International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(10s) 2026 pp. 893-922
https://doi.org/10.70917/ijcisim-2026-3683
Article

IMAO-DrMACN: Threat Detection in IoT Networking for Secure Communication in Wireless Networks

Kapil Dnyaneshwar Dere[1], Pushpalata G. Aher[2]

[1]Department of Computer Engineering School of Engineering & Technology, Sandip University, Trimbak Road, Nashik, Maharashtra, India, 422002, Email: kapilddere@gmail.com, ORCID: 0000-0002-9432-9434

[2]Department of Computer Engineering School of Engineering & Technology, Sandip University, Trimbak Road, Nashik, Maharashtra, India, 422002, Email: pushpalata.aher@sandipuniversity.edu.in, ORCID ID: 0000-0001-9841-7215

Corresponding Author Email: kapilddere@gmail.com

Abstract: The rapid growth of the Internet of Things (IoT) has made securing wireless networks increasingly difficult. As IoT devices interconnect, the network becomes more exposed to security threats that affect confidentiality, integrity and availability. Existing threat detection systems remain limited by high false-positive rates, poor adaptability to emerging attack patterns, and degradation of performance when the statistical properties of network traffic shift over time. This research addresses the adaptability and feature-learning limitations of existing detectors by introducing a framework named Intelligent Migration Avian Optimization enabled Drift-enhanced Modified Activation-based Convolutional Neural Network (IMAO-DrMACN). The proposed IMAO algorithm performs hyperparameter tuning, so that the model can be re-tuned as the IoT environment changes; the drift mechanism allows the model to respond to significant changes in the data distribution rather than degrading silently; and the modified activation function, which introduces a learnable channel-wise slope, improves feature learning in the convolutional blocks. The framework is evaluated on the DAPT2020 and BoT-IoT datasets under a 90% training / 10% testing split. On DAPT2020 it attains an accuracy of 97.531%, a recall of 97.476%, a precision of 97.687% and an F1-score of 97.417%; on BoT-IoT it attains 96.905%, 97.763%, 95.004% and 96.364% respectively. These correspond to accuracy gains of 2.67 and 2.95 percentage points over the strongest of the eight comparison methods (AM-DrCN) on the two datasets. The results indicate that coupling adaptive hyperparameter search with a drift-aware convolutional classifier is an effective strategy for threat detection in dynamic IoT environments.

Keywords: Internet-of-Things, Threat Detection, Concept Drift, Intelligent Migration Avian Optimization, Modified Activation, Convolutional Neural Network.

1. Introduction

The Statista report forecasts that the global number of Internet of Things (IoT) devices could reach 30 billion by 2030. As these devices become more widespread, cyber risks also increase, posing significant challenges to the safety of various systems and networks[1]. While IoT offers enhanced convenience, it also raises serious concerns regarding privacy and security. These networks are increasingly becoming prime targets for hackers. Research from Palo Alto Networks' Unit 42 indicates that 98% of IoT device communications are unencrypted, and 40% of cyberattacks take advantage of vulnerabilities in these devices[2]. The rapid growth of IoT technology has led to a vast number of interconnected devices, creating an extensive web of systems. However, this increased connectivity brings new security risks, as these devices are often vulnerable to threats such as malware, data breaches, and network interference[3]. Cybercriminals are attracted to IoT systems for several reasons, with three main factors standing out: first, the inherent security weaknesses in sensor devices, often due to manufacturing flaws; second, the insufficient defense mechanisms in edge network zones; and third, the high value of the data stored within these systems, making them appealing targets[4]. Cyber threats include any malicious activities aimed at stealing information or damaging resources. These threats have resulted in financial losses and compromised IoT-based infrastructures, highlighting the importance of threat detection as a critical area of research for IoT networks[5].

The Industrial Internet of Things (IIoT) brings together IoT innovations with industrial control systems (ICSs), which have been vital for managing machinery and processes in critical infrastructure for a long time. However, the enormous amount of data generated by sensors in IIoT networks has turned these systems into prime targets for global cyber threats[6]. Attacks on IIoT networks can come in various forms and intensities, including cyber warfare (high-level), espionage (mid-level), and cybercrime (low-level)[7]. These attacks often target essential components like switches, sensors, and CCTV cameras, which are crucial for machine-to-machine (M2M) and industrial communication networks [8]. Conventional cybersecurity measures, such as Intrusion Detection Systems (IDS), user authentication, firewalls, antivirus software, and encryption[9], aim to safeguard users and devices. Although Machine Learning (ML) models have shown effectiveness in spotting unusual activities and harmful traffic, relying solely on IDS solutions is often insufficient. Rule-based IDS methods, which were once widely used, have lost their effectiveness due to the complexity and scale of contemporary IoT systems [10]. In the past, threat detection depended heavily on manual statistical rules and data analysis, which were labor-intensive processes [11] [12]. Therefore, it is essential to develop strategies for identifying and mitigating such traffic to improve intrusion detection and prevention systems[13].

The Internet of Things (IoT) ecosystems have been recognized as having vulnerabilities that make them susceptible to various types of cyberattacks. In addition to the potential loss of critical data, issues like data confidentiality, user privacy, and system accessibility heighten these risks. By monitoring IoT devices and pinpointing vulnerable assets, it becomes possible to anticipate the types of attacks that may target cost-effective IoT products[14]. Moreover, it is crucial to consistently monitor and assess IoT security measures and networks to reduce damage to system components, ensuring strong protection by considering the societal implications and ethical use of these technologies [15]. Utilizing robust computational and storage resources allows for the analysis of historical data, identification of potential threats, prediction of future network behavior, and improvement of situational awareness. Implementing security measures within edge computing platforms, such as IoT Edge[16], can significantly decrease the chances of attacks and prevent these systems from being used to target other network components. This strategy also guarantees the smooth operation of IoT functions while reducing the excessive resource use that security measures can cause[17]. Furthermore, even with the implementation of established hardware and software protections like anti-malware programs, firewalls, intrusion detection systems (IDS), and intrusion prevention systems, many organizations still face digital threats. This vulnerability arises from the inadequate resilience of these solutions in addressing both contemporary and well-known cyber risks. As a result, it is essential to create an advanced threat detection framework specifically designed to protect IoT-integrated systems from a range of cyber threats[18].

To tackle these challenges that arise in existing threat detection models, this study presents an innovative approach for IoT threat detection by combining a customized convolutional neural network (CNN) architecture with the improved IMAO algorithm. The approach starts with the collection of a reliable, comprehensive dataset, which forms the foundation for accurate analysis. The preprocessing phase ensures the data quality by identifying missing values through the K-nearest neighbor imputation and normalizing the data with the quantile transformation, thus enhancing its consistency. The core contribution of IMAO-DrMACN lies in its DrMACN model, which adapts dynamically to changing data patterns, continuously refining feature extraction for higher accuracy in threat detection. To ensure sustained performance, the IMAO algorithm fine-tunes the model, optimizing its adaptability to evolving conditions. This unique combination of advanced data preprocessing, dynamic adaptation, and optimization enables the IMAO-DrMACN model to deliver a robust, scalable solution that is capable of detecting complex threats in real-time, making it a crucial advancement in the evolving field of IoT security.

The key spotlights are listed as follows:

- ✓ **Drift-enhanced Modified Activation-based CNN(DrMACN):** The drift mechanism allows the model to identify the changes that occur in data patterns, maintaining reliable performance in dynamic IoT settings. The modified activation function boosts the CNN's capacity to learn intricate features, enhancing detection accuracy for diverse threats. Together, these techniques tackle the shortcomings of conventional methods, providing a stronger and more adaptable solution for precise IoT threat detection.
- ✓ **Intelligent Migration Avian Optimization enabled Drift enhanced Modified Activation-based Convolutional Neural Network (IMAO-DrMACN):** The use of IMAO lies in its ability to dynamically tune hyperparameters, allowing the model to adapt to frequent changes of data distribution and achieve optimal performance. This minimizes the need for manual intervention, boosts efficiency, and enhances detection accuracy in dynamic IoT environments. The drift mechanism-based modified activation CNN

effectively addresses data drift and captures complex patterns, ensuring robust and accurate threat detection over time.

The remainder of the paper is organised as follows. Section 2 reviews the related literature, identifies the research gaps and states the problem addressed. Section 3 presents the system model. Section 4 describes the proposed IMAO-DrMACN framework, covering data acquisition, pre-processing, the drift-enhanced modified activation-based CNN classifier and the IMAO algorithm. Section 5 reports the experimental setup, datasets, evaluation metrics and results, including the comparative, statistical and convergence analyses. Section 6 concludes the paper, states its limitations and outlines directions for future work.

2. Literature Review

Dey, A.K.*et al.* [18] created a network intrusion detection system (NIDS) that uses metaheuristic techniques to spot cyber threats by combining the selection of attributes with ensemble categorization methods. Their framework operates in three phases: data preprocessing, attribute selection, and ensemble-based classification, which helps in making quick decisions in IoT-connected environments. Although the system showed a better positive detection rate, it faced challenges in identifying zero-day attacks coming from IoT network traffic.

Alturki, N. *et al*. [19] designed a framework based on a Regression Network (R-Net) model to enhance the security of cyber-physical systems, particularly satellite networks and IoT-assisted aerial vehicles. The methods employed include machine learning algorithms, such as deep learning and anomaly detection techniques, integrated with real-time monitoring systems to identify and mitigate potential threats. A key advantage of this framework is its ability to process large volumes of data, enabling swift detection and response to security breaches. However, a notable drawback is the reliance on high computational resources, which may limit its applicability in resource-constrained environments.

Hasan, T.[6] Designed a lightweight framework for enhanced intrusion detection in Industrial IoT (IIoT) networks. The framework leverages an autoencoder-based feature learning technique to identify potential intrusions in real-time. The key methods utilized in this approach include unsupervised learning, anomaly detection, and feature reduction, enabling the detection of both known and unknown threats. However, a notable limitation is its potential vulnerability to attacks with highly complex patterns, as the autoencoder model may struggle to detect sophisticated adversarial strategies.

Javed, S.H. *et al.* [8] introduced a system designed to identify and counter Advanced Persistent Threats (APTs) in I-IoT environments. The authors utilize a mix of ML techniques, including deep learning models and anomaly detection algorithms, to scrutinize network traffic and system behavior. A key benefit of this approach is its capacity to adapt to changing attack patterns, ensuring strong protection for essential industrial infrastructure. However, the system's dependence on large-scale data means that the model has not yet been applied to real-life congestion analysis in I-IoT contexts. A significant highlight of the system is its impressive detection accuracy, reported to exceed 95%, showcasing its capability to identify APTs while maintaining a low rate of false positives.

Cam, N.T., and Trung, N.G. [4] employed supervised learning techniques such as Random Forest and Support Vector Machines (SVM), along with Principal Component Analysis (PCA), to identify anomalies. By applying PCA, they were able to reduce the data's dimensionality, which improved the model's predictive performance. Although the model showed optimal efficiency and shorter training times, it faced challenges with multiclass classification. Nevertheless, the framework achieved an impressive detection accuracy of 98.7%, demonstrating its effectiveness in accurately identifying IoT-related threats.

Ferrag, M.A. *et al.* [1] presented a method that uses a lightweight BERT-based model specifically designed for IoT and IIoT environments. The focus is on applying natural language processing (NLP) techniques to analyse network traffic data, converting it into textual formats that the model can interpret. This strategy prioritizes privacy by reducing the risk of exposing sensitive information while still achieving high detection accuracy. A notable success of this model is its detection accuracy of 95.3%, highlighting its capability to identify cyber threats effectively. However, a potential limitation is its dependence on simulated data, which might not fully reflect the complexities of actual attack patterns.

Alkhonaini, M.A. *et al.* [10] combined the Sine-Cosine Algorithm (SCA) and Chimp Optimization Algorithm (ChOA) for feature selection, integrated with a deep learning model to improve threat detection in IoT sensor networks. The main advantage of this approach is its ability to achieve high detection accuracy while reducing dimensionality. However, a potential downside is the increased complexity involved in tuning the hybrid optimization

algorithm, which may demand significant computational resources during the training phase. The research employs benchmark IoT threat detection datasets to assess the model's performance. One notable achievement reported is an enhanced threat detection accuracy of over 98%, showcasing the effectiveness of the hybrid method in identifying malicious activities within IoT networks.

Alrayes, F.S. *et al.* [12] introduce a new method that combines the Enhanced Gorilla Troops Optimizer (EAGTO) algorithm with deep learning techniques to improve cybersecurity threat detection. By harnessing optimization, the approach significantly boosts the ability to identify cyber threats. A major benefit of this method is its high accuracy in threat detection, achieved by effectively balancing exploration and exploitation through the GTO algorithm. On the downside, the computational complexity involved in optimizing deep learning models can lead to longer training times. Nevertheless, the framework boasts an impressive accuracy rate of 99.4% on the NSL-KDD dataset, showcasing its effectiveness in detecting cybersecurity threats when compared to traditional methods.

Alblehai, F. [20] introduced an Intelligent Cybersecurity System for IoT using Self-Attention Deep Learning and Metaheuristic Algorithms. The system utilizes Bidirectional LSTM with self-attention for threat detection and Improved Tuna Swarm Optimization (ITSO) for feature selection, with hyperparameter tuning performed through the Hunger Games Search (HGS) technique. The framework achieved an impressive 99.37% accuracy on the TON IoT and Edge-IIoT datasets, demonstrating robust detection performance in complex IoT environments. However, the system's reliance on metaheuristic tuning may hinder real-time deployment, and it has only been tested on a limited number of datasets, raising concerns about its generalization to broader network scenarios.

