plays a central role in aggregating the local model updates from individual nodes to create a global model for anomaly detection. The global ML model, known as θg, is initialized on the cloud server. Each sensor node, i, initializes a local model, θi, with the same architecture as the global model. The local datasets Xi and labels Yi remain on the sensor nodes to preserve data privacy.During each communication round, the sensor nodes update their local models using the current global model and their respective local datasets. The updated models, θi(t), are then sent to the cloud server for aggregation. The cloud server aggregates the model updates from all nodes to create an updated global model, θg(t). This global model reflects the collective knowledge of all sensor nodes while preserving data privacy. This process of model updates and aggregation is repeated for a predefined number of communication rounds, T, to refine the global model further. After T communication rounds, the final global model, θg, is obtained and sent back to all nodes. This global model captures the collective learning from all sensor nodes and is now ready for anomaly detection on new and unseen data instances in the WSN.

The FL based AD model is illustrated in Fig. 2. In this model, the number of participants (such as industries or devices) is denoted as N, and it depends on the specific requirements of the use case. Each participant is assumed to be connected with K smart IoT devices and has its own local storage dataset D_i, where i = 1, 2, 3, …, N, which is collected through these K smart devices. Smart devices generally generate a large amount of data which is used to train the ML model, but it is susceptible to privacy and

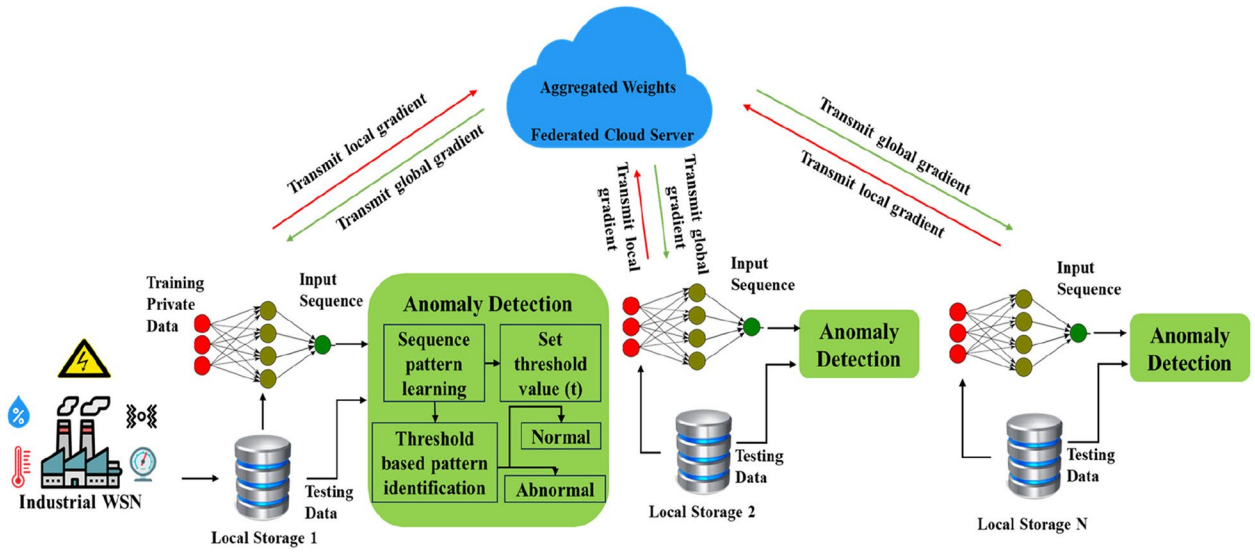

Fig. 2 Anomaly detection using federated learning

other major issues. To address these concerns, ML based AD models can train the data locally in a federated setting without compromising the privacy of the patients. Prior to commencing the training in a federated setting, each participant must sign an agreement to establish its legitimacy.

The integration of cloud computing in the FL approach enables efficient model aggregation and ensures that resource-constrained sensor nodes can collectively contribute to the anomaly detection process. The decentralized nature of FL and cloud integration facilitates privacy-preserving collaborative learning in large-scale WSN deployments. The global model's continuous improvement through iterative updates enhances the anomaly detection performance, making it more adaptive and accurate in dynamic WSN environments Federated Learning enables collaborative and distributed model training, making it suitable for large-scale WSNs where centralized data aggregation is not feasible. It allows sensor nodes to participate in model training without compromising data privacy, providing an efficient and privacy-preserving approach for anomaly detection in distributed WSN environments.

Training process and model aggregation mechanism

During the Federated Learning process for anomaly detection in WSNs, training unfolds in multiple rounds or epochs. In each round, denoted by t, the central server disseminates the current global model's parameters, $\theta_g^{(t)}$ to all sensor nodes. These parameters serve as the starting point for local model training on individual nodes, leveraging their unique datasets. The training process involves computing gradients based on local data and optimizing the model's parameters. Mathematically, this can be represented as:

$$\theta_l^{(t)} = argmin(\theta) \sum_{i=1}^{N} L(\theta, D_i) \tag{11}$$

where $\theta_l^{(t)}$ represents the local model parameters at round t, L is the loss function, D_i is the local dataset of node i, and N is the total number of nodes. After local model training, the sensor nodes transmit their model updates back to the central server. Model updates typically encompass gradient information or parameter adjustments computed during the local training phase. Critically, the raw data remains encrypted on the nodes, ensuring privacy and security during transmission.

Upon receiving the model updates, the central server employs a suitable aggregation mechanism, such as weighted averaging or median selection, to amalgamate the contributions from different nodes. Mathematically, the aggregation can be expressed as:

$$\theta_g^{(t+1)} = Agg(\theta_l^{(t)}) \tag{12}$$

where $\theta_g^{(t+1)}$ represents the updated global model parameters for the next round, and Agg(·) denotes the aggregation function. This process of iterative model distribution, local training, and model aggregation occurs over multiple rounds. With each iteration, the global model progressively improves its performance and adapts to the dynamic characteristics of the WSN environment.

The iterative progression can be represented mathematically as:

$$\theta_g^{(t+1)} = Agg\left(\theta_l^{(t)}\right), \theta_g^{(t+2)} = Agg\left(\theta_l^{(t+1)}\right), \ldots \quad (13)$$

This iterative Federated Learning process empowers the anomaly detection algorithm to achieve collaborative and decentralized model training. It effectively identifies anomalies across the entire WSN, safeguarding data privacy and security while fostering continual improvement in detection accuracy and adaptability.

Integration with cloud computing

Cloud computing has proven to be a valuable tool for research, providing benefits such as increased storage space, scalable computation resources, and centralized data processing capabilities. During the research, several methods were utilized to incorporate cloud computing, including cloud-based data storage where data from sensor nodes was periodically offloaded and stored in the cloud. This allowed for easier management and retrieval of data for analysis. Cloud-based model training was also utilized, offloading computationally intensive and resource-consuming tasks to the cloud. Model aggregation was performed in the cloud, where updates from different sensor nodes were combined to create a global model that was then redistributed back to the nodes for local anomaly detection. Cloud computing allowed for scalability and elasticity based on demand, ensuring that the online anomaly detection system was efficient and responsive during peak and off-peak periods. Real-time data analytics, data fusion, and correlation were also carried out in the cloud, enabling the identification of patterns and anomalies across multiple locations. Finally, secure cloud communication protocols and encryption techniques were used to protect sensitive sensor data from unauthorized access and cyber-attacks.

