MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

Intelligent Intrusion Detection in Industrial IoT Using Nature-Inspired Optimization and Machine Learning

Swetha A [1], Dr. Ramesh Sekaran [2], Dr.Annamalai S [3]
[1]Research Scholar ,Department of Computer Science and Engineering ,JAIN (Deemed- to-be- University), Bangalore,Karnataka,India
[2]Professor, Department of Computer Science and Engineering, JAIN (Deemed- to-be- University), Bangalore, Karnataka, India.
[3]Associate Professor, Department of Computer Science and Engineering, JAIN (Deemed- to-be- University), Bangalore, Karnataka, India.
Email: [1]swethaashok28@gmail.com , [2]drsramesh2015@gmail.com ,[3]annamalaiphd@gmail.com
Orcid id: [1]https://orcid.org/0009-0001-8175-0844 ,[2]https://orcid.org/0000-0002-6668-2142 , [3]https://orcid.org/ 0000-0001-6336-119

***Abstract*—** The rapid expansion of the Industrial Internet of Things (IIoT) has enabled the integration of smart sensors, industrial controllers, and cyber-physical systems across critical infrastructures such as manufacturing, energy, and automation. However, the heterogeneous and resource-constrained nature of IIoT networks introduces significant security challenges, particularly in detecting sophisticated cyber intrusions in real time. To address these challenges, this paper proposes an optimized intrusion detection framework for IIoT environments using machine learning combined with nature-inspired optimization techniques. The proposed methodology performs data preprocessing, including handling missing values and normalization, followed by feature selection using the Cyber Range Ant Optimization Algorithm to identify the most relevant industrial network attributes. The selected features are then utilized for model training and classification using the Cyber Ant Opto Boost Algorithm, which integrates XGBoost with Ant Colony Optimization for hyperparameter tuning.The model is implemented in a Python environment and evaluated using industrial network traffic datasets containing both normal and malicious IIoT activities. Experimental results demonstrate that the proposed approach improves detection accuracy while maintaining low computational overhead, making it suitable for resource-constrained industrial devices. The framework achieves superior performance compared to traditional machine learning and deep learning models in terms of accuracy, precision, recall, and F1-score.The proposed IIoT intrusion detection system offers a scalable, efficient, and real-time security solution for Industry 4.0 environments. By combining machine learning with nature-inspired optimization, the model enhances threat detection capabilities and ensures reliable protection of critical industrial infrastructures, making it well-suited for deployment in smart factories, industrial automation systems, and cyber-physical networks.

***Index Terms*—Intrusion detection, Industrial Internet of things, Machine Learning, Cyber Range Ant Optimization, Cyber Ant Opto Boost Algorithm, Hyperparameter Tuning**

I. INTRODUCTION

The rapid evolution of Industry 4.0 has significantly accelerated the adoption of the Industrial Internet of Things (IIoT) across critical sectors such as smart manufacturing, energy systems, healthcare, transportation, and industrial automation. IIoT enables seamless connectivity between industrial devices, sensors, actuators, and control systems, allowing real-time monitoring, intelligent decision-making, and automated operations. While these advancements improve operational efficiency and productivity, they also introduce serious cybersecurity risks due to increased connectivity, heterogeneous communication protocols, and resource-constrained industrial devices. As industrial systems become more interconnected, they become vulnerable to various cyber threats such as Distributed Denial of Service (DDoS) attacks, malware injections, unauthorized access, and data manipulation, which may disrupt critical industrial processes.

Intrusion detection in IIoT environments has become a vital requirement to ensure system reliability, operational safety, and data integrity. However, designing an efficient intrusion detection system for IIoT networks is challenging due to high-dimensional industrial data, dynamic traffic patterns, and limited computational capabilities of edge devices. Traditional rule-based security mechanisms are often insufficient to detect evolving cyber threats in industrial networks. Machine learning techniques have emerged as promising solutions due to their ability to learn complex patterns from large-scale data and detect unknown attacks. Nevertheless, the performance of machine learning models largely depends on optimal feature selection, parameter tuning, and computational efficiency, which are critical factors in IIoT environments.

To address these challenges, this paper proposes an optimized intrusion detection framework for IIoT systems using machine learning integrated with nature-inspired optimization algorithms. The proposed method incorporates data preprocessing, feature selection using the Cyber Range Ant Optimization Algorithm, and classification using the Cyber Ant Opto Boost Algorithm with Ant Colony Optimization-based hyperparameter tuning. This hybrid approach enhances detection accuracy while reducing computational overhead, making it suitable for resource-constrained industrial devices. The proposed model is evaluated using industrial network traffic datasets containing both normal and malicious activities.

The main contributions of this work are summarized as follows:

- A hybrid machine learning-based intrusion detection framework tailored for IIoT environments.
- Integration of nature-inspired optimization algorithms for efficient feature selection and hyperparameter tuning.
- Improved detection accuracy with reduced computational complexity suitable for industrial applications.
- A scalable and real-time security solution for Industry 4.0 and cyber-physical industrial systems.

The remainder of the paper is organized as follows. Section 2 reviews related work on intrusion detection in IIoT and industrial environments. Section 3 describes the proposed methodology, including preprocessing, feature selection, and classification. Section 4 presents experimental results and performance evaluation. Finally, Section 5 concludes the paper and outlines future research directions.

II. Related Works

The Industrial Internet of Things (IIoT) has introduced new security challenges due to the integration of cyber-physical systems, industrial controllers, and smart sensors. Several researchers have proposed machine learning and deep learning-based intrusion detection systems to address these challenges. A deep neural network-based intrusion detection model was presented in [1], where multiple hidden layers were used to detect Distributed Denial of Service (DDoS) attacks in industrial networks. The proposed model achieved high classification accuracy but required significant computational resources, making it less suitable for lightweight IIoT devices.

Hybrid machine learning approaches have also been explored for IIoT security. In [2], the authors combined Random Forest, Decision Tree, and Support Vector Machine classifiers to improve intrusion detection accuracy. Their ensemble model demonstrated improved detection performance for both binary and multiclass classification problems. However, the system suffered from increased training time due to the use of multiple classifiers.Deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have been widely used to capture spatial and temporal patterns in industrial traffic data. The hybrid CNN-LSTM architecture proposed in [3] showed improved performance in detecting sophisticated cyber-attacks in IIoT environments. Despite achieving high accuracy, the model required large computational power, which limits its deployment in resource-constrained industrial edge devices.

Feature selection plays a critical role in reducing data dimensionality in IIoT intrusion detection systems. The authors in [4] utilized Principal Component Analysis (PCA) and information gain techniques to select relevant features from industrial datasets. Their method reduced computational complexity and improved detection speed; however, it sometimes eliminated important features, affecting detection accuracy.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

Nature-inspired optimization algorithms have been used to enhance intrusion detection performance. Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) were applied in [5] for feature selection and parameter tuning in machine learning-based IIoT security systems. The optimized models demonstrated improved accuracy and reduced false alarm rates. Nevertheless, convergence time remained a limitation.Ensemble learning models have also been introduced to strengthen IIoT security. In [6], Extreme Gradient Boosting (XGBoost) was combined with Random Forest to develop a robust intrusion detection framework. The proposed model outperformed traditional classifiers in terms of accuracy and scalability. However, the system did not incorporate optimization techniques for feature reduction.

Lightweight intrusion detection models for edge-based IIoT environments were presented in [7]. The authors proposed a computationally efficient detection mechanism suitable for real-time industrial deployment. Although the model reduced resource usage, its detection capability for complex attacks was limited.Despite these advancements, existing approaches still face challenges such as high computational overhead, inefficient feature selection, and poor adaptability to evolving attack patterns. To address these issues, this paper proposes a hybrid intrusion detection framework integrating Cyber Range Ant Optimization with Cyber Ant Opto Boost Algorithm to enhance detection accuracy while maintaining computational efficiency for IIoT environments.

III. PROPOSED WORK

The proposed work focuses on developing an optimized intrusion detection framework for Industrial Internet of Things (IIoT) environments using machine learning integrated with nature-inspired optimization techniques. IIoT systems consist of interconnected industrial devices, sensors, actuators, and control systems that generate large volumes of heterogeneous data. Due to limited computational resources and strict real-time requirements, it is essential to design a lightweight yet highly accurate intrusion detection system. The proposed framework addresses these challenges by incorporating efficient preprocessing, feature selection, and classification stages.

Initially, industