Journal of Computer Science

Research Article

Employing Hybrid Serial Cascaded Adaptive Network for Anomaly Detection and Prevention in IoT Time Series Data With Optimal Interdomain Routing

Kante Satyanarayana and K. Venkatesh

Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Kattankulathur, 603203, Chennai, India

Article history
Received: 21-03-2025
Revised: 18-06-2025
Accepted: 25-07-2025

Corresponding Author:
K. Venkatesh
Department of Networking and Communications, School of Computing, SRM Institute of Science and Technology, Kattankulathur, 603203, Chennai, India
Email: venkateshk3420@gmail.com

Abstract: Nowadays, emerging Internet of Things (IoT) in data processing becomes interested research topic and is used for various applications. This varies from big data processing and analytics for accumulating entire sensor data to detect and generate long-term trends. Nevertheless, this results in the requirement for resources like the execution of power, memory, and bandwidth, in which resource-constrained IoT tools suffers during the data transmission in order to improve effective operations. Hence, these results in risk and poor availability of data, and accepted by an insider like a malignant system administrator, without leaving any hints of their activities. Henceforth, a novel anomaly detection and prevention model over IoT time series data using Deep Learning (DL) is presented to identify and mitigate the attacks. Primitively, the required data is assembled from the accurate websites. The anomalies are detected from the obtained data utilizing the developed Hybrid Serial Cascaded Adaptive Network (HSCAN). This method is constructed by the concatenation of the Spatial-Temporal Attention (STA-AE) based Autoencoder and Long Short-Term Memory (LSTM). Once the detection of anomalies tends to be completed, it is eliminated for the superior operation of the data. Furthermore, the efficacy of the prevention and the detection phase is improved by fine-optimizing the parameters from the offered model through the Iterative Concept of Peregrine Falcon Optimization (ICPFO) algorithm. After processing the detection and the prevention stage, the routing stage is taken over by the developed ICPFO by assuming the energy, latency, and Packet Delivery Ratio (PDR). Hence, the efficacy of the recommended approach in anomaly detection and processing the routing is estimated and compared to the classical methods.

Keywords: IoT Time Series Data, Long Short-Term Memory, Anomaly Detection, Spatial-Temporal Attention Autoencoder, Hybrid Serial Cascaded Adaptive Network, Optimal Interdomain Routing, Iterative Concept of Peregrine Falcon Optimization

Introduction

IoT is a new network system combined with communication technologies, computers, and control that has a crucial impact on the economy and society. These IoT depend on Wireless Sensor Network (WSN), universal sensors that are spread in the real-time environment, accumulate data from neighborhood areas, and share data to the processing center through the internet or mobile communication networks (Sharma et al., 2021). By using big data technology and cloud computing and the valuable information from these data enables better services. The IoT technology has been extensively enforced in transportation, smart homes, medical, and various sectors (Sivasankari and Kamalakkannan, 2022). Therefore in the computing field, this developing trend upgraded our daily life objects to communicate and connect without any human intervention. Although IoT devices are helpful in many areas, it has very limited security features, and also most of the IoT devices are designed with hard-coded or fixed key passwords and default usernames, that cannot be changed by the user (Malki et al., 2022). These security problems allow hackers to easily access IoT devices and

administer them. However, various cyber attacks, such as malicious attacks, networking viruses, malicious eavesdropping, and so on constitute a crucial threat to women's property protection and information security (Kumar and Singh, 2024). Consequently, both communities and people have grown by depending on communication and information technology protection. Hence, firewalls are installed and used widely as a fundamental protective measure (Liu et al., 2021).

In general, attacks mainly aim at the energy consumption and usability of a node that is interconnected to the heavy data flow (Wu et al., 2020). In IoT system, the attack detection is a crucial role to enable security measures. The objective of source-side attacks is for the malicious node, which poses a threat to our highly sensitive network architecture, to induce energy consumption, process congestion, and excessive memory usage, thereby disrupting the stability of Quality of Service (QoS) within the network (Sana et al., 2024). Flood attacks, recognized as common prevalent types of attacks, seek to supply nodes by compromising the actions of malicious nodes. The anomaly-based detection method involves analyzing the typical behavior of network traffic and establishing a baseline profile for each device engaged in communication within the system (Chen et al., 2022). Any notable deviation from this baseline is identified as an anomaly. This anomaly-based detection method is divided into two categories: Statistics-based detection, which identifies anomalies through the statistical distribution of intrusions, and Machine learning-based detection, which recognizes irregularities related to packet and payload characteristics (Zhao et al., 2021). These machine-learning methods primarily focus on detecting and preventing potential attacks by utilizing machine-learning models. Additionally, the Knowledge-based detection method identifies anomalies based on the established profile or prior knowledge of the network, which is developed through various test scenarios to uncover abnormalities within the network (Fang et al., 2021).

In the instance of anomaly detection methods, the machine learning model has revealed drastic processing in recent years (Truong et al., 2022). The detection methods based on machine learning concepts are trained in the datasets to train and differentiate the pattern and behavior of attack and normal traffic (Lai et al., 2024). Therefore, the traditional botnet detection technique utilizes machine learning is restricted to some datasets on which they are trained; hence this is because of different types of botnet attacks that take place in different datasets. Also, the features utilized in detecting botnet attacks from one dataset are not enough to effectively analyze the botnet from other datasets because of different types of botnet attacks (Zhang et al., 2020). Consequently, these detection solutions tend to underperform when employed in different datasets because of variability in attack patterns. Anomaly detection can essentially be viewed as a classification task that employs historical time-series data to ascertain precise outcomes (Jiang et al., 2021). The concept of drift is relevant, as it can enhance performance over time with the help of data sampling process. Additionally, IoT applications typically function over extended periods, leveraging the detection process to analyze a vast array of IoT communication records and statuses (Huang et al., 2022). Malware attacks represented as significant threats to cyber security, making the development of effective malware detection methods a critical concern. Machine learning algorithms have previously proven to be a promising solution to this challenge (Amarbayasgalan et al., 2020). Numerous proposed methods involve converting malware executable code into image pixels and employing Convolutional Neural Networks (CNNs) for classification (Guan et al., 2022).

The significant contributions of the recommended detection and prevention model of anomalies are detailed as follows:

- To accomplish a new architecture for anomaly detection and prevention using the IoT time series data with optimal interdomain routing. This facilitates detecting malicious activities in the network easily. Hence, it can be even prevented effectively using an optimal routing strategy. For better performance execution, deep learning methodologies are also included
- To propose a network named HSCAN for the anomaly detection process. This network is deployed to mitigate computational complexities and problems found in the data. Thus, it supports training the model more efficiently. Moreover, this network is the combination of the STA-AE and LSTM. Using these two networks for our work may find solutions for the computational issues affected, high energy consumption, and latency
- To present an ICPFO algorithm for estimating the tuned factor that elevates the efficiency and functionality of the network. This algorithm is the interpreted optimizer from the traditional PFO, with novel depictions of random factors for identifying the optimized evaluations. Among the examined arbitrary variables, the effectiveness of the algorithm is enhanced to receive accurate results
- To introduce the routing mechanisms using ICPFO that contemplates PDR, latency, and energy in it. Introducing this routing mechanism for mitigating the anomaly to perform better data transmission. Hence, the effective performance is attained in the suggested framework than the existing approaches

The implemented anomaly detection and prevention model is detailed as follows. Part II elaborates on the existing review and their research gaps. Part III conveys the description of collected data, developed network, and algorithm. Part IV declares the different algorithms utilized in this model like STA-AE, LSTM, and the optimized HSCAN. Part V describes the optimal routing and its multi-objective formulation deploying the ICPFO. Part VI evaluates the results and discussion section, which also contains the evaluation setup and the validation metrics. Consequently, Part VII estimates the conclusion and the future work that will be introduced based on anomaly detection and prevention.

Related Work

In Cakir et al. (2020) have developed a Gated Recurrent Unit (GRU) network model based on deep learning to predict and mitigate Hello Flooding (HF) attacks on the Routing Protocol for Low-Power and Loss Networks (RPL) within the IoT environments. The Support Vector Machine (SVM) and logistic regression techniques were taken into account during various experiments. The outcomes validated the expected performance of the model regarding source efficacy and IoT security. Furthermore, the detection of attacks was achieved with a significantly lower error rate compared to the existing works.

In Husain et al. (2021) have designed a comprehensive dataset encompassing generic scanning and Distributed Denial of Service (DDoS) attacks. Furthermore, the researchers incorporated samples from different datasets to enhance attack scenarios, thereby improving the performance in suggested model. Subsequently, they introduced a dual-phase machine learning strategy aimed at the prevention and detection of IoT botnet attacks. A cutting-edge deep learning model, ResNet-18 was employed to screen the activities, which reduced the risk of IoT botnet attacks. Another ResNet-18 model was utilized for the identification of DDoS attacks that mainly focus on detecting IoT botnet attacks. To validate the efficacy of this dual-phase approach, three additional ResNet-18 models were trained by three different datasets of DDoS attacks, whereas the performance was compared against the developed dual-phase method.

In Miranda et al. (2022) have introduced a Reinforcement Learning (RL) agent to support and enhance a Software-Defined Networks (SDN) controller to obtain cost-effective route optimization and Quality of Service (QoS) management. The experimental analysis confirmed that this method successfully controls rank attacks, thereby acceptable levels of were maintained in the network.

In Yin et al. (2022) have developed an integrated model of the CNN and recurrent autoencoders for anomaly detection. An easy integration of CNN and autoencoders does not enhance the performance of classification, particularly in the context of time series data. For resolving the issues, a two-stage sliding window approach was adopted during the process of data pre-processing to facilitate improved representation learning. Following the data-pre-processing, spatial and temporal features were extracted using recurrent and CNN autoencoders, which were utilized within the fully connected network for classification. Empirical results have indicated that the proposed model demonstrated superior performance against various classification metrics and achieved better outcomes in anomaly detection.

In Nagaraju et al. (2022) have implemented a hybrid optimization technique and deep learning approach to detect the prevention attacks in IoT. Initially, a cyber security warning system was implemented, after that, the index factors were measured, and as a last step situation assessment was accomplished. Many nature-inspired techniques were used to reduce the dimensions of data in Intrusion Detection System (IDS) by deleting unwanted noise. Grey Wolf Optimization (GWO) was introduced for maintaining the efficiency of IDS by the identification of both abnormal and regular congestion of the network. Malware specimens have been gathered from the mailing database for testing. Hence to analyze attack prevention in IoT, deep convolution network and Whale with GWO (WGWO) were used. Experimental results have shown that the recommended technique possessed better classification outcomes.

In Xu et al. (2020) have introduced an innovative concept called the Drift Adaptive method to enhance the accuracy of anomaly detection, taking into consideration the temporal influences that alter sample distribution over time. Additionally, an AI-driven improved method was developed that incorporated utilized a novel smooth activation function enhance the performance of anomaly detection. Ultimately, the comprehensive experimental outcomes reveal that I-LSTM surpassed all metrics, representing an optimal integration of communication security and artificial intelligence.

In Aditya Sai Srinivas and Manivannan (2020) have established a robust model for the detection and prevention of HELLO flooding attacks via a tuned deep-learning approach. This model incorporated several key steps, such as generation of k-paths, selection of cluster heads, and selection of optimal shortest path. The presence of a HELLO flood attack was further confirmed using an optimized Deep Belief Network (DBN) that facilitates the removal of malicious nodes from the network. The objective constraints considered for optimal path selection included inter-node distance, node trust, packet loss ratio, and transmission delay. Ultimately, the experimental analysis of various performance metrics demonstrated the proposed model efficiency.

In Tukur et al. (2021) have introduced an edge-based blockchain-enabled anomaly detection method aimed at mitigating the inside threats within the IoT. Additionally, it incorporates elements of sequence-based anomaly detection and integrated distributed edge computing with blockchain technology, which utilizes smart contracts to identify and rectify anomalies in incoming sensor data. Evaluation for this method was conducted using actual IoT system datasets, which demonstrated significant results in achieving its objectives to strengthen the performance of the IoT system.

Research Gaps

IoT is an emerging technology that connects and communicates devices with one another without the need for human intervention. Even though IoT gadgets are very helpful to us, it does not offer high security. Additionally, a lot of IoT devices have a hard-coded default username and password or a set key that the user is unable to alter. Hackers can easily take advantage of these security flaws in IoT devices to invade the user's privacy. Therefore, various researchers have developed new methods to detect anomalies in IoT time series data. Some of the problems in existing models are given in the following:

- Existing models need more computational resources and do not focus on the data transmission process while detecting the attacks in IoT time series data
- On behalf of the increasing volume of data generated by IoT devices, existing models struggle to maintain the scalability of the models while detecting anomalies
- One of the biggest problems in some of the prior models is that they only detect the anomalies without taking any steps to prevent them. However, preventing anomalies is crucial for enhancing the overall quality of the operations performed in IoT devices
- Current techniques developed for anomaly detection do not concentrate on proper routing, thus it results in network congestion and increased latency

These limitations are rectified by using a developed deep learning-aided anomaly detection model. Various features and challenges of conventional anomaly detection approach in IoT time series data are detailed in Table 1.