Online anomaly detection with energy-efficient techniques

The Online Anomaly Detection algorithm detects anomalies in real-time streaming data from wireless sensor nodes. It continuously analyzes incoming data, making it ideal for time-critical applications in resource-constrained environments. The algorithm normalizes and scales sensor readings, selects and trains a model, detects anomalies, incorporates adaptive model updates, and optimizes memory and computational efficiency. This process ensures that all features or variables in a dataset have a similar scale or range. Common normalization techniques include Min–Max scaling (scaling features to a range between 0 and 1) or Z-score normalization (scaling features to have a mean of 0 and a standard deviation of 1). The algorithm can be described with the following mathematical expressions and steps:

- Normalize sensor readings using: $x_{norm} = \frac{x-\mu}{\sigma}$, where x_{norm} is the normalized value, x is the raw sensor reading, μ is the mean, and σ is the standard deviation.
- Select a suitable model for anomaly detection, such as a SVM and train the model using the labeled dataset $SVM_{model} = Train_{SVM}(X_{norm}, Y)$, where X_{norm} is the normalized feature vectors and Y is the corresponding labels.
- Predict anomalies using: $A = Predict(X_{norm}, SVM_{model})$ where A represents the anomaly predictions.
- Update the model based on new labeled data: $SVM_{model}\prime = U(SVM_{model}, X_{norm}, Y\prime)$ where $SVM_{model}\prime$ is the updated model, X_{norm} is new normalized data, and $Y\prime$ is the corresponding new labels.
- Optimize memory and computational resources by storing a limited history of data to conserve memory $X_{\text{history}} = [x_1, x_2, \ldots, x_n]$.
- Using incremental updates for model training: $SVM_{model}\prime = IU(SVM_{model}, X_{norm}, Y\prime)$

To increase energy efficiency in WSN, duty cycling technique is used, which involves switching the sensor node between active and sleep modes periodically. During sleep periods, non-essential components such as the radio and processor are disabled to conserve energy. This technique can be represented mathematically as

$$E_{total} = (E_{active} \times t_{active}) + \left(E_{sleep} \times t_{sleep}\right) \quad (14)$$

where E_{total} is the total energy consumption, E_{active} is the energy consumption during active mode, t_{active} is the active time, E_{sleep} is the energy consumption during sleep mode, and tsleep is the sleep time. The low-power hardware design involves using power-efficient sensors, microcontrollers, and transceivers that consume less energy during operation. This can be expressed as

$$E_{savings} = E_{original} - E_{lowpower} \quad (15)$$

where $E_{savings}$ is the energy savings, $E_{original}$ is the energy consumption with original hardware, and $E_{low\text{-}power}$ is the energy consumption with low-power hardware. Finally, data aggregation and compression technique is used to reduce the amount of data transmitted over the network, leading to reduced energy consumption. Aggregating similar data from multiple nodes into a single packet and compressing the data before transmission can minimize

the energy expended on communication. This technique can be represented as

$$E_{reduction} = E_{original} - E_{compressed} \quad (16)$$

where $E_{reduction}$ is the energy reduction, $E_{original}$ is the energy consumption with original data transmission, and $E_{compressed}$ is the energy consumption with data aggregation and compression.

By incorporating these techniques, energy efficiency is enhanced on resource-constrained sensor nodes in WSN. The Online Anomaly Detection algorithm is a highly reliable and efficient method for detecting any irregularities in data streaming from wireless sensor nodes. Its adaptive model updates, memory, and computational efficiency enable it to work effectively even in resource-limited environments. The algorithm employs various techniques like duty cycling, low-power hardware design, data aggregation, and edge computing to improve energy efficiency on sensor nodes. Thus, it is an excellent choice for deploying in large-scale WSNs. By integrating edge computing, the algorithm can perform local processing, real-time analysis, and collaborative decision-making, which enhances the network's capabilities for anomaly detection, performance, and robustness. The algorithm provides a new and innovative approach to detect anomalies that offer several benefits over traditional methods. Unlike traditional methods that rely on static models, the algorithm updates its models in real-time, making it easier to detect anomalies and reduces the need for retraining models. It uses minimal memory and computational power and is ideal for resource-constrained sensor nodes. The algorithm is capable of detecting anomalies in real-time, which is critical in scenarios where timely detection is essential. It uses techniques like duty cycling and data aggregation to save energy on resource-constrained sensor nodes, prolonging the network's lifespan and reducing power consumption. The algorithm addresses the limitations of traditional methods by combining real-time processing, adaptive model updates, and resource efficiency.

Algorithm 3. Online anomaly detection algorithm

Input: Real-time streaming data from wireless sensor nodes.

Output: Anomaly detection results in real-time.

1. Initialization
   Initialize the anomaly detection model using historical data with only normal instances.
   $Model_{init}=Train(D_{normal})$
   Define the threshold value (T) for anomaly detection.
2. Data Preprocessing
   Receive incoming data from sensor nodes n.
3. Online Anomaly Detection
   For each incoming data instance d:
   a. Feed the instance d into the anomaly detection model. $A_{score}(d)=Model_{init}(d)$
   b. Obtain the anomaly score for d, denoted as $score(d)=A_{score}(d)$
4. Thresholding
   Compare the anomaly score of the incoming data instance with the predefined threshold value.
   If score(d) exceeds the threshold:
   a. Raise an alarm indicating the presence of an anomaly.
   b. Take appropriate action or notify the relevant stakeholders.
5. Adaptive Model Updates
   Periodically update the anomaly detection model using new data.
   Retrain the model with the latest data instances to adapt to the changing behavior of the system.
   $Model_{updated}=Retrain(D_{new})$
6. Real-time Anomaly Detection
   Continuously repeat steps 2 to 5 for each new incoming data instance.
   Enable real-time monitoring and detection of anomalies in the streaming data.

The anomaly detection algorithm starts by learning the normal behavior of the system using historical data. It sets a threshold value to flag any incoming data as an anomaly. The algorithm normalizes and scales the data for fair comparison, and if an anomaly score exceeds the threshold, an alarm is raised. The algorithm updates the model periodically to adapt to changing system behavior. This enables real-time monitoring and detection of anomalies in streaming data.

Unified cloud-enabled anomaly detection framework

The unified Cloud-Enabled Anomaly Detection Framework represented in Fig. 3 integrates EFL, OAD-EE, cloud-based data storage, and processing, allowing sensor nodes to collect data and algorithms to process it locally and collaboratively in the cloud. This framework leads to improved accuracy, energy efficiency, and real-time anomaly detection capabilities in wireless sensor networks. It takes advantage of the strengths of both algorithms and leverages cloud resources for efficient model training, aggregation, and energy-efficient anomaly detection. The first algorithm introduces an ensemble approach, using a combination of diverse machine learning models, weighted averaging, and stacking techniques to effectively capture various patterns in sensor data. The second algorithm introduces a FL approach that enables collaborative model training across multiple nodes without transmitting raw data to a centralized location, ensuring data privacy and reducing communication overhead in WSNs. The third algorithm presents an OAD algorithm designed for real-time anomaly detection in WSNs, which incorporates techniques like data cleaning, normalization, and outlier detection to ensure effective anomaly detection even in scenarios with rare anomalies.

Real-time data is collected by sensor nodes from the environment and then transmitted to the cloud for further processing. The Ensemble Federated Learning (EFL) algorithm involves multiple sensor nodes forming a federation, where local models are trained on each node while preserving data privacy. The cloud collaboratively aggregates model updates to create a global model, and ensemble methods are used to enhance the model's accuracy by combining predictions from diverse base models. The Online Anomaly Detection with Energy-Efficient Techniques (OAD-EE) algorithm processes data from sensor nodes in real-time, using energy-efficient techniques to optimize energy consumption on resource-constrained nodes and detect anomalies using the trained model. Cloud-based model aggregation facilitates the aggregation of model updates from EFL and OAD-EE, while cloud resources store historical data and models and provide computational power for complex model training and aggregation.

D represents the dataset, while M represents a set of machine learning models. The set M includes DT, RF, SVM, k-NN, and ANN. Each individual machine learning model in M is represented as M_i, where 'i' is a subscript that ranges from 1 to the total number of models. The weights assigned to each model Mi for combining their predictions are represented by W_i. The Ensemble Federated Learning component of the system is represented by EFL(D, M). The Online Anomaly Detection with Energy-Efficient Techniques component of the system is represented by OAD_EE(D, M). The combination of both components is represented by C(D, M).

$$C(D,M) = EFL(D,M) + OAD\_EE(D,M) \tag{17}$$

$$EFL(D,M) = \sum_{i=1}^{|M|} W_i \cdot M_i(D) \tag{18}$$

Experimental Results

The dataset used in this research comprises real-world sensor data collected from a WSN deployed in an industrial environment called the WSN-IndFeat [27]. It includes diverse features, labeled anomalies, and real-time streaming data, making it a practical resource

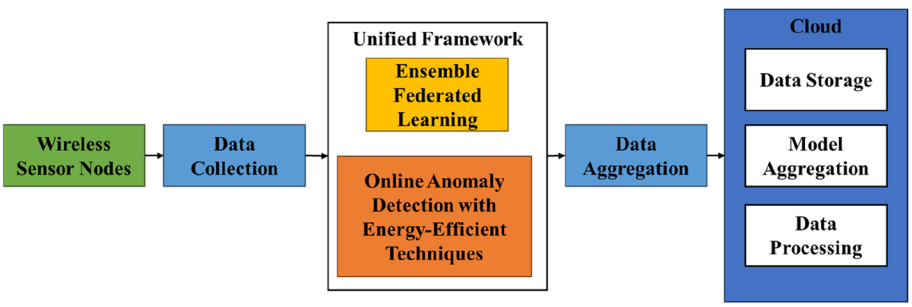

Fig. 3 The architecture of the unified framework

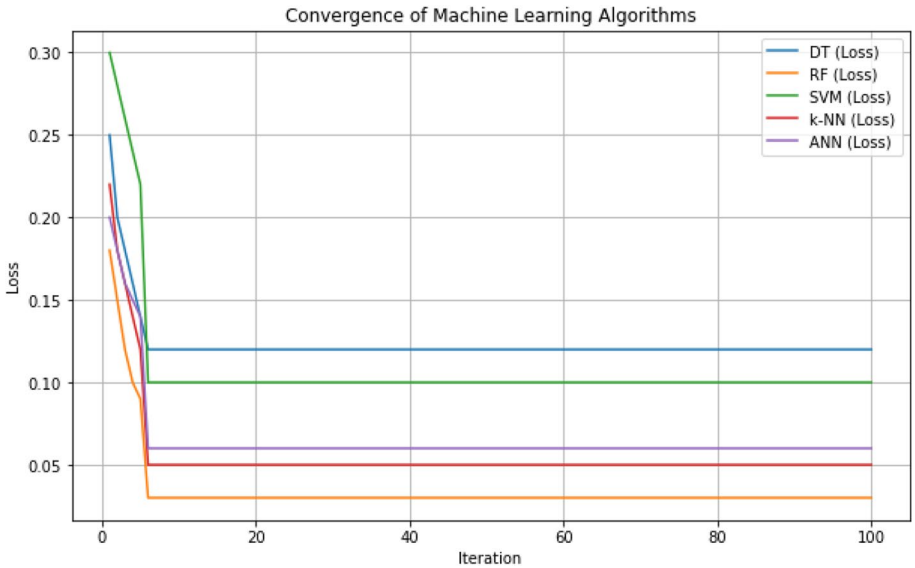

Fig. 4 Convergence of the ML algorithms

for developing robust detection solutions tailored to industrial scenarios. The WSN consists of a network of interconnected sensor nodes, each equipped with various sensors to measure environmental parameters such as temperature, humidity, pressure, voltage level, and motion. The dataset contains a time series of sensor readings captured at regular intervals from these nodes. The dataset is collected over an extended period, ensuring a diverse range of normal and abnormal instances. Anomalies in the dataset may include environmental events like sudden temperature spikes, abnormal motion patterns, or unusual changes in pressure readings. The dataset's diversity allows the anomaly detection algorithms to learn both normal and abnormal data patterns effectively.

