

Fog-Aware Hierarchical Autoencoder with Latent-Space Gating and Density-Based Clustering for Privacy-Preserving Threat Detection in IoT-Enabled Smart Farming Environments

Manikandan Thirumalaisamy

Anna University, Chennai

Sumendra Yogarayan

sumendra@mmu.edu.my

Multimedia University

Md. Shohel Sayeed

Multimedia University

Siti Fatimah Abdul Razak

Multimedia University

Ramesh Shunmugam

Anna University, Chennai

Research Article

Keywords: Anomaly Detection, Unsupervised learning, Fog Computing, DBSCAN Cluster-ing, Agricultural IoT security, CIC IoT-DIAD 2024

Posted Date: September 18th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7449865/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Smart farming's IoT-driven automation and real-time analytics enhance productivity but simultaneously expose heterogeneous, resource-limited devices to advanced cyber threats. Flow-based, unsupervised threat detection offers a privacy-preserving alternative to signature-based methods, yet key challenges remain: autoencoders may lose subtle anomaly cues during compression, reconstruction-error thresholds can fail under dynamic traffic conditions, and density-based clustering often degrades in noisy, high-dimensional spaces. To address these issues, we propose a fog-aware, two-stage hierarchical autoencoder (AE) with latent-space gating, followed by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for unsupervised attack categorization. In Stage 1, a shallow AE performs progressive dimensionality reduction, producing a compact 21-dimensional latent representation that preserves nonlinear patterns while meeting fog-node constraints. In Stage 2, a deep AE applies reconstruction-error scoring to detect anomalies, inherently denoising and isolating malicious behaviors. Only high-error latent vectors are forwarded to DBSCAN, improving separation between threat types, mitigating noise, and avoiding predefined cluster counts. Evaluated on the CIC IoT-DIAD 2024 dataset, the framework achieves 98.99% accuracy, 0.9897 F1-score, an Adjusted Rand Index (ARI) of 0.895, and a Davies–Bouldin Index (DBI) of 0.019, surpassing state-of-the-art methods in detecting and categorizing DoS, scanning, botnet, and man-in-the-middle attacks without labeled data or fixed signatures.

1 Introduction

Smart farming, or precision agriculture, marks a transformative leap in agricultural practices through the integration of advanced digital technologies—particularly the Internet of Things (IoT), artificial intelligence (AI), big data analytics, and cloud computing—into traditional operations [1]. This paradigm shift addresses the increasing demand for sustainable agricultural models that maximize crop yields, optimize resource use, and minimize environmental impact [2]. At the core lies IoT-enabled infrastructure, enabling seamless communication among distributed sensors, actuators, and intelligent devices deployed across the agricultural landscape. These devices continuously capture critical environmental and operational data—such as soil moisture, temperature, humidity, crop health indices, pest presence, and livestock well-being [3]. Processing occurs either in centralized cloud servers or, increasingly, at fog and edge nodes closer to the source, ensuring low-latency, context-aware decision-making. IoT-based irrigation systems activate only when soil moisture falls below defined thresholds, conserving water while maintaining yield. Predictive maintenance can detect machinery anomalies before failure, minimizing downtime and associated costs [4].

While IoT adoption offers agility and operational advantages over traditional farming [5], [6], it also transforms smart farms into complex cyber–physical systems highly exposed to cybersecurity threats. The heterogeneous architecture—comprising wireless sensor networks, actuators, fog/edge nodes, and cloud services—broadens the attack surface [7]. Common threats include wireless jamming, spoofing, eavesdropping, and denial-of-service (DoS) attacks [8]. More severe are control manipulation attacks,

which can hijack autonomous machinery, greenhouse climate control, or irrigation scheduling [9], leading to economic losses and physical damage. Data breaches can expose sensitive field maps, proprietary farming techniques, or livestock health records [10] often due to weak authentication, insecure communication protocols, or outdated firmware [11].

The threat environment is intensified by the shortage of cybersecurity expertise in agriculture [12]. Many farms operate with legacy devices lacking built-in protections, rely on fragmented vendor-specific security tools [13], and have limited capacity for incident response. IoT-targeted malware, ransomware, and botnets highlight the urgency for robust solutions—most notably, the Mirai botnet incident demonstrated how compromised endpoints can be exploited for massive distributed denial-of-service (DDoS) campaigns. Proactive defense in this context requires secure hardware, segmented network architectures, strong encryption, fine-grained access controls, and behavioral threat monitoring. Real-time intrusion detection capable of recognizing anomalies and zero-day exploits is especially vital. Traditional signature-based intrusion detection systems (IDS) fail against novel threats, while deep packet inspection is computationally intensive and privacy-invasive, making it unsuitable for fog/edge IoT nodes [14].

Flow-based anomaly detection has emerged as a scalable, privacy-preserving alternative, analyzing aggregated flow-level metadata (IP addresses, ports, protocols, byte/packet counts, temporal statistics) rather than payloads [15]. Such methods can be deployed in fog-aware architectures, reducing latency and computational demands while protecting data privacy.

In agricultural IoT, the scarcity of labeled datasets makes unsupervised learning an attractive option. Yet, deep technical challenges remain in the two most promising building blocks for unsupervised detection—autoencoders (AEs) and density-based clustering (DBSCAN). Autoencoders, while powerful for nonlinear dimensionality reduction, can lose subtle anomaly cues during compression, particularly in high-dimensional, noisy traffic. DBSCAN, though capable of discovering arbitrarily shaped clusters and handling noise, struggles when applied directly to high-dimensional raw features and is sensitive to parameter selection in noisy environments.

To overcome these limitations, we propose a fog-aware, two-stage hierarchical autoencoder with latent-space gating, followed by DBSCAN clustering. In Stage 1, a shallow autoencoder performs progressive dimensionality reduction, compressing high-dimensional flow data into a compact 21-dimensional latent space that preserves critical nonlinear feature interactions while meeting the computational constraints of fog nodes. In Stage 2, a deep autoencoder reconstructs these latent vectors to compute reconstruction-error scores, effectively distinguishing benign from anomalous flows and inherently denoising latent representations. Only high-error latent vectors are forwarded—via a latent-space gating mechanism—to the final clustering step. DBSCAN then categorizes anomalies without requiring a predefined number of clusters. This gated approach reduces DBSCAN's exposure to benign noise, enhances cluster separability for different attack patterns, and improves resilience to encrypted or obfuscated traffic. By integrating latent-space representation learning with selective clustering, the

proposed method enables adaptive, privacy-preserving, and highly accurate threat detection in heterogeneous agricultural IoT networks. The framework is evaluated on the CIC IoT-DIAD 2024 dataset, chosen for its relevance to smart farming network scenarios.

This research makes the following key contributions to the field of smart farming cybersecurity:

1. Two-stage hierarchical AE-gated detection: Progressive shallow dimensionality reduction followed by deep reconstruction-error scoring to detect subtle, high-dimensional anomalies.
2. Fog-aware design: Latent-space processing optimized for real-time execution on resource-constrained fog nodes.
3. Noise-resilient clustering: DBSCAN applied exclusively to high-error latent vectors, improving anomaly separation and reducing sensitivity to noise.
4. Extensive evaluation: Achieved 98.99% accuracy, 0.9897 F1-score, 0.895 ARI, 0.019 DBI on CIC IoT-DIAD 2024—surpassing state-of-the-art methods.
5. Broad-spectrum detection: Effective against DoS, scanning, botnet, and MitM attacks without labeled data or fixed signatures.

Unlike existing approaches that treat dimensionality reduction and clustering as independent steps, our framework integrates them through a latent-space gating mechanism, addressing AE compression loss, adapting to dynamic IoT traffic, and improving DBSCAN's noise robustness—critical for privacy-preserving intrusion detection in encrypted, resource-limited smart farming environments.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of existing intrusion detection techniques, emphasizing their strengths and limitations. Section 3 describes the proposed methodology, including the system architecture. Section 4 outlines the experimental setup, evaluation metrics, and performance analysis, followed by a comparison of the proposed model with existing approaches. Finally, Section 5 concludes the study and highlights future research directions.

2 Related Survey

Intrusion Detection Systems (IDS) represent a cornerstone of current-day cybersecurity frameworks, especially against the backdrop of mounting complexity and sophistication of cyber threats across cloud, IoT, and mobile environments. Recent research has thoroughly analyzed a broad range of IDS approaches, from traditional machine learning to sophisticated deep learning and hybrid frameworks, all with focus on maximizing detection accuracy, scalability, and responsiveness. This section presents an exhaustive synthesis of the latest IDS advancements, focusing on their architectural approaches, utilized datasets, essential contributions, and built-in limitations.

The paper in [16] proposes a hybrid IDS framework that synergistically combines machine learning (ML) and deep learning (DL) paradigms namely XGBoost, Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks. The suggested pipeline utilizes XGBoost and CNN for feature extraction with stability, then uses LSTM for sequential classification. Experimental assessments over

four benchmark datasets—CIC IDS 2017, UNSW NB15, NSL KDD, and WSN-DS—illustrate substantial improvements in detection precision and lower false positives. Nonetheless, the method is afflicted by high computational overhead and architectural complexity, which can hamper its deployment in latency-critical or resource-limited environments.

In [17], the authors introduce an IDS designed for cloud environments through a hybrid of Graph Neural Networks (GNN) and Leader K-means clustering, further refined through the Grasshopper Optimization Algorithm (GHO). The model integrates AES encryption and steganographic methods to ensure robust data protection. On the ADFA IDS dataset, the system provides improved classification accuracy (79.84%) and clustering performance (85.91%). Despite such improvements, the method requires excessive memory and processing power, especially at the clustering and classification phases, which might hamper its applicability in dynamic, massive-scale cloud implementations.

A GAN-based IDS that is specific to the Social Internet of Things (SIoT) in Collaborative Edge Computing (CEC) settings is suggested in [18]. The system processes both one and multi-attack cases of detection using a multi-GAN framework for threat detection and feature learning. Performance on the CSE-CIC-IDS2018 and CIC-DDoS2019 datasets shows high accuracy, precision, and recall, with low false alarm rates. The approach does, however, show lower effectiveness against low-volume, concealed attacks like WebDDoS and Brute Force-XSS and could have difficulty handling high-dimensional inputs, which might cause real-time application in edge environments to be challenging.

The work in [19] introduces an unsupervised IDS that detects zero-day Distributed Denial of Service (DDoS) attacks on IoT networks. The model uses Gaussian Random Projection (GRP) for dimensionality reduction followed by an ensemble of K-means, Stochastic Gradient Descent One-Class SVM (SGD-OCSVM), and Gaussian Mixture Models (GMM) that are combined through hard voting. Evaluated on the CIC-DDoS2019 dataset, the ensemble recorded a significant detection accuracy of 94.5%. Though effective, its computational requirements and scalability issues make it difficult for deployment in real-time, high-volume environments.

In [20], the authors present an adaptive botnet detection model using a hybrid unsupervised DBSCAN-Grey Wolf Optimizer (GWO) mechanism for IoT networks. In this, the DBSCAN algorithm is adaptively adjusted through GWO to fine-tune the eps parameter for every incoming data stream, allowing botnet detection in real-time without labels. The evaluation standard was the N_BaloT dataset, comprising six different IoT device traffic, with the model recording an impressive accuracy level of 98%. There is a limitation pointed out in the sensitivity of the model towards noisy data, which can negatively impact cluster precision and reliability.

The paper in [21] investigates a better phishing detection method with an optimized K-Means clustering algorithm that uses CfsSubsetEval for the best feature selection. Using an 11,430 URL dataset with 87 unique features, the research measures model performance for different sample sizes (2,000; 7,000; and 10,000) with accuracy rates of 89.2%, 86.62%, and 86.25%, respectively, and performs better than the conventional kernel K-Means method. As the research emphasizes the need for proper feature selection,

it also addresses issues regarding dataset diversity and the scalability of the algorithm under real-time conditions.

In [22], the researchers aim mobile network security via anomaly detection and predictive modeling using K-means clustering on Call Detail Records (CDRs). The data set contains 14 million CDRs gathered in a one-year span. The process combines deseasonalization and Mahalanobis distance metrics to improve discrimination of anomalies, with 96% accuracy in detection. While the methodology evinces effective pattern discovery and predictive performance, its precision and recall values suggest possible improvement in subtle threat detection cases.

In [23], a federated transfer learning (FTL)-based IDS is introduced for cooperative smart farming infrastructures. It enables distributed edge nodes to train locally on IoT traffic and share only model updates, preserving privacy. MobileNetV2-based transfer learning is used for feature extraction, with federated aggregation reducing data transfer needs. Evaluations on CIC-IDS2017 and NSL-KDD datasets achieved > 98% accuracy, though the reliance on stable connectivity and the use of generalized datasets limit direct agricultural applicability.

The authors in [24] propose a hybrid feature selection method combining Mutual Information and ReliefF, integrated with shallow and deep artificial neural networks for anomaly detection in smart farming traffic. Using CIC-IDS2017 subsets, the hybrid selection reduced training time by 35% and increased F1-score by ~ 4%, but the offline, supervised approach and non-agricultural dataset reduce real-time relevance.

Despite significant progress in intrusion detection methodologies across various domains, including cloud, IoT, and edge environments, several critical challenges remain unaddressed—especially when these methods are applied to smart farming’s unique cyber-physical ecosystem. The constrained computational resources of fog and edge nodes, combined with the dynamic and encrypted nature of agricultural IoT traffic, complicate traditional anomaly detection approaches. High-dimensional and noisy data further degrade clustering performance, while fixed reconstruction-error thresholds in autoencoder-based models struggle to adapt to fluctuating network behaviors. These persistent limitations highlight a pressing need for innovative IDS frameworks that are lightweight, adaptive, and privacy-preserving, yet capable of robustly detecting and categorizing threats in real-time within the heterogeneous, resource-limited smart farming environment.

Overall, the examined literature demonstrates the accelerated evolution and differentiation of IDS methods, with considerable advances in hybrid modeling, adaptive optimization, and unsupervised learning. These methods cover a wide range of computational paradigms and domains of application. Common challenges persist, however, including effective management of high-dimensional input data, real-time processing constraints on resource-limited nodes, and the reliable identification of complex or low-signature attack patterns. These challenges are particularly acute in precision agriculture’s heterogeneous, resource-constrained IoT environments, where privacy concerns limit payload inspection, network dynamics complicate fixed thresholding, and noisy data hinders traditional clustering

techniques. This gap underscores the urgent need for lightweight, adaptive, and noise-resilient intrusion detection frameworks—capable of preserving user privacy while maintaining high detection accuracy in encrypted, dynamic smart farming networks.

3 Proposed Methodology

The proposed methodology, illustrated in Fig. 1, presents a fog-aware detection and categorization framework for smart farming IoT environments, leveraging the CIC IoT-DIAD 2024 dataset. The pipeline begins with data preprocessing—including integration, cleaning, and normalization—to ensure uniform feature scaling and compatibility for learning. In the first stage of the hierarchical autoencoder pipeline, a shallow autoencoder performs progressive dimensionality reduction, compressing high-dimensional flow features into a compact 21-dimensional latent space. This preserves nonlinear traffic patterns while reducing computational load for deployment in resource-constrained fog nodes. In the second stage, a deep autoencoder reconstructs these latent representations to compute reconstruction-error scores, distinguishing benign from anomalous flows while inherently denoising the latent features. Only high-error latent vectors are passed—via a latent-space gating mechanism—to the final categorization step. The final step applies density-based clustering (DBSCAN) exclusively to the gated anomaly vectors, producing coherent attack categories without requiring pre-defined cluster counts. This coupling of two-stage representation learning with noise-resilient clustering enhances detection accuracy, improves separation between threat types, and mitigates DBSCAN's sensitivity to noise—ultimately strengthening the cybersecurity posture of smart agricultural systems.

3.1 Dataset Overview

The CIC IoT-DIAD 2024 dataset [25] is a recent, flow-based intrusion detection dataset developed by the Canadian Institute for Cybersecurity (CIC) to represent realistic network traffic within IoT ecosystems. It captures diverse communication patterns among various IoT devices—such as smart sensors, actuators, and automation hubs—under both benign and malicious conditions. The dataset includes multiple attack vectors such as Denial of Service (DoS), scanning, man-in-the-middle (MitM), botnet operations, and command-and-control (C2) behavior. Each instance is represented as a network flow with 84 features, making it suitable for traffic-based anomaly detection without relying on payload inspection. This dataset is particularly relevant for smart farming cybersecurity, where IoT-based automation and wireless sensing are widely used in crop monitoring, irrigation control, and autonomous machinery. Its flow-based format makes it ideal for fog-layer deployment, where intermediate computing nodes can analyze network flows close to the data source while balancing real-time detection needs and computational feasibility. The diversity and labeling of attacks in CIC IoT-DIAD provide a robust foundation for training and evaluating unsupervised models intended to operate in fog-enabled smart agricultural infrastructures.

3.2 Data Preprocessing

3.2.1 Feature Selection and Cleaning.

The raw (r_D) CIC IoT-DIAD 2024 dataset contains numerous features, including identifiers and static attributes that offer little value for anomaly detection. During preprocessing, non-numeric and constant-value features are removed. Additionally, fields such as IP addresses, timestamps, and protocol flags are excluded to reduce noise and prevent overfitting. Additionally, missing or infinite values are handled through imputation and consistency checks to ensure data integrity. This step refines the dataset (C_D)

into a clean, feature-rich subset suitable for downstream learning, thereby improving the model's ability to focus on meaningful behavioral patterns in flow-based IoT traffic.

3.2.2 Flow-Based Feature Engineering.

The dataset's core strength lies in its flow-level granularity, which allows for behavioral modeling of IoT device communication. During feature engineering, selected attributes are evaluated to reflect temporal, volumetric, and statistical patterns of traffic flows. This includes flow duration, packet counts, byte rates, and inter-arrival times. These engineered features help capture anomalies related to scanning, flooding, or botnet coordination. By focusing on flow-level characteristics rather than packet-level data, the framework enables scalable analysis at the fog layer while maintaining robustness against encryption and payload obfuscation – common in real-world smart farming cyber threats.

3.2.3 Normalization and Transformation.

To ensure uniform scale across features, the selected numeric attributes are normalized using Z-score standardization (Z-scalar). This technique (1) transforms each feature to have a mean of zero and a standard deviation of one, which is essential for stable training of deep learning models like autoencoders. It prevents features with large numeric ranges from dominating the loss function and supports faster convergence. After normalization, the data (n_D) is ready for dimensionality reduction through the first encoder layer of the autoencoder, which captures essential information in a compressed latent representation while preserving patterns indicative of cyber anomalies.

$$n_D = \frac{(c_D - \mu)}{\sigma}$$

1

3.3 Dimensionality Reduction Using Autoencoder.

High-dimensional data is particularly challenging to the efficient and timely analysis in the context of intrusion detection for smart farming IoT networks, especially in fog-based systems where computational and energy resources are limited. The CIC IoT-DIAD 2024 dataset employed in this work includes 80 preprocessed and normalized flow-based features per instance. While the features are able to capture a wide amount of information relating to traffic behavior, most are redundant or inter-related,

which creates unnecessary complexity in the data. Dimensionality reduction is hence an important preprocessing operation to demote the input prior to being inserted into

clustering models. For this purpose, a deep autoencoder (deep-AE)—an unsupervised neural network—is used to compress the initial 80-dimensional (nD) feature space into the more convenient 21-dimensional (RD) latent space. Whereas most deep autoencoders would employ several stacked layers for both the encoder and decoder, the model utilized herein only makes use of the first fully connected layer of a deep-AE, essentially applying a single encoder layer and a single decoder layer. This minimalistic approach is specifically designed to facilitate light deployment in fog computing scenarios where real-time execution and power consumption are paramount.

The suggested model adopts a hybrid representation learning approach, known as Progressive Dimensionality Reduction with Hierarchical Anomaly Representation, with the dimensionality reduction carried out by the initial hidden layer of the deep autoencoder and the rest of the layers providing hierarchical abstraction to aid in anomaly detection. The architecture includes an 80-neuron input layer directly connected to the first fully connected hidden layer of the encoder, which is the latent space for dimensionality reduction with 21 neurons. The decoder has this same setup in its final hidden layer, decoding the original 80 features from the compressed 21-dimensional form. This initial hidden layer acquires the most descriptive low-dimensional representation of the input data without the overcomplication of the multiple transformation phases. Figure 2 illustrates the deep-AE architecture. The dataset is pre-processed, and normalization is applied using the Z-score prior to training to make sure that all features have unit variance and zero mean to avoid the dominance of the learning process by features having higher ranges. The autoencoder learns to reduce the mean squared error (MSE) between the input and reconstructed output, forcing the model to retain as much information as possible during compression of the latent space. ReLU activation functions are used for non-linear transformations within the encoder, and sigmoid activation is used in the decoder to reconstruct the input. The Adam optimizer is used to enable more rapid convergence and adaptive learning rate adjustment. After training, the encoder alone is maintained. The incoming flow records are mapped using this encoder to generate concise 21-dimensional vectors, which are computationally light for downstream analysis. These latent vectors are then input to detection and clustering algorithms like a deeper autoencoder (deepAE) and DBSCAN to identify and classify cyber threats. This first-layer autoencoder methodology balances representational capability and computational ease. It efficiently reduces high-dimensional input and captures the essential structure of the data, supporting effective anomaly detection. Also, its ease of use and minimal overhead ensure that it is well-suited for implementation on fog-level nodes of smart agriculture networks, where bandwidth, memory, and energy are in short supply.

3.4 Anomaly Detection Using Deep Autoencoder Architecture

The core of the proposed framework is a deep autoencoder (deep-AE), as shown in Fig. 3, which is a symmetric neural network architecture designed to learn compact, meaningful representations of high-dimensional input data in an unsupervised manner. In the context of this study, the deep-AE [26] is leveraged to extract latent features from IoT network flows and facilitate the separation of benign and anomalous traffic patterns through clustering.

3.4.1 Architecture Layers

The envisioned autoencoder model is intended to successfully learn concise representations of network traffic patterns for anomaly detection in fog-enabled smart farming scenarios. It consists of four major elements: the input layer, encoder, bottleneck (latent space), and decoder. The input layer takes in normalized flow-based features obtained from the CIC IoT-DIAD 2024 dataset. After going through a thorough preprocessing pipeline, every data instance is converted to a fixed-size vector of 80 numerical attributes. The features are pruned using dimensionality reduction and feature selection methods to remove redundancy and irrelevance to allow the model to be efficient and generalizable.

The encoder module consists of a sequence of fully connected (dense) layers with progressively decreasing dimensionality. In this work, a deep autoencoder (deepAE) architecture comprising four hidden layers is employed. Each layer incrementally compresses the input, capturing hierarchical abstractions and latent correlations among the flow-level features. As detailed in Section 3.3, the initial hidden layer executes the first stage of compression, producing an intermediate feature embedding that facilitates scalable, low-latency processing on fog nodes. Central to the network is the bottleneck layer, alternatively known as the latent space or code layer. This layer represents the most information-dense and compact representation of the input, retaining the underlying structural and statistical properties of the input while removing noise and unnecessary components. The latent vector produced here is used as the basis for downstream

tasks of anomaly detection, clustering, and threat classification in the overall framework. The decoder module is architecturally identical to the encoder, with dense layers having increasingly higher dimensionality. It transforms the latent representation back into the original input vector, with the output layer having the same original input dimensionality (i.e., 80 features). The error in reconstruction—calculated as the difference between the input and the output of the decoder—is used as the anomaly score, where larger errors correspond to possible departures from learned normal behavior. The entire anomaly detection process, including data preprocessing, encoding, reconstruction, and anomaly scoring, is shown in Fig. 3, outlining the end-to-end procedural flow of the suggested detection mechanism.

3.4.2 Activation Functions and Loss Function

To ensure non-linearity and enhance representational capacity, the ReLU (Rectified Linear Unit) activation function (2) is applied in all encoder and decoder layers, except for the final output layer. ReLU is

computationally efficient and helps address the vanishing gradient problem during training, which is especially important for deep networks.

$$ReLU = \max(0, R_D)$$

2

For the output layer, a Sigmoid activation function (3) is used to produce continuous-valued reconstructions of input features. This design choice enables the model to capture subtle variations in network traffic behavior, which is critical for accurate anomaly detection.

$$Sigmoid = \frac{1}{(1 + e^{(-R_D)})}$$

3

The loss function used for training the autoencoder is Mean Squared Error (MSE), which measures the reconstruction error (4) between the original input and its reconstruction. Minimizing this error helps the model learn compact representations that preserve the most informative aspects of the data. Flows with significantly high reconstruction errors are considered anomalous, as the autoencoder has not learned to accurately reproduce them—indicating deviation from normal behavior.

$$RC_D = g((W' * R_D) + b')$$

4

3.4.3 Training Optimization

The deep autoencoder is trained using the Adam optimizer, which combines the advantages of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam enables efficient gradient-based optimization by dynamically adjusting the learning rate for each parameter, making it particularly effective for high-dimensional data such as IoT traffic flows. Early stopping is employed with a defined patience value to prevent overfitting and reduce unnecessary training epochs. This adaptive learning mechanism not only accelerates convergence but also enhances training stability, especially in deep and basic autoencoder architectures.

3.4.5 Threshold-Based Anomaly Identification

To differentiate between benign and anomalous traffic, a threshold is applied to the reconstruction error generated by the autoencoder. Instead of relying on fixed statistical percentiles, this study adopts a standard deviation-based thresholding approach, which is both robust and data-driven.

The threshold is computed using the mean (μ) and standard deviation (σ) of reconstruction errors obtained from the training set. A data point is flagged as anomalous if its reconstruction error exceeds a

threshold (T) defined in (5):

$$T = \mu + k * \sigma$$

5

where k is a configurable hyperparameter that controls the sensitivity of the detection.

This approach ensures that only data points exhibiting significantly higher reconstruction errors than the typical range are considered anomalous. By filtering based on this threshold, the framework minimizes false positives and focuses clustering efforts (e.g., using DBSCAN) on genuinely suspicious traffic patterns, thus improving the accuracy and interpretability of cyber threat detection. Overall, this deepAE architecture is central to the unsupervised anomaly detection pipeline, enabling the transformation of high-dimensional flow-based IoT traffic into a compact latent space suitable for clustering and threat categorization at the fog level.

3.5 Latent-Space Clustering Using DBSCAN

After extracting compact and meaningful latent representations (R_D) from flow-based IoT traffic using a AE, and detecting the anomaly from deepAE, the next crucial step involves clustering these detected anomalous from the encoded data points to identify and categorize anomalous behaviors. In this study, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [27] algorithm is adopted as the primary clustering technique due to its suitability for detecting arbitrarily shaped clusters and handling noise and requires no pre-defined number of clusters in unsupervised settings. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a non-parametric clustering algorithm particularly well-suited for detecting outliers and grouping anomalies in unlabeled data. Unlike K-Means, DBSCAN does not require the number of clusters to be specified in advance and is capable of identifying noise points that do not belong to any cluster: a valuable feature for real-world anomaly detection scenarios.