Table 1: Benefits and drawbacks of traditional anomaly detection approach in IoT time series data

Reference	Model	Benefits	Drawbacks
Cakir et al. (2022)	GRU	• It effectively reduced the consumption of more energy and memory • It provides better security for IoT networks by effectively detecting and preventing HF attacks	• It does not have the ability to solve the scalability issues. • Implementing and maintaining the model is complex and time-consuming
Hussain et al. (2021)	ResNet-18	• It prevents botnet attacks by identifying the attack in its premature stage • It can analyze and detect various attack patterns	• Degradation issues need to be solved
Miranda et al. (2022)	RL	• It prevents the rank attack with cost-efficient route optimization • It minimizes the overhead and latency issues	• The complexity of the system is high
Yin et al. (2022)	CNN	• It retrieves high-level features for an effective anomaly classification process	• Its accuracy is limited due to hardware resource issues • The training time as well as the computational resource requirement is high
Nagaraju et al. (2022)	DCNN	• It reduces the noise in input for enhancing the quality of data used to detect threats	• The requirement for computational resources is high
Xu et al. (2020)	I-LSTM	• It protects the user's privacy. • It offers communication security by identifying different kinds of anomalies	• The robustness of the classification and detection outcomes is low
Srinivas and Manivannan (2021)	DBN	• It identifies optimal shortest path selection for effectively preventing the HELLO flooding attacks in IoT systems	• The algorithm's convergence rate is not sufficient for reliable performance • It does not consider link quality
Tukur et al. (2021)	Edge-based Blockchain technique	• It minimizes the latency and bandwidth requirements • It avoids single points of failure issues and maintains the integrity of data	• Performing full-fledged anomaly detection with data mining approaches is needed

Table 2: Types of Anomalies and Data Characteristics

Types of Anomaly	Description	Characteristics
Art-Daily-FlatMiddle	Flat or constant signals in the mid-region — possibly representing sensor freezes	It comprised of several key characteristics like trends, cyclicality, seasonality and noise. The underlying patterns evolving over time based on its dynamic nature of time series data. Moreover, understanding the regular patterns (daily, weekly or yearly cycles) in the time series data facilitates to detect the anomalies
Art-Daily-JumpsDown	Abrupt downward changes — indicative of possible signal drops or failures	
Art-Daily-JumpsUp	Sudden upward surges — often seen during overloads or false reporting	
Art-Daily-NoJump	Gradual drift or long-term anomaly without sharp transitions	
Arts-Increase-Spike-Density	Increased frequency of spikes — may reflect noise injection or instability	
Arts-Load-Balancer-Spikes	Spiky patterns resembling load balancing shifts in high-load environments	

Development of Proposed Model and Data Collection Details for Anomaly Detection and Routing

Experimented Data Details

For the developed anomaly detection and prevention mechanism, the data are accumulated from"https://www.kaggle.com/code/joshuaswords/time-series-anomaly-detection/input: Access data: 2024-05-10". This dataset is named Kaggle's repository. This dataset contains 44,363 time-series records, out of which 20,165 samples are no anomaly and 24,198 samples are found to be anomaly. It also has 2 classes, namely no anomaly and anomaly. This dataset consists of nearly 50 labeled artificial time series and real-world data files along with a new grading method. The data gathered from the data source and it is represented as A_{DP} . A dataset description of the anomaly types and dataset characteristics is included in Table 2. Here, the six labeled types of anomalies are represented in this dataset, whereas each can simulate in diverse real-world behaviors of the IoT systems.

Elucidating the Novel Concept of Detection and Routing

IoT comprises a platform of interconnected with various devices to enable proper communication. The application layer in the IoT performs and detects this data for various objectives, such as data visualization, decision-making, automation, and control. Since huge volumes of data are incorporated in IoT, maintaining the integrity and reliability of information is essential. IoT systems face significant security and efficiency challenges in outliers, deviations, and malicious data patterns. Early identification of these anomalies allows proactive measures and timely interventions to reduce the potential complexities. Furthermore, several techniques are employed to detect abnormal activities in IoT systems, and recommended preventive measures. The ever-evolving landscape of IoT implementations, coupled with the wide variety of IoT devices and applications, necessitates the use of robust data anomaly detection methods. These methods should be scalable, flexible, and proficient in managing substantial data volumes. Ensuring reliable and secure deployment of IoT systems across different sectors, it is essential to employ sophisticated deep learning-based data anomaly detection and prevention approaches. Figure 1 depicts the detailed representation of the implemented technique for anomaly detection and prevention tasks in IoT time series data.

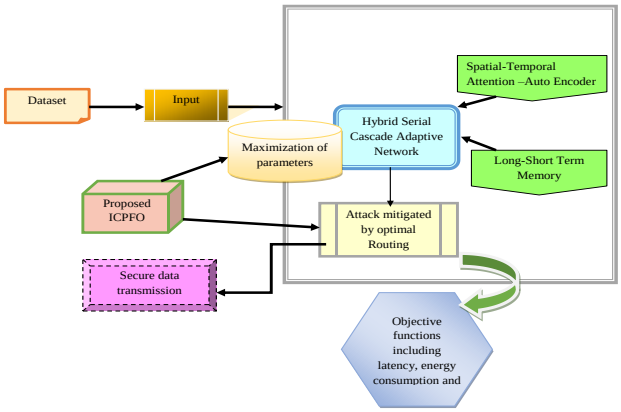


Fig. 1: Detailed representation of the implemented model for anomaly detection and prevention task using IoT time series data

A comprehensive network for anomaly detection and prevention of IoT time series data is proposed, by employing advanced deep learning techniques aimed at identifying and mitigating potential attacks. The initial phase of this process involves the systematic accumulation of data sourced from reliable and standardized websites. This data serves as the foundation for identifying irregular patterns that have different types of anomalies. To effectively detect these anomalies, the proposed methodology utilizes the Hybrid Serial Cascaded Adaptive Network (HSCAN). This innovative network design is a formation of two robust components: A Spatial-Temporal Attention-based Autoencoder and Long Short-Term Memory (LSTM) units. Autoencoder

is specifically designed for allowing to learn complex spatial and temporal dependencies in the data. Meanwhile, the LSTM analyzes sequential data, which is highly effective for time series analysis in IoT settings. The concatenation of these components enhances the ability of the developed model that automatically recognize deviations from expected behavior accurately. Once the HSCAN successfully identifies anomalies within the dataset, the next step is to implement measures for preventing these detected threats. This prevention strategy is aimed at facilitating better management of the overall data integrity. To maximize the efficiency of both the detection and prevention processes, an optimization technique known as ICPFO is employed. This technique focuses on fine-tuning the parameters of the HSCAN, enhancing its ability to recognize the anomalies effectively. Following the successful detection and prevention of anomalies, the routing process is also optimized using the ICPFO. This routing will take into consideration critical metrics such as energy consumption, PDR, and latency. The overall efficiency of the designed framework in both detecting anomalies and managing routing processes is evaluated against conventional models.

Enhanced Heuristic Algorithm: ICPFO

A novel algorithm ICPFO is designed to attain the optimum solutions that can be utilized to enhance anomaly detection and prevention in IoT time series data.

Inspiration: The developed algorithm is a nature-inspired swarm intelligent approach by mimicking the predation and migration characteristics of peregrine falcons. The behavior of PFO (Wu and Liu, 2023) achieves better performance in solving a complex optimization problem and finds optimal solutions by effectively traversing the search space. However, the PFO does not attain the convergence values as expected, and also sometimes it falls into local optima issues. In some instances, a few operators fail to solve multi-objective problems because of their un-adaptivity nature. These issues are addressed and solved by the proposed ICPFO. The mathematical model of the proposed ICPFO is shown below.

Step 1- Population Initialization: At first the population is taken as peregrine and it is randomly located over the searching space. The position of peregrine is attained using Eq. (1):

$$Y = (Y_1, Y_2, ..., Y_N)'$$
$$Y_i = lob + rand(1, \dim) \times (upb - lob). j = 1, 2, ..., N \quad (1)$$

Step 2: Partner Strategy: Peregrine are monogamous and reside in the nest of the partner. The position of Peregrine is enhanced by the partner's position.

Step 3: Aerial Predation: The peregrine randomly locates their prey because of their excellent vision, and constantly hovers in the air. The selection of prey in the search space is random, therefore the exploration capability is increased in this phase, and as a result, the optimal area is determined. The new position of Peregrine Falcon is expressed in Eq. (2):

$$Y_{new} = Y + F \times r \times \alpha \times (u_1 \times Y - u_2 \times Y_m) + F \times T \times (u_{best} - u_2 \times Y) \times e^{sf} \times \cos(2 \times \pi \times sf) \quad (2)$$

Here, the new position of the peregrine is denoted as Y_{new}, f is a flag that changes the search direction, the step factor is denoted as a, and the adaptive parameter that varies with iteration number is denoted as sJ.

Step 4: Ground predation phase: Once the peregrine attacks the prey, it takes it to an organized place and starts to eat. Owing to its high speed, the peregrine chases, and in most situations, it catches the prey at the end. The exploitation of local search is improved in this phase.

Step 5: Migration strategy: Finally, a strategy of migration was introduced to adjust the position of the peregrine based on their migration behavior.

Novelty: ICPFO is developed by refining the existing PFO. The PFO derives the optimal solution quickly and also solves the complex optimization problem very effectively.PFO uses an independent random number of uniform distribution in 0 and 1. This uniform distribution does not improve the adaptability of parameters to solve multi-objective problems. Therefore, addressing the limitation is crucial for achieving optimal solutions. Hence PFO is modified by formulating a new random variable in the suggested PFO. This helps to mitigate the local optima issue by using a random variable at intervals 0 and 1 based on the total number of iterations. The proposed random variable r is evaluated by Eq. (3):

$$r = -t * \left(\frac{(-1)}{\max iter} \right) \quad (3)$$

Here, $\max iter$ the maximum iterations taking place in the ICPFO and the variable t mean the current iteration value. The random variable r is replaced in Eq. (2). Further, the pseudo-code of recommended ICPFO is shown here.

Algorithm 1: Implemented ICPFO

Input: Hidden neuron and number of epoch count in AE and LSTM, and number of nodes.
Output: Optimized Value
Begin
Initialize the population
Set the iteration limit as $S < S_{\max}$
Revise the position in the aerial predation step with the adaptive formulation of a random variable as given in Eq. (3).
Upgrade the global optimal function.
End

Anomaly Detection Using a Hybrid Serial Cascaded Deep Learning Network With Parameter Optimization

Autoencoder

An autoencoder (Alaghbari et al., 2023) is a specialized type of neural network designed to learn a compact and efficient representation of input data. Its primary function is to reconstruct the original input, thereby ensuring that required features are preserved during the compression process. The framework of an autoencoder contains three main components including, encoder network, hidden latent vector, and decoder. At first, encoder network processes input data and compress into a lower dimensional space represented by the hidden latent vector. Following these, the decoder network takes the latent vector and reconstructs back to original format. The deep learning approach enhances the autoencoder's ability for analyzing the relationship among the data points. An autoencoder consisting single hidden layer in the encoder and decoder is illustrated in Eq. (4):

$$b = \sigma\left|(V_{encoder} y + c_{encoder}\right) \tag{4}$$

Here, the input vector is represented as y, the low dimensional feature space of the bottleneck is represented as b, the activation function is denoted as σ, and the bias, weight indicates c, V. The decoder network recovers the original input y from the predicted low-dimension features b by using Eq. (5):

$$\hat{y} = \sigma(V_{decoder} b + c_{decoder}) \tag{5}$$

In reconstruction process, the AE model diminishes the errors. This is represented in Eq. (6):

$$L(y, \hat{y}) = \left\| y - \hat{y} \right\|_2^2 \tag{6}$$

Here, $\|\bullet\|_2$ indicates the Euclidian norm. To determine the threshold value, the AE model is performed to execute the outcomes. The schematic diagram of AE for the proposed system is given in Fig. 2.

Long Short-Term Memory

LSTM (Shende, 2020) is a significant advancement over the RNN architecture. LSTM is specifically designed to address the limitation of backpropagation gradient explosion that is faced by RNN. This is achieved by developing a gating mechanism in LSTM, that enables the model to retain information over a longer period. Consequently, LSTMs are highly apt for tasks that require processing sequential data with long-range dependencies. The LSTM cell consists of an input, forget, and output gate, in which manages the information flow. Initially, the unwanted data are identified and the decision is made to throw those data from the cell state. This decision is made by the sigmoid layer known as the forget gate layer. This is expressed in Eq. (7):

$$g_t = \sigma(V_g.[h_{t-1}, y_t] + c_g) \tag{7}$$

Here, the output obtained from the previous stamp is represented as h_{t-1}, the new input is denoted as y_t, and c_g denotes the bias.