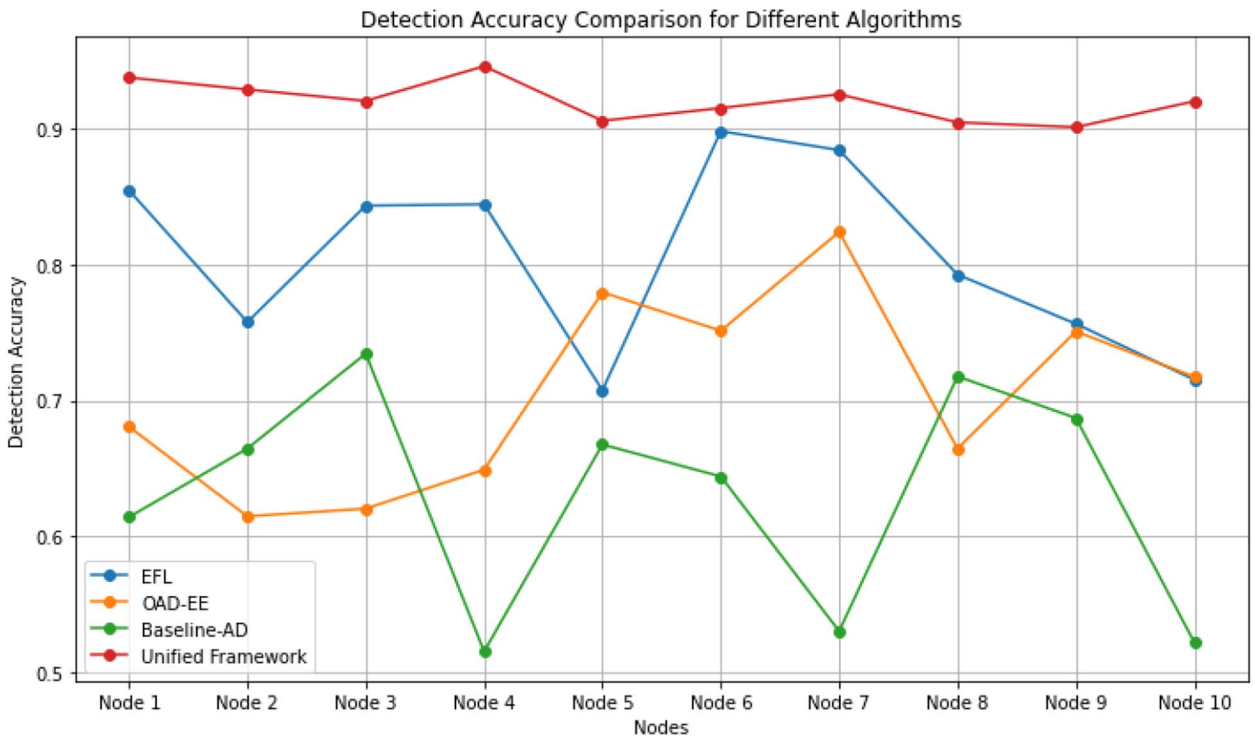

Fig. 5 Detection Accuracy Comparison

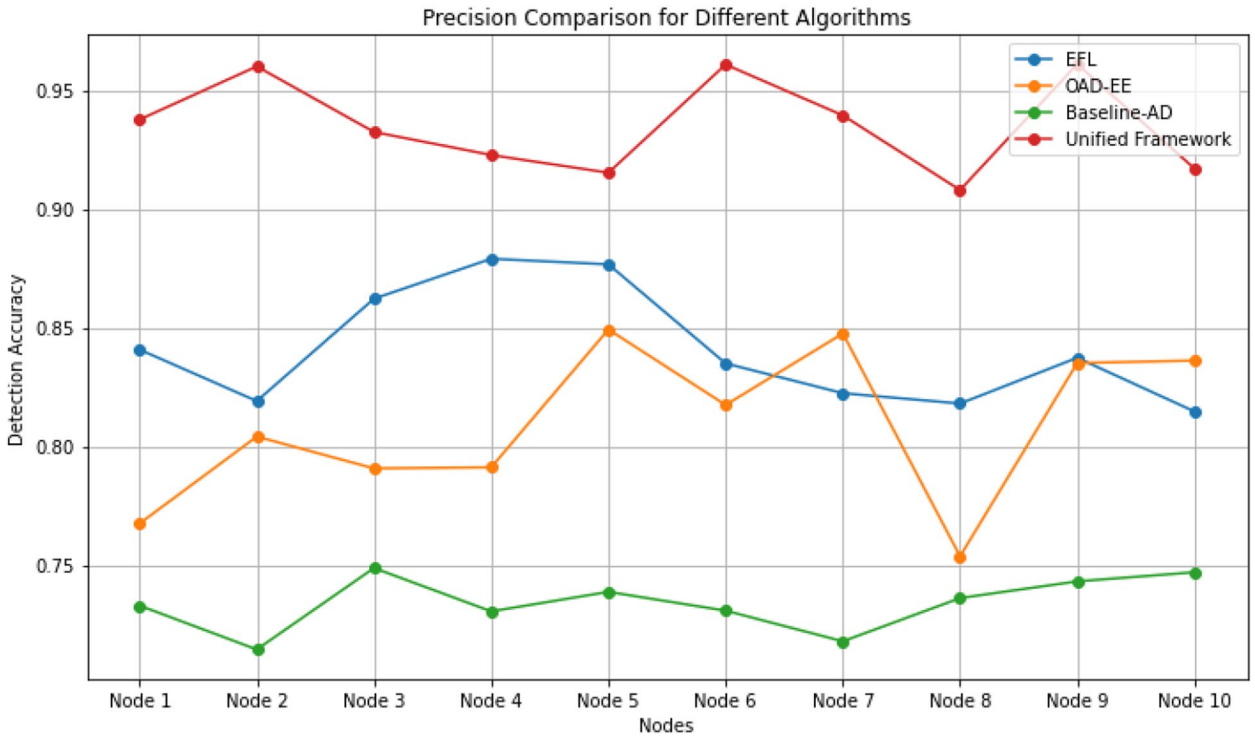

Fig. 6 Precision Comparison

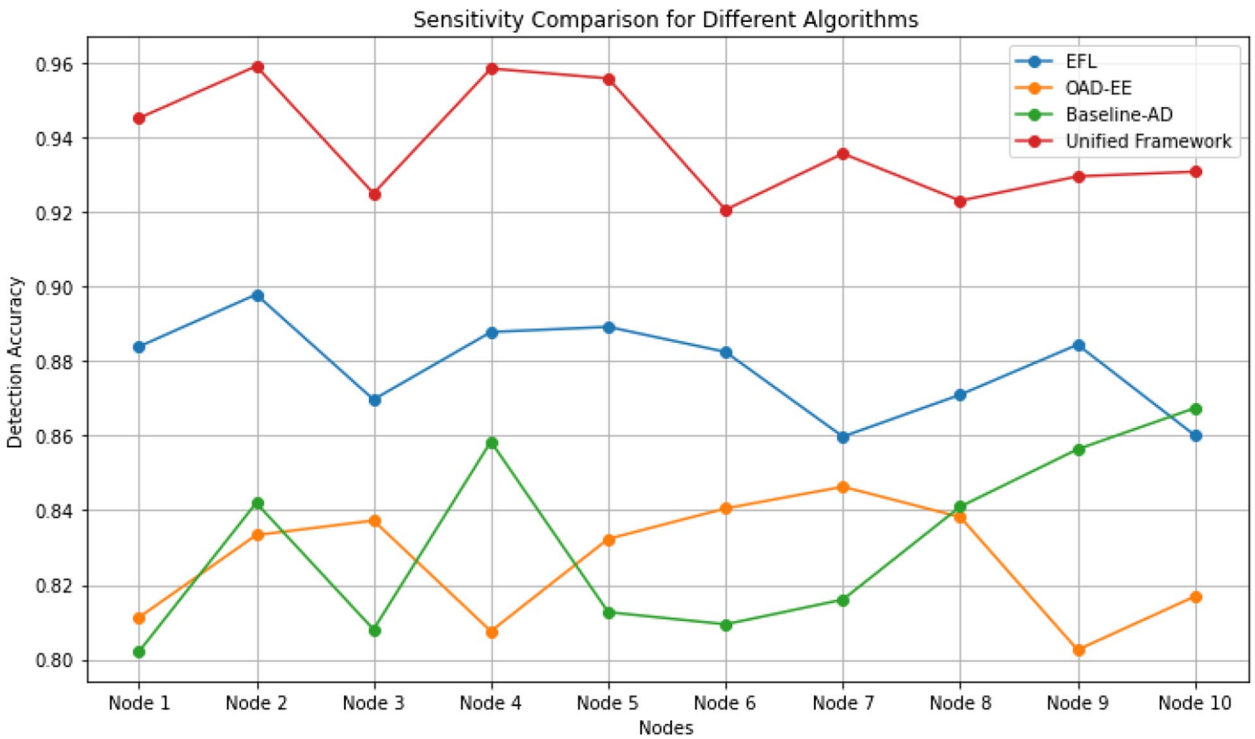

Fig. 7 Sensitivity Comparison

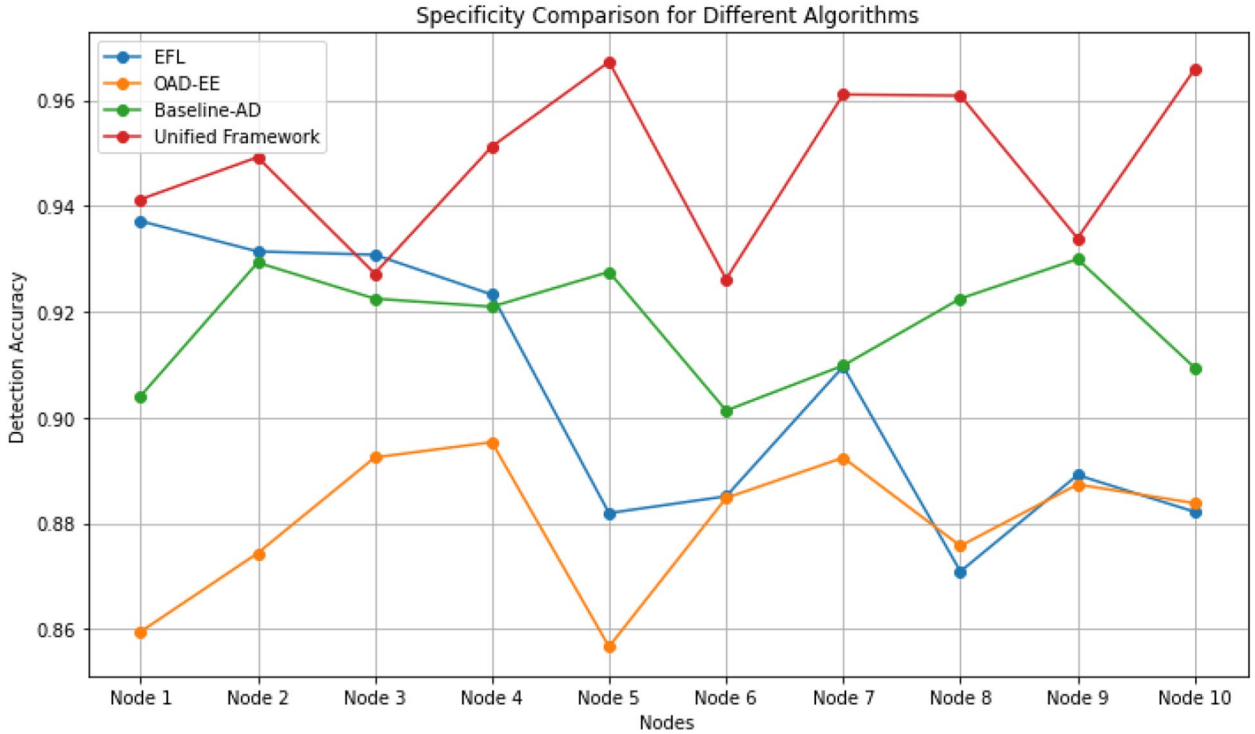

Fig. 8 Specificity Comparison

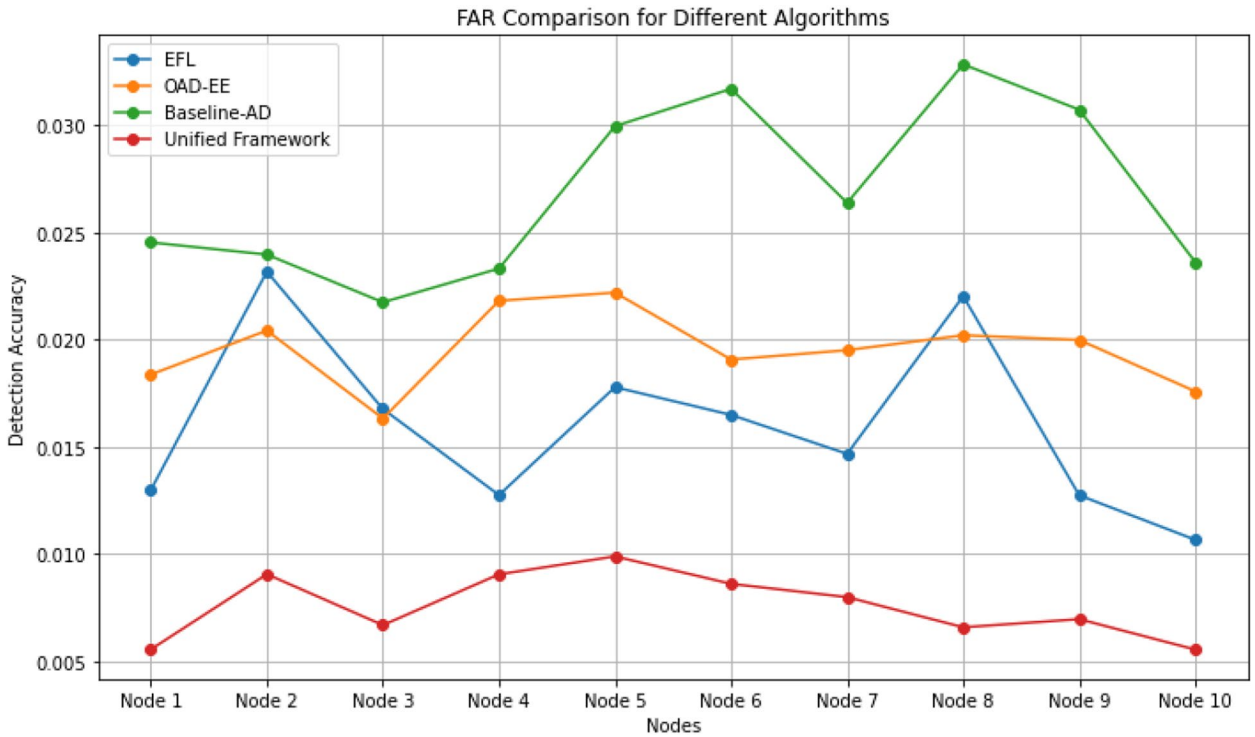

Fig. 9 False Alarm Rate Comparison

Table 1 Comparison of parameters

Technique	Accuracy	Precision	Recall	Specificity	F1 Score	AUC
EFL	0.95	0.96	0.94	0.97	0.79	0.79
OAD-EE	0.87	0.92	0.93	0.93	0.72	0.63
Baseline-AD	0.79	0.85	0.90	0.92	0.69	0.72
Unified framework	0.96	0.97	0.95	0.97	0.86	0.78

The data is prepared for analysis through a series of steps, including data cleaning to address missing or damaged data points, normalization to ensure consistency in sensor readings, feature selection to identify important features for anomaly detection, segmentation to facilitate online detection, and outlier detection to handle extreme or noisy data instances. Segmentation involves dividing a dataset or data stream into smaller segments or subsets. In the context of anomaly detection, segmentation might involve dividing the data stream into smaller chunks to analyze and detect anomalies more effectively within each segment. The dataset is divided into training and testing sets to evaluate the performance of the anomaly detection model. In addition, techniques such as oversampling or undersampling have been used to balance the dataset for effective anomaly detection in real-world scenarios where anomalies are rare compared to normal instances.