Algorithm 1 Latent-Space Clustering with DBSCAN for Anomaly Categorization

Input:

- R_D : Original flow-based IoT traffic data
- AE: Trained deep autoencoder model
- ϵ : Neighborhood radius parameter for DBSCAN
- MinPts: Minimum points to form a cluster in DBSCAN
- Threshold: Reconstruction error cutoff to select anomalies

Output:

- Cluster labels for anomalous data points (including noise)

Steps:

1. Encode data to latent space

$R_D = \text{AE.encode}(D)$

2. Compute reconstruction errors for each data point

$\text{Recon_}R_D = \text{AE.decode}(R_D)$

$\text{Errors} = \text{compute_reconstruction_error}(R_D, \text{Recon_}R_D)$

3. Filter latent vectors corresponding to anomalies

$\text{AnomalyIndices} = \{ i \mid \text{Errors}[i] > \text{Threshold} \}$

$R_D^P_{\text{anomalous}} = R_D[\text{AnomalyIndices}]$

4. Apply DBSCAN clustering on filtered latent vectors

$\text{ClusterLabels} = \text{DBSCAN}(R_D^P, \epsilon, \text{MinPts})$

5. Assign noise label (-1) to non-anomalous data points

For i in $\{0 \dots \text{length}(R_D^P)-1\}$ do

If i not in AnomalyIndices then

ClusterLabels_full[i] = -1

Else

ClusterLabels_full[i] = ClusterLabels[corresponding index]

End If

End For

Algorithm 1 Latent-Space Clustering with DBSCAN for Anomaly Categorization

6. Return ClusterLabels_full

In this framework, DBSCAN is applied to the latent vectors generated by the first hidden layer of the deep autoencoder. These vectors represent reduced-dimensional embeddings of the original input and emphasize the underlying structure of the data. The focus is on those latent representations corresponding to data points flagged as anomalous, identified based on high reconstruction error — calculated as the difference between the input and the reconstructed output of the autoencoder. A higher reconstruction error indicates a greater deviation from normal learned patterns, serving as a reliable anomaly score. By filtering and clustering only those data points with reconstruction errors above a defined threshold, the framework enhances DBSCAN's ability to isolate dense anomaly clusters and noise points. This targeted clustering approach improves the interpretability and categorization of malicious behavior in flow-based IoT traffic. The DBSCAN algorithm, detailed in Algorithm 1, uses two key parameters: ϵ (epsilon), which defines the neighborhood radius, and MinPts, the minimum number of points required to form a dense region. These parameters are initially chosen based on heuristics and later fine-tuned through empirical experimentation to optimize cluster quality and improve the separation of abnormal traffic patterns.

4 Experimental Results and Discussion

After extracting dense and semantically informative latent representations (referenced as RD) from flow-based IoT traffic through the autoencoder (AE), and subsequent detection of anomalous points through the deep autoencoder (deepAE), the next crucial step is to cluster these anomalous points. This clustering is unavoidable in the process of discovering unique malicious patterns and enabling categorization of heterogeneous

cyber threats. DBSCAN algorithm is used here as the main clustering approach. DBSCAN is especially favorable for the task in this scenario because it can identify clusters of any shape, is robust against noise and outliers, and does not require a priori knowledge about the number of clusters—thus being particularly adequate for unsupervised anomaly detection applications in dynamic IoT systems. Since it is a non-parametric clustering algorithm, DBSCAN clusters data points based on the local neighborhood density. In contrast to centroid-based algorithms like K-Means, DBSCAN automatically detects noise points that don't fit into any dense region, which is particularly useful in real-world intrusion detection, where unusual behavior tends to vary substantially from typical traffic distributions.

In the proposed framework, DBSCAN is applied specifically to the latent vectors obtained from the first hidden layer of the deepAE, which provides an intermediate-level feature embedding that captures the structural essence of the input data. These vectors correspond to those input samples that exhibit high reconstruction error—a metric computed as the discrepancy between the original input and its reconstructed output from the autoencoder. A large reconstruction error indicates large deviation from the normal traffic learned distribution, hence making it a strong anomaly score. To further improve the

accuracy and interpretability of clustering, only those latent vectors corresponding to reconstruction errors above a specified threshold are transmitted to DBSCAN. This filtered clustering reduces attention to significant anomalies, allowing DBSCAN to better differentiate dense clusters of bad behavior and extract noise points that could correspond to new or low-frequency attack categories.

DBSCAN is based on two hyperparameters with vital importance:

1. ϵ (epsilon): determines the size of the neighborhood surrounding each point.
2. MinPts: determines the minimum number of neighboring points needed to create a dense cluster.

These values are first established via domain-driven heuristics and then further optimized via empirical testing to ensure optimization of clustering quality and improved separation of varied types of anomalies in flow-based IoT traffic. This combination of DBSCAN clustering with autoencoder-based latent representation learning allows context-aware categorization of threats, providing better granularity in distinguishing among various classes of attacks and still being robust to noise and the dynamic nature of cyber threats in smart farming situations.

4.1 Experimental setup

All experiments in this research were conducted using Google Colab, a cloud-based platform that facilitates access to powerful computational resources. The implementation was carried out in Python 3.10, utilizing libraries such as NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow/Keras for data preprocessing, model development, and visualization. To accelerate the training process, a NVIDIA Tesla T4 GPU

was employed through Colab's hosted runtime environment. The combination of a high-performance GPU and a robust Python-based ecosystem enabled efficient handling of deep learning workflows and large-scale datasets. Table 1 presents the dataset configuration and class distribution employed in this study, highlighting the composition of training and validation data along with the feature dimensions and instance counts. Table 2 outlines the hyperparameter settings used during the training process, detailing the configuration choices that guided the optimization and model development.

Table 1
Dataset Configuration and Class
Distribution

Parameter	Value
Training Class	Benign
Validation Class	Benign + Attacks
Feature Dimension	80
Benign Instances	183,595
Attack Instances	231,395

4.2 Performance evaluation metrics

To evaluate the effectiveness of the proposed Deep-AE model, four widely used classification metrics were employed: precision, recall, F1-score, and accuracy. These metrics, derived from the confusion matrix, provide a comprehensive assessment of the model's classification performance. Precision (6) measures the proportion of correctly predicted positive instances among all instances predicted as positive. Recall (7), also known as sensitivity, indicates the proportion of correctly identified positive instances among all actual positives. The F1-score (8) is the harmonic mean of precision and recall, offering a balanced measure that accounts for both false positives and false negatives. Accuracy (9) reflects the overall correctness of the model by measuring the proportion of all correctly classified instances. For the evaluation of the unsupervised DBSCAN model, two clustering-specific metrics—Davies–Bouldin Index (DBI) and

Table 2
Experimental Hyperparameters

Parameter	Setting
Activation Function	ReLU
Output Activation Function	Sigmoid
Hidden Layers	4
Epochs	49
Batch Size	32
Threshold	0.0239
Patience	7
Learning Rate	0.001
Optimizer	Adam
Epsilon	0.2
MinPts	3000

Adjusted Rand Index (ARI)—were used in addition to the standard evaluation metrics: precision, recall, F1-score, and accuracy. The DBI assesses the quality of clustering by evaluating intra-cluster compactness and inter-cluster separation—lower DBI values indicate more distinct and compact clusters, which is desirable. The ARI measures the similarity between the clustering results and ground truth labels, adjusted for chance. ARI values range from - 1 (poor agreement) to 1 (perfect clustering), with higher values reflecting better clustering performance.

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F - Measure = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{8}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{9}$$

4.3 Convergence Behavior of the Autoencoder’s First Hidden Layer

The loss curve presented in Fig. 4 illustrates the training and validation loss over 49 epochs for the Autoencoder’s first hidden layer, trained on the CIC IoT-DIAD 2024 dataset. Both losses decrease sharply

during the initial epochs, indicating that the model quickly captures the underlying patterns in the data. Around epoch 10, the rate of decrease begins to slow, and by epoch 20, both training and validation losses stabilize at low values, suggesting that the model has effectively converged. The close alignment between the training and validation losses throughout the training process reflects strong generalization capability with no signs of overfitting. By epoch 49, the loss values remain minimal and stable, confirming that the Autoencoder has accurately learned to reconstruct normal input data, as further illustrated in Fig. 5. This stable convergence at the final epoch validates the effectiveness of the first hidden layer as a foundational component for the subsequent stages of the anomaly detection pipeline.

4.4 Attack Detection Performance of the Deep Autoencoder Model

The performance of the deep autoencoder (deep-AE) model for attack detection was evaluated using a confusion matrix on the CIC IoT-DIAD 2024 dataset, as illustrated in Fig. 6. The model demonstrates high classification accuracy, correctly identifying 180,382 benign instances and 228,618 attack instances. It resulted in a relatively low number of misclassifications, with 3,213 benign instances incorrectly labeled as attacks (false positives) and 2,777 attack instances misclassified as benign (false negatives).

As shown in Fig. 7, the model achieved an overall accuracy of approximately 98.56%, with a precision of 98.61%, a recall of 98.80%, and an F1-score of 98.71%. These results indicate that the deep-AE model is highly effective in distinguishing between benign and malicious traffic, making it a robust solution for real-time intrusion detection in smart farming environments. The low false positive and false negative rates further underscore the model's reliability and practical applicability in enhancing cybersecurity for smart farming systems.

4.5 Comparative Evaluation of the Proposed Deep Autoencoder with State-of-the-Art Models

The proposed Deep-AE model was rigorously evaluated on the CIC IoT-DIAD 2024 dataset and benchmarked against five prominent state-of-the-art intrusion detection methods: CNN + LSTM, Deep Neural Network (DNN), Graph Neural Network with Grey Wolf Optimizer (GHO + GNN), Stacked Convolutional Autoencoder with SVM (SCAE + SVM), and an Ensemble Model. As summarized in Table 3, the proposed model achieves the highest performance across all four-evaluation metrics, attaining a

precision of 0.9861, recall of 0.9880, F1-score of 0.9871, and accuracy of 0.9856. Although SCAE + SVM produced a slightly higher precision (0.9897), its recall (0.9390) was considerably lower, resulting in a reduced F1-score (0.9637) and implying more missed attack instances. Similarly, the GHO + GNN model reached a reasonable F1-score (0.9490) but fell short of the proposed model in both precision and accuracy. The CNN + LSTM approach, despite achieving a high accuracy of 0.9695, showed the lowest recall (0.7668), indicating limited effectiveness in detecting all types of attacks. The Ensemble Model offered a good recall (0.9530) but suffered from lower precision (0.9330), leading to a higher false

positive rate. In contrast, the proposed Deep-AE model consistently maintained an optimal balance between precision and recall, resulting in the highest F1-score and accuracy among all compared methods. This balance highlights its capability to minimize both false alarms and undetected threats—an essential feature for real-time and resource-constrained IoT security deployments. The results, further illustrated in Fig. 8, confirm that the proposed framework offers a more robust and reliable intrusion detection solution for smart IoT environments than existing approaches.

Table 3
Performance comparison of the proposed deep autoencoder with existing classification methods

Models	Precision	Recall	F1-Score	Accuracy
CNN + LSTM [5]	0.8796	0.7668	0.7792	0.9695
DNN [15]	0.9355	0.9267	0.9263	0.9267
GHO + GNN [17]	0.9410	0.9570	0.9490	0.9480
SCAE + SVM [18]	0.9897	0.9390	0.9637	0.9371
Ensemble Model [19]	0.9330	0.9530	0.9430	0.9450
Proposed	0.9861	0.9880	0.9871	0.9856

4.6 Clustering Performance Analysis of the Proposed Deep-AE + DBSCAN Model

As shown in Fig. 9, the Adjusted Rand Index (ARI) score of 0.895 indicates that DBSCAN was highly effective in clustering the data in a manner that closely aligns with the actual labels. This demonstrates strong performance in distinguishing between attack and benign behavior patterns within the dataset. Additionally, the Davies-Bouldin Index (DBI) score of 0.019 signifies that the formed clusters are highly compact and well-separated—an essential characteristic for accurate anomaly or intrusion detection. The combination of a high ARI and an exceptionally low DBI confirms that DBSCAN achieved outstanding clustering performance on the CIC IoT DIAD 2024 dataset. These results highlight the algorithm’s capability to reliably differentiate between normal and malicious traffic in IoT environments, underscoring its potential as an effective unsupervised intrusion detection technique for smart IoT-based systems.

4.7 Comparative Evaluation of proposed deepAE + DBSCAN Against Existing Clustering-Based Intrusion Detection Methods

The performance of the proposed Deep-AE + DBSCAN model was compared against three existing clustering-based approaches: DBSCAN + GWO, CfsSubsetEval + K-Means, and K-Means + CDR. As shown in Table 4, the proposed model significantly outperforms the others across all evaluation metrics—

precision (0.9872), recall (0.9902), F1-score (0.9897), and accuracy (0.9899). Compared to the next best model, DBSCAN + GWO, which achieved an F1-score of 0.94 and accuracy of 0.98, the proposed approach shows notable improvements, especially in precision and recall. This reflects a more balanced and effective detection of both attack and benign instances, resulting in fewer false positives and false negatives. CfsSubsetEval + K-Means and K-Means + CDR exhibited moderate performance, with F1-scores of 0.863 and 0.61, respectively. However, they fall considerably short in precision and overall accuracy when compared to the proposed model. In particular, K-Means + CDR, despite achieving a relatively high recall of 0.818, suffers from low precision (0.473), suggesting a high rate of false alarms. Overall, these results, as visualized in Fig. 10, demonstrate that the proposed Deep-AE + DBSCAN model delivers superior detection accuracy and

robustness, making it a highly effective and reliable solution for intrusion detection in smart IoT environments.

Table 4
Performance comparison of the proposed deep autoencoder with DBSCAN clustering against existing clustering methods

Models	Precision	Recall	F1-Score	Accuracy
DBSCAN + GWO [20]	0.9100	0.9800	0.9400	0.9800
CfsSubsetEval + K-Means [21]	0.8570	0.8700	0.8630	0.8625
K-Means + CDR [22]	0.4730	0.8180	0.6100	0.9670
Proposed	0.9872	0.9902	0.9897	0.9899

Table 5 presents a comparative analysis between state-of-the-art intrusion detection techniques and the proposed fog-aware hierarchical deep autoencoder (AE) with latent-space gated DBSCAN. While existing approaches—such as hybrid ML–DL models, graph neural networks, clustering–optimization hybrids, and federated learning frameworks—achieve competitive accuracy levels ranging from 85.91–98.00%, our proposed model surpasses them with an accuracy of 98.99% on the CIC IoT-DIAD 2024 dataset.

This improvement is achieved without sacrificing computational efficiency, a critical factor for fog-based smart agriculture environments. Many high-performing prior methods are either dataset-specific, require extensive labelled data, or are sensitive to noise and parameter tuning. In contrast, our two-stage framework—combining dimensionality reduction, deep feature reconstruction, and density-based clustering—addresses these limitations by effectively separating normal and abnormal traffic, even in noisy conditions, without prior knowledge of the number of attack categories.

The integration of clustering quality metrics (Adjusted Rand Index of 0.895 and Davies–Bouldin Index of 0.019) further demonstrates the ability of our approach to produce compact, well-separated anomaly clusters, a performance aspect not evaluated in most prior studies. Overall, this comparison validates that the proposed model not only achieves the highest detection accuracy among evaluated methods

but also offers greater adaptability, robustness, and suitability for real-time, resource-constrained IoT security deployments.

Table 5
Performance comparison of intrusion detection method from state-of-the-art method and our proposed fog-aware hierarchical autoencoder with latent-space gated DBSCAN model

Ref	Year	Model / Method	Dataset(s)	Accuracy
[16]	2024	Hybrid ML + DL: XGBoost + CNN + LSTM	CIC IDS 2017, UNSW NB15, NSL KDD, WSN-DS	0.9840
[17]	2024	GNN + Leader K-means + GH0 clustering	ADFA IDS	0.8591
[18]	2022	SCAE + SVM	CIC-DDOS2019	
[19]	2024	Ensemble: GRP + K-means + SGD- OCSVM + GMM	CIC-DDoS2019	0.9450
[20]	2023	DBSCAN + GWO	N_BaloT	0.9800
[21]	2024	Optimized K-Means + CfsSubsetEval	URL dataset (11,430 URLs)	0.8625
[22]	2024	K-Means clustering + Mahalanobis distance	Mobile network CDRs (14M)	0.9600
[23]	2025	Federated Transfer Learning + MobileNetV2	CIC-IDS2017, NSL-KDD	0.9800
[24]	2025	Hybrid MI + ReliefF + Shallow and Deep ANN	Smart-Farm-IDS	0.9382
[5]	2024	EBWO-HDLID + Hybrid CNN-LSTM	ToN-IoT	0.9790
Proposed		Fog-aware hierarchical deep AE + latent-space gated DBSCAN	CIC IoT-DIAD 2024	0.9899

4.8 Discussion

The experimental results clearly demonstrate the superior performance of the proposed two-stage fog-aware Deep Autoencoder (Deep-AE) and Deep-AE combined with latent-space gated DBSCAN models across multiple unsupervised evaluation metrics. On the CIC IoT-DIAD 2024 dataset, the Deep-AE alone achieved an impressive F1-score of 98.71%, precision of 98.61%, recall of 98.80%, and overall accuracy of 98.56%. These findings confirm the model's ability to accurately distinguish between benign and malicious network traffic—an essential requirement for practical deployment in

resource-constrained smart farming environments, where reliable and timely threat detection is critical.

Notably, the Deep-AE exhibited a low false positive rate (FPR) and false negative rate (FNR), balancing detection sensitivity and specificity effectively. When compared with several state-of-the-art methods—including CNN-LSTM, Optimized Neural Networks (ONN), Stacked Convolutional Autoencoder with SVM

(SCAE + SVM), and ensemble models—our Deep-AE demonstrated a better balance between precision and recall. Although SCAE + SVM reported marginally higher precision, it suffered from lower recall, implying higher missed attack rates. In contrast, the Deep-AE minimized both false alarms and missed detections, providing a robust and balanced intrusion detection solution.

In the fully unsupervised scenario, integrating latent-space gated DBSCAN clustering with the Deep-AE further enhanced performance. Applying DBSCAN to the Deep-AE's latent representations yielded an Adjusted Rand Index (ARI) of 0.895 and a Davies–Bouldin Index (DBI) of 0.019, indicating well-separated, dense clusters that closely correspond to ground truth attack categories. This validates DBSCAN's capability to uncover intrinsic data structure without reliance on labeled samples. Compared with other clustering algorithms—such as DBSCAN with Grey Wolf Optimizer (DBSCAN + GWO), CfsSubsetEval + K-Means, and K-Means combined with Cluster-based Dimensionality Reduction (K-Means + CDR)—our proposed Deep-AE + DBSCAN framework outperformed them significantly, achieving a maximum F1-score of 98.97% and accuracy of 98.99%. This superior performance reflects not only enhanced detection ability but also increased robustness to false alarms.

Methods like K-Means + CDR, while attaining high recall, exhibited substantially lower precision (47.3%), which is problematic for high-security smart farming scenarios due to excessive false positives. Our approach avoids this pitfall by eliminating the need to predefine cluster counts and efficiently handling noisy data through the latent-space gating mechanism and density-based clustering.

Together, the Deep-AE and Deep-AE + DBSCAN models form a powerful hybrid framework capable of learning compact, informative latent representations of complex IoT traffic and effectively clustering anomalous behaviors. This hierarchical design supports real-time intrusion detection adaptable to evolving attack patterns and dynamic smart farming environments. The latent-space gating is critical for filtering out benign flows before clustering, improving noise resilience and privacy preservation by operating on flow-level metadata rather than payloads.

However, some limitations remain. The current anomaly threshold within the autoencoder is static and may require adaptive tuning for deployment across varying datasets or highly dynamic environments. While DBSCAN performed well in this study, its computational complexity could become a bottleneck with larger-scale or higher-dimensional datasets. Future work could explore scalable density-based clustering variants or approximate methods to maintain efficiency. Additionally, although the CIC IoT-DIAD 2024 dataset offers a comprehensive testbed relevant to smart farming, further validation on diverse real-world datasets will be necessary to ensure broad generalizability. Lastly, despite optimizations for fog nodes, the inherently resource-intensive nature of deep learning models may pose challenges for deployment on highly

constrained edge IoT devices, motivating continued research on lightweight model compression and pruning techniques.

5 Conclusion

This study introduces a novel, lightweight, and highly efficient anomaly detection framework designed for fog-based smart agriculture environments. By employing a two-stage hierarchical autoencoder—combining shallow dimensionality reduction with deep reconstruction-based anomaly scoring—our approach effectively balances computational feasibility and robust threat detection on resource-constrained fog nodes. Using unsupervised learning with Z-score normalization and density-based clustering (DBSCAN) on latent representations, the framework achieves superior performance in detecting and categorizing IoT threats without requiring labeled data. Evaluated on the CIC IoT-DIAD 2024 dataset, the Deep Autoencoder alone delivered an F1-score of 98.71% and accuracy of 98.56%. Incorporating latent-space gated DBSCAN further enhanced clustering quality, with an F1-score of 98.97%, accuracy of 98.99%, and strong cluster validity metrics (ARI 0.895, DBI 0.019). Compared with state-of-the-art AI-based intrusion detection techniques, including CNN-LSTM, ONN, SCAE + SVM, and ensemble models, our framework consistently outperforms in accuracy and noise resilience while maintaining low false positives—critical for operational deployment in smart farming. The integration of latent-space gating with density-based clustering addresses key challenges in high-dimensional, noisy IoT traffic typical in agriculture. Limitations include the static reconstruction-error threshold and DBSCAN’s computational demands on large-scale or high-dimensional datasets. Future research will focus on adaptive thresholding, scalable clustering algorithms, extensive evaluation on diverse real-world agricultural IoT data, and federated learning integration to enhance privacy and distributed training capabilities. Overall, this work contributes a practical, privacy-preserving AI solution that advances real-time intrusion detection for emerging precision agriculture systems, enabling more secure and resilient smart farming infrastructures.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no conflict of interest.

Funding

No Financial support was provided for this research

Author Contribution

Manikandan and Sumendra analyzed the data, authored or reviewed drafts of the article, and approved the final draft. Manikandan, Sumendra, Ramesh, Siti Fatimah Abdul and Md. Shohel conceived and designed the experiments, performed the experiments, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data availability

The dataset is publicly available (NSL-KDD datasets)

References

1. Kethineni K, Pradeepini G. Intrusion detection in internet of things-based smart farming using hybrid deep learning framework. *Cluster Comput.* 2024;27(2):1719–32. 10.1007/s10586-023-04052-4.
2. Eleftheriadis C, Andronikidis G, Kyranou K, Pechlivani EM, Hadjigeorgiou I, Batzos Z. ‘Machine Learning for Cybersecurity Frameworks in Smart Farming’, in 2024 28th International Conference on Information Technology (IT), 2024, pp. 1–5. 10.1109/IT61232.2024.10475711
3. Computing I, Arai K, editors. Noor Muaaz and Sithungu, ‘An Anomaly Detection Framework for IIoT-Based Smart Farming Systems’. Cham: Springer Nature Switzerland; 2024. pp. 396–409.
4. Bui HT et al. ‘Agriculture 4.0 and beyond: Evaluating cyber threat intelligence sources and techniques in smart farming ecosystems’, May 01, 2024, Elsevier Ltd. 10.1016/j.cose.2024.103754
5. Aburasain RY. Enhanced Black Widow Optimization With Hybrid Deep Learning Enabled Intrusion Detection in Internet of Things-Based Smart Farming. *IEEE Access.* 2024;12:16621–31. 10.1109/ACCESS.2024.3359043.
6. Sajid J, Hayawi K, Malik AW, Anwar Z, Trabelsi Z. ‘A Fog Computing Framework for Intrusion Detection of Energy-Based Attacks on UAV-Assisted Smart Farming’, *Applied Sciences* (Switzerland), vol. 13, no. 6, Mar. 2023, 10.3390/app13063857
7. Rupnik R, Kukar M, Vračar P, Košir D, Pevec D, Bosnić Z. AgroDSS: A decision support system for agriculture and farming. *Comput Electron Agric.* 2019;161:260–71. <https://doi.org/10.1016/j.compag.2018.04.001>.
8. Kaliyaperumal P, Karuppiyah T, Perumal R, Thirumalaisamy M, Balusamy B, Benedetto F. ‘Enhancing cybersecurity in Agriculture 4.0: A high-performance hybrid deep learning-based framework for DDoS attack detection’, *Computers and Electrical Engineering*, vol. 126, p. 110431, 2025, <https://doi.org/10.1016/j.compeleceng.2025.110431>
9. Yazdinejad A et al. Aug., ‘A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures’, *Applied Sciences* (Switzerland), vol. 11, no. 16, 2021, 10.3390/app11167518
10. Prodanović R et al. Nov., ‘Wireless sensor network in agriculture: Model of cyber security’, *Sensors* (Switzerland), vol. 20, no. 23, pp. 1–22, 2020, 10.3390/s20236747

11. Guembe B, Azeta A, Misra S, Osamor VC, Fernandez-Sanz L, Pospelova V. 'The Emerging Threat of Ai-driven Cyber Attacks: A Review', 2022, Taylor and Francis Ltd. 10.1080/08839514.2022.2037254
12. Demestichas K, Peppes N, Alexakis T. 'Survey on Security Threats in Agricultural IoT and Smart Farming', *Sensors*, vol. 20, no. 22, 2020, 10.3390/s20226458
13. Mahlous AR. 'Security Analysis in Smart Agriculture: Insights from a Cyber-Physical System Application', *Computers, Materials and Continua*, vol. 79, no. 3, pp. 4781–4803, 2024, 10.32604/cmc.2024.050821
14. Vatambeti R, Venkatesh D, Mamidisetti G, Damera VK, Manohar M, Yadav NS. 'Prediction of DDoS attacks in agriculture 4.0 with the help of prairie dog optimization algorithm with IDSNet', *Sci Rep*, vol. 13, no. 1, Dec. 2023, 10.1038/s41598-023-42678-x
15. Ferreira R, Bispo I, Rabadão C, Santos L, de Costa RL. Farm-flow dataset: Intrusion detection in smart agriculture based on network flows. *Comput Electr Eng*. 2025;121:109892. <https://doi.org/10.1016/j.compeleceng.2024.109892>.
16. Sajid M, et al. Enhancing intrusion detection: a hybrid machine and deep learning approach. *J Cloud Comput*. Dec. 2024;13(1). 10.1186/s13677-024-00685-x.
17. Dugyala R, et al. Secure cloud computing: leveraging GNN and leader K-means for intrusion detection optimization. *Sci Rep*. Dec. 2024;14(1). 10.1038/s41598-024-81442-7.
18. Nie L, et al. Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach. *IEEE Trans Comput Soc Syst*. Feb. 2022;9(1):134–45. 10.1109/TCSS.2021.3063538.
19. Roopak M, Parkinson S, Tian GY, Ran Y, Khan S, Chandrasekaran B. Sep., 'An unsupervised approach for the detection of zero-day distributed denial of service attacks in Internet of Things networks', *IET Networks*, 2024, 10.1049/ntw2.12134
20. Mustafa DH, Husien IM. Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection. *Int J Intell Eng Syst*. 2023;16(4):409–21. 10.22266/ijies2023.0831.33.
21. Al-Sabbagh A, Hamze K, Khan S, Elkhodr M. 'An Enhanced K-Means Clustering Algorithm for Phishing Attack Detections', *Electronics (Switzerland)*, vol. 13, no. 18, Sep. 2024, 10.3390/electronics13183677
22. Aziz Z, Bestak R. Mar., 'Insight into Anomaly Detection and Prediction and Mobile Network Security Enhancement Leveraging K-Means Clustering on Call Detail Records', *Sensors*, 24, 6, 2024, 10.3390/s24061716
23. Praharaj L, Gupta D, Gupta M. Efficient federated transfer learning-based network anomaly detection for cooperative smart farming infrastructure. *Smart Agricultural Technol*. Mar. 2025;10. 10.1016/j.atech.2024.100727.
24. Ileri K. A hybrid feature selection method for anomaly detection using shallow and deep ANN classifiers in smart farming. *J Ambient Intell Smart Environ* Jul. 2025. 10.1177/18761364251359885.

25. Rabbani M, et al. Device Identification and Anomaly Detection in IoT Environments. IEEE Internet Things J. 2025;12(10):13625–43. 10.1109/JIOT.2024.3522863.

26. Kumar A, Radhakrishnan R, Sumithra M, Kaliyaperumal P, Balusamy B, Benedetto F. 'A Scalable Hybrid Autoencoder–Extreme Learning Machine Framework for Adaptive Intrusion Detection in High-Dimensional Networks', Future Internet, vol. 17, no. 5, p. 221, May 2025, 10.3390/fi17050221

27. Jain P, Bajpai MS, Pamula R. A Modified DBSCAN Algorithm for Anomaly Detection in Time-series Data with Seasonality. Int Arab J Inform Technol. Jan. 2022;19(1):23–8. 10.34028/iajit/19/1/3.

Figures

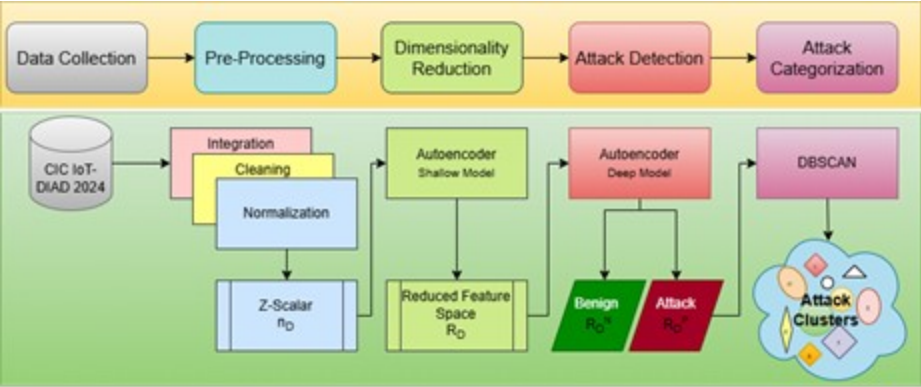


Figure 1

Architecture of the proposed Cyber Threat Detection and categorization framework

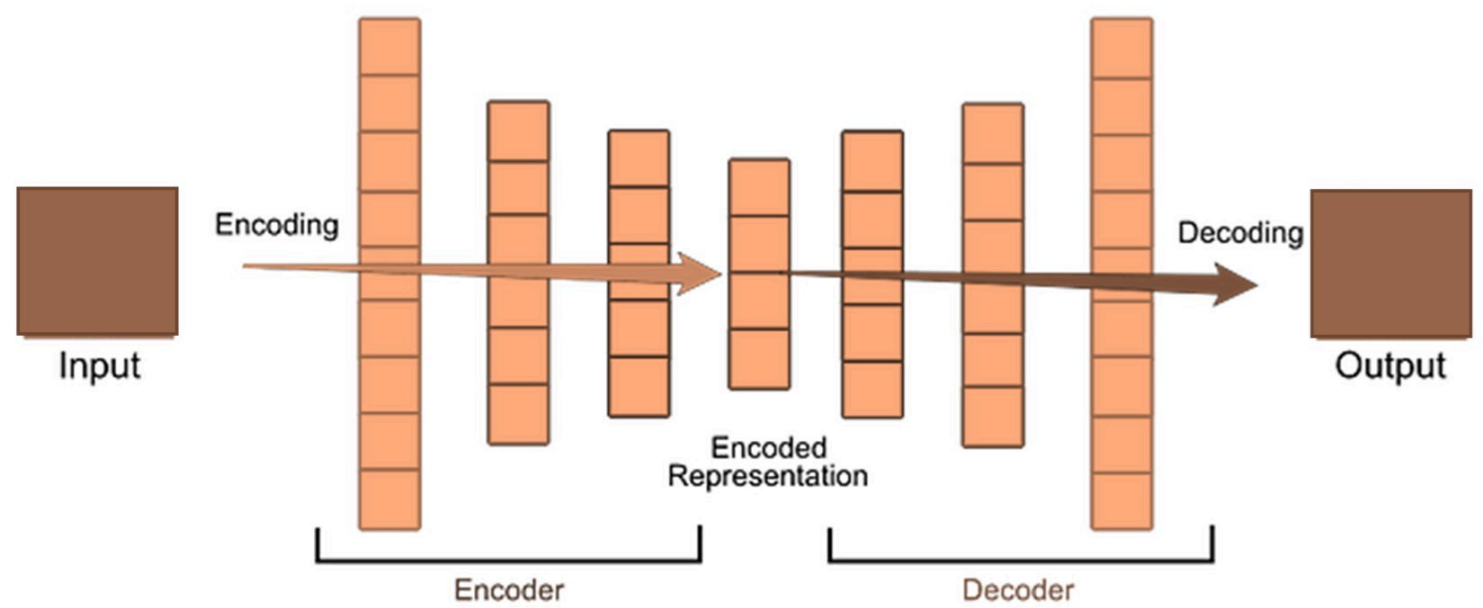


Figure 2

Architecture of the Deep Autoencoder

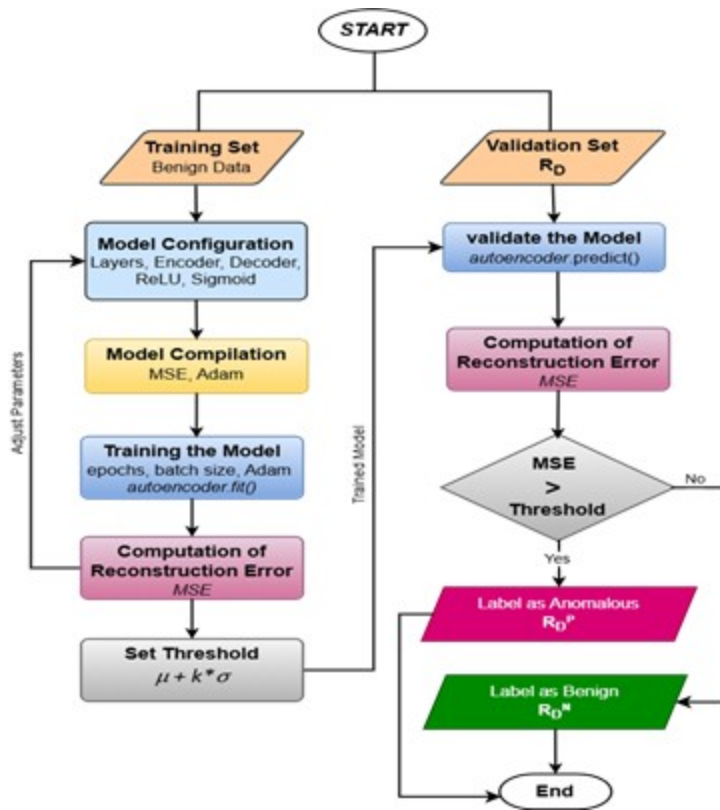


Figure 3

Step-by-step procedure for detecting anomalies

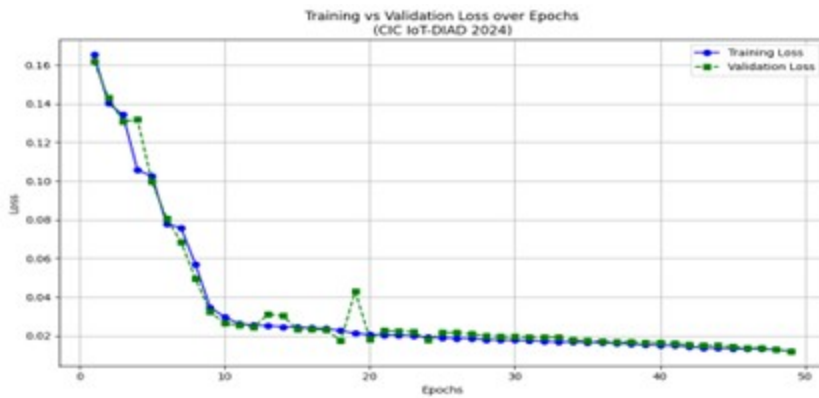


Figure 4

Loss function of the proposed autoencoder using MSE

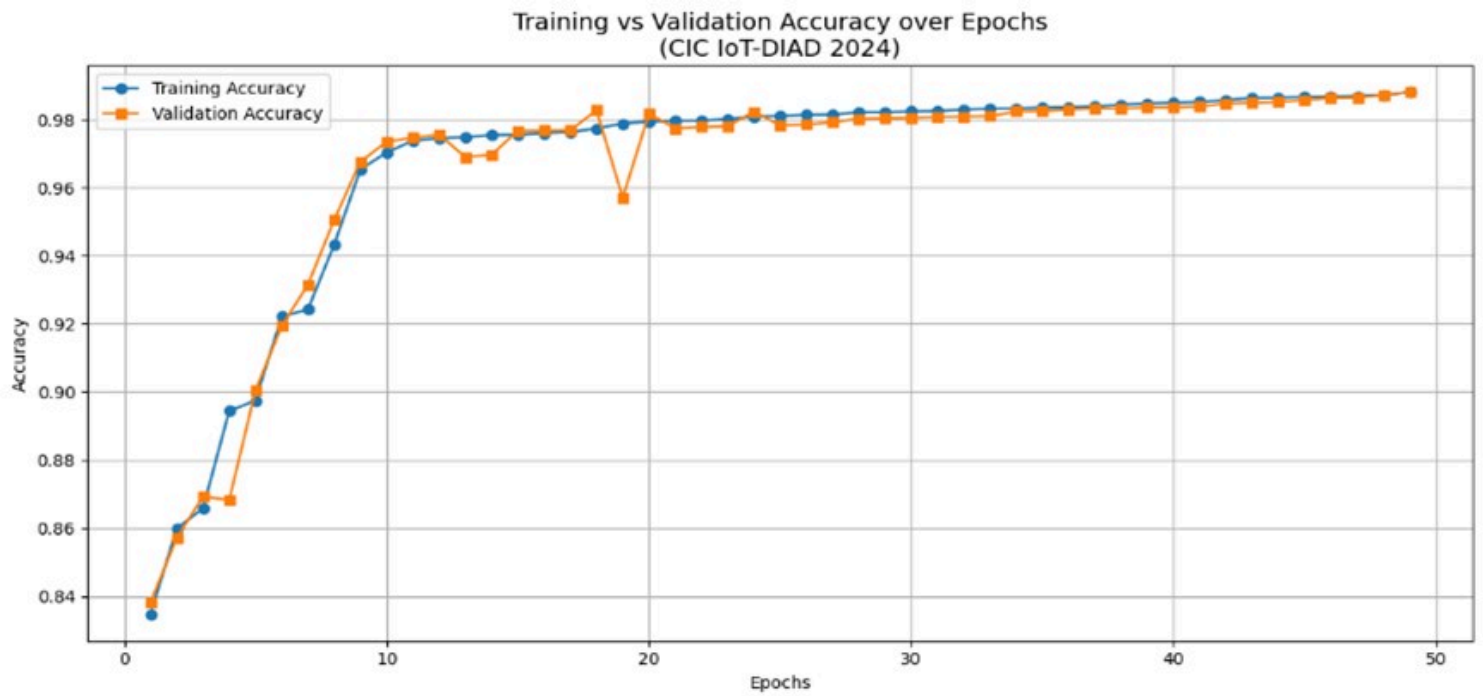


Figure 5

Accuracy curve of the proposed autoencoder using MSE

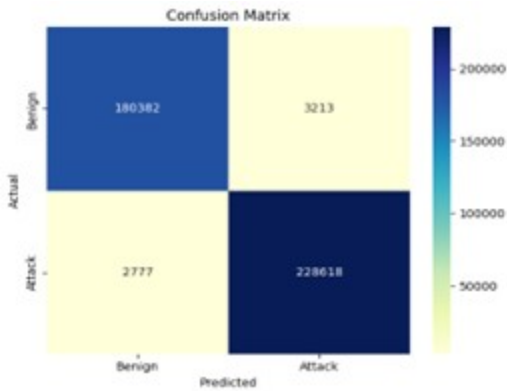


Figure 6

Confusion matrix of the proposed deep autoencoder

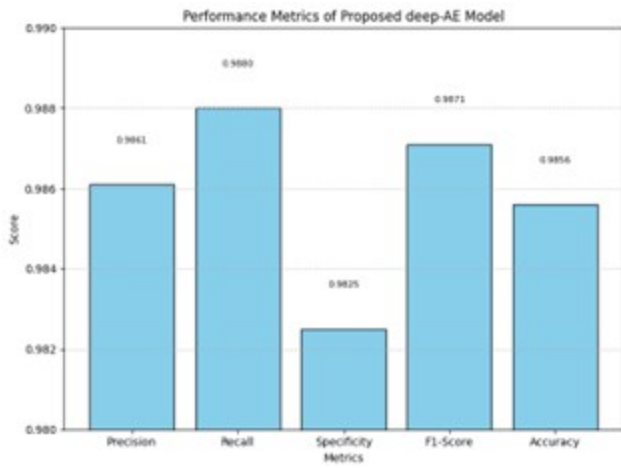


Figure 7

Performance of the proposed deep autoencoder model

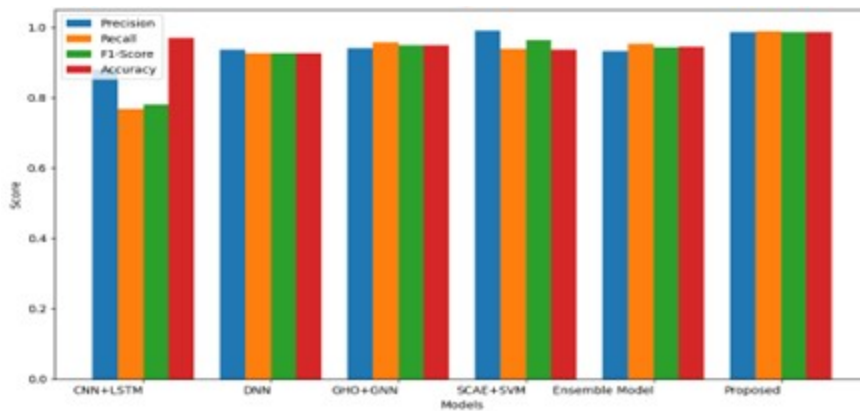


Figure 8

Performance comparison of the proposed deep-AE with existing methods

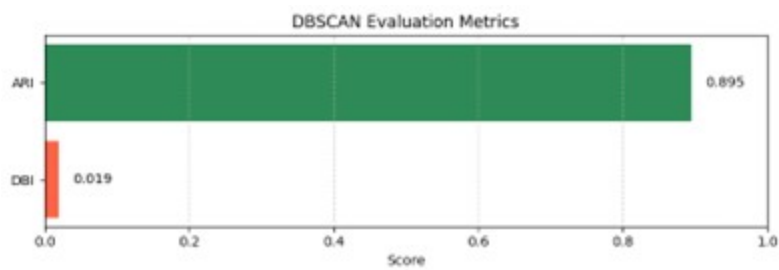


Figure 9

Performance of the proposed deep autoencoder with DBSCAN clustering

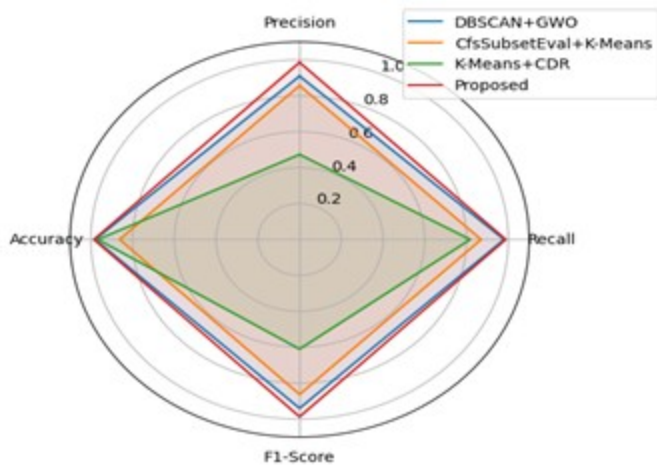


Figure 10

Performance comparison of the proposed deep autoencoder with DBSCAN clustering against existing methods