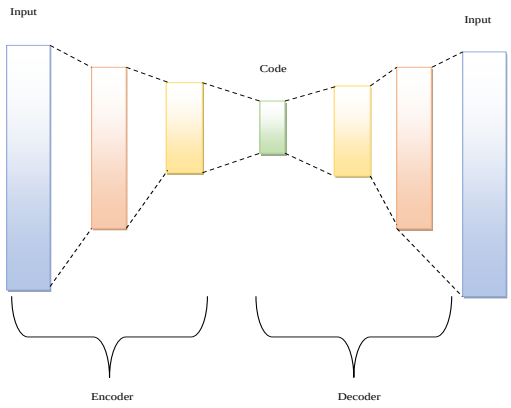


Fig. 2: Schematic diagram of Autoencoder for the proposed model

The decision-making process has two distinct components: One is the input layer, which uses a sigmoid function, and the other is called the tanh layer generates a new vector value that could be integrated into the cell state. This is mathematically illustrated in Eqs. (8-9):

$$j_t = \sigma(V_j \cdot [h_{t-1}, y_t] + c_j) \tag{8}$$

$$\tilde{D}_t = \tanh(V_d.[h_{t-1}, y_t] + c_d) \tag{9}$$

Here $\tilde{D}_t$ is a temporary new state; the weight matrix is represented as V_d, V_j, V_o.

Followed by the decision-making phase, the output stage is determined in which the cell state is passed through a sigmoid layer that decides which component of cell state is going to be a final output. Further, cell state is processed through the tanh function and output obtained from the tanh function is multiplied by the results from the sigmoid state. This multiplication ensures that only the relevant information is included in the final output. The output is expressed in Eq. (10):

$$o_t = \sigma(V_0.[h_{t-1}, y_t] + c_0)$$
$$h_t = o_t * \tanh(\tilde{D}_t) \quad (10)$$

The diagrammatic structure of LSTM is illustrated in Fig. 3.

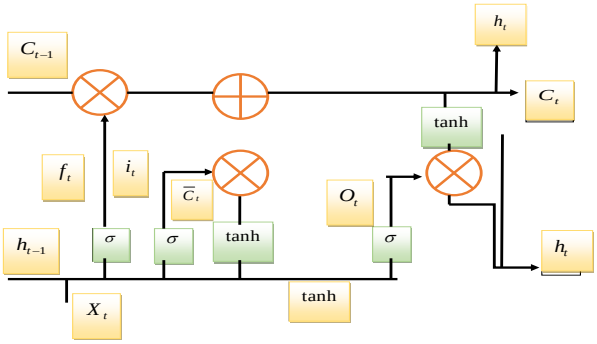


Fig. 3: Diagrammatic illustration of LSTM

Proposed HSCAN for Detection

Initially, the raw data M_j^{raw} is taken as input for this network. Then this data is subjected to a STA-AE network to extract the relevant features required for detection. STA (Li et al., 2022) comprises spatial and temporal attention mechanisms**.** The SA demonstrates the relative contributions of different input variables in multivariate time series forecasting and also enables to recognition and analysis of the correlations that exist between different locations. The features extracted using STA are not accurately representative and the errors found cause inefficient results. Hence, AE is preferred to extract the features for detecting the anomaly. The AE compresses the input data to extract features and reconstructs the original data from the encoded data. All the extracted features in the encoding phase are not used further for detecting the anomalies. In this instance, a spatial-temporal attention mechanism is infused with the AE, which only selects the most relevant features that are essential for the classification process. Subsequently, the extracted features are fed into the LSTM network, where the detection mechanism takes place to attain a classified outcome. This LSTM aids in learning longer sequence data and also captures information from earlier time steps and remembers it over a long period. Thus HSCAN is implemented by integrating both methods; hence, by using the STA-AE and LSTM, efficiency of the model is being improved. Though the model offers better performance, the HSCAN uses more network parameters such as epoch count, and hidden neurons. This improper count of parameters results in minimized accuracy and more computation time for performing the task. Therefore an adaptive concept called ICPFO is utilized for optimizing these attributes. The ICPFO is an enhanced algorithm that escapes from local optima issues and has higher convergence values. The fitness function is illustrated in Eq. (11):

$$F = \underset{\{hn^{AE}, ep^{AE}, hn^{L}, ep^{L}\}}{\arg\min} \left[\left(\frac{1}{ACC + MCC} \right) + FDR \right] \quad (11)$$

From the above equation, the terms ACC describe the accuracy, MCC state the Matthews Correlation Coefficient, and FDR denote the False Discovery Rate. Further, the term hn^{AE} is hidden neuron count noted in the AE between the ranges [5-255], ep^{AE} denotes the epoch count noted in the AE between the ranges [5-50], hn^{L} indicates hidden neuron count present in the LSTM in the range [5-255], and ep^{L} indicates the epoch count present in the LSTM between the ranges [5-50].

Statistical Decisions Influence the Results Using Parameter Selection

In anomaly detection and prevention, the statistical decision is the crucial aspects, especially in selecting the accurate parameters in the neural network. Focusing on the right parameters enables reliable and accurate outcomes in IoT time series data. Selecting the right parameters in the neural networks can precisely capture the underlying patterns to easily identify the normal and abnormal anomalies. Moreover, analyzing the number of hidden neurons represents model's complexity. For effectively learn the intricate patterns, the model trains with a sufficient number of neurons to detect the accurate anomalies. However, training too many neurons can easily increase the overfitting issues. In this context, the research work trains a significant number of hidden neurons to capture the underlying patterns without degrading the model's performance. Here, the hidden neuron of the AE and LSTM model is considered in the interval [5-255]. Considering the number of epochs initiates the amount of time taken while training the datasets. Training the excessive number of epoch could have the ability to refine the parameters and improve the model performance taken from [5-50] in the AE and LSTM model. Choosing the right parameters could strengthen the model accuracy and effectiveness to improve the decision-making process. In this context, the statistical decision is made by analyzing the underlying data patterns. Poor statistical decisions could impact the false positives and false negatives. Incorrect identification of normal data points are anomalies leads to unnecessary alerts that increase the chances of system failure. Thus, it impacts the result during the training and testing process. The research work demonstrates to show improved accuracy rate by choosing exact parameters and minimize the error rate. Validating the sensitivity and specificity correctly identifies

the anomalies and truly identifies the normal data points. This makes the developed model provides better statistical decisions that provides robust performance. The expressions for accuracy, MCC, and FDR are detailed below.

Accuracy: Accuracy is given in Eq. (12):

$$ACC = \frac{AB + AC}{B + C} \tag{12}$$

From Eq. (12), the term ACC denotes the accuracy, AB is the true positive value, and AC is the true negative value.

MCC: The elaboration of MCC is examined in Eq. (13):

$$MCC = \sqrt{(AB \times AC \times EF \times GH)} - \sqrt{(DO \times FP \times FO \times FD)} \tag{13}$$

From Eq. (13), the terms GH mean the negative predicted value, FO state the false omission rate, and MCC Matthews's correlation coefficient.

FDR: The estimation of FDR is formulated in Eq. (14):

$$FDR = \frac{DN}{EF} \tag{15}$$

From the above equation, the terms EF mean the positive predicted value and FDR state the false discovery rate. The illustration of developed HSCAN is given in Fig. 4.

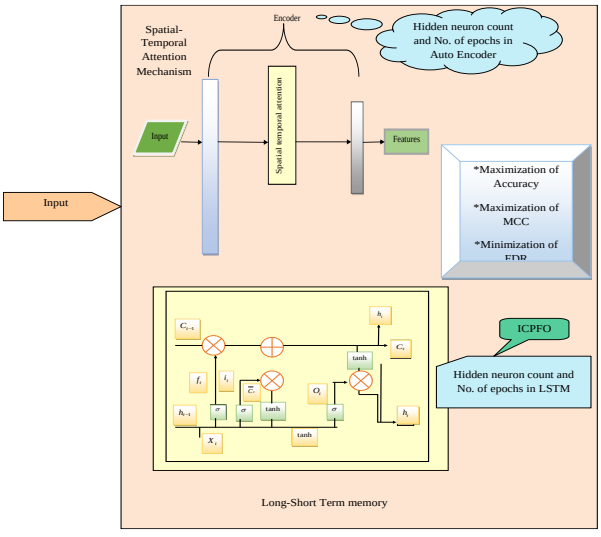


Fig. 4: Pictorial diagram of the developed HSCAN for anomaly detection and prevention

Performing Optimal Routing Using the ICPFO Approach and its Multi-Objective Formulation

Optimal Routing Mechanism

Routing is a technique of choosing a path for network traffic that utilizes deep learning methods to tune the decisions of routing. It evaluates the optimal path among the nodes. It is optimal, only if it reduces some cost function that validates the functionality of the system. It always identifies the shortest path among any couple of nodes in any of the integrated networks. The shortest path can be found by using an enormous number of nodes. These nodes help in providing an efficient path for the data transformation. A smaller number of nodes present help in providing the shortest path for information sharing. It assists in mitigating the network failure by handling the data traffic, and hence a network can utilize as much of its ability as possible without generating congestion. Moreover, it can deal with complex and huge data and it has the capacity to maintain structured and unstructured data, missing data, and so on. However, some of the methods, in which routing is used, suffer from large computational challenges, transferring a set of nodes in a particular order while facing tuned variables. Furthermore, heuristics are frequently assigned for routing problems, but developing them can be more challenging. Routing can be optimally selected from the total number of nodes within the network. The pictorial view for the solution encoding-based routing mechanism is provided in Fig. 5.

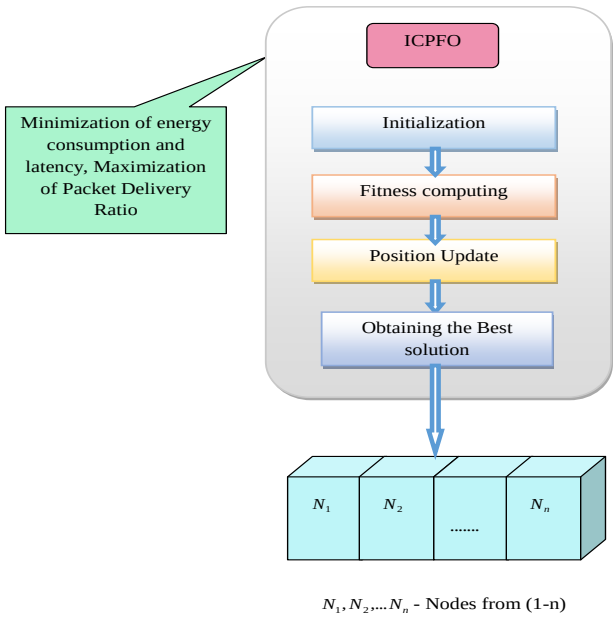


Fig. 5: Illustrative view of solution encoding-based routing mechanism

Advantages

- ✓ This optimal routing can mitigate the risks and achieve the functionality of the model
- ✓ It enables the process to perform highly, whether that is computing huge amounts of data, or running the tasks more rapidly
- ✓ It assists in reducing the network traffic, by eliminating the congestion
- ✓ It allows the network to decrease the fault tolerance and provide a sufficient availability of nodes in the network

Multi-Objective Function With Constraint Specification

Generally, the routing process is performed to mitigate the attacked node. After mitigation, routing is taken place by the remaining nodes available. The major goal is to find out, whether any malicious activities are occurring in the nodes. If any activities are identified, the network provides a high penalty to the respective path. On the other hand, due to the selection of the shortest path, the optimal routing is achieved through ICPFO. Further to focus on such aspects, the different constraints are considered to attain better results via the multi-objective formulation, and in addition to using the proposed algorithm ICPFO, the routing process performed well. Moreover, if any malicious activities are found among the nodes, the penalty is expressed as mentioned in Eq. (15):

$$obj = \arg\min_{\{N_n\}} \left[\left(\frac{1}{pdr} \right) + EC + LT + PT \right] \tag{15}$$

From Eq. (15), the terms obj refer to the objective function, EC mean the Energy Consumption, LT describe the Latency, and PT state the penalty expression respectively. Here, the nodes are optimized, which ranges from (Sharma et al., 2021; Sivasankari and Kamalakkannan, 2022; Malki et al., 2022; Kumar and Singh, 2024; Liu et al., 2021; Wu et al., 2020; Sana et al., 2024; Chen et al., 2022; Zhao et al., 2021; Fang et al., 2021) respectively. Sharma et al. (2021); Sivasankari and Kamalakkannan (2022); Malki et al. (2022); Kumar and Singh (2024); Liu et al. (2021); Wu et al. (2020); Sana et al. (2024); Chen et al. (2022); Zhao et al. (2021); Fang et al. (2021) refers to the total number of nodes, which is represented in a variable N_n. Hence, by optimizing this parameter, various objectives such as latency and energy consumption are minimized and the PDR is maximized accordingly. Therefore, by using ICPFO, better optimal routing is performed efficiently.

Latency: It evaluates time delay between which the AI network gets an input and produces efficient outcomes. It particularly calculates the lag for a network to perform inputs and computes its inference logic to generate decisions or forecasting. Expression for latency can be written as in Eq. (16):

$$L = \frac{1}{TR} + PD + QD \tag{16}$$

From Eq. (16), the variable L defines the latency, and the terms TR state the transmission rate, QD define the queuing delay, and PD mean the propagation delay respectively.

Energy consumption: It is the calculation of the huge amount of energy used by the computing tools. This can be estimated in Eq. (17):

$$EC = ET + ER + EI + ES \tag{17}$$

Here, the terms EC state the energy consumed, ET and ER convey the time of transmission and reception, and the terms EI ES denote the node which is idle and sleeping.

PDR: PDR is a measure utilized to calculate the success rate of giving the data packets from a transmitter to a receiver. It is validated by splitting number of packets received by the total count of packets sent, and further multiplying by 100 to provide a percentage. PDR can be estimated in Eq. (18):

$$PDR = \frac{PR}{PS} \times 100\% \tag{18}$$

From Eq. (18), the terms PDR refer to the packet delivery ratio; PS mean the packets sent, and PR describe the packet received.