In order to combine the machine learning models, random search technique is used. Preprocessing the data, including handling missing values, encoding categorical features, and scaling/normalizing numeric features, is performed. The dataset is split into training (70%) and testing (30%) sets for model evaluation. Multiple machine learning models, including DT, RF, SVM, k-NN, and ANN, have been selected for combination. Hyperparameters for each individual model have been tuned using random search to find the best set of hyperparameters within specified ranges. After tuning individual models, their predictions have been combined using stacking, a weighted averaging method. These are ensemble learning techniques used to combine predictions from multiple models. Weighted averaging assigns different weights to the predictions of individual models before combining them, while stacking involves training a meta-learner that learns how to combine the predictions of the base models.

The weights for each model have been defined to determine the importance of their prediction in the final ensemble, ensuring their sum is equal to 1. The weights

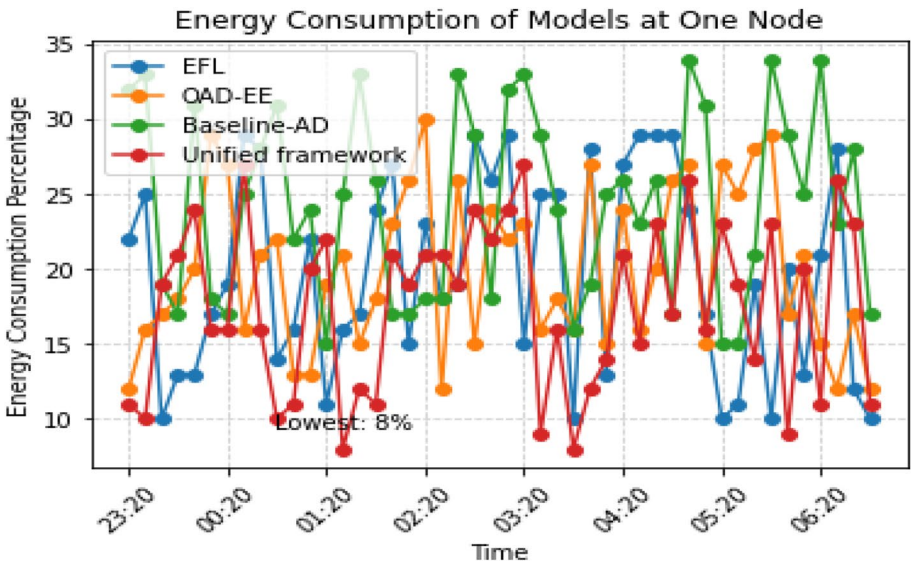

Fig. 10 Energy consumption comparison

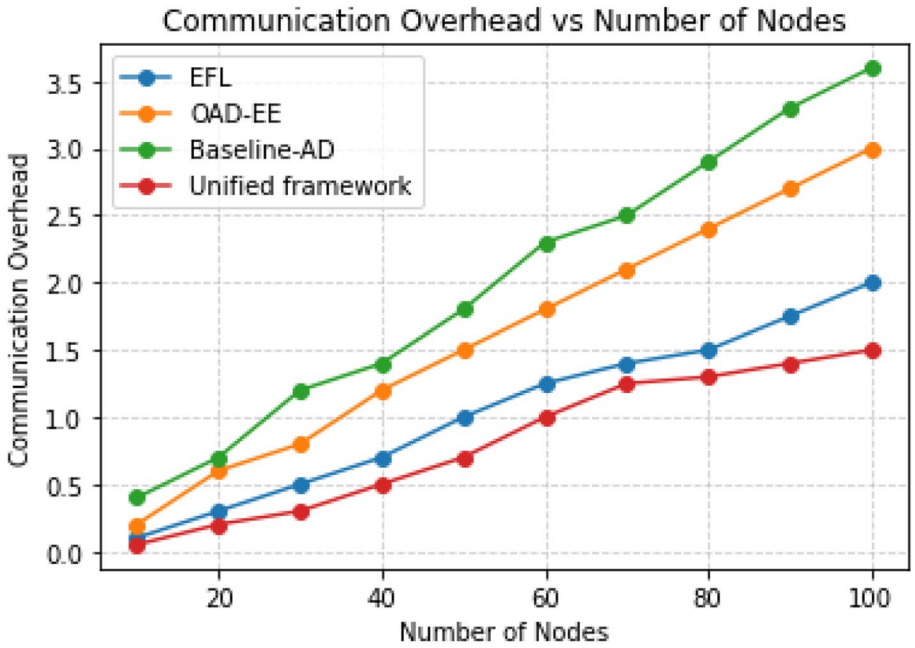

Fig. 11 Communication overhead comparison

are assigned as follows: DT (0.1), RF (0.2), SVM (0.2), k-NN (0.2), and ANN (0.3). With the defined weights, the chosen ensemble method has been applied to combine the predictions of individual models. For weighted averaging, a weighted average of the predictions has been calculated. For stacking, a meta-learner has been trained on top of the predictions from individual models. The performance of the combined model has been evaluated using appropriate evaluation metrics, such as accuracy, F1 score, and AUC. Cross-validation has been used to obtain a more reliable estimate of the model's performance. The convergence of the algorithms is analyzed for 100 iterations, as shown in Fig. 4. This refers to the point where iterative algorithms reach stability or stop changing significantly with further iterations. Analyzing the convergence involves observing how the performance or behavior of algorithms changes over a specific number of iterations or epochs.

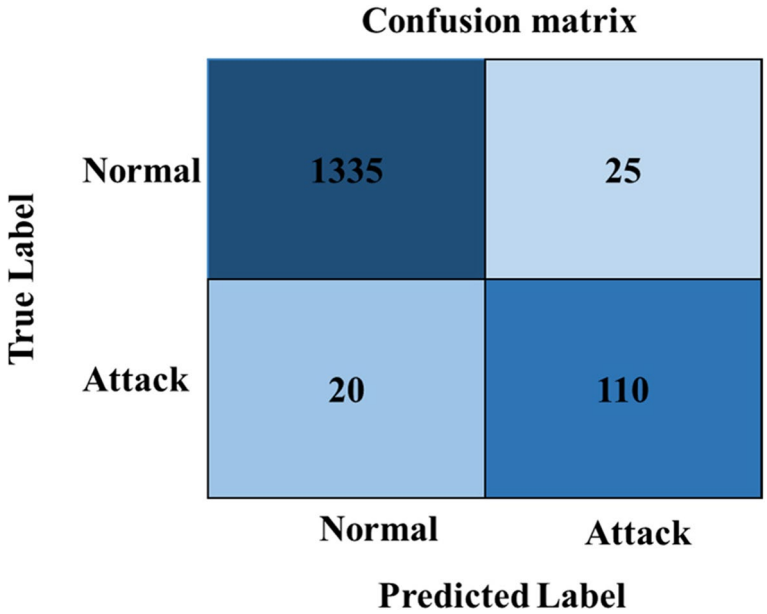

Fig. 12 Confusion Matrix of the unified framework

Table 2 Comparison of EFL with state-of-the-art federated learning models

Model	Model Acccuracy	Communication Overhead	Privacy Preservation	Training Convergence
Federated Learning with Secure Agg [16]	91%	1.2 MB	High	34 epochs
Federated Averaging (FedAvg) [32]	88%	1.5 MB	Low	28 epochs
EFL	95%	1.0 MB	High	9 epochs

The experimental results of the proposed EFL, OAD-EE, and a benchmark Baseline-AD are presented below. Figure 5 compares the detection accuracy achieved by each algorithm for 10 sample nodes. Higher accuracy indicates better anomaly detection performance. It is observed that EFL offers better detection accuracy among others. Figure 6 provides the comparison of precision. A higher precision represents that the algorithm has fewer false alarms, indicating that the detected anomalies are more likely to be true anomalies. A higher precision value suggests that the algorithm's detections are reliable and trustworthy.

Figure 7 provides the comparison of sensitivity (recall). A higher sensitivity means that the algorithm is effective at capturing a larger proportion of actual anomalies. It indicates the algorithm's ability to detect true anomalies among all the anomalies present. Figure 8 compares the sensitivity. A higher specificity means that the algorithm is better at correctly identifying normal instances as normal. This is important to avoid unnecessary alerts for normal operations.

Figure 9 provides the comparison of false alarm rate. The false alarm rate is crucial in assessing the accuracy of anomaly detection systems in WSNs. It measures the frequency of false alarms or false positives generated by the system. A lower false alarm rate suggests higher reliability and accuracy. Striking a balance between detecting real anomalies and minimizing false alarms is essential. A high false alarm rate can lead to disruptions, waste of resources and loss of trust in the system. In wireless sensor networks, controlling the false alarm rate is vital to ensure efficient resource utilization and effective anomaly detection. It is observed that the EFL model outperforms the other models in this parameter.

Table 1 provides the comparison of various parameters involved in the performance analysis of the proposed model. The Unified framework beats individual techniques in detecting anomalies. It has higher accuracy, precision, recall, and F1 Score. Despite slightly lower AUC, the unified framework offers best performance.

Figure 10 provides the comparison of energy consumption in percentage for all the models. Lower energy consumption indicates more energy-efficient anomaly detection. The analysis is performed for seven hours at a single node. It is observed that the OAD-EE technique optimizes energy by approximately 25% when compared to the baseline approach. The unified framework optimizes maximum energy. Figure 11 provides a comparison of the communication overhead of the algorithms. Lower communication overhead indicates reduced data transmission in WSN. It is observed that the unified framework offers the least communication overhead.

Figure 12 provides the confusion matrix of the unified framework for anomaly detection. The anomaly detection results are evaluated using incident labels. All true positive and true negative ratios are increased while all false alarms are decreased. Tables 2 and 3 provides the comparison the EFL and OAD-EE with state-of-the-art federated learning and anomaly detection models as tested with a common dataset used for this research.

Table 4 provides a comparison of the computational complexity of the three algorithms. Further, the scalability analysis, cloud resource utilization, and cost–benefit analysis are performed for the proposed model. It is observed that the unified framework offers improved performance in all these analyses.

Implementing and fine-tuning ensemble learning, as well as combining various machine learning models, can be complex and computationally intensive. This can be especially challenging in resource-constrained environments like WSNs. Additionally, the effectiveness of the proposed ensemble approach and machine learning

Table 3 Comparison of OAD-EE with state-of-the-art anomaly detection models

Model	Detection Accuracy	Energy Efficiency	Scalability	Latency (ms)
OAD-EE	87%	High	High	9
Isolation Forest [33]	82%	Medium	Medium	18
One-Class SVM [34]	78%	Low	Low	14
Recurrent Neural Networkd (RNN) [35]	87%	Low	Medium	13

Table 4 Comparison of computational complexity

Algorithm	Training Time (s)	Testing Time (s)	Number of model parameters
EFL	345	9	1335
OAD-EE	452	15	12,674
Baseline-AD	524	17	18,465
Unified Framework	376	15	5673

models may heavily depend on hyperparameter tuning, which might require expertise and significant computational resources. While cloud integration can improve scalability and efficiency, it may also raise concerns about data privacy and security. Therefore, it's essential to ensure secure data transmission and processing in the cloud. The performance of the proposed techniques might vary across different WSN environments. Therefore, it's important to validate the models' adaptability to diverse scenarios and their generalization beyond the dataset used for evaluation. Deploying complex algorithms and cloud-integrated systems might pose challenges in practical implementation, maintenance, and real-time adaptability in industrial settings.

Conclusion and Future Scope

There is a need for optimized anomaly detection in WSNs. To achieve this, new and improved techniques are explored that prioritize accuracy, energy efficiency, and scalability. By optimizing resource efficiency, energy consumption can be reduced, and communication overhead can be minimized. The results of experiments show that integrating EFL and Online Anomaly Detection with Energy-Efficient Techniques with cloud computing outperforms traditional anomaly detection methods in terms of accuracy, false positive rate, energy consumption, and communication overhead. This approach leverages the power of cloud computing to improve scalability, resource utilization, and real-time response in detecting anomalies in WSNs. The unified framework offers 96% accuracy, 97% precision, 95% recall, 97% specificity, 0.86 F1 score and 0.78 AUC. These findings demonstrate that using machine learning and federated learning techniques in conjunction with cloud computing can significantly improve anomaly detection performance while considering resource constraints in WSN environments. The inclusion of cloud computing in this research enhances the system's overall efficiency. It enables seamless model training, aggregation, and real-time data analytics. The cloud's scalability and elastic resources empower the system to handle larger WSNs and adapt to changing workloads effectively. Additionally, cloud-based data storage ensures seamless long-term data retention and easy retrieval, facilitating comprehensive analysis and insights. The impressive accuracy and energy efficiency of EFL makes it a compelling and viable approach for real-world anomaly detection in WSNs. This research lays the foundation for future investigations to explore the adaptability and performance of these algorithms in more extensive and complex WSN deployments. Furthermore, optimizing cloud integration and security measures can safeguard sensitive sensor data and ensure privacy during cloud-based data processing.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13677-024-00595-y.

Additional file 1.

Authors' contributions

The author S.Gayathri the author contributed data analysis, took part in the paper's background research, and supported the mathematical derivations. On paper, the author made an effort to organise it and incorporated Reviewer comments also. Technically, contributed data analysis as well. The author Dr.D.Surendran incorporated Reviewer comments also technically, participated, reviewed the facts, and assisted with text editing prepared tables and figures as well participated and contributed to the creation of the mathematical equation.

Funding

No funding was received to assist with the preparation of this manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Received: 17 August 2023 Accepted: 9 January 2024
Published online: 23 February 2024

References

1. Liu Y, Garg S, Nie J, Zhang Y, Xiong Z, Kang J, Hossain MS (2020) Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach. IEEE Internet Things J 8(8):6348–6358. https://doi.org/10.1109/JIOT.2020.3011726
2. Mahesh B (2020) Machine learning algorithms-a review. International Journal of Science and Research (IJSR) 9(1):381–386
3. Gebremariam GG, Panda J, Indu S (2023) Blockchain-based secure localization against malicious nodes in IoT-based wireless sensor networks using federated learning. Wirel Commun Mob Comput 2023:1–27. https://doi.org/10.1155/2023/8068038
4. Deng B, Chen X, Chen X, Pei S, Wan and SK Goudos Trusted Edge Computing System Based on Intelligent Risk Detection for Smart IoT, in IEEE Transactions on Industrial Informatics https://doi.org/10.1109/TII.2023.3245681
5. Lu J, Pan B, Seid AM, Li B, Hu G and Wan S Truthful Incentive Mechanism Design via Internalizing Externalities and LP Relaxation for Vertical Federated Learning, in IEEE Transactions on Computational Social Systems https://doi.org/10.1109/TCSS.2022.3227270.
6. Lin Y et al (2023) DRL-Based Adaptive Sharding for Blockchain-Based Federated Learning. IEEE Trans Commun 71(10):5992–6004. https://doi.org/10.1109/TCOMM.2023.3288591

7. Xie M, Han S, Tian B, Parvin S (2011) Anomaly detection in wireless sensor networks: A survey. J Netw Comput Appl 34(4):1302–1325. https://doi.org/10.1016/j.jnca.2011.03.004
8. Dwivedi RK, Rai AK, & Kumar R (2020) A study on machine learning based anomaly detection approaches in wireless sensor network. In Data Science and Engineering (Confluence) 10th International Conference on Cloud Computing, 2020 (pp. 194–199). IEEE Publications. https://doi.org/10.1109/Confluence47617.2020.9058311
9. Cauteruccio F, Fortino G, Guerrieri A, Liotta A, Mocanu DC, Perra C, Terracina G, Torres Vega MT (2019) Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance. Information Fusion 52:13–30. https://doi.org/10.1016/j.inffus.2018.11.010
10. Stephanie V, Khalil I, Atiquzzaman M, & Yi X (2022) Trustworthy privacy-preserving hierarchical ensemble and federated learning in healthcare 4.0 with blockchain. IEEE Transactions on Industrial Informatics, 19(7), 7936–7945. https://doi.org/10.1109/TII.2022.3214998
11. Huang Z, Wu Y, Tempini N, Lin H, Yin H (2022) An energy-efficient and trustworthy unsupervised anomaly detection framework (EATU) for IIoT. ACM Transactions on Sensor Networks 18(4):1–18. https://doi.org/10.1145/3543855
12. Hu W, Liao Y, Vemuri R (2003) Robust support vector machines for anomaly detection in computer security 168–174
13. Singh VK, & Govindarasu M (2018) Decision tree based anomaly detection for remedial action scheme in smart grid using pmu data. In IEEE Power and Energy Society General Meeting (PESGM). IEEE Publications, 2018, (1–5). https://doi.org/10.1109/PESGM.2018.8586159
14. Biswas P, Samanta T (2021) Anomaly detection using ensemble random forest in wireless sensor network. Int J Inf Technol 13(5):2043–2052. https://doi.org/10.1007/s41870-021-00717-8
15. Lorenz J, Silva MI, Aparício D, Ascensão JT, & Bizarro P (2020). Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. In Proceedings of the First ACM International Conference on AI in Finance (pp. 1–8). https://doi.org/10.1145/3383455.3422549
16. Mothukuri V, Khare P, Parizi RM, Pouriyeh S, Dehghantanha A, Srivastava G (2021) Federated-learning-based anomaly detection for IoT security attacks. IEEE Internet Things J 9(4):2545–2554. https://doi.org/10.1109/JIOT.2021.3077803
17. Ifzarne S, Tabbaa H, Hafidi I, & Lamghari N (2021) Anomaly detection using machine learning techniques in wireless sensor networks. In Journal of Physics: Conference Series (Vol. 1743, No. 1, p. 012021). IOP Publishing, 1743(1). https://doi.org/10.1088/1742-6596/1743/1/012021
18. Kaur A, Pal SK, Singh AP (2018) Hybridization of K-means and firefly algorithm for intrusion detection system. International Journal of System Assurance Engineering and Management 9(4):901–910. https://doi.org/10.1007/s13198-017-0683-8
19. Emy Dorfman LE, Leite JCL, Giugliani R, Riegel M (2015) Microarray-based comparative genomic hybridization analysis in neonates with congenital anomalies: Detection of chromosomal imbalances. Jornal de Pediatria 91(1):59–67. https://doi.org/10.1016/j.jped.2014.05.007
20. Li J (2008) Distributed estimation in resource-constrained wireless sensor networks. Georgia Institute of Technology. http://hdl.handle.net/1853/26633
21. Barr JR, Shaw P, Abu-Khzam FN, & Chen J (2019) Combinatorial text classification: The effect of multi-parameterized correlation clustering. In First International Conference on Graph Computing (GC), 2019 (pp. 29–36). IEEE Publications. https://doi.org/10.1109/GC46384.2019.00013
22. Khan RA, Mohammadani KH, Soomro AA, Hussain J, Khan S, Arain TH et al (2018) An energy efficient routing protocol for wireless body area sensor networks. Wireless Pers Commun 99. https://doi.org/10.1007/s11277-018-5285-5
23. Ferreira AC, Vilaça MA, Oliveira LB, Habib E, Wong HC, Loureiro AA (2005) On the security of cluster-based communication protocols for wireless sensor networks 449–458. https://doi.org/10.1007/978-3-540-31956-6_53
24. Ali H, Tariq UU, Hussain M, Lu L, Panneerselvam J, Zhai X (2021) ARSH-FATI: A Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks. IEEE Syst J 15(2):2386–2397. https://doi.org/10.1109/JSYST.2020.2986811
25. Ali H, Tariq UU, Hardy J, Zhai X, Lu L, Zheng Y, Bensaali F, Amira A, Fatema K, Antonopoulos N (2021) A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics. Computer Science Review 41:100416
26. OReilly, C., Gluhak, A., Imran, M. A., & Rajasegarar, S. (2014) Anomaly detection in wireless sensor networks in a non-stationary environment. IEEE Communications Surveys and Tutorials 16(3):1413–1432. https://doi.org/10.1109/SURV.2013.112813.00168
27. Haque SA, Rahman M, Aziz SM (2015) Sensor anomaly detection in wireless sensor networks for healthcare. Sensors 15(4):8764–8786. https://doi.org/10.3390/s150408764
28. Rajasegarar S, Leckie C, Palaniswami M (2008) Anomaly detection in wireless sensor networks. IEEE Wirel Commun 15(4):34–40. https://doi.org/10.1109/MWC.2008.4599219
29. Schneble W, Thamilarasu G (2019) Attack detection using federated learning in medical cyber-physical systems
30. Ghimire B, Rawat DB (2022) Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. IEEE Internet Things J 9(11):8229–8249. https://doi.org/10.1109/JIOT.2022.3150363
31. Chen Z, Lv N, Liu P, Fang Y, Chen K, Pan W (2020) Intrusion detection for wireless edge networks based on federated learning. IEEE Access 8:217463–217472. https://doi.org/10.1109/ACCESS.2020.3041793
32. Ihekoronye VU, Nwakanma CI, Kim DS, Lee JM (2023) DATA-FedAVG: delay-aware truncated accuracy-based federated averaging for intrusion detection in UAV network. J Kor Inst Commun Inform Sci 48:648–668. https://doi.org/10.7840/kics.2023.48.6.648
33. Yang X, Chen Y, Qian X, Li T, Lv X (2021) BCEAD: a blockchain-empowered ensemble anomaly detection for wireless sensor network via isolation forest. Security and Communication Networks 2021:1–10
34. Trinh V-V, Tran KP, Huong T (2017) Data driven hyperparameter optimization of one-class support vector machines for anomaly detection in wireless sensor networks 6–10. https://doi.org/10.1109/ATC.2017.8167642
35. Haque A, Chowdhury NU, Soliman H, Hossen MS, Fatima T, Ahmed I (2023) Wireless sensor networks anomaly detection using machine learning: a survey. https://doi.org/10.48550/arXiv.2303.08823

Publisher's Note