Khan, M.Z., Reshi, A.A., Shafi, S. [21] developed an Adaptive Hybrid Framework for IIoT Intrusion Detection, combining ANN for classification with feature optimization via Genetic Algorithm (GA). The framework demonstrated 99.5% test accuracy on a large IIoT traffic dataset, achieving efficient feature reduction and excellent generalization. However, the system was tested primarily on generic network traffic rather than strictly wireless IIoT, and the computational overhead of GA introduced some performance challenges, particularly in terms of processing time.

Poonguzhali, C. *et al.* [22] presented a comprehensive survey on Intrusion Detection Systems in IoT-Enabled Wireless Sensor Networks, exploring various detection techniques such as anomaly detection, signature-based detection, and potential approaches. The paper discusses the challenges associated with implementing and deploying IDS in IoT WSNs. These networks are specified by a large number of interconnected devices with fewer resources, which makes them exposed to various security threats.

P. Rajesh Kanna[23] proposed a BWO-CONV-LSTM model for intrusion detection. It combines CNN and LSTM for spatial-temporal feature learning, with ABC for feature selection and BWO for hyperparameter tuning. The model shows high accuracy on multiple datasets. A key limitation is its complexity for real-time use. The OCNN–HMLSTM model [24] proposed by Rajesh Kanna and Santhi combines CNN and LSTM to detect network attacks using both spatial and temporal features. It uses Lion Swarm Optimization to improve CNN performance and shows high accuracy on multiple datasets. However, the model is complex and needs more time and resources to train.

Rajesh Kanna et al. proposed a PCA-based Defensive Mechanism [25] that uses Principal Component Analysis to reduce complexity and improve detection of DoS attacks by analyzing network traffic features. It achieves high accuracy on the KDD Cup 99 dataset and performs well compared to other methods. However, it cannot detect certain attacks like Land, Teardrop, and Neptune, limiting its coverage. Rajesh Kanna and Santhi present a multi-level distributed approach[26] using attack graphs and reconfigurable virtual systems to enhance network security and attack detection. Their signature-based intrusion detection method shows better precision, recall, accuracy, and efficiency compared to other techniques. However, this method may not fully address complex attacks like insider threats or malware infiltration.

The IMAO-DrMACN model introduces an innovative approach to IoT threat detection by integrating a modified CNN architecture with the IMAO algorithm, enabling dynamic hyperparameter adjustment to effectively manage data drift. Existing threat detection models often struggle with key limitations such as static parameter settings, poor adaptability to evolving data patterns, limited feature extraction capabilities, and reduced accuracy when dealing with large-scale IoT data. These issues lead to degraded performance over time, especially in environments with frequent behavioral changes and complex attack types. The proposed IMAO-DrMACN model addresses these challenges by enhancing feature learning and allowing the model to adapt in real-time to shifting data distributions. As a result, it improves both the accuracy and adaptability of IoT threat detection systems. The model demonstrates

superior performance compared to traditional machine learning techniques and advanced deep learning models, achieving higher accuracy and robustness on benchmark datasets such as DAPT2020 and Bot-IoT.

Some of the earlier ways to identify threats are described with their drawbacks, are represented in Table 1.

Table 1. Literature review

Author(s)	Year	Method	Methodology	Key Findings	Limitations
Dey, A. K. et al. [18]	2023	Decision Tree, AdaBoost , and Random Forest (RF)	Data preprocessing, attribute selection (metaheuristics), ensemble-based classification	Better positive detection rate in IoT-connected environments	Difficulty in detecting zero-day attacks in IoT traffic
Alturki, N. et al. [19]	2023	(R-Net)	Deep learning, anomaly detection, and real-time monitoring for satellite & aerial IoT systems	Processes large data volumes, swift detection, and response	High computational resource requirements are not fit for resource-constrained environments
Javed, S. H. et al. [8]	2022	Deep learning	Deep learning and anomaly detection, analyzing network traffic and system behavior	Adaptive to evolving attack patterns; detection accuracy > 95%, low false positives	It relies on large-scale data and has not yet been applied to real-world congestion analysis in IoT environments.
Cam, N. T. & Trung, N. G. [4]	2023	RF (Random Forest), SVM (Support vector machine) PCA (Principal Component Analysis)	Supervised learning: Random Forest, SVM, PCA for dimensionality reduction	Efficient, fast training; detection accuracy 98.7%	Challenges in handling multiclass classification
Ferrag, M. A. et al. [1]	2024	natural language processing (NLP)	NLP-based conversion of network traffic to text and a lightweight BERT model	Detection accuracy ~95.3%; privacy-preserving approach	Dependence on simulated data may not reflect real-world attack complexity
Alkhonaini, M. A. et al. [10]	2024	Deep learning	Metaheuristic hybrid optimization (SineCosine Algorithm and Chimp Optimization) and deep learning	High threat detection accuracy > 98%, reduced dimensionality	Complexity in tuning the hybrid algorithm; significant computational resources needed
Hasan, T.*et.al*[6]	2025	Machine Learning	Autoencoder-based feature learning, anomaly detection, unsupervised learning, feature reduction	Low computational overhead for intrusion detection in resource-constrained IIoT environments	Vulnerability to sophisticated, complex attacks and adversarial strategies
Alrayes, F. S.*et.al*.[12]	2023	Deep learning	Converted binary files to images; cascaded GRU (CGRU) for classification; hyperparameters tuned via Enhanced Artificial Gorilla Troops	Achieved 99.47% accuracy in distinguishing benign vs. malicious samples	Tested on a single binary-image dataset (malicious vs benign); conversion may not generalize to other file types; CGRU and metaheuristic tuning increase

			Optimizer (EAGTO) metaheuristic		computational overhead
Alblehai, *et al.* F.[20]	2025	LSTM with self-attention	Bidirectional LSTM with self-attention; feature selection using improved Tuna Swarm Optimization (ITSO); hyperparameters tuned via Hunger Games Search (HGS)	Achieved 99.37% accuracy on TONIoT and EdgeIIoT datasets; balanced precision/recall, and demonstrated robust detection in complex IoT environments.	Heavy reliance on metaheuristic tuning may hinder real-time deployment; only been tested on two datasets, not validated across broader network scenarios.
Khan, M. Z., *et al*.[21]	2025	ANN	ANN for classification; feature selection via Genetic Algorithm	Reached 99.5% test accuracy; efficient feature reduction and high generalization on a large IIoT traffic dataset	Moderate computational overhead from GA
Poonguzhali, C. *et al.* [22]	2024	Machine learning and Advanced Deep Learning	Survey of various IDS techniques in IoT-enabled WSNs	Identifies detection approaches and challenges in WSN-based IDS	Resource limitations and complexity in IDS deployment
Rajesh Kanna *et al.* [23]	2021	CNN + LSTM hybrid	BWO-CONV-LSTM model combining CNN and LSTM for spatial-temporal feature learning, with ABC for feature selection and BWO for hyperparameter tuning	High accuracy on multiple datasets	Complex on real-time deployment
Rajesh Kanna and Santhi [24]	2021	CNN + LSTM with Lion Swarm Optimization	OCNN-HMLSTM model combining CNN and LSTM for attack detection using both spatial and temporal features	High accuracy in detecting network attacks using spatial-temporal features	Needs more time and resources for training
Rajesh Kanna et al. [25]	2017	PCA-based defensive mechanism	PCA-based defensive mechanism for DoS attack detection using Principal Component Analysis to reduce complexity	High accuracy on KDD Cup 99 dataset, effective in DoS detection	Cannot detect certain attacks (e.g., Land, Teardrop, Neptune)
Rajesh Kanna and Santhi [26]	2021	Multi-level distributed approach	Multi-level distributed approach with attack graphs and reconfigurable virtual	Better precision, recall, accuracy, and efficiency than other methods	May not fully address complex attacks like insider threats or malware infiltration

			systems for enhanced network security		

2.1 Research Gaps

The issues inherent in the identification of threats are outlined from the aforementioned approaches.

- The intelligent NIDS framework of [18] cannot identify zero-day threats in network traffic, which limits its ability to detect attacks in real time.
- The APT detection system of [8] was not tested in real-world IIoT environments; as a result, it is not established that it can handle network congestion or the complexities introduced by diverse datasets.
- The method combining machine learning with PCA for DDoS attack detection [4] has not been validated on real-world datasets. It also performs poorly on multiclass classification, which limits its practical application.
- The GA-mADAM-IIoT model faces significant computational overhead [6], making it difficult to deploy in resource-constrained IIoT environments due to its complexity.
- The LLM-based IoT security framework [7] shows promise but needs further evaluation in diverse real-world IoT environments. Its adaptability to new, unseen attack vectors and scalability in large-scale IoT networks remains unclear.

2.2 Problem Statement

The rapid expansion of IoT networks has introduced significant security challenges, particularly in wireless communication environments, where traditional intrusion detection systems (IDS) struggle due to the dynamic and heterogeneous nature of IoT networks. Existing models face limitations such as high false positive rates, limited adaptability to new attack patterns, and inefficiencies in real-time threat detection. For instance, the intelligent NIDS framework presented by Dey et al [18] cannot identify zero-day threats, reducing its ability to detect attacks in real-time. The proposed IMAO-DrMACN framework addresses this by leveraging a modified CNN architecture, which enables it to capture emerging attack patterns and adapt to evolving threats. Similarly, the APT detection system [8] struggles with network congestion and diverse datasets, whereas the IMAO-DrMACN uses dynamic hyperparameter tuning to continuously adjust to shifting network conditions, ensuring robustness in complex environments. The PCA-based method for DDoS detection [4] has limitations in multiclass classification and lacks real-world validation. In contrast, IMAO-DrMACN is evaluated on multiclass classification using two public benchmark datasets, DAPT2020 and BoT-IoT, on which it attains higher accuracy, precision, recall and F1-score than the eight comparison methods considered in Section 5.

The general process for threat detection in IoT consists of several stages, beginning with the collection of data from IoT datasets that can be arithmetically represented as

$$R = \{r_1, r_2, ..., r_s, ..., r_t\} \tag{1}$$

where R indicates the database, in which denotes a single data sample r_s with various parameters, and r_t denotes the entire data of the database. Further, the vector r_s that contains a range of parameters m such as packet size, bandwidth, protocol type, uptime, timestamps, attack label, and Other relevant parameters can be determined as,

$$r_s = \left\langle \{r_{s1}, ..., r_{sd}, ..., r_{sm}\} \right\rangle, t_s \in \{0, 4\} \tag{2}$$

where r_{sd} denotes one of the various parameters in the entire database, and r_{sm} denotes the total parameters. Further, the exact values of the database can be represented as t_s belonging to [0,4]. The data after collection goes through preprocessing, which involves cleaning, normalization, and eliminating noise, and preparing the dataset for training, and is represented as r_{sd}^*. Next, the outcome is employed for the model training phase, where advanced deep learning techniques are utilized to identify patterns and differentiate between benign and malicious activities. The performance is more precisely enhanced by diminishing the function of error. The model's error rate is measured using the categorical cross-entropy loss and is shown as,

$$\Delta_{loss} = -\sum_{s=1}^{t} t_s \log(c_s) \tag{3}$$

where, t_{s1} and t_{s2} determines the exact outputs from the multiple datasets, whereas c_s determines the output that is predicted by the model.

$$t_{s1} = \begin{cases} s = 0, if\ c_s\ beningn \\ s = 1, if\ c_s\ data\ exfiltration \\ s = 2, if\ c_s\ establish\ foothold \\ s = 3, if\ c_s\ lateral\ movement \\ s = 4, if\ c_s\ reconnaissance \end{cases} \tag{4}$$

$$t_{s2} = \begin{cases} s = 0, if\ c_s\ DoS \\ s = 1, if\ c_s\ DDoS \\ s = 2, if\ c_s\ reconnaissance \\ s = 3, if\ c_s\ normal \\ s = 4, if\ c_s\ theft \end{cases} \tag{5}$$

Finally, the system's effectiveness is assessed and categorized based on its type and vulnerability.

3. System Model

IoT networks, which consist of interconnected devices that gather and share data, have become essential in various modern applications across different industries. However, the widespread use of these networks has brought about significant security challenges, especially in identifying threats like malware, DDoS attacks, and unauthorized access. Detecting threats often falls short of addressing the dynamic and diverse nature of IoT environments, highlighting the need for more advanced solutions. This system model suggests a structured approach to detecting threats in IoT, utilizing deep learning techniques to improve accuracy and efficiency. The process starts with collecting data from IoT device logs, network traffic, or publicly available datasets. The raw data undergoes pre-processing to manage missing values, normalize features, and eliminate noise, ensuring that the input for the model is of high quality. Following this, the cleaned data is input into a deep learning model, which is trained to recognize patterns that indicate potential threats. The model's performance is validated using a separate test dataset. Predictions are then made on new data to identify possible threats, and the system's effectiveness is evaluated using performance metrics. A visual representation of this workflow is depicted in Figure 1, which demonstrates the integration of processes carried out for detecting threats, offering a comprehensive solution for securing IoT network.

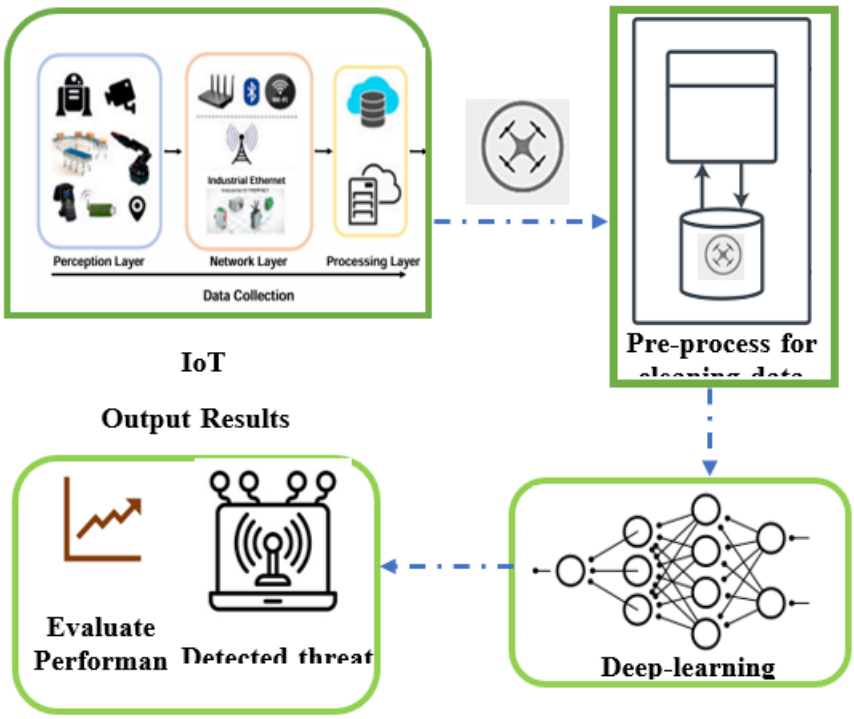


Figure 1. Systematic approach used for detecting threats in IoT networks.

4. Intelligent Migration Avian Optimization enabled Drift-enhanced Modified Activation-based Convolutional Neural Network (IMAO-DrMACN)

IoT networks, which consist of interconnected devices, are transforming industries by facilitating smooth data exchange. However, their swift growth has exposed them to cyber threats such as malware, DDoS attacks, and data breaches. Conventional detection methods frequently struggle because of the ever-changing landscape of IoT. The IMAO-DrMACN framework was designed to address key challenges in IoT threat detection, particularly the issues of data drift and hyperparameter optimization, which hinder the effectiveness of existing models. Unlike traditional methods, which often rely on static or manually tuned models, IMAO-DrMACN combines a modified CNN architecture with a dynamically adaptive IMAO algorithm. This integration enables continuous learning and real-time adaptation to evolving network traffic and emerging threats. By optimizing hyperparameters in real-time, the framework ensures that the model remains resilient to shifts in data patterns and attack characteristics. Traditional models, such as Random Forest and deep learning-based approaches, are limited by their inability to dynamically adjust to these changes, leading to decreased accuracy and increased susceptibility to new attack types. IMAO-DrMACN's enhanced optimization mechanism addresses this gap, enabling automated fine-tuning without manual intervention. The combination of adaptive optimization and a robust CNN structure provides superior detection accuracy and scalability, particularly in dynamic and high-dimensional IoT environments. The process begins by gathering input data for threat detection. This collected data is then passed to the pre-processing phase, where the KNN imputation method is applied to clean the dataset and fill in any missing values. Once pre-processed, the data is fed into the DrMACN classifier, designed to accurately identify intrusions within IoT systems. The classifier combines a Drift network with a MACN to handle vast amounts of unstructured data, while the modified activation efficiently diminishes the gradients by upgrading the learning capability. Figure 2 represents the block diagram for the proposed IMAO-DrMACN framework.

4.1 Accumulate IoT data

The DAPT2020[27] and the BOT-IoT dataset [28] offer a rich collection of input data gathered from various IoT ecosystems, encompassing network traffic details, device-specific metadata, and activity logs. This diverse information allows for the precise detection of irregularities that could signal security risks. To utilize the data in the proposed model for threat detection, it can be extracted following the order that is formulated in equations 1 and 2.

4.2 Pre-processing IoT traffic data

Pre-processing is a crucial step in preparing data for the models, particularly when addressing missing values and adjusting the parameters to achieve a consistent or normal distribution. These tasks can be effectively handled using the two methods discussed below.

4.2.1 Handling Missing Data using KNN Imputation

KNN imputation[29] is a widely used method for addressing missing data by estimating the missing values based on the nearest neighbors in the feature space. This approach assumes that similar data points, like neighbors, tend to have similar values, making it effective for datasets with both continuous and categorical features. In the realm of IoT threat detection, datasets may have missing values due to issues like sensor malfunctions, transmission errors, or incomplete logs. KNN imputation effectively fills these gaps by utilizing the relationships between data points, ensuring the dataset is complete and ready for analysis. To find the nearest neighbors, a distance metric $f\left(r_s, r_n\right)$ called the Euclidean distance is calculated between the sample r_s and its nearest sample r_n as

$$f\left(r_s, r_n\right) = \sqrt{\sum_{d=1}^{m}\left(r_{sd} - r_{nd}\right)^2} \tag{6}$$

where n denotes the index of the neighbor ranging from 1 to N, r_{sd} and denotes the value of d^{th} the feature for the n^{th} neighbor. This action endorses that missing values are ignored in the distance calculation.

For a sample r_s with a missing value in a feature d, the n^{th} nearest neighbors are identified based on the distance metric. The missing value r_{sd} is then imputed as the weighted average for the continuous features or the mode for categorical features of the corresponding feature values from the n neighbors.

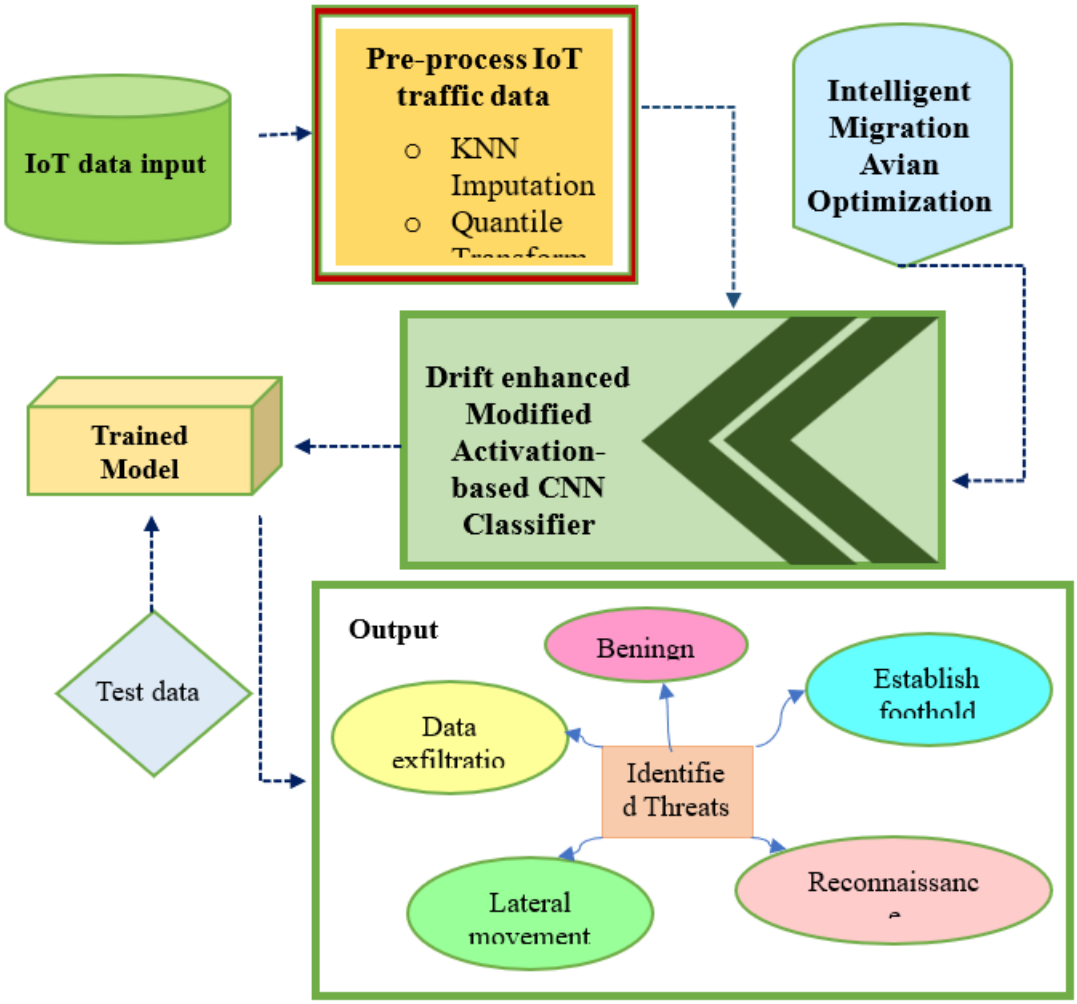


Figure 2. Pictorial representation of the IMAO-DrMACN framework.

$$r_{sd}^{*} = \frac{\sum_{n=1}^{N} M_n \cdot r_{nd}}{\sum_{n=1}^{N} M_n} \tag{7}$$

where r_{sd}^{*} denotes the imputed output after the utilization of k-nearest neighbors. M_n is the weight assigned to n^{th} the neighbor, which can typically be the inverse of distance and can be represented as,

$$M_n = \frac{1}{f\left(r_s, r_n\right)} \tag{8}$$

where $f\left(r_s, r_n\right)$ implies the Euclidean distance on every parameter n.

4.2.2 Scaling of features using quantile transformation

In threat detection, IoT datasets frequently include features with skewed distributions, such as packet size, bandwidth usage, and connection duration. These non-Gaussian distributions can negatively impact model

performance. One effective solution is Quantile transformation [30], a non-linear technique that maps the distribution of features to a uniform or normal distribution. This transformation minimizes the influence of outliers, making the data more appropriate for subsequent tasks. It operates by transforming feature values according to their empirical cumulative distribution function (ECDF). For the given feature d , the ECDF $E_d(r)$ is determined and can be mathematically represented as follows.

$$E_d(r) = \frac{1}{t}\sum_{s=1}^{t} I(r_{sd} \le r) \tag{9}$$

Here, the I indicator function that equals 1 if $(r_{sd} \le r)$, otherwise 0, r_{sd} gives the proportion of samples in the dataset with feature values less than or equal to r . The quantile transform maps the original feature values to a target distribution, either a uniform or normal distribution, using the inverse of the target distribution's CDF. The transformed r_{sd}^{*} feature of the sample r_s is computed as,

$$r_{sd}^{*} = Z^{-1}(E_d(r)) \tag{10}$$

Where $E_d(r)$ is the ECDF value r_{sd} for the feature d , and Z^{-1} denotes the inverse quantile function of the target distribution.

4.3 Drift-enhanced Modified Activation-based CNN Classifier

The IMAO-DrMACN framework was developed to provide an effective solution for IoT threat detection by incorporating a dynamic approach to hyperparameter optimization and a modified CNN architecture. Traditional models, such as Random Forest [4] and deep learning methods \[18], often struggle to adapt to evolving network conditions, leading to reduced performance when facing new attack scenarios. This is primarily due to their inability to continuously update their parameters in response to data drift. In contrast, the IMAO-DrMACN framework integrates the IMAO mechanism to dynamically tune hyperparameters, allowing the model to adapt to shifting traffic patterns and identify emerging threats. Furthermore, the modified CNN architecture enables the system to capture complex patterns and interactions in high-dimensional IoT traffic, which previous models like R-Net \[19] and GSGW-DT \[8] fail to handle effectively. By combining these techniques, the model ensures improved accuracy, scalability, and real-time detection, addressing the limitations of conventional systems. The motivation behind this approach is to create an adaptive, scalable, and efficient model that can continuously learn from both past and present data, offering reliable security in dynamic IoT environments. Data drift [31] happens when the statistical characteristics of input data change over time. This can occur for various reasons, including changes in data collection methods, adjustments in preprocessing techniques, or shifts in the original data sources. When data streams are the main source of information, the volume of data can lead to challenges with evolving data patterns. Consequently, in areas like data mining, deep learning, pattern recognition, and information extraction, the issue of concept drift has received considerable attention. This problem is particularly important in dynamic settings such as IoT networks, where data distributions can vary due to changes in user behavior, updates to devices or software, the rise of new threats, or fluctuations in environmental conditions. Without effective strategies to manage drift, a model's predictive accuracy can decline over time, rendering it less effective.

A drift-enabled CNN provides a way to tackle these issues by continuously monitoring network traffic data for changes in distribution and updating the model to maintain high accuracy. Implementing drift in a CNN means allowing the model to adjust to changing data patterns. This is achieved when there exist deviations while comparing the difference between predicted and actual data with the threshold, Δ_m such that the value of Δ_m is set to 0.1. The actual data will be referred to as t_s, while the predicted output obtained from the model is denoted as c_s. When $(t_s - c_s) > \Delta_m$, then the concept of drift is activated by training the model repeatedly with the predicted output. A drift detection mechanism [32] is incorporated into the CNN to monitor shifts in the input data distribution, allowing for adaptation to diverse environments. The main benefit of a drift-enabled CNN is its ability to sustain accuracy and reliability even when the data conditions change. By adjusting dynamically to new patterns, the model can automatically respond to emerging threats, minimizing the need for manual intervention and improving its effectiveness in practical applications.

Moreover, the Dr-MACN architecture features four convolutional layers, each using the modified activation function introduced in Equation (13), along with three pooling layers. The convolutional block is a vital part of the IMAO-DrMACN framework, as it helps extract feature maps and reveal hidden patterns in the input data. This design effectively manages a diverse range of inputs due to the reduced number of weights in each layer. The operation can be represented as,

$$Co_s = w_s + r_{sd}^* + b_s \tag{11}$$

where Co_u implies the operation of convolution w_s and b_s indicates the factors of weight and bias. The ReLU activation function is crucial because it allows to learn complex relationships between the input and output data. By using a non-linear activation like ReLU, the model can create curved decision boundaries, which enhances its ability to classify data accurately. This non-linearity enables the network to capture intricate patterns that linear models struggle to represent. However, one drawback of ReLU is its tendency to return an output value of zero for any negative input, which can cause such inputs to become inactive. This issue is referred to as the "dying ReLU problem", which makes a significant part of the network ineffective. To tackle this, alternative activation functions such as ELU, leaky ELU, PReLU, and many more have been introduced. Unlike ReLU, ELU can produce negative outputs, which increases its flexibility. The ELU operation is mathematically defined as follows:

$$e(y) = \begin{cases} y, if\ y \geq 0 \\ \alpha\left(ex^y - 1\right), if\ y < 0 \end{cases} \tag{12}$$

Here, the $e(y)$ denotes the output generated by elu activation α denotes the coefficient factor that handles negative values, and y indicates some input applied to ELU. It produces the same output value if the input is greater than 0; otherwise, it outputs $ex^y - 1$ a fixed positive integer. However, ELU has a limitation: when the output falls within the range $(0, \infty)$, it can cause the activation function to explode for inputs greater than 0. On the other hand, Leaky ReLU considers negative values, although to a smaller degree. It allows backpropagation with negative inputs by introducing a slight positive slope in the negative region. Yet, in certain scenarios, it struggles to adequately handle negative values, leading to suboptimal performance. Similarly, PReLU is an adaptation of Leaky ReLU, where the slope for negative inputs becomes a learnable parameter. To address gradient-related issues and improve the effectiveness of the aforementioned methods, an activation function is modified to enhance performance. This can be expressed mathematically as follows:

$$re\left(r_{sd}^*\right) = \begin{cases} r_{sd}^*, if\ r_{sd}^* \geq 0 \\ a\left(ex^{\alpha r_{sd}^*} - 1\right), if\ r_{sd}^* < 0 \end{cases} \tag{13}$$

In this context, the input from the layer r_{sd}^* is passed to the activation function re. Each layer shares a common slope parameter, denoted as a, along with the coefficient factor α that ranges from α 1. For CNNs, this parameter corresponds to the number of channels. Learning this parameter enhances the model's accuracy without adding significant computational costs. Using the modified version of the ReLU activation does not impose a heavy burden on the neural network's learning process. This is because the additional parameters to be learned are limited to the number of channels, which is approximately tiny compared to the total number of weights. This adaptation of the ReLU activation leads to a notable improvement in model accuracy when compared to alternative activation functions. After the activation, the max pooling layer processes each channel of the feature maps, reducing dimensionality while preserving critical information, which can be derived as,

$$c_s = mx\left(Co_s\right)_{s=1}^t \tag{14}$$

Subsequently, the output produced by the dense layer c_s is trained on extensive labeled attack data. The model's hyperparameters are optimized using the IMAO algorithm, which is implemented before the flattened layer. Additionally, the model's performance is significantly improved by minimizing the error rate, as outlined in equation

(3). The architectural representation of the model is depicted in Figure 3. The stages of the IMAO algorithm, detailed in the next section, enable the IMAO-DrMACN model to iteratively assess all potential solutions, resulting in superior detection accuracy.

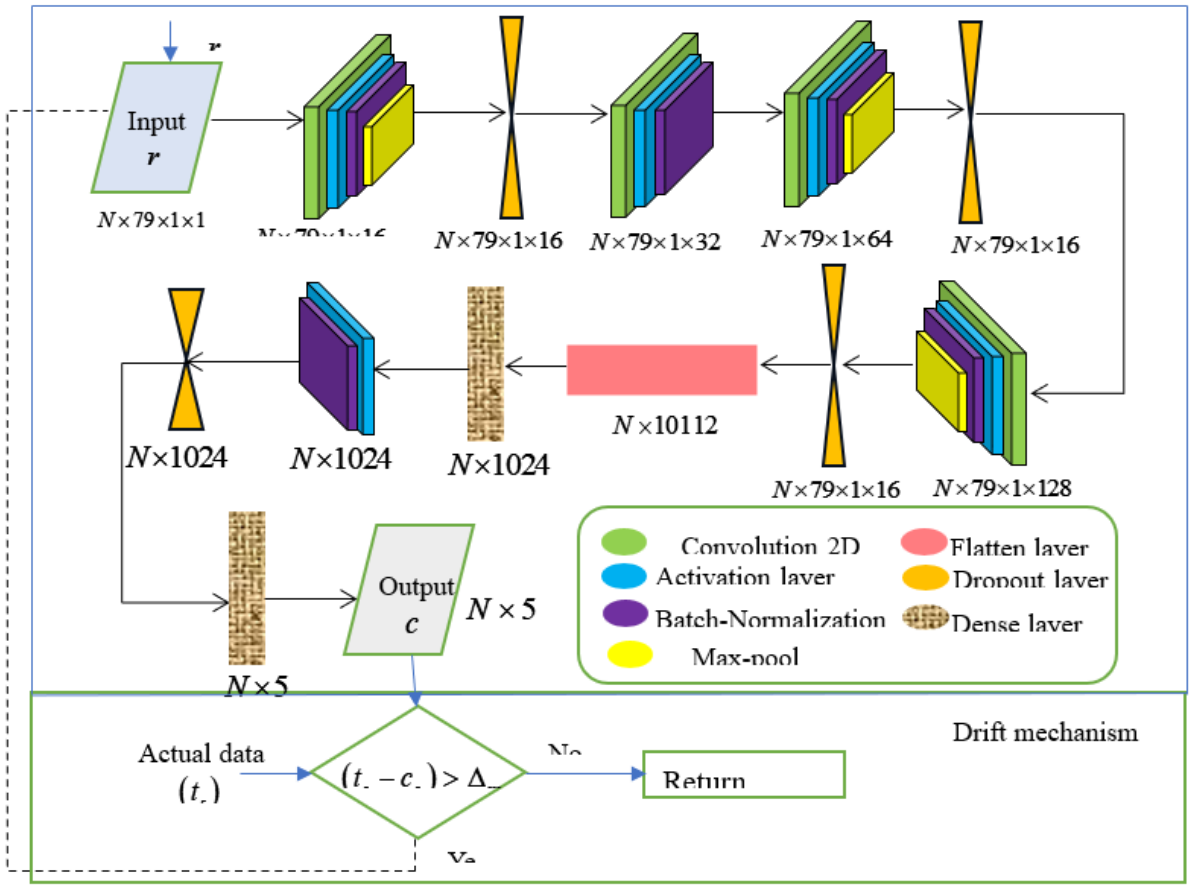


Figure 3. Architectural Structure of the IMAO-DrMACN Framework

4.4 Intelligent Migration Avian Optimization

IMAO is an innovative metaheuristic algorithm that combines the natural behaviours of migrating birds [35] and hummingbirds [34] to tackle complex optimization challenges, especially in threat detection systems. IMAO merges the exploratory strengths of AMO with the precision and agility of HO, resulting in a powerful and efficient optimization framework. Migrating birds are renowned for their ability to cover vast distances and adapt to changing environments, while hummingbirds are known for their remarkable precision and speed in foraging. By integrating these traits, IMAO strikes a balance between global exploration and local exploitation, making it particularly effective for addressing dynamic and high-dimensional problems. Hyperparameter tuning is critical in adapting the model to the dynamic and evolving nature of IoT networks, ensuring high detection accuracy. By modifying hyperparameter tuning dynamically, the proposed IMAO-DrMACN model ensures adaptability to shifting data patterns and changing threat characteristics. This modification allows the model to fine-tune its parameters continuously, ensuring enhanced detection accuracy and real-time responsiveness, crucial for maintaining robust threat detection in diverse and evolving IoT environments

Motivation

The impressive strategies of migratory birds and the agility of hummingbirds. Migratory birds showcase collective intelligence, optimizing their flight paths by forming dynamic formations, conserving energy, and adapting to changing environmental conditions. On the other hand, hummingbirds are known for their remarkable precision and agility, quickly adjusting their positions to efficiently explore areas with abundant resources. IMAO merges these strengths, balancing global exploration through adaptive migration with local exploitation through refined searching

techniques. The dynamic leadership within the flock promotes diverse solution exploration, while the hummingbird-inspired focused search helps avoid premature convergence. This hybrid strategy boosts optimization efficiency and robustness, making IMAO particularly suitable for complex, multi-dimensional challenges. By emulating nature's dual-tiered intelligence, IMAO presents an innovative framework for addressing issues of threats.

a) Initialize the solution

The solution can be effectively initialized by determining the randomized parameters along the iterations and can be represented as,

$$F = \left\{ F_1, F_2, .., F_e, ..., F_f \right\} \tag{15}$$

where F_e denotes the e^{th} solution in the space with f the number of entire solutions $F_e \in \left\{ W_s, b_s \right\}$. The topmost and lowermost limits can be represented F_e as,

$$F_e = F_{ra} + F_{uf} + \left(F_{uf} - F_{lf} \right) * j \tag{16}$$

where F_{ra} indicates the randomized place, F_{lf} and F_{uf} indicates the topmost and lowermost limits, and j implies the random value in the range of $\left[0,1 \right]$.

b) Calculation of fitness

The fitness measure is essential to obtain an effective solution. The higher range of this value indicates a greater choice of attaining better solutions.

$$g\left(F^c \right) = mx\left(acc\left(F^c \right) \right) \tag{17}$$

where g denotes the fitness, $mx\left(acc\left(F^c \right) \right)$ and denotes the highest maximum accuracy attained at the c maximized solutions.

c) Solution update: The solution gets updated based on the probabilistic factor A that is attained from the migrating characteristics of birds. The rule of updation can be illustrated as,

$$A = \left(2 j_1 - 1 \right) \cdot r \tag{18}$$

where j_1 denotes the arbitrary factor and denotes the coefficient vector that linearly varies along the range. The phases based on the factor A are discussed in the subsequent section.

Case (i): Improved Exploration phase $if\ A > 1$, if the specified state gets satisfied, the exploration phase gets initialized.

The solution involves three types of movements to determine the optimal region. Exploring with the three different strategies allows the algorithm to attain the optimal regions and thereby reduce the computational cost of finding the frequent regions in the context of threats. The exploration strategies are expressed based on the factor D_b obtained by,

$$D_b = \sum_{e=1}^{f} \frac{g\left(F_e^c \right)}{f} \tag{19}$$

where D_b the average factor is based on the fitness level of all solutions f.

Subcase (i) $if\ D_b < 0.5$

The three movements that the migrating birds are involved in during the food search can be initiated as Q follows. The conditions of the movements can be mathematically derived as,

$$Q = \{Q_1, Q_2, Q_3\} \tag{20}$$

$$Q_1 = \begin{cases} 1, if\ e = ra\, e[1,u]\ where\, e = \{1,...,u\} \\ 0, else \end{cases}$$

$$Q_2 = \begin{cases} 1, if\ e = B(g); g \in [1,v]\ , where\, B = ra\ per(v), \\ \qquad\qquad\qquad\qquad\qquad\qquad v \in \left[2, \lceil j_1 \cdot (u-2) \rceil + 1\right] \\ 0, else \end{cases} \tag{21}$$

$$Q_3 = 1$$

Here, Q_1, Q_2, Q_3 denotes the three types of movement, $ra\, e[1,u]$ denotes the random integer that is generated in the range of $[1,u]$, u denotes the dimension variable, B denotes the factor that contains a random permutation of the integers along $[1,u]$, v generates the values in the specified criteria $2, \lceil j_1 \cdot (u-2) \rceil + 1$, and j_1 indicates the arbitrary factor along $[0,1]$.

The movements Q_1, Q_2, Q_3 carried out by the hummingbirds determine the diverse types of searching for the optimal solutions. According to these conditions, the specific movement here, the solution is obtained by searching along the space and is determined as follows,

$$Q = \begin{cases} Q_1, if\ D_b \le 1/3 \\ Q_3, if\ D_b \le 2/3\ \&\ D_b > 1/3 \\ Q_2, otherwise \end{cases} \tag{22}$$

where Q_1, Q_2, Q_3 denotes the axial, diagonal, and omnidirectional movement toward the search space. These movements are carried out by the algorithm to attain optimal locational points in the search space. Thus, the updation rule is determined as,

$$F_e^{c+1} = \frac{1}{2} Q \left[F_{best} \left(1 - \frac{c}{c_{\mathrm{mx}}} \right) + \left(F_{mean} - F_{best} * ra \right) + Levy(F) + ra \right] \tag{23}$$

where F_{best} implies the better solution, F_{mean} implies the average generated with the f solutions, c_{mx} implies the maximized iterations, and $Levy(F)$ denotes the Levy-flight factor inherited by migrating birds. The factor Q that employs different movements allows the algorithm to search all the possible regions. This strategy of movements enhances the chances of variational such that both the least and best regions are obtained within the minimum time constraints.

Subcase (ii) $if\ D_b \ge 0.5$

The value D_b decides to initialize movements accordingly to obtain the best solutions. The following conditions determine the regions that the algorithm explores to attain the precise solution.

$$F_e^{c+1} = \frac{1}{2}\left[F_{best}\left(1 - \frac{c}{c_{\mathrm{mx}}}\right) + \left(F_{mean} - F_{best} * ra\right) + Levy\left(F\right) + ra \right] \tag{24}$$

Case (ii): Avian Migratory phase *if* $A = 1$. If the specified state gets satisfied, the avian migration phase gets initialized.

During this phase, the migration of avian strategy is more likely to decide whether to search for food or be satisfied with the attained food. That capability is utilized to make fine-refining or make global searches towards the entire search space, respectively. The mathematical update rule based on the migrative phase that balances both phases can be determined as,

$$F_e^{c+1} = F_{lf} + j \cdot \left(F_{uf} - F_{lf}\right) \tag{25}$$

This condition describes that the solution focuses more on global search rather than refining the attained results.

Case (iii): Exploitation phase *if* $A < 1$. If the specified state gets satisfied, the phase of the exploit gets initialized.

During this phase, the algorithm attains topmost solutions according to the velocity and quality indices that join with the nearest values in the space, but the level of obtaining local or global solutions is limited. The highest fitting condition for the solution is determined as,

$$F_e^{c+1} = W \cdot F_{best}^{c} - \left(h - F^{c} \times ra_1\right) - \left(k * Levy\left(Q\right) + ra_2 \times h \cdot k\right) + ra_3 * \left(k - F^{c}\right) + T^{c+1} * ra \tag{26}$$

where, h and k denote the global and local solutions ra_2 , and ra_3 are the random positions of the solutions, where $ra_1 = ra_2 = ra_3 \cong \left[-1,1\right]$. F^{c} and F_{best}^{c} imply the current solution and best solution attained so far, and W indicates the index based on the quality factor. Thus, the proposed algorithm focuses more on finding the optimal threats or the abnormal changes accordingly by propagating backward with the help of gradients of the loss function and increasing its performance.

d)Termination

After the exploration and exploitation phases, the feasibility of the resulting solution is re-evaluated. If the solution fails to meet the necessary criteria, the iterative process continues. Once the conditions are fulfilled, the final value is returned, and the algorithm completes its execution. The flowchart representation of the IMAO algorithm is discussed in Figure 4.

5. Results and Evaluation

The performance of the threat detection system utilizing the IMAO-DrMACN framework is evaluated and discussed in the subsequent sections.

5.1 Experimental Setup

The experiment is conducted on a Windows 11 platform equipped with 16 GB of RAM, 100 GB of storage, and a CPU operating at 1.7 GHz. The implementation is carried out using Python, ensuring a robust and efficient development environment for testing and analysis. The dataset is split into a training set and a testing set, where 90% of the data is used for training and 10% of the data is used for testing. The model is trained using a batch size of 32 and a learning rate of 0.001 for 100 epochs. A dropout rate of 0.2 is applied to prevent overfitting, while the Adam optimizer is used for efficient weight updates. The loss function employed is categorical cross-entropy, suitable for multi-class classification tasks. The hyperparameter settings used for the proposed model are given in Table 2.

Table 2. Hyperparameter Details

Batch Size	**32**

Learning Rate	0.001
Number of Epochs	100
Dropout Rate	0.2
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Activation Function	Modified ReLU (learnable channel-wise slope, Eq. 13)
Population Size	50

5.2 Dataset Description

The input samples are network traffic records drawn from the DAPT2020 [27] and BoT-IoT [28] datasets. A comprehensive overview of these datasets is provided in the following section. The DAPT2020 dataset [27] is a thorough compilation of network traffic data aimed at detecting APTs. It encompasses a variety of attack scenarios, including malware infections, data exfiltration, and lateral movement, in addition to normal traffic. The BOT-IoT dataset [28] is centered around IoT-based botnet attacks, simulating different malicious activities like DDoS attacks, data theft, and keylogging. It merges legitimate IoT device traffic with attack traffic, providing a well-rounded view of both normal and malicious behaviors.

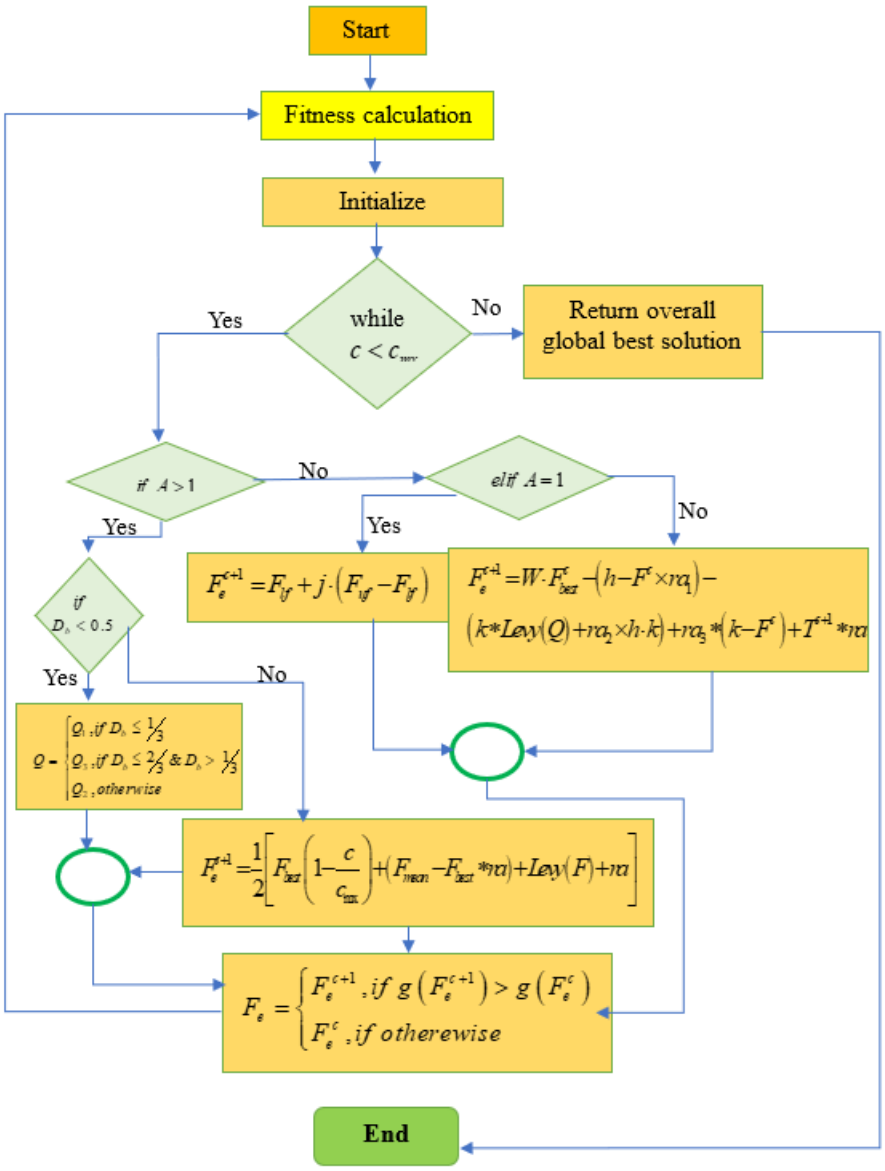
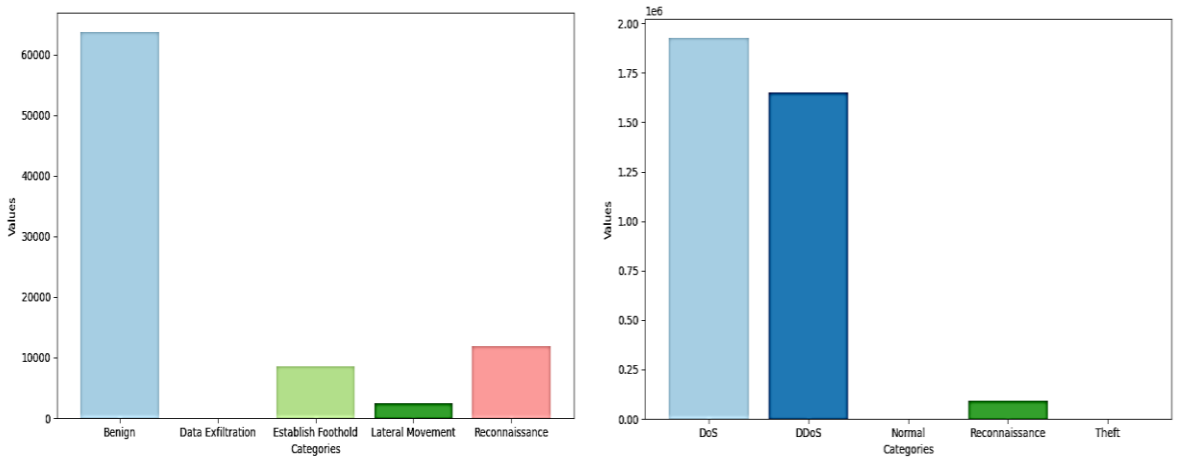


Figure 4. Illustration of IMAO using the Flow-chart representation

5.3 Dataset Visualization

Depending on the classes of prediction on [27] and [28], the levels of detection performance are analyzed. Figure 5 shows the visualisation of the classes of the IMAO-DrMACN framework.



DAPT 2020 Dataset **BoT-IoT Dataset**

Figure 5. Dataset Visualization of the IMAO-DrMACN Framework.

5.4 Evaluation Metrics

The performance assessment of the IMAO-DrMACN model utilized several evaluation metrics, including Accuracy, Recall, Precision, and F1-Score. The mathematical formulas for these metrics are detailed in Table 3.

Table 3. Evaluation Metrics

Metrics	Expression
Accuracy	$\frac{U_\lambda + U_\mu}{U_\lambda + U_\mu + V_\lambda + V_\mu}$
Recall	$\frac{U_\lambda}{U_\lambda + V_\mu}$
Precision	$\frac{U_\lambda}{U_\lambda + V_\lambda}$
F1-score	$\frac{2 \cdot precision - recall}{precision + recall}$

Here, U_λ and U_μ represent true positive and true negative values, while V_λ and V_μ correspond to false positive and false negative values, respectively. These metrics collectively provide a comprehensive measure of the model's effectiveness in threat detection.

5.5 Performance Evaluation

The performance analysis of the IMAO-DrMACN model for threat detection was conducted using the [27] and [28] datasets, assessed through metrics such as Accuracy, Recall, Precision, and F1-Score. During evaluation, the maximum training percentage was set to 90%, with epochs ranging from 20 to 100. For the DAPT2020 dataset at epoch 100, the model achieved an accuracy of 97.531%, a recall of 97.476%, a precision of 97.687%, and an F1-score of 97.417%. The results demonstrate a consistent improvement in performance as the number of epochs increases, with 90% of the data used for training and the remaining 10% for testing. Figure 6 demonstrates the visual results of performance attained through the DAPT 2020 dataset.

Performance Evaluation using DAPT2020 Dataset

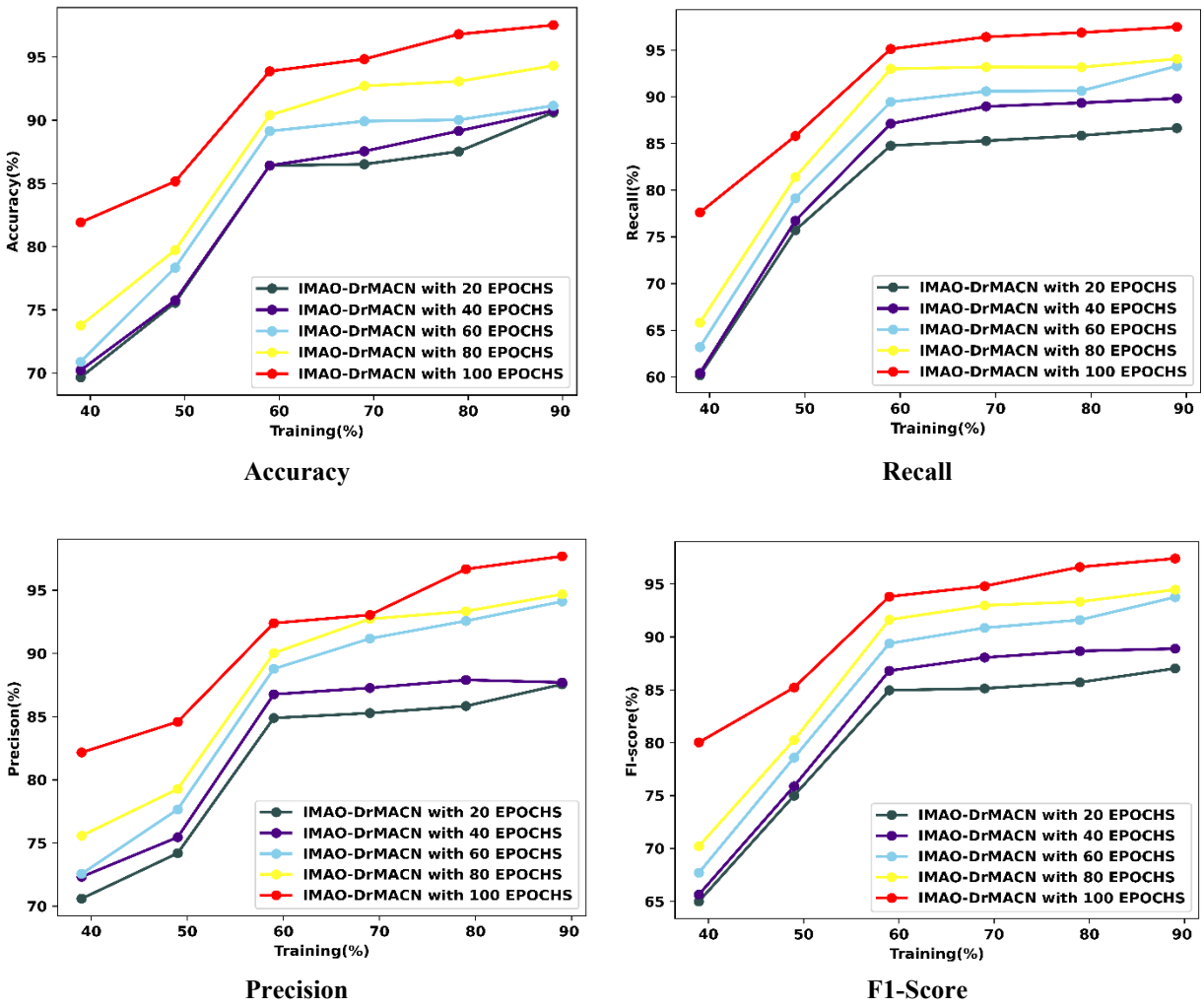


Figure 6. Calculation of performance of the IMAO-DrMACN framework with the DAPT 2020 dataset

Similarly, when evaluated on the BOT-IoT dataset with a maximum training percentage of 90% at epoch 100, the model achieved an accuracy of 96.905%, a recall of 97.763%, a precision of 95.004%, and an F1-score of 96.364%. These findings highlight the model's robust performance across both datasets. The overall effectiveness of the IMAO-DrMACN model is visually summarized in Figure 7, showcasing its capability to deliver high detection accuracy and reliability.

Performance Evaluation using BOT-IoT Dataset

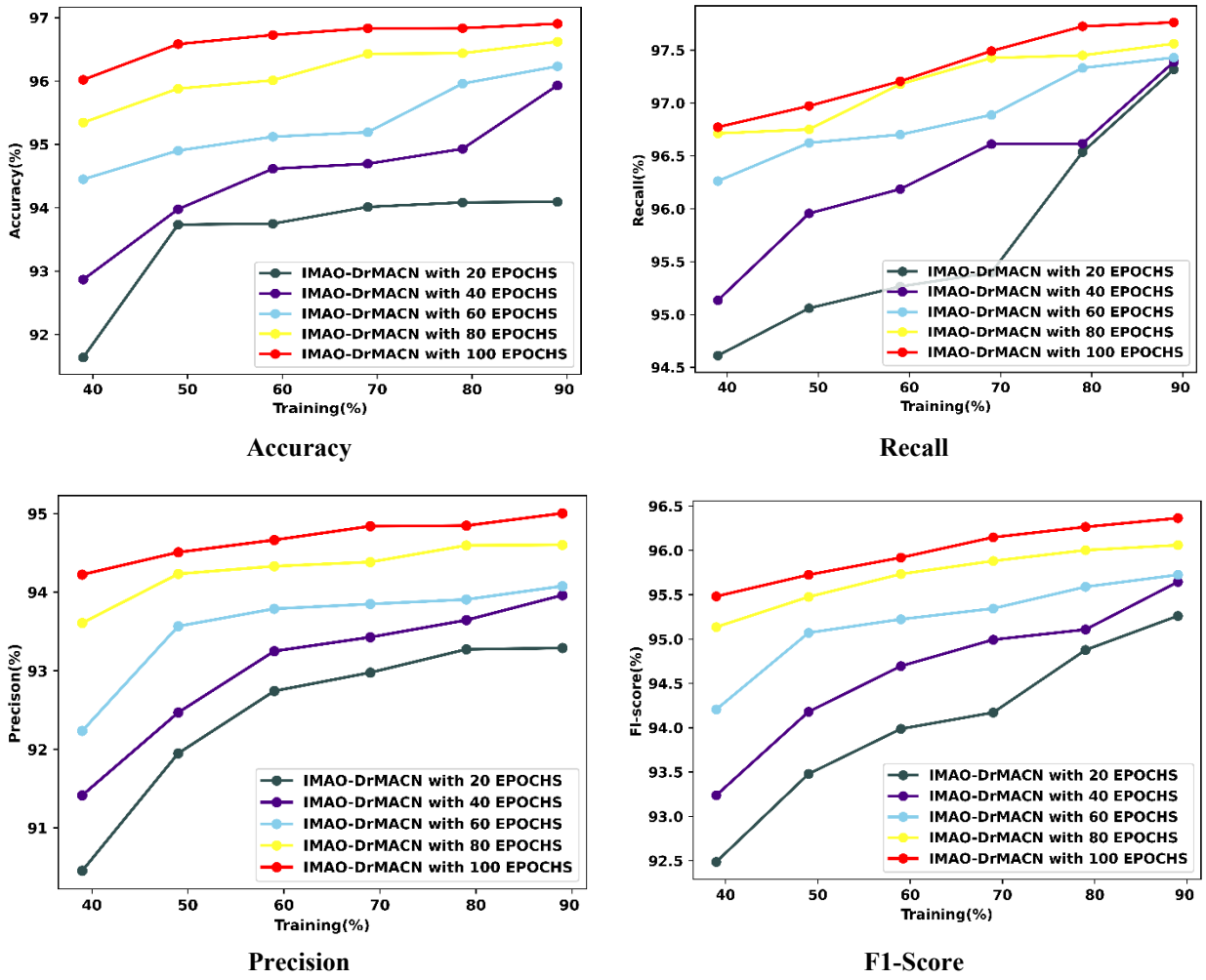


Figure 7. Calculation of performance of the IMAO-DrMACN framework on the BOT-IoT dataset

5.6 Comparative Methods

The effectiveness of the IMAO-DrMACN model is evaluated against several established techniques, including RF [4], LR-MLP [18], GSGW-DT [8], R-Net [19], MB-DrCN [35], PI-DrCN, and AM-DrCN [33]. This comparison is conducted using the DAPT2020 and BOT-IoT datasets, with a focus on achieving a training percentage rate of up to 90%. The detailed results and analysis are presented in the following section.

5.6.1 Comparative evaluation with DAPT2020 dataset

The performance comparison of IMAO-DrMACN using the DAPT2020 dataset is illustrated in Figure 8. During evaluation, the IMAO-DrMACN model demonstrated superior performance, achieving an accuracy of 97.531% at a training rate of 90%. This represents significant improvements over RF, LR-MLP, and R-Net by margins of 6.713%, 8.173%, and 4.590%, respectively. In terms of recall, the IMAO-DrMACN model achieved 97.476%, outperforming MB-DrCN by 3.407%, PI-DrCN by 3.149%, and AM-DrCN by 2.758%. For precision, the IMAO-DrMACN model reached 97.687%, surpassing RF by 10.534%, LR-MLP by 8.717%, GSGW-DT by 9.798%, and R-Net by 4.762%. Lastly, in the F1-Score metric, the IMAO-DrMACN approach achieved 97.417%, exceeding the performance of MB-DrCN, PI-DrCN, AM-DrCN, and Dr-MACN by 3.282%, 3.059%, 2.627%, and 2.843%, respectively. The improved performance of the IMAO-DrMACN model on the DAPT2020 dataset is primarily due to its integration of Intelligent Migration Avian Optimization (IMAO) and a drift-enhanced modified activation-based CNN. The IMAO algorithm dynamically tunes hyperparameters, enabling better generalization and adaptability to changing IoT environments. The drift mechanism ensures consistent performance by addressing concept drift, while the modified activation function enhances the model's ability to learn complex threat patterns. Additionally, advanced

pre-processing techniques like KNN imputation and quantile normalization improve data quality, leading to higher detection accuracy. These innovations collectively result in significant improvements in accuracy, precision, recall, and F1-score compared to traditional models. These results highlight the robustness and effectiveness of the IMAO-DrMACN model in threat detection compared to conventional methods.

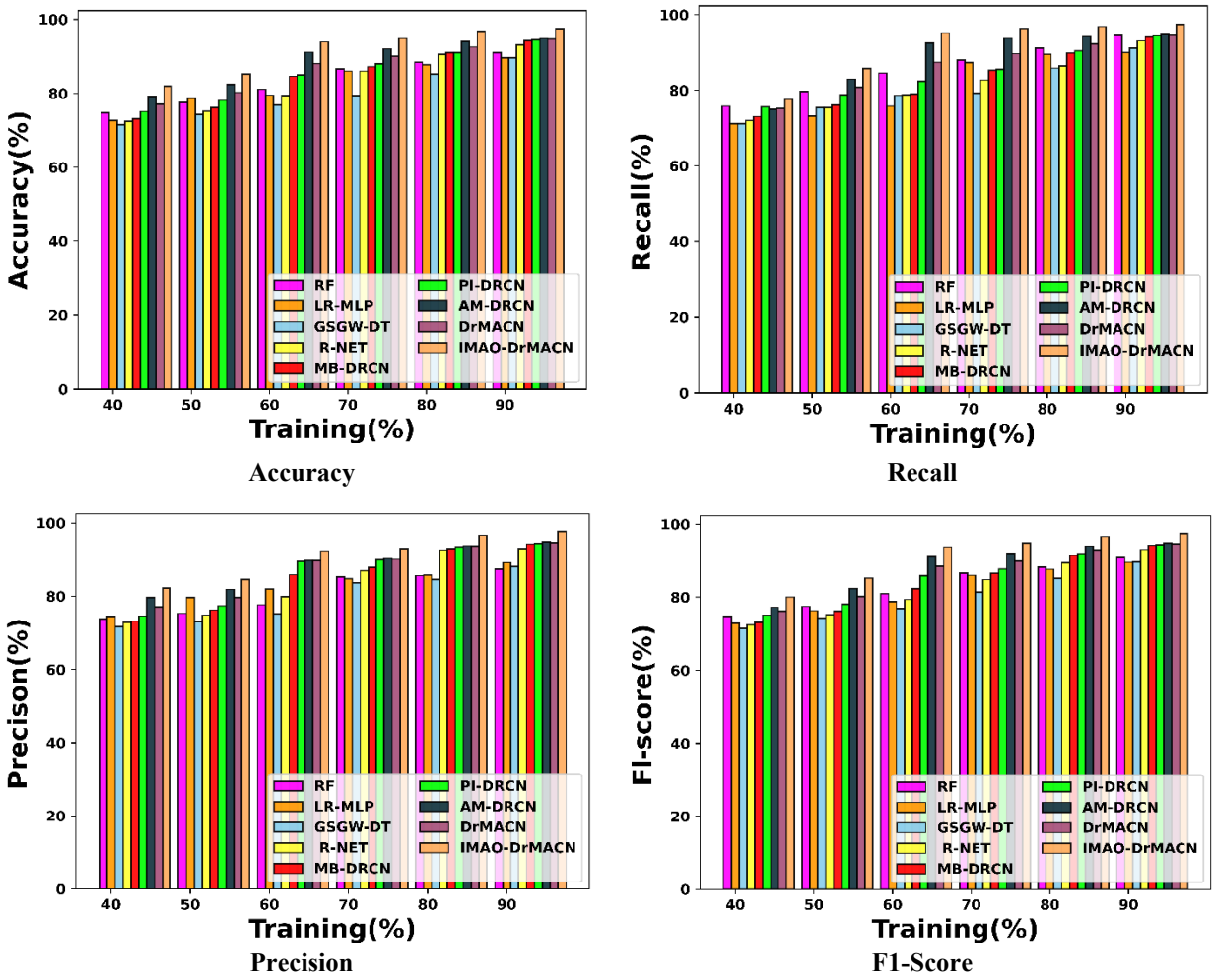


Figure 8. Comparison of IMAO-DrMACN with existing methods on the DAPT2020 dataset

5.6.2 Comparative evaluation with BOT-IoT dataset

The comparative evaluation of IMAO-DrMACN under the BOT-IoT dataset is described using Figure 9. For the accuracy metric, IMAO-DrMACN model gained 96.905%at training percentage 90%, which shows better improvement than RF, LR-MLP, GSGW-DT, R-Net, as 6.789%, 5.375%, 4.957%, 4.762%. Similarly, for the recall metric, the attained value for the IMAO-DrMACN model is 97.763%, which is 3.905% effective than MB-DrCN, 3.324% greater than PI-DrCN, and 1.632% improved than the AM-DrCN model. Furthermore, the achieved value for the IMAO-DrMACN model under the precision metric is 95.004%, which is more effective than other conventional metrics as 6.346%, 5.693%, 4.368%, 4.095%, for RF, LR-MLP, GSGW-DT, and R-Net. At last, for the F1-Score, the IMAO-DrCN achieved 96.364% for 90% training, which is higher than MB-DrCN by 3.842%, PI-DrCN by 2.789%, AM-DrCN by 1.899%, and Dr-MACN by a margin of 2.605% respectively.

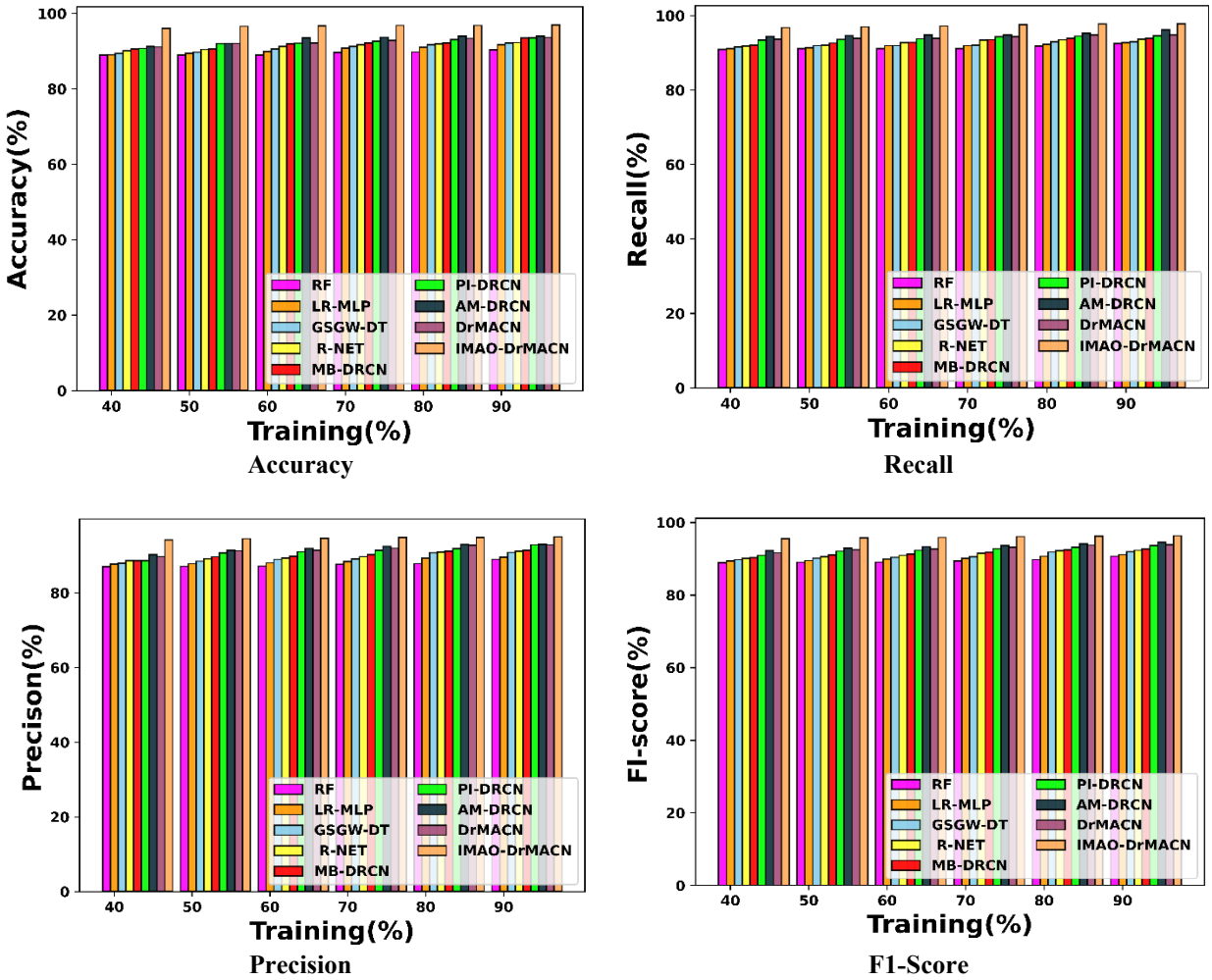


Figure 9. Comparison of IMAO-DrMACN with existing methods on the BOT-IoT dataset

5.7 Statistical Analysis

The statistical analysis for the IMAO-DrMACN approach focuses on assessing its performance through best, mean, and variance using both datasets are displayed in Tables 4 and 5. The best value signifies the highest performance achieved, the mean represents the average result across various trials, and the variance indicates the approach's consistency. This analysis provides a thorough understanding of the model's effectiveness and reliability in identifying anomalies and threats in different IoT environments.

Table 4. Statistical results obtained on the DAPT2020 Dataset for IMAO-DrMACN

Methods vs Attained Results		**DAPT2020 Dataset**								
		Training Percent (90%)								
		RF	LR-MLP	GSGW-DT	R-Net	MB-DrCN	PI-DrCN	AM-DrCN	Dr-MACN	**IMAO-DrMACN**
Best	*Accuracy (%)*	90.983	89.559	89.663	93.054	94.220	94.437	94.858	94.647	**97.531**
	Precision (%)	87.397	89.172	88.116	93.035	94.285	94.467	94.928	94.698	**97.687**
	Recall (%)	94.570	89.946	91.210	93.073	94.155	94.407	94.788	94.597	**97.476**

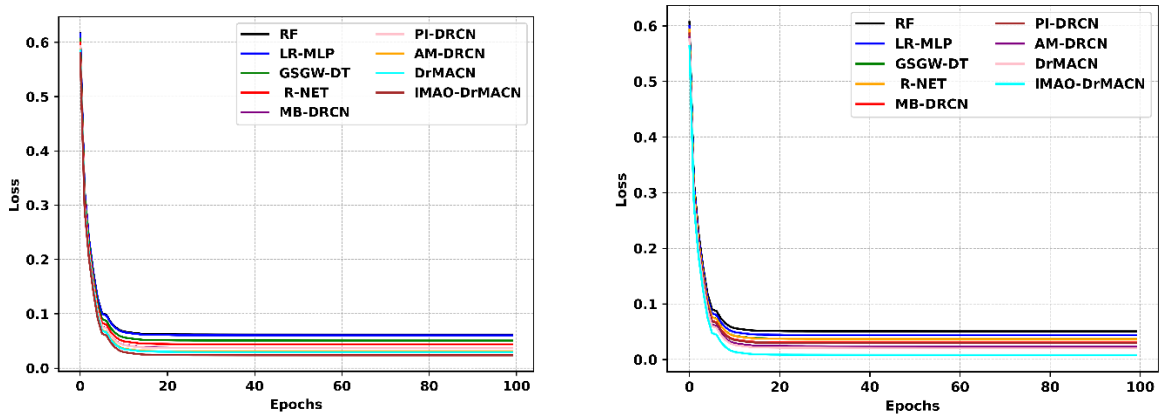
	F1-Score (%)	90.842	89.557	89.636	93.054	94.220	94.437	94.858	94.647	**97.417**
Mean	*Accuracy (%)*	83.232	82.344	79.482	82.760	84.379	85.268	88.932	87.100	**91.688**
	Precision (%)	80.847	82.624	79.374	83.394	85.069	86.576	88.380	87.478	**91.091**
	Recall (%)	85.616	81.156	80.266	81.405	82.899	84.527	88.852	86.690	**91.541**
	F1-Score (%)	83.153	81.832	79.795	82.371	83.952	85.512	88.596	87.054	**91.314**
Variance	*Accuracy (%)*	34.362	34.856	39.113	58.831	57.154	46.664	35.487	40.334	**35.512**
	Precision (%)	29.328	22.197	39.650	64.433	62.556	59.841	33.048	45.368	**33.700**
	Recall (%)	41.362	62.645	43.352	49.008	56.667	41.802	54.563	44.552	**54.522**
	F1-Score (%)	34.092	38.729	39.606	54.938	58.130	48.338	42.348	44.644	**41.403**

Table 5. Statistical results obtained on the BOT-IoT Dataset for IMAO-DrMACN

Methods vs Attained Results		**BOT-IoT Dataset**								
		Training Percent (90%)								
		RF	LR-MLP	GSGW-DT	R-Net	MB-DrCN	PI-DrCN	AM-DrCN	Dr-MACN	**IMAO-DrMACN**
Best	*Accuracy (%)*	90.326	91.696	92.101	92.290	93.454	93.516	93.951	93.638	**96.905**
	Precision (%)	88.353	88.488	89.650	89.981	90.131	91.065	92.259	91.790	**95.720**
	Recall (%)	89.847	90.438	90.590	91.580	92.241	92.585	93.407	93.251	**97.722**
	F1-Score (%)	89.094	89.453	90.118	90.774	91.174	91.819	92.830	92.515	**96.710**
Mean	*Accuracy (%)*	89.443	90.329	90.788	91.267	91.808	92.308	93.062	92.534	**96.651**
	Precision (%)	83.213	84.127	85.356	86.222	86.854	87.390	88.732	87.903	**93.229**
	Recall (%)	86.716	87.323	87.762	88.235	89.058	89.704	90.623	90.239	**94.881**
	F1-Score (%)	84.928	85.694	86.541	87.217	87.942	88.531	89.667	89.055	**94.048**
Variance	*Accuracy (%)*	0.268	0.858	1.014	0.640	1.003	0.792	1.076	0.742	**0.090**
	Precision (%)	7.113	7.600	7.990	7.198	5.936	6.836	5.877	6.555	**3.032**
	Recall (%)	4.215	4.410	4.458	4.737	4.718	3.646	3.929	3.889	**4.136**
	F1-Score (%)	5.495	5.892	6.053	5.922	5.277	5.148	4.856	5.165	**3.552**

5.8 Convergence Analysis

The convergence analysis of the IMAO-DrMACN shows a steady decrease in loss value as the number of epochs rises from 20 to 100. Figure 10. presents the graphical representation of the convergence analysis for both datasets, where the graph illustrates a smooth and consistent decline in the loss curve, reflecting effective learning and optimization during the training process.



DAPT2020 Dataset **BOT-IoT Dataset**

Figure 10. Convergence of the IMAO-DrMACN training loss on the DAPT2020 and BoT-IoT datasets

5.9 Confusion Matrix

Figures 11 and 12 illustrate the confusion matrix of the IMAO-DrMACN with two different datasets. The visual representation highlights the labels of prediction with the truth labels to showcase the effectiveness. The class labels are defined in Equations (4) and (5).

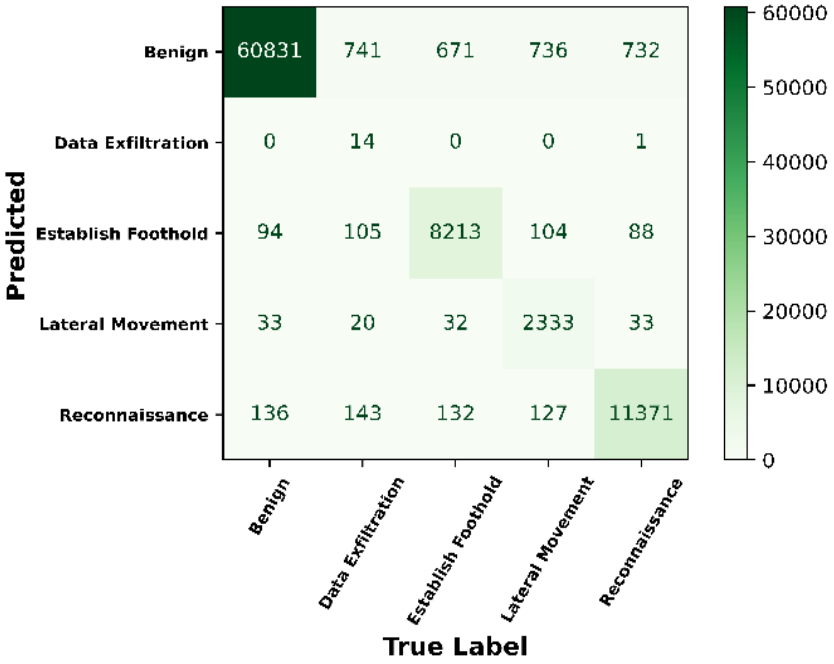


Figure 11. Confusion matrix of the DAPT 2020 dataset

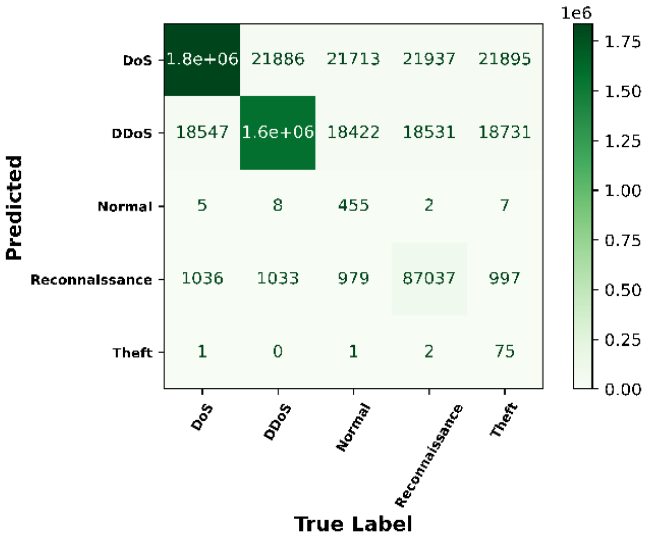


Figure 12. Confusion matrix of BoT-IoT dataset.

5.10 Comparative Discussion

Threat detection in IoT networks has been approached in various ways, each with its strengths and weaknesses. The performance of the threat detection system utilizing the IMAO-DrMACN framework with the DAPT2020 and BOT-IoT dataset is evaluated in terms of accuracy, F1 score, Precision, and recall. The results were compared with existing models, including RF [4], LR-MLP [18], GSGW-DT [8], R-Net [19], MB-DrCN [35], PI-DrCN, and AM-DrCN [33] under the same conditions. Traditional machine learning techniques like RF [4] are known for their robustness and ability to handle high-dimensional data, but they struggle with imbalanced datasets and lack interpretability. In contrast, LR-MLP [18] is straightforward but often fails to capture the complex, non-linear patterns present in IoT traffic, which limits its accuracy. GSGW-DT [8] improves feature selection but tends to be computationally intensive and can overfit the data. R-Net[19] is effective for analyzing sequential data but requires large datasets and substantial computational resources, making it less suitable for real-time IoT applications. More advanced methods like MB-DrCN[35] and Pl-DrCN tackle non-linear patterns and concept drift, but are hindered by high computational demands and slow training times. AM-DrCN [33] offers some optimization but still struggles with efficiently managing large-scale, diverse IoT data. Our proposed IMAO-DrMACN addresses these challenges. By integrating a modified CNN architecture with IMAO, it dynamically adjusts hyperparameters and adapts to data drift, ensuring strong performance and achieving an accuracy of 97.53% in the DAPT-2020 dataset, 96.90% accuracy in the BOT-IoT dataset, showing a significant improvement in model performance. The integration of the IMAO algorithm with a drift-aware CNN-based architecture allowed the model to capture underlying patterns more effectively than the existing models, thus enhancing the accuracy in threat detection. Furthermore, the drift-enhanced CNN effectively captures complex patterns in high-dimensional data, while the modified activation function improves feature learning. On the two datasets considered, the approach attains higher accuracy, precision, recall and F1-score than the eight comparison methods. Claims regarding scalability and computational efficiency are not evaluated in this study and are identified as future work in Section 6. Table 6 depicts the discussion table with values attained through evaluation on both [27] and [28] datasets.

Table 6. Comparison table of the IMAO-DrMACN framework

Methods	**Training Percent (90%)**							
	DAPT2020 Dataset				**BOT-IoT Dataset**			
	Accuracy (%)	*Recall (%)*	*Precision (%)*	*F1-Score (%)*	*Accuracy (%)*	*Recall (%)*	*Precision (%)*	*F1-Score (%)*
RF	90.983	94.570	87.397	90.842	90.326	92.558	88.975	90.731
LR-MLP	89.559	89.946	89.172	89.557	91.696	92.773	89.596	91.157
GSGW-DT	89.663	91.210	88.116	89.636	92.101	93.036	90.855	91.933

R-Net	93.054	93.073	93.035	93.054	92.290	93.616	91.114	92.348
MB-DrCN	94.220	94.155	94.285	94.220	93.454	93.945	91.412	92.661
PI-DrCN	94.437	94.407	94.467	94.437	93.516	94.513	92.854	93.676
AM-DrCN	94.858	94.788	94.928	94.858	93.951	96.167	92.954	94.533
Dr-MACN	94.647	94.597	94.698	94.647	93.638	94.868	92.859	93.853
IMAO-DrMACN	**97.531**	**97.476**	**97.687**	**97.417**	**96.905**	**97.763**	**95.004**	**96.364**

6. Conclusion

The IMAO-DrMACN model marks a notable step forward in the realm of IoT threat detection. By utilizing IMAO, this model can dynamically adjust its hyperparameters, allowing it to adapt to changing data patterns effectively. The drift-enhanced CNN architecture is particularly adaptable for managing data drift, which is a frequent issue in IoT settings, while the modified activation function improves feature learning and representation. This combined approach results in achieving higher performance, reducing error rates, and enhancing detection capabilities. The IMAO mechanism effectively enhances detection by tuning hyperparameters adaptively, while the modified DrMACN architecture addresses concept drift and improves the generalization on dynamic traffic patterns. Together, the IMAO mechanism and the DrMACN architecture provide an accurate and stable detector for the IoT threat-detection tasks evaluated here. The model was evaluated on the DAPT2020 and BoT-IoT datasets, on which it attains higher detection accuracy than the eight comparison methods considered. The results obtained on the two datasets are summarised as follows. For the DAPT2020 dataset, it achieved an accuracy of 97.531%, a recall of 97.476%, a precision of 97.687%, and an F1-score of 97.417%. In the BOT-IoT dataset, it recorded an accuracy of 96.905%, a recall of 97.763%, a precision of 95.004%, and an F1-score of 96.364%. This study has several limitations, which also define its future scope. The evaluation uses a single 90%/10% split of two static datasets, and results are reported without repeated runs or statistical significance testing. No explicit data augmentation or class-balancing technique was applied, which may affect generalisation to under-represented threat classes, particularly on the highly imbalanced BoT-IoT dataset. False-positive rate, per-class performance, inference latency, model size and the computational cost of the IMAO search were not measured, and the drift mechanism was evaluated on static partitions rather than on a time-ordered, non-stationary stream. Accordingly, future work will (i) adopt a train/validation/test protocol with repeated runs and significance testing, (ii) report false-positive rate, macro-averaged F1 and computational cost, (iii) evaluate the drift mechanism on streaming data against established drift detectors, (iv) test generalisation to unseen attack families and to further datasets such as TON_IoT and CICIoT2023, and (v) integrate augmentation and balancing strategies while optimising energy efficiency for resource-constrained deployments.

Declaration Statements:

Funding: This research did not receive any specific funding

Conflict of Interest: The authors declare no conflict of interest

Acknowledgements: I would like to express my very great appreciation to the co-authors of this manuscript for their valuable and constructive suggestions during the planning and development of this research work.

Informed consent: Not Applicable

Ethical approval: Not Applicable

Author Contribution: Kapil Dnyaneshwar Dere conceived the presented idea and designed the analysis. Also, he carried out the experiment and wrote the manuscript with support from Dr. Pushpalata Aher. All authors discussed the results and contributed to the final manuscript. All authors read and approved the final manuscript.

Data Availability Statement:

The data underlying this article are available in the BOT IOT dataset: “https://research.unsw.edu.au/projects/bot-IoT-dataset”, DAPT2020 dataset: “https://www.kaggle.com/datasets/sowmyamyneni/dapt2020”, accessed in February 2025.

References

1. Ferrag, M. A., Ndhlovu, M., Tihanyi, N., Cordeiro, L. C., Debbah, M., Lestable, T., Thandi, N. S.: Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices. IEEE Access. 12, 23733-23750 (2024). DOI: https://doi.org/10.48550/arXiv.2306.14263

2. Derhab, A., Guerroumi, M., Gumaei, A., Maglaras, L., Ferrag, M. A., Mukherjee, M., Khan, F. A.: Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security Sensor. 19(14), 3119 (2019). https://doi.org/10.3390/s19143119
3. Santhadevi, D., Janet, B.: Stacked deep learning framework for edge-based intelligent threat detection in IoT network J Supercomput. 79(11), 12622-12655 (2023). DOI: https://doi.org/10.1007/s11227-023-05153-y
4. Cam, N. T., Trung, N. G.: An intelligent approach to improving the performance of threat detection in iot. IEEE Access. 11, 44319-44334 (2023). DOI: 10.1109/ACCESS.2023.3273160
5. Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., Djukic, P.: Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. ACM Comput Surv. 55(5), 1-37 (2022). DOI: https://doi.org/10.1145/3530812
6. Hasan, T., Hossain, A., Ansari, M. Q., Syed, T. H.: Enhanced Intrusion Detection in IIoT Networks: A Lightweight Approach with Autoencoder-Based Feature Learning. arXiv preprint arXiv. 2501.15266 (2025). DOI: https://doi.org/10.48550/arXiv.2501.15266
7. Otoum, Y., Asad, A., Nayak, A.: Llm-based threat detection and prevention framework for iot ecosystems. arXiv preprint arXiv. 2505.00240 (2025).
8. DOI: https://doi.org/10.48550/arXiv.2505.00240
9. Javed, S. H., Ahmad, M. B., Asif, M., Almotiri, S. H., Masood, K., Ghamdi, M. A. A.: An intelligent system to detect advanced persistent threats in industrial internet of things (I-IoT). Electron. 11(5), 742 (2022). https://doi.org/10.3390/electronics11050742
10. Yao, C., Yang, Y., Yin, K., Yang, J.: Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network. IEEE Access. 10, 103136-103149 (2022). DOI: 10.1109/ACCESS.2022.3210189
11. Alkhonaini, M. A., Al Mazroa, A., Aljebreen, M., Hassine, S. B. H., Allafi, R., Dutta, A. K., Khamparia, A.: Hybrid Sine-Cosine Chimp optimization based feature selection with deep learning model for threat detection in IoT sensor networks. Alexandria Eng J. 102, 169-178 (2024). https://doi.org/10.1016/j.aej.2024.05.051
12. Alohali, M. A., Al-Wesabi, F. N., Hilal, A. M., Goel, S., Gupta, D., Khanna, A.: Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment. Cognit Neurodyn. 16(5), 1045-1057 (2022). DOI: 10.1007/s11571-022-09780-8
13. Alrayes, F. S., Alotaibi, N., Alzahrani, J. S., Alazwari, S., Alhogail, A., Al-Sharafi, A. M., Hamza, M. A.: Enhanced Gorilla Troops Optimizer with Deep Learning Enabled Cybersecurity Threat Detection. Comput Syst Sci Eng. 45(3), 3037-3052 (2023). DOI:10.32604/csse.2023.033970
14. Alrowais, F. M., Althahabi, S., Alotaibi, S. S., Mohamed, A., Hamza, M. A., Marzouk, R.: Automated Machine Learning Enabled Cybersecurity Threat Detection in Internet of Things Environment. Comput Syst Sci Eng 45(1), 687-700 (2023). DOI:10.32604/csse.2023.030188
15. Torres, N., Pinto, P., Lopes, S. I.: Security vulnerabilities in LPWANs—an attack vector analysis for the IoT ecosystem. Appl Sci. 11(7), 3176 (2021). https://doi.org/10.3390/app11073176
16. Mishra, S., Albarakati, A., Sharma, S. K.: Cyber threat intelligence for IoT using machine learning. Processes. 10(12), 2673 (2022). https://doi.org/10.3390/pr10122673
17. Li, H., Ota, K., Dong, M.: Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Netw. 32(1), 96-101 (2018). DOI: 10.1109/MNET.2018.1700202
18. Zhao, Y., Cheng, G., Duan, Y., Gu, Z., Zhou, Y., Tang, L.: Secure IoT edge: Threat situation awareness based on network traffic. Comput Netw. 201, 108525 (2021). https://doi.org/10.1016/j.comnet.2021.108525
19. Dey, A. K., Gupta, G. P., Sahu, S. P.: A metaheuristic-based ensemble feature selection framework for cyber threat detection in IoT-enabled networks. Decis Anal J. 7, 100206 (2023). https://doi.org/10.1016/j.dajour.2023.100206
20. Chen, Z., Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H. T., Djukic, P.: Machine learning-enabled iot security: Open issues and challenges under advanced persistent threats. ACM Comp Surv. 55(5), 1-37 (2022). https://doi.org/10.1145/3530812
21. Alblehai, F.: Artificial intelligence-driven cybersecurity system for internet of things using self-attention deep learning and metaheuristic algorithms. Sci Rep. 15(1), 13215 (2025). DOI: https://doi.org/10.1038/s41598-025-98056-2
22. Khan, M. Z., Reshi, A. A., Shafi, S., Aljubayri, I.: An adaptive hybrid framework for IIoT intrusion detection using neural networks and feature optimization using genetic algorithms. Discover Sustainability, 6(1), 382 (2025). DOI: https://doi.org/10.1007/s43621-025-01141-9
23. Poonguzhali, C., BR, T. B., Kalaivani, K., Kaviya, M., Madhubala, S., Chrysolite, S.: A Comprehensive Survey on Intrusion Detection Systems in IoT-Enabled Wireless Sensor Networks: Techniques, Challenges and future directions. In 2024 International Conference on Smart Technologies for Sustainable Dev Goals. 1-7 (2024). DOI:10.1109/ICSTSDG61998.2024.11026314
24. Kanna, P. R., Santhi, P.: Hybrid intrusion detection using mapreduce based black widow optimized convolutional long short-term memory neural networks. Expert Syst Appl. 194, 116545 (2022). https://doi.org/10.1016/j.eswa.2022.116545
25. Kanna, P. R., Santhi, P.: Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features. Knowledge Based Syst. 226, 107132 (2021). https://doi.org/10.1016/j.knosys.2021.107132
26. Kanna, P.R., Sindhanaiselvan, K., Vijaymeena, M.K.: A defensive mechanism based on PCA to defend denial-of-service attacks. Int J Secur and its Appli. 11(1),71-82 (2017).

27. Rajesh Kanna, P., Santhi, P.: Exploring the landscape of network security: a comparative analysis of attack detection strategies. J Ambient Int Hum Comp. 15(8), 3211-3228 (2024). DOI: https://doi.org/10.1007/s12652-024-04794-y
28. DAPT2020 dataset, “https://www.kaggle.com/datasets/sowmyamyneni/dapt2020”, accessed in February 2025.
29. The BOT-IoT dataset, “https://research.unsw.edu.au/projects/bot-iot-dataset”, accessed in February 2025.
30. Duraz, R. (2024). Trustable machine learning for intrusion detection systems (Doctoral dissertation, Ecole nationale supérieure Mines-Télécom Atlantique).
31. Katipoğlu, O. M., Pekin, M. A., Akil, S.: Indonesian Journal of Earth Sciences. Indones J Earth Sciences, 3(2), 821 (2023).
32. Suárez-Cetrulo, A. L., Quintana, D., Cervantes, A.: A survey on machine learning for recurring concept drifting data streams. Expert Syst Appl. 213, 118934 (2023). https://doi.org/10.1016/j.eswa.2022.118934
33. Ali Abdu, N.A., Basulaim, K.O.: Machine learning in concept drift detection using statistical measures. Int J Comp Appl. 46(5), 281-291 (2024). https://doi.org/10.1080/1206212X.2023.2289706
34. Dere, K.D. and Aher, P.: AM-DRCN: Adaptive Migrating Bird Optimization-based Drift-Enabled Convolutional Neural Network for Threat Detection in Internet of Things. In 2024 Int Conf IoT Based Control Netw Intell Syst. 1085-1091 (2024). DOI: 10.1109/ICICNIS64247.2024.10823167
35. Ramadan, A., Ebeed, M., Kamel, S., Ahmed, E.M., Tostado-Véliz, M.: Optimal allocation of renewable DGs using artificial hummingbird algorithm under uncertainty conditions. Ain Shams Eng J. 14(2), 101872 (2023). https://doi.org/10.1016/j.asej.2022.101872
36. Alp, G., Alkaya, A.F.: Hyperheuristic-based migrating birds optimization algorithm for a fairness-oriented shift scheduling problem. Math Probl Eng 2021(1), 6756588 (2021). https://doi.org/10.1155/2021/6756588