Experimental Findings

Simulation Setup

To provide comprehensive outcomes, the designed network was evaluated, utilizing the device named Python. The ICPFO algorithm was executed by deploying maximum iteration at 50, the total count of the population as 10, and the length of chromosome 4. Hence, several classical methods were utilized in this work for contrast such as Peregrine Falcon Optimization (PFO) (Jaganathan et al., 2025), Whale Optimization Algorithm (WOA) (Siahmarzkooh and Alimardani, 2020), Kookaburra Optimization Algorithm (KOA) (Waghmode and Patil, 2024), Gold Rush Optimizer (GRO) (Sarjamei et al., 2021), Deep Belief Network (DBN) (Zhang et al., 2022), Convolutional Neural Network (CNN) (Liu and Wang, 2023), and Long-Short Term Memory (LSTM) (Almahadin et al., 2024). Configuration details of all network comparisons are given in Table 3.

Table 3: Configuration representation for all network comparison

DBM	Learning Rate	0.01
	Number of epochs	10
CNN	Learning Rate	0.01
	Number of steps per epoch	5
	Number of epochs	10
LSTM	Learning Rate	0.01
	Number of Steps Per Epoch	10
HSCAN	learning rate	0.01
	Number of steps per epoch	10

Evaluation Metrics

The derivation of Accuracy is given in Eq. (12).

MCC is expressed in Eq. (13).

FDR is estimated in Eq. (14).

Specificity: The elaboration of specificity is depicted in Eq. (19):

$$Spc = \frac{AB}{AB + FP} \tag{19}$$

Here, FP denotes the false positive value respectively.

FPR: The evaluation of FPR is defined in Eq. (20):

$$FP = \frac{DN}{DO} \tag{20}$$

Here, the term DO refers to the false negative rate.

F1-score: The examination of the F1-score is given in Eq. (21):

$$F1-s = \frac{2AB}{2AB + FP + DO} \tag{21}$$

FOR: The detailed formula of FOR is given in Eq. (22):

$$FO = \frac{DO}{GH} \tag{22}$$

Sensitivity: The formula for sensitivity is represented in Eq. (23):

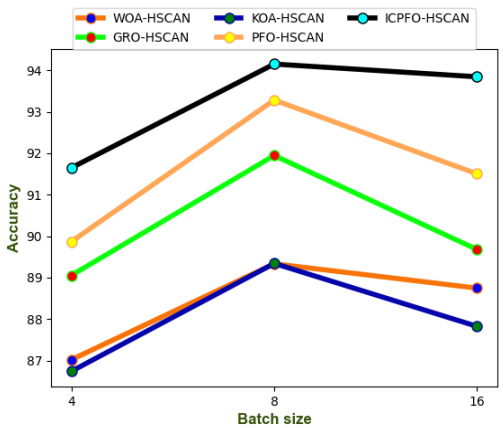
$$Sn = 1 - DO \tag{23}$$

Comparative Estimation of ICPFO Over Traditional Algorithms and Methods

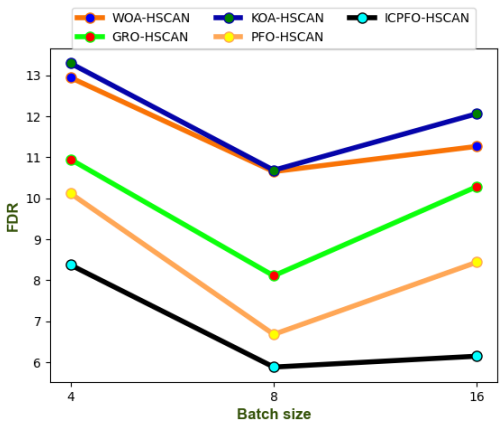
The comparative estimation of proposed optimizers and classifiers are given in Figs. 6 and 7. On taking the batch size at 8 of MCC from Fig. 6 (c), the results obtained are 11.13% for KOA, 11.25% for WOA, 5.56 % for GRO, and 2.27% for PFO respectively, which is lesser than the proposed work. Therefore, the developed model acquires greater performance from both the methods and algorithms. From the estimated graph, it is verified that in contrast to the other conventional methods and the algorithms, the resulting values got superior efficacy.

Evaluation of the Confusion Matrix

The given Fig. 8 represents the confusion matrix evaluation for the proposed anomaly detection and prevention of the given dataset. The confusion matrix utilized the true and the predicted labels for finding the effectiveness of the network. The experiment of the confusion matrix verifies that the offered network efficiently identifies the anomaly even if it is difficult and new to the model. Furthermore, it has been assured that the developed anomaly detection process elevates the security and privacy of the IoT time series than the classical techniques.



(a)



(b)

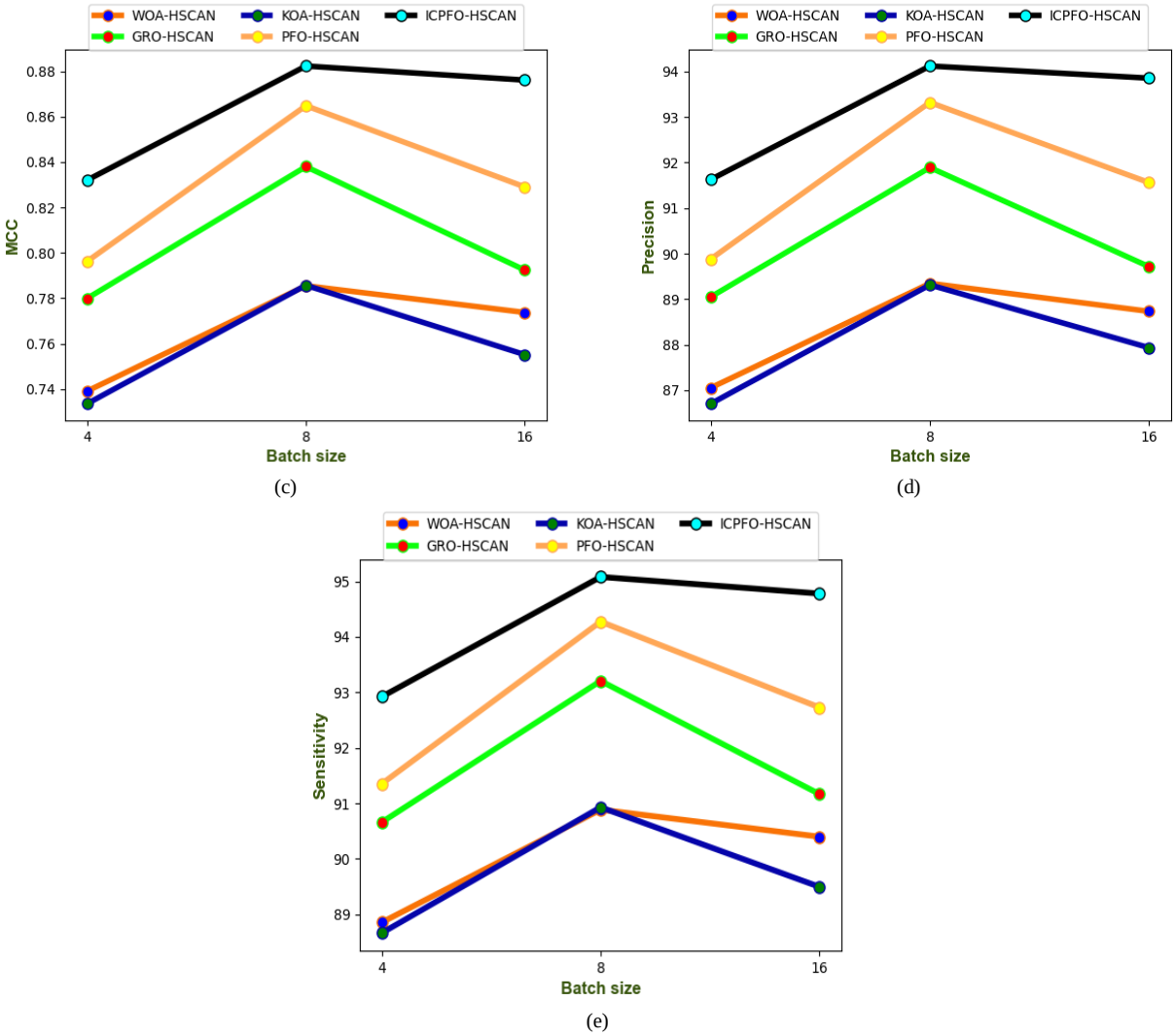
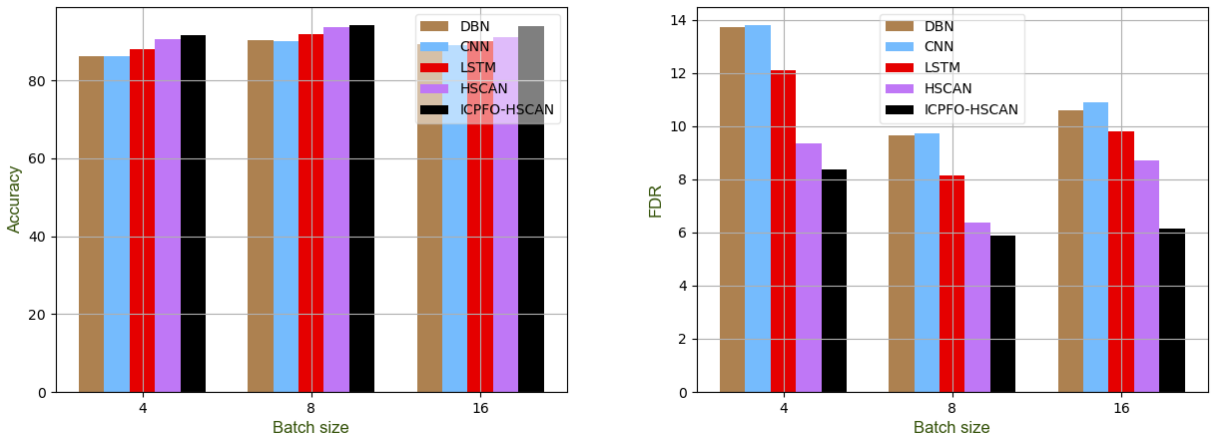


Fig. 6: Comparative estimation of the detected model for the given dataset over classical algorithms concerning "(a) Accuracy, (b) FDR, (c) MCC, (d) Precision, (e) Sensitivity"



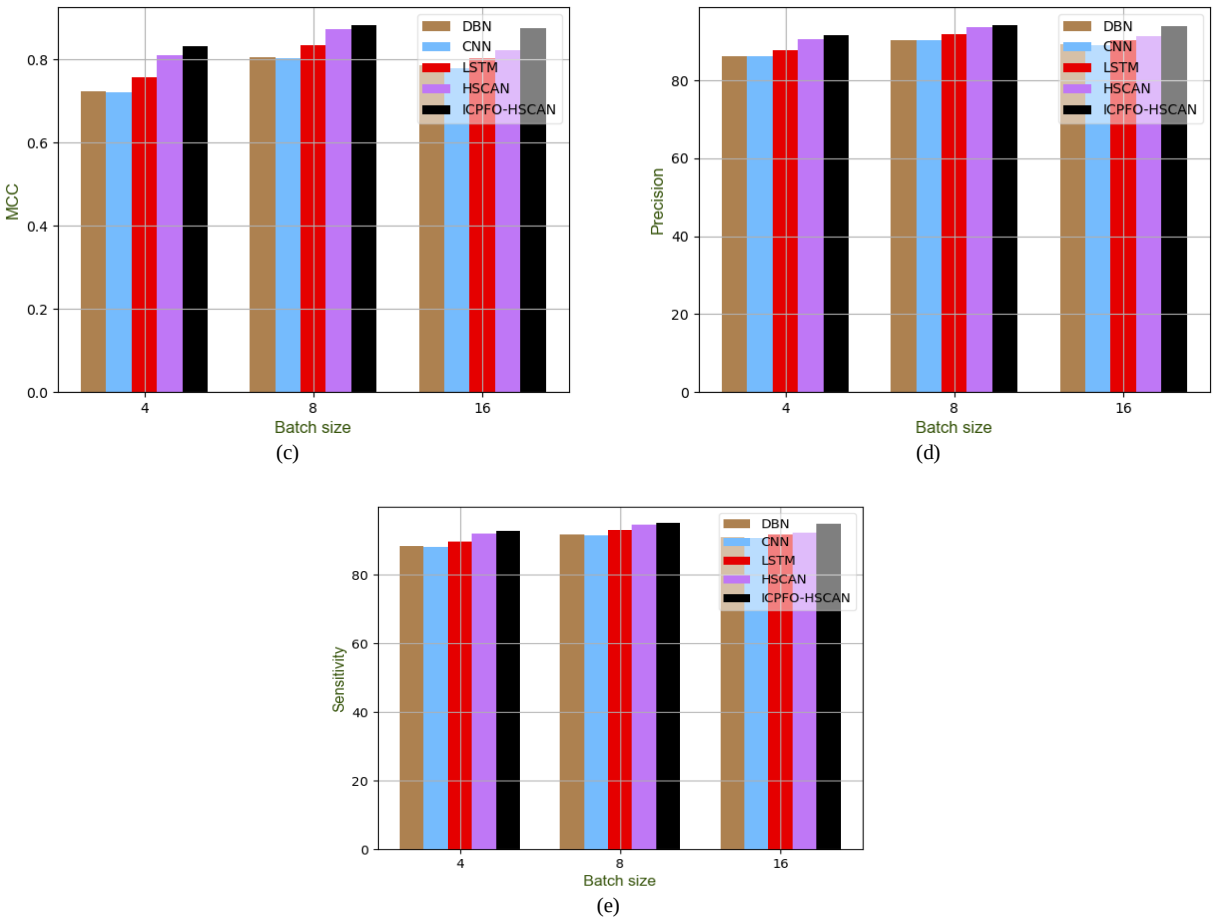


Fig. 7: Comparative estimation of detected model for given dataset over classical methods concerning (a)" Accuracy, (b) FDR, (c) MCC, (d) Precision, (e) Sensitivity"

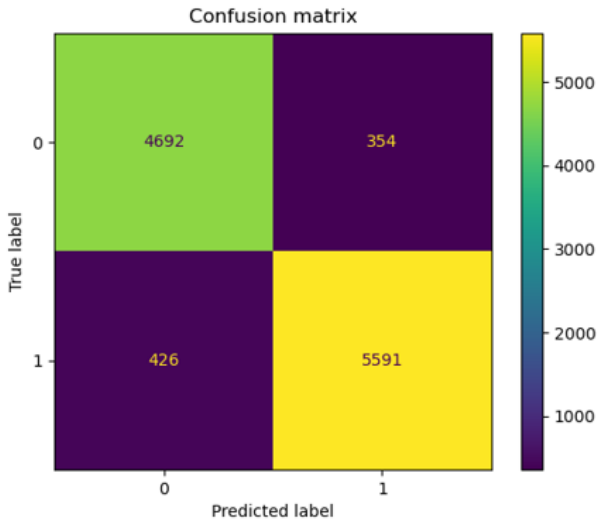


Fig. 8: Confusion matrix of the given dataset for anomaly detection and prevention

ROC Evaluation

The anomaly detection and prevention process is evaluated with ROC, which is given in Fig. 9 for the provided database. The ROC experiment is estimated using false positive and true positive rates effectively. Here, the HSCAN is used as the contrast to the existing methods and calculated the efficiency. Taking the 0.4th FPR evaluation from Fig. 10, the values obtained for LSTM are 6.25%, HSCAN is 1.25%, DBN is 2.5%, and CNN is 3.75 %, the performance of the designed model is enhanced over traditional techniques approximately. Thus, the efficiency of proposed network is superior to old techniques. In addition, this experiment described that the designed system succeeded accurately in smaller error values while assuming the conventional techniques.

Convergence Analysis ICPFO-Based Routing

Figure 10 evaluates the convergence function of routing through different numbers of nodes such as 50,

100, 150, and 200. The convergence evaluation is analyzed in this model for the ICPFO by utilizing the iteration values. This evaluation is functionalized by differentiating the total number of iterations. It is performed to reveal the cost rate in providing high-tuned value. On considering the iterated value of node 50, from Fig. 6 (a), the convergence value for the developed technique is validated as 3.36% for WOA, 2.17% for KOA, 3.15% for GRO, and 3.26% for PFO approximately. Hence, the tuned parameters support designing the detection and prediction process efficiently. Here, the superior performance is obtained by the ICPFO. The outcomes said that the designed algorithm improved a better-tuned evaluation over conventional algorithms. Therefore, from the overall calculation higher performance and efficiency has estimated.

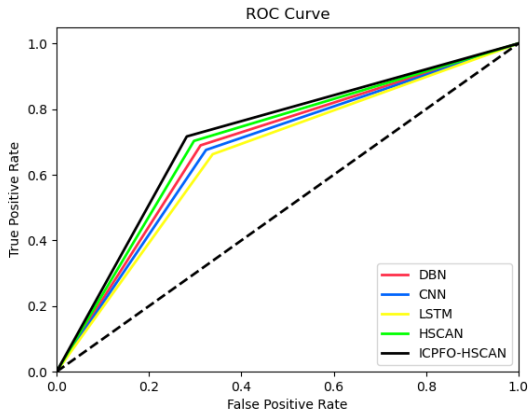


Fig. 9: ROC analysis for anomaly detection and prevention concerning a given dataset

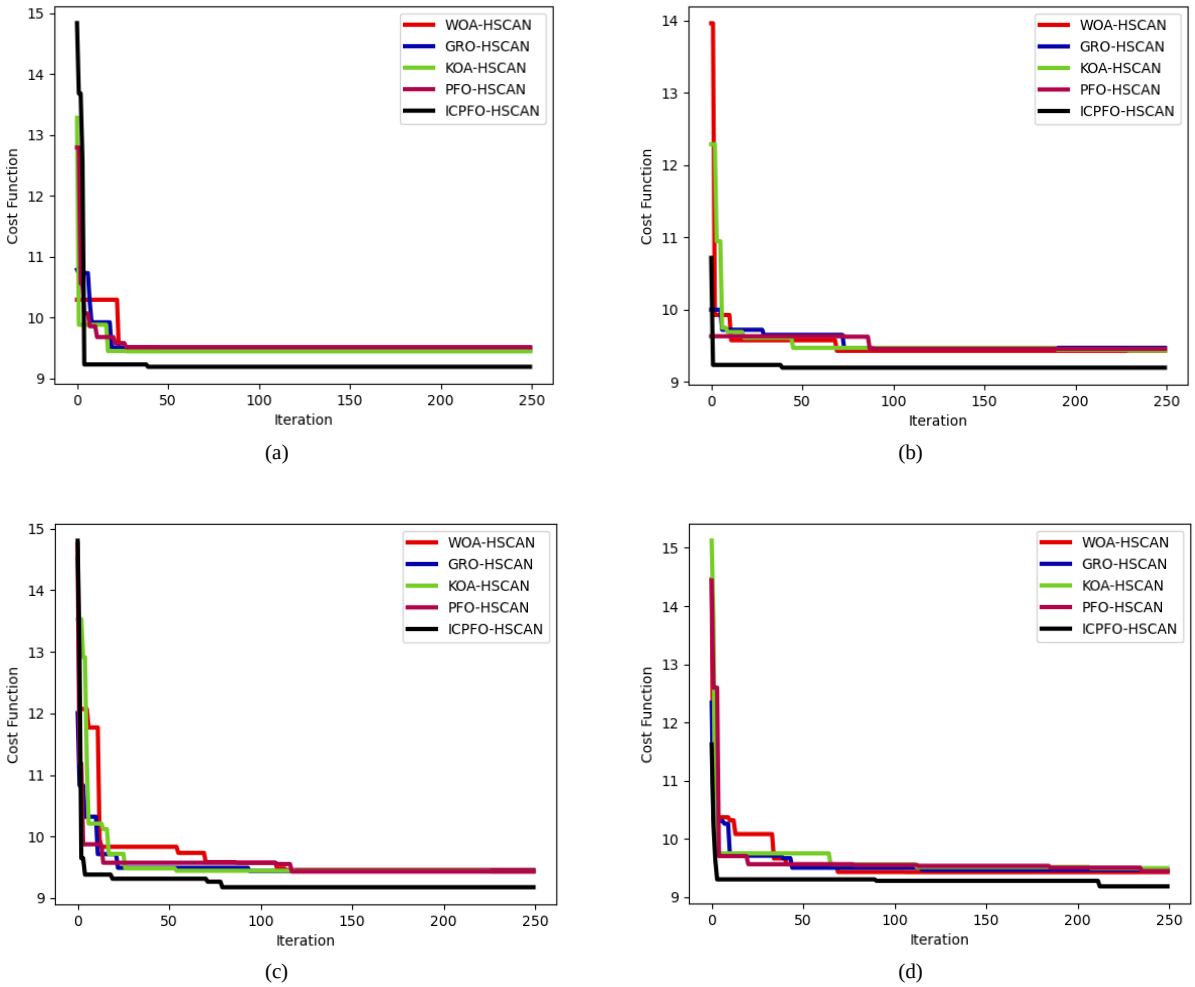


Fig. 10: The cost function for the developed model over classical optimizers by having the number of nodes "(a) 50, (b) 100, (c) 150, and (d) 200"

Evaluation of Optimal Routing Using ICPFO for Various Methods

Figure 11 shows the optimal routing mechanism using the proposed algorithm for several models. By considering the PDR (d) at a number of nodes at 100 from Fig. 11, the outputs obtained 16.8% for PFO, 20% for WOA, 20% for GRO, and 17.8% for KOA effectively. However, the proposed technique obtained a higher performance. Therefore, the optimal path is obtained using various methods for the routing analysis. Consequently, from the calculated values, it has been proved that by comparing to the classical models, the proposed one received higher performance and efficiency.

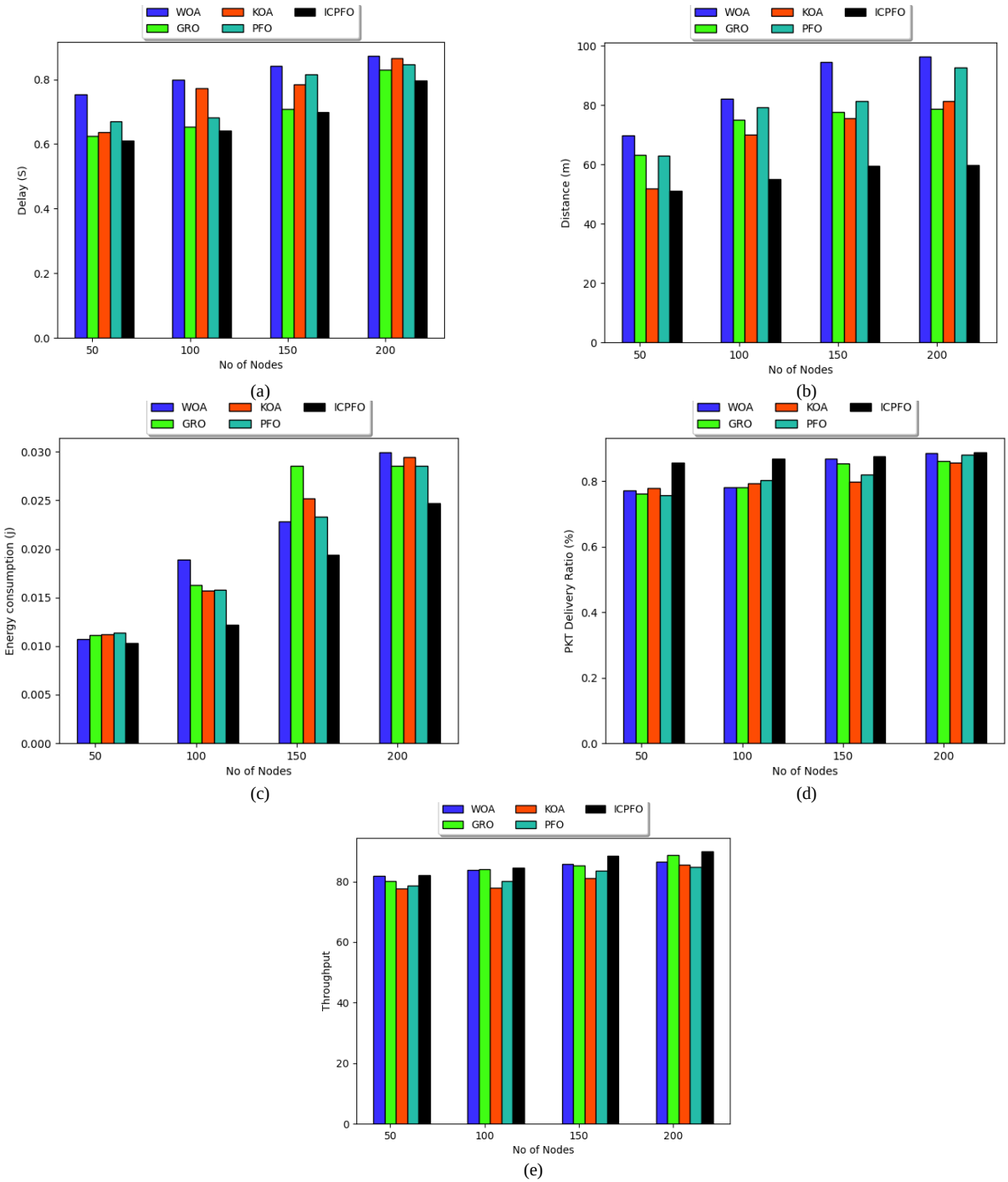


Fig. 11: Evaluation of optimal routing using ICPFO for various methods concerning (a) "Delay, (b) Distance, (c) Energy Consumption, (d) PKT, and (e) Throughput"

Overall Statistical Evaluation of the Detection Algorithms With 5 Different Folds

Collecting the data from benchmark datasets often causes data imbalance issues. This could generate the false outcomes and affect the overall performance model. To enhance the performance in anomaly detection in IoT-based time series data, the K-fold validation is selected to monitor the continuous performance and estimates the model's stability in wide range of applications. In particular, the K-fold analysis in the anomaly detection model enables more robust to generate model performance, especially in imbalanced datasets. The K-fold analysis is comprised of multiple folds, and each fold can perform training and testing for minimizing data bias. This could lead to provide robust performance and improves the generalization ability. Based on this advantageous performance in K-fold, the research validates by comparing with baseline models. It evaluates how the model effectively generalizes on unseen data. This can be achieved by considering the five sets of folds as (1-fold, 2-fold, 3-fold, 4-fold and 5-fold). During the process of training, if the model uses the k-2 fold, the remaining fold is performed in the testing process. Overall this process gets repeated with multiple times by effectively utilizes the different folds in the validation process until it reaches k. Moreover, it averages the outcomes from the several train-test splits that minimize the bias and enables more relevant and reliable outcomes in the anomaly detection and prevention process. To generate robust evaluation, analyzing the K-fold value effectively prevents from the issue of overfitting. Determining the single train-test split often leads to a better understanding of the model's capabilities and generalizability. Table 4 details the statistical examination of the proposed algorithm has 5 different folds. Here, enormous metrics such as Accuracy, Specificity, FPR, FDR, F1-score, FOR, and MCC are considered to provide efficient outcomes. Hence from the table, let us consider the FDR measure of the 5th Fold for evaluation. The evaluation provided superior outcomes of 65%, 75%, 101.8%, and 16% than the GRO, KOA, WOA, and PFO. Thus, by comparing to the traditional algorithms, the proposed algorithms received higher performance with efficient outcomes.

Overall Statistical Evaluation of Detection Models With 5 Different Folds

Table 5 conveys an overall statistical evaluation of proposed methods having 5 different folds. Here, enormous metrics such as Specificity, FOR, FPR, F1-score, MCC, FDR, and Accuracy, are assumed to offer effective outputs. Therefore, from the given table, let us take the accuracy metric of the 4th Fold for estimating the value. Hence, the estimation has given the higher performed results as 5.89% for DBN, 5.7% for CNN, 3.18% for LSTM, and 2.78% for HSCAN respectively. In contrast to the older methods, the implemented method acquired superiority.

Overall Statistical Estimation of the Detection Model With Several Nodes

Table 6 shows overall statistical estimation of the developed anomaly detection and prevention model for different nodes 50, 100, 150, and 200 over classical methods. Several parameters like best, worst, median, mean and standard deviation are considered and carried out in this analysis. The given table, considering the mean values of the 50th node offers efficient results as 100%, 2.37%, 3.13 %, and 3.23% than the conventional algorithms WOA, KOA, GRO, and PFO accordingly. Therefore, in comparison, the developed model is better than the conventional model.

Statistical Test Analysis of the Developed Model

Table 7 depicts the statistical test analysis by considering the p-values. Here, the statistical test analysis is conducted by comparing with diverse algorithms. Analyzing the p-value in the statistical hypothesis test is the statistical tool that effectively identifies the deviations from the expected patterns under the null hypothesis. The common practice of significance level is set into 0.05 whereas, considering the data points less than 0.05 examines to provide statistically significant. Moreover, the small p-value represents the strong likelihood that determines the real anomaly. Thus, it facilitates to control the false alarms in real-time scenarios.

Analysis of Computational Complexity, Time and Memory Usage

Table 8 examines the computational complexity, time and memory usage is examined against diverse baseline models. Here, the number of population and the chromosome length denotes n *and* $Chlen$. The terms k and T denotes the number of kernel and sequence length (time steps), whereas the hidden unit is represented as h. Training time in the developed model shows a superior outcome of 35.2 (s) than the existing techniques. Within the limited time, the recommended ICPFO-HSCAN technique can accurately detect the anomalies in the time series data. CNN, on the other hand, the training time shows 44.8 (s) that degrades the system's performance. This causes delays to take recommended actions. Inference time analysis in the anomaly detection facilitates to evaluate the speed and efficiency during the training process. Moreover, it helps to understand the model quickly and analyzing the new data from the expected patterns. Thus, it helps to reduce the processing time and takes immediate actions to resolve from the issues. Moreover, the inference time shows 1.8 (ms) in the recommended technique ensures robustness analysis in the anomaly detection framework.

Table 4: Overall statistical evaluation of proposed algorithms over classical algorithms having 5 different folds

TERMS	WOA-HSCAN (Siahmarzkooh and Alimardani, 2020)	GRO-HSCAN (Sarjamei et al., 2021)	KOA-HSCAN (Waghmode and Patil, 2024)	PFO-HSCAN (Jaganathan et al., 2025)	ICPFO-HSCAN
1 Fold					
Specificity	83.34944418	84.85290542	81.74143753	86.46415553	90.52993131
FOR	13.90414378	12.88067898	15.40189715	11.18322516	7.888167748
FDR	14.22084623	12.85861713	15.62435501	11.49638803	7.966976264
Accuracy	85.92249463	87.13139758	84.47633036	88.64535081	92.0686928
F1_score	86.96379996	88.1143692	85.61256545	89.51048951	92.7027027
MCC	0.717028335	0.741113188	0.688005207	0.7716238	0.840284126
FPR	16.65055582	15.14709458	18.25856247	13.53584447	9.470068695
2 Fold					
FOR	14.20369446	9.660509236	11.60758862	7.588617074	6.265601598
MCC	0.716992541	0.801793176	0.767383547	0.847077571	0.872104648
Accuracy	85.93379279	90.14800588	88.44198396	92.40763756	93.65043498
FDR	13.95252838	10.01031992	11.51702786	7.595459236	6.418988648
FPR	16.43569171	11.81773879	13.61307636	9.041769042	7.648794884
Specificity	83.56430829	88.18226121	86.38692364	90.95823096	92.35120512
F1_score	87.00824377	90.90909091	89.34041888	93.01890713	94.16407061
3 Fold					
Accuracy	85.59484804	88.58886002	86.84894362	90.15930403	92.41893571
Specificity	83.18347509	86.53658537	84.37348815	88.27838828	91.02091021
FPR	16.81652491	13.46341463	15.62651185	11.72161172	8.979089791
FOR	14.55317024	11.43285072	12.93060409	9.760359461	7.638542187
FDR	14.28276574	11.39318885	13.33333333	9.907120743	7.533539732
F1_score	86.69241207	89.47478116	87.82681447	90.92802833	93.03291455
MCC	0.710186379	0.770351101	0.735637961	0.801947387	0.847275307
4 Fold					
Specificity	82.02898551	85.21485798	85.21400778	87.84841076	91.21472393
MCC	0.692484812	0.748908059	0.747702673	0.79303009	0.852801181
FPR	17.97101449	14.78514202	14.78599222	12.15158924	8.785276074
FOR	15.22715926	12.38142786	12.5312032	10.3095357	7.214178732
Accuracy	84.70229353	87.51553497	87.45904418	89.71867586	92.69009152
F1_score	85.83089159	88.46194006	88.41819699	90.52675411	93.27512733
FDR	15.35603715	12.56965944	12.54901961	10.25799794	7.389060888
5 Fold					
FOR	11.3330005	9.186220669	9.935097354	6.465302047	5.591612581
Specificity	86.57080185	88.97040841	88.30151738	92.17712177	93.2675709
FPR	13.42919815	11.02959159	11.69848262	7.822878229	6.7324291
FDR	11.37254902	9.308565531	9.865841073	6.563467492	5.634674923
F1_score	89.52361097	91.47496617	90.88449532	94.0089295	94.8449331
MCC	0.771515336	0.813737824	0.800739064	0.868683066	0.886853183
Accuracy	88.64535081	90.74680827	90.10281324	93.4809626	94.38481528

Table 5: Overall statistical evaluation of the proposed models over classical models having 5 different folds

TERMS	DBN (Zhang et al., 2022)	CNN (Liu and Wang 2023)	LSTM (Almahadin et al., 2024)	HSCAN (Liu and Wang 2023) Almahadin et al., 2024)	ICPFO-HSCAN
1 Fold					
Accuracy	87.65111287	87.07490679	87.94486499	89.62829059	92.0686928
Specificity	85.44658068	84.97067449	85.92031288	87.75061125	90.52993131
FPR	14.55341932	15.02932551	14.07968712	12.24938875	9.470068695
FDR	12.34262126	12.69349845	11.88854489	10.34055728	7.966976264
F1_score	88.59914468	88.08829654	88.8912025	90.44347283	92.7027027
MCC	0.751542876	0.73977999	0.757299106	0.791208967	0.840284126
FOR	12.3564653	13.20519221	12.25661508	10.40938592	7.888167748
2 Fold					
Accuracy	88.37419501	89.45881821	90.12540956	91.92181674	93.65043498
Specificity	86.20773534	87.55805427	88.25109917	90.43991153	92.35120512
FPR	13.79226466	12.44194573	11.74890083	9.560088474	7.648794884
FDR	11.70278638	10.50567595	9.927760578	8.028895769	6.418988648

Table 5: Continued

F1_score	89.26447574	90.28630921	90.89772964	92.57297185	94.16407061
MCC	0.766125559	0.787802387	0.801256566	0.837279293	0.872104648
FOR	11.53270095	10.58412381	9.810284573	8.13779331	6.265601598
3 Fold					
Accuracy	88.61145633	88.12563552	88.82612134	92.21556886	92.41893571
Specificity	86.63245357	85.94016054	86.74786845	90.73937607	91.02091021
FPR	13.36754643	14.05983946	13.25213155	9.260623925	8.979089791
FDR	11.28998968	11.92982456	11.22807018	7.78121775	7.533539732
F1_score	89.50437318	89.03495044	89.6882494	92.84155844	93.03291455
MCC	0.770738013	0.761119226	0.775174344	0.843212514	0.847275307
FOR	11.50773839	11.80728907	11.10833749	7.788317524	7.638542187
4 Fold					
Accuracy	87.23308101	87.33476443	89.74127217	90.1141114	92.69009152
Specificity	84.85215705	85.03764877	87.85434995	88.36069591	91.21472393
FPR	15.14784295	14.96235123	12.14565005	11.63930409	8.785276074
FDR	12.8998968	12.71413829	10.25799794	9.803921569	7.389060888
F1_score	88.1922675	88.29731705	90.545606	90.89963599	93.27512733
MCC	0.743289866	0.745247932	0.793501266	0.800929168	0.852801181
FOR	12.60609086	12.60609086	10.25961058	9.985022466	7.214178732
5 Fold					
Accuracy	88.71313976	90.76940459	91.1196475	93.83120551	94.38481528
Specificity	86.64391908	88.99486427	89.46078431	92.65285996	93.2675709
FPR	13.35608092	11.00513573	10.53921569	7.347140039	6.7324291
FDR	11.31062951	9.287925697	8.875128999	6.150670795	5.634674923
F1_score	89.58615657	91.49578432	91.82612313	94.33609959	94.8449331
MCC	0.772880879	0.814193113	0.821185316	0.875696654	0.886853183
FOR	11.25811283	9.161258113	8.886669995	6.190713929	5.591612581

Table 6: Overall statistical evaluation of proposed models over conventional models with different nodes

TERMS	WOA-HSCAN (Siahmarzkooh and Alimardani, 2020)	GRO-HSCAN (Sarjamei et al., 2021)	KOA-HSCAN (Waghmode and Patil, 2024)	PFO-HSCAN (Sarjamei et al., 2025)	ICPFO-HSCAN
No of nodes - 50					
Mean	9.536719264	9.554270677	9.486767967	9.56644752	9.266098163
Worst	10.29077664	10.78520617	13.2812184	12.79252708	14.83669455
Std	0.240152893	0.223464909	0.263095307	0.314028257	0.573681781
Best	9.453696889	9.496905818	9.443303165	9.511460369	9.188564663
Median	9.453696889	9.496905818	9.443303165	9.511460369	9.188564663
No of nodes -100					
Median	9.430953704	9.471378954	9.471635916	9.457801365	9.197782261
Worst	13.96001162	9.998941346	12.2870016	9.632214897	10.71515893
Std	0.41247817	0.116664079	0.348177335	0.082560012	0.096369964
Mean	9.518841942	9.538758505	9.542161909	9.518236425	9.209594581
Best	9.430953704	9.471378954	9.435539339	9.457801365	9.197782261
No of nodes -150					
Std	0.596585051	0.263041614	0.566115921	0.36450672	0.432009573
Median	9.457258873	9.440397997	9.438330543	9.431017208	9.176956304
Mean	9.687599554	9.518973739	9.571572896	9.543525352	9.262457658
Best	9.457258873	9.440397997	9.432646848	9.431017208	9.176956304
Worst	14.74810999	12.00484043	13.532144	14.52684112	14.80531415
No of nodes -200					
Best	9.429947389	9.469886633	9.497391919	9.447339999	9.184974059
Median	9.429947389	9.469886633	9.517032924	9.536924965	9.280556535
Mean	9.571348364	9.553398207	9.620795971	9.601542457	9.289239533
Std	0.402623238	0.242158039	0.449461412	0.455072542	0.166230214
Worst	14.46151565	12.34057659	15.12515686	14.43737922	11.62349634

Table 7: Statistical test analysis of developed anomaly detection model

Algorithms	Statistic	Adjusted p-value	Result
WOA vs ICPFO-HSCAN	1.31245	0.5421	
KOA vs WOA	1.68921	0.72365	
GRO vs ICPFO-HSCAN	1.80512	0.60133	
KOA vs GRO	0.73219	0.79441	
WOA vs PFO	0.92568	0.98101	
ICPFO-HSCAN vs PFO	0.59218	0.39927	H0 is accepted
KOA vs ICPFO-HSCAN	0.74315	0.6752	
GRO vs WOA	0.83421	1	
GRO vs PFO	0.50371	1	
KOA vs PFO	0.67852	1	

Table 8: Analysis of computational complexity, time and memory usage

Algorithms	Training Time (s)	Inference Time (ms)	Resource Usage (CPU %)	Memory Usage (MB)	Space Complexity
WOA (Siahmarzkooh and Alimardani, 2020)	41.2	5.1	Medium	130	$O(Iter(n^2 * Chlen + 1))$
GRO (Sarjamei et al., 2021)	40.6	4.9	Medium	125	$O(Iter(n\log n * Chlen))$
KOA (Waghmode and Patil, 2024)	43.5	5.6	High	140	$O(Iter(n^2 * Chlen))$
PFO (Jaganathan et al., 2025)	44.3	6	High	150	$O(Iter(n * Chlen + 1))$
DBN (Zhang et al., 2022)	45	3.5	Very High	260	$O(n * h)$
CNN (Liu and Wang, 2023)	44.8	2.9	High	230	$O(n * T^2)$
LSTM (Almahadin et al., 2024)	45	3.1	Very High	270	$O(n * T^2)$
HSCAN (Liu and Wang, 2023) (Almahadin et al., 2026)	43.9	2.5	Very High	290	$O(n * T + 2)$
ICPFO-HSCAN	35.2	1.8	Low	180	$O(n * T)$

Robustness Analysis

The robustness analysis of the recommended technique is executed in Fig. 12. Moreover, the robustness analysis is further validated, whereas the model's accuracy is inversely related to the Gaussian noise level. If the noise level increases, then detecting the anomalies in IoT time series data becomes decreases leads to affect the model's accuracy. Moreover, the presence of higher noise levels often leads to misclassification. From the given graph analysis, the Gaussian noise level is gradually decreased at the level of 0.2. At the noise level of 0.4 to 1.0, the Gaussian noise is minimized to improve the model accuracy. Determining the lower noise level, the model can learn complex patterns from the normal and abnormal data and ensures a higher accuracy rate for precisely identifying the anomalies in time series data. This validation helps the model to maintain the higher accuracy even, if the noise is present. Ensuring higher robustness analysis helps the model to easily adapt in several environmental conditions. Thus, it is a significant aspect to easily adapt in the real-world applications.

Confidence Interval of the Developed Model

Table 9 evaluates the confidence interval of the developed model, by comparing it against diverse techniques. Moreover, the analysis of confidence interval is specifically considered as anomalous, if the values fall outside the upper and lower bound of the predictive intervals. This prediction interval has the specified range, whereas the expected values fall within the specified range of 95%. In this context, the values attained from below the lower bound and above the upper bound are considered as anomalous. In Table 9, the developed ICPFO-HSCAN model attains 0.8411 (lower bound) and 0.9489 (upper bound) within the specified interval of 95%. Here, the observed value falls in the specified region. In essence, confidence interval in anomaly detection is a statistical framework that effectively identifies the normal data patterns from deviations that facilitates to detect more anomalies and take appropriate actions.

Comparative Analysis of Relevant Algorithms from this Research Field

Table 10 examines the comparative analysis of the relevant algorithms from this research field is compared to enhance the model's performance. This table analysis is compared against diverse batch size as 4, 8, 16, 32 and 48, respectively. Validating the batch size in anomaly detection can improvise the training efficiency and enhance better generalization ability in the intrinsic

patterns. Training with sufficient number of batch size can merely improve the model performance from neglicting the information loss. Selecting the relevant algorithms in this research field facilitates to strengthen the models accuracy and minimize the falsey outcomes. Here, the relevant algorithms of this research field is considered as Chameleon Swarm Optimization Algorithm (CSOA) (Shanmuganathan and Suresh, 2025) Genetic Algorithm with Attention Mechanism and Modified Adam Optimized LSTM (GA-mADAM-IIoT) (Saheed et al., 2025), Improved Bacterial Foraging Optimization with Optimum Deep Learning for Anomaly Detection (IBFO-ODLAD) (Khayyat, 2023) and Improved Grey Wolf Optimization (IGWO) (Manokaran and Vairavel, 2023). Moreover, the developed model achieves 94.54% at the 48th batch size. Hence, the precision of the developed model achieves 13.47, 16.73, 13.6 and 6.04% improved performance than CSOA, GA-mADAM-IIoT, IBFO-ODLAD and IGWO, respectively. Throughout the performance analysis, the developed model shows significant outcomes rather than the existing baseline algorithms.

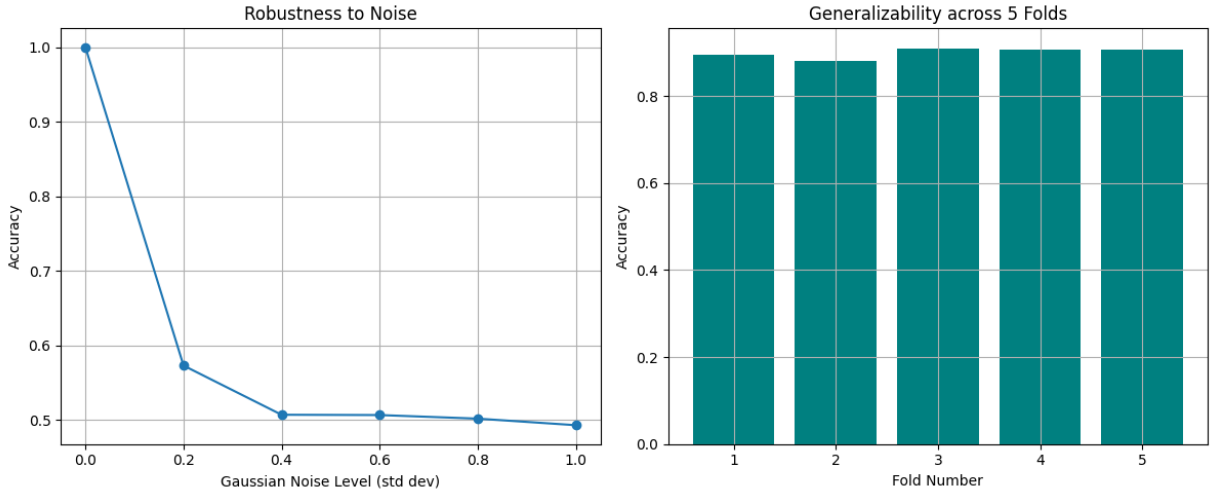


Fig. 12: Robustness analysis of the developed model

Table 9: Analysis of confidence interval of developed anomaly detection model

			Confidence Interval	
Model	Mean Acc	Std Dev	95% CI Lower	95% CI Upper
DBN (Zhang *et al*., 2022)	0.861	0.0318	0.8168	0.9052
CNN (Liu and Wang, 2023)	0.874	0.0402	0.8182	0.9298
LSTM (Almahadin *et al*., 2024)	0.88	0.0167	0.8568	0.9032
HSCAN (Liu and Wang, 2023)				
(Almahadin *et al*., 2024)	0.873	0.0242	0.8394	0.9066
Proposed ICPFO-HSCAN	0.895	0.0389	0.8411	0.9489

Table 10: Comparative analysis of relevant algorithms in this research field

TERMS	CSOA (Shanmuganathan and Suresh, 2025)	GA-mADAM-IIoT (Saheed et al., 2025)	IBFO-ODLAD (Khayy et al., 2023)	IGWO (Manokaran and Vairavel, 2023)	ICPFO-HSCAN
Accuracy					
4	85.40	83.52	84.30	89.57	93.21
8	86.17	84.79	85.81	90.24	93.48
16	87.56	86.13	87.45	90.90	93.84
32	88.93	87.71	88.89	91.51	94.30
48	90.00	89.12	90.50	92.28	94.54
Precision					
4	74.53	71.80	72.91	81.14	87.29
8	75.71	73.55	75.05	82.14	87.79
16	77.87	75.69	77.78	83.32	88.36
32	79.93	78.15	80.04	84.34	89.07
48	81.83	80.43	82.67	85.72	89.60

Table 10: Continued

FDR					
4	25.47	28.20	27.09	18.86	12.71
8	24.29	26.45	24.95	17.86	12.21
16	22.13	24.31	22.22	16.68	11.64
32	20.07	21.85	19.96	15.66	10.93
48	18.17	19.57	17.33	14.28	10.40
MCC					
4	68.67	64.77	66.42	77.34	85.09
8	70.26	67.46	69.60	78.79	85.67
16	73.16	70.18	72.89	80.16	86.44
32	76.05	73.44	75.92	81.46	87.46
48	78.26	76.39	79.31	83.08	87.97

Potential Errors/Uncertainity in these Measurements

In anomaly detection, training the insufficient number of training data, limiting the model's ability on generalizing the new data and inappropriate model assumptions can impact the model's performance. Uncertainity measurements in the anomaly detection is susceptible to noise affects entire accuracy of the system. However, discrepancies among the recorded data as well as the actual data affects the measurement quality. Yet, the presence of noise in the time series data can obscure the true nature of anomalies. These potential errors/uncertainity in the measurements have the possibilities to increases the false positives (normal data points incorrectly identified as anomalies) and false negatives (actual anomalies are incorrectly identified as normal data points). Increasing these false errors can tremendously affects the generalization performance of the unseen data. These potential errors is mitigated by collecting the data from the benchmark dataset, in which the whole dataset is split into training as well as testing phase. Implementing the robust anomaly detection model is helped for effectively identifying the noise and errors to enhance the model's performance. Based on this process, the occurrence of false positives and false negatives is highly reduced to improve the accuracy rate of the model. Various validation measures are taken for the experimental analysis to improves the model's generalizability of the unseen data.

Discussion on Performance of Hybrid Model Performs Better than Existing Techniques

In order to detect the precise anomaly detection, the HSCAN model is compared against with existing techniques to show better outcomes. The comparative analysis with diverse performance measures is further validated in Figs. 6 and 7 with diverse classifiers and algorithms. In particular, the analysis shows valuable insights to examine the overall performance of the model. Here, the existing KOA algorithm shows poor accuracy rate at the batch size of 16. Limited accuracy rate leads to false positives and negatives. HSCAN model, on the other hand, examines and provides reliable outcomes because of its higher accuracy rate. Increasing the amount of accuracy leads to enable earlier disease detection, minimize downtime and enhance the overall efficiency of the developed model. Moreover, the developed model can takes timely corrective actions to prevent from system failures. Representing the confusion matrix in the developed model in Fig. 8 shows actual and predicted outcomes. Moreover, the hybrid developed model helps to visualize the model's performance by correctly identifying the false positives and false negatives. Thus, this analysis helps to effectively understand the model's behaviour for precisely analyzing the anomalies. Fig. 9 examines the ROC analysis in the developed ICPFO-HSCAN model to show the comprehensive view of classifier performance to obtain the optimal thresholds. Moreover, the hybrid model offers the quantitative measure whereas, the higher ROC analysis indicates the developed model shows stronger performance by identifying various types of patterns and anomalies. Enabling the faster convergence speed is essential to get the optimal and desired outcomes. The existing algorithm easily stucks from the local optima issues due to the poor convergence rate. Also, it is sensitive to noise by considering the irrelevant and inconsistent data leads to false alarms. Unlike traditional optimization algorithms, the developed ICPFO-HSCAN model identifies the relevant data points that could effectively enables the timely detection of anomalies in the IoT system. Fig. 11 validates the delay, distance, energy consumption, PKT and throughput by comparing with the developed model against existing models. Increasing the throughput performance have the credentiability to transfer the data without any delay and minimize the potential threats. Overall performance of the developed model achieves 94.38% in terms of accuracy at 5th fold. Validating the K-fold helps the developed model have the ability to reduce the class imbalance issues and neglects overfitting issues. The performance enhancement in the developed model could have the possibilities to detect the anomalies and takes recommended actions to enhance the data quality. This helps the developed hybridized model ensure more reliable outcomes than the existing frameworks.

Discussion of Real-World IoT Systems

In IoT time series data, detecting anomalies is a crucial task for maintaining the reliability and security among the interconnected devices and systems. Moreover, the IoT time series data can be applied in various sectors, like smart cities, healthcare, cybersecurity and industrial IoT. In real-world systems, earlier detection of anomaly in IoT helps to prevent from the potential issues of security breaches, fraudulent activities and system failures. Owing to this, it facilitates diminishing the negative consequence:

- Smart cities: In recent times, the IoT has been incorporated in various devices like sensors, cameras, smart homes and connected vehicles, which generates more amount of data and potentially leads to cyberattacks. Anomaly detection facilitates to effectively identifies the malicious activity to prevent from the unauthorized access based on its traffic patterns and device behavior. In smart cities, the unusual traffic patterns are detected by preventing from the network congestion and potential accidents and takes proactive measures, especially in traffic management. Identifying the unusual energy consumption in smart homes and buildings further indicates the malfunctions and energy wastage
- Healthcare: In the healthcare sector, the anomaly detection is emphasized in patient monitoring, equipment monitoring, cybersecurity in healthcare. In a patient monitoring system, healthcare professionals can provides potential health risks based on the vital signs of heart rate, blood pressure and temperature and ensures better treatment planning. Monitoring the equipment in the healthcare can susceptible to prevent from costly breakdown and ensures patient safety. Incorporating the IoT in the healthcare can susceptible to cyberattacks further, detecting the anomalies helps to prevent from these cyberattacks
- Industrial IoT: Generally, industrial IoT could have the possibility to generate a large amount of time series data. Furthermore, detecting the anomalies in sensor data can effectively predict from equipment failures and allows from proactive maintenance and minimize downtime. Identifying the anomalies earlier in the industrial IoT can suggest the safety measurements to prevent from accidents and make sure the safety of the worker to enhance overall model efficiency

Conclusion

To detect and prevent anomaly issues, a new model framework has been explored in this work using IoT time series. From various websites, essential data was collected for this work. Then, the assembled data was given as input to the HSCAN for anomaly detection and prevention. Here, the proposed network HSCAN was a detailed combination of STA-AE and LSTM methods. Hence, using this network the anomaly was detected correctly. Further, the anomalies were ignored for the greater functioning of the data. Here, various parameters were used and optimized using the proposed ICPFO. Once the detection stage was completed, the prevention stage took place and it was carried out by routing, which included PDR, energy, and latency. The routing was tuned by the developed optimizer ICPFO. On the other side, the classical methods obtained 58.85, 101.25, 75.2, and 63.8% than the LSTM, DBN, HSCAN, and CNN respectively. Hence, when compared to the classical method results, the developed methods received the optimum outcomes. Moreover, while deploying the deep learning-oriented approaches for anomaly detection and prevention, the major benefits acquired by this model were the potential to deal with high-dimensional data, complex programming, and recognized anomalies in multiple data types such as time series and images.

Limitations and Future Scope

Hence the model improved high functionality when contrasted with the classical methods and algorithms. Still, it also faced difficulties in interpreting the results of models, data quality problems, high dimensionality, and data imbalance. While deploying the IoT time series data in real-time, it faces significant challenges like computational cost, redundant data and data inconsistencies leads to impact the model performance in terms of scalability, accuracy and efficiency. Sometimes, the real-time data from various IoT sources becomes complex and redundant affecting the quality of the data. However, the developed model needs to be incorporated in the real-time advanced AI system that enables faster detection and response to address this challenge. On the other hand, the scalability issues occur while managing the massive amount of data, which is generated by the IoT devices. Handling the massive number of data is crucial making it difficult to process and access the data in real-time system. These issues needs to be addressed by implementing the novel ensemble techniques, ensuring better valuable insights and provides decision-making performance. Henceforth, in future work, the developed anomaly detection and prevention process will be fortified by deploying adaptive and enhanced techniques along with the algorithms for achieving the issues affected by data during computation.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for

the opportunity to contribute to the field of research through this publication.

Funding Information

The authors have not received any financial support or funding to report**.**

Authors Contributions

Both the authors have equally contributed to this manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Aditya Sai Srinivas, T., & Manivannan, S. S. (2020). Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm. *Computer Communications*, *163*, 162–175. https://doi.org/10.1016/j.comcom.2020.03.031

Alaghbari, K. A., Lim, H.-S., Saad, M. H. M., & Yong, Y. S. (2023). Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction in IoT Networks. *IoT*, *4*(3), 345–365. https://doi.org/10.3390/iot4030016

Almahadin, G., Subburaj, M., Hiari, M., Sathasivam Singaram, S., Kolla, B. P., Dadheech, P., Vibhute, A. D., & Sengan, S. (2024). Enhancing Video Anomaly Detection Using Spatio-Temporal Autoencoders and Convolutional LSTM Networks. *SN Computer Science*, *5*(1). https://doi.org/10.1007/s42979-023-02542-1

Amarbayasgalan, T., Pham, V. H., Theera-Umpon, N., & Ryu, K. H. (2020). Unsupervised Anomaly Detection Approach for Time-Series in Multi-Domains Using Deep Reconstruction Error. *Symmetry*, *12*(8), 1251. https://doi.org/10.3390/sym12081251

Cakir, S., Toklu, S., & Yalcin, N. (2020). RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning. *IEEE Access*, *8*, 183678–183689. https://doi.org/10.1109/access.2020.3029191

Chen, Z., Chen, D., Zhang, X., Yuan, Z., & Cheng, X. (2022). Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT. *IEEE Internet of Things Journal*, *9*(12), 9179–9189. https://doi.org/10.1109/jiot.2021.3100509

Fang, L., Li, Y., Liu, Z., Yin, C., Li, M., & Cao, Z. J. (2021). A Practical Model Based on Anomaly Detection for Protecting Medical IoT Control Services Against External Attacks. *IEEE Transactions on Industrial Informatics*, *17*(6), 4260–4269. https://doi.org/10.1109/tii.2020.3011444

Guan, S., Zhao, B., Dong, Z., Gao, M., & He, Z. (2022). GTAD: Graph and Temporal Neural Network for Multivariate Time Series Anomaly Detection. *Entropy*, *24*(6), 759. https://doi.org/10.3390/e24060759

Huang, D., Shen, L., Yu, Z., Zheng, Z., Huang, M., & Ma, Q. (2022). Efficient time series anomaly detection by multiresolution self-supervised discriminative network. *Neurocomputing*, *491*, 261–272. https://doi.org/10.1016/j.neucom.2022.03.048

Hussain, F., Abbas, S. G., Pires, I. M., Tanveer, S., Fayyaz, U. U., Garcia, N. M., Shah, G. A., & Shahzad, F. (2021). A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks. *IEEE Access*, *9*, 163412–163430. https://doi.org/10.1109/access.2021.3131014

Jaganathan, R., Rajendran, K., & Sam Ponnukumar, P. (2025). Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET). *International Journal of Computing and Digital Systems*, *17*(1), 1–18. https://doi.org/10.12785/ijcds/1571111848

Jiang, J.-R., Kao, J.-B., & Li, Y.-L. (2021). Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning. *Applied Sciences*, *11*(15), 6698. https://doi.org/10.3390/app11156698

Khayyat, M. M. (2023). Improved bacterial foraging optimization with deep learning based anomaly detection in smart cities. *Alexandria Engineering Journal*, *75*, 407–417. https://doi.org/10.1016/j.aej.2023.05.082

Kumar, A., & Singh, D. (2024). Detection and prevention of DDoS attacks on edge computing of IoT devices through reinforcement learning. *International Journal of Information Technology*, *16*(3), 1365–1376. https://doi.org/10.1007/s41870-023-01508-z

Lai, T., Farid, F., Bello, A., & Sabrina, F. (2024). Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis. *Cybersecurity*, *7*(1). https://doi.org/10.1186/s42400-024-00238-4

Li, C., Zhang, H., Wang, Z., Wu, Y., & Yang, F. (2022). Spatial-Temporal Attention Mechanism and Graph Convolutional Networks for Destination Prediction. *Frontiers in Neurorobotics*, *16*. https://doi.org/10.3389/fnbot.2022.925210

Liu, H., & Wang, H. (2023). Real-Time Anomaly Detection of Network Traffic Based on CNN. *Symmetry*, *15*(6), 1205. https://doi.org/10.3390/sym15061205

Liu, Y., Garg, S., Nie, J., Zhang, Y., Xiong, Z., Kang, J., & Hossain, M. S. (2021). Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach. *IEEE Internet of Things Journal*, *8*(8), 6348–6358. https://doi.org/10.1109/jiot.2020.3011726

Malki, A., Atlam, E.-S., & Gad, I. (2022). Machine learning approach of detecting anomalies and forecasting time-series of IoT devices. *Alexandria Engineering Journal*, *61*(11), 8973–8986. https://doi.org/10.1016/j.aej.2022.02.038

Manokaran, J., & Vairavel, G. (2023). IGWO-SoE: Improved Grey Wolf Optimization Based Stack of Ensemble Learning Algorithm for Anomaly Detection in Internet of Things Edge Computing. *IEEE Access*, *11*, 106934–106953. https://doi.org/10.1109/access.2023.3319814

Miranda, C., Kaddoum, G., Boukhtouta, A., Madi, T., & Alameddine, H. A. (2022). Intrusion Prevention Scheme Against Rank Attacks for Software-Defined Low Power IoT Networks. *IEEE Access*, *10*, 129970–129984. https://doi.org/10.1109/access.2022.3228170

Nagaraju, R., Pentang, J. T., Abdufattokhov, S., CosioBorda, R. F., Mageswari, N., & Uganya, G. (2022). Attack prevention in IoT through hybrid optimization mechanism and deep learning framework. *Measurement: Sensors*, *24*, 100431. https://doi.org/10.1016/j.measen.2022.100431

Saheed, Y. K., Omole, A. I., & Sabit, M. O. (2025). GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data. *Sensors International*, *6*, 100297. https://doi.org/10.1016/j.sintl.2024.100297

Sana, L., Nazir, M. M., Yang, J., Hussain, L., Chen, Y.-L., Ku, C. S., Alatiyyah, M., Alateyah, S. A., & Por, L. Y. (2024). Securing the IoT Cyber Environment: Enhancing Intrusion Anomaly Detection With Vision Transformers. *IEEE Access*, *12*, 82443–82468. https://doi.org/10.1109/access.2024.3404778

Sarjamei, S., Sadegh Massoud, M., & Esfandi Sarafraz, M. (2021). Gold Rush Optimization Algorithm. *International Journal of Optimization in Civil Engineering*, *11*(2), 291–327.

Shanmuganathan, V., & Suresh, A. (2025). Parallel Residual Stacked Bidirectional Long Short-Term Memory Network Optimized with Chameleon Swarm Optimization Algorithm for Time-Series Sensor Data. *IETE Journal of Research*, *71*(4), 1422–1431. https://doi.org/10.1080/03772063.2024.2448588

Sharma, D. K., Dhankhar, T., Agrawal, G., Singh, S. K., Gupta, D., Nebhen, J., & Razzak, I. (2021). Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks. *Ad Hoc Networks*, *121*, 102603. https://doi.org/10.1016/j.adhoc.2021.102603

Shende, S. (2020). Long Short-Term Memory (LSTM) Deep Learning Method for Intrusion Detection in Network Security. *International Journal of Engineering Research And*, *9*(6). https://doi.org/10.17577/ijertv9is061016

Siahmarzkooh, A. T., & Alimardani, M. (2020). A Novel Anomaly-based Intrusion Detection System using Whale Optimization Algorithm WOA-Based Intrusion Detection System". *International Journal of Web Research*.

Sivasankari, N., & Kamalakkannan, S. (2022). Detection and prevention of man-in-the-middle attack in iot network using regression modeling. *Advances in Engineering Software*, *169*, 103126. https://doi.org/10.1016/j.advengsoft.2022.103126

Truong, H. T., Ta, B. P., Le, Q. A., Nguyen, D. M., Le, C. T., Nguyen, H. X., Do, H. T., Nguyen, H. T., & Tran, K. P. (2022). Light-weight federated learning-based anomaly detection for time-series data in industrial control systems. *Computers in Industry*, *140*, 103692. https://doi.org/10.1016/j.compind.2022.103692

Tukur, Y. M., Thakker, D., & Awan, I. (2021). Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things. *Transactions on Emerging Telecommunications Technologies*, *32*(6). https://doi.org/10.1002/ett.4158

Waghmode, S., & Patil, B. M. (2024). Adaptive load balancing in distributed cloud environment: Hybrid Kookaburra-Osprey optimization algorithm. *Intelligent Decision Technologies*, *18*(3), 1933–1954. https://doi.org/10.3233/idt-240672

Wu, D., Jiang, Z., Xie, X., Wei, X., Yu, W., & Li, R. (2020). LSTM Learning With Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT. *IEEE Transactions on Industrial Informatics*, *16*(8), 5244–5253. https://doi.org/10.1109/tii.2019.2952917

Wu, J., & Liu, H. (2023). Peregrine Falcon Optimization: A Novel Nature-Inspired Meta-Heuristic Algorithm for Global Optimization and Engineering Design Problems. *SSRN*.

Xu, R., Cheng, Y., Liu, Z., Xie, Y., & Yang, Y. (2020). Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services. *Future Generation Computer Systems*, *112*, 228–242. https://doi.org/10.1016/j.future.2020.05.035

Yin, C., Zhang, S., Wang, J., & Xiong, N. N. (2022). Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(1), 112–122. https://doi.org/10.1109/tsmc.2020.2968516

Zhang, L., Cheng, B., & Lin, F. (2022). Hyperspectral anomaly detection via fractional Fourier transform and deep belief networks. *Infrared Physics & Technology*, *125*, 104314. https://doi.org/10.1016/j.infrared.2022.104314

Zhang, M., Guo, J., Li, X., & Jin, R. (2020). Data-Driven Anomaly Detection Approach for Time-Series Streaming Data. *Sensors*, *20*(19), 5646. https://doi.org/10.3390/s20195646

Zhao, P., Chang, X., & Wang, M. (2021). A Novel Multivariate Time-Series Anomaly Detection Approach Using an Unsupervised Deep Neural Network. *IEEE Access*, *9*, 109025–109041. https://doi.org/10.1109/access.2021.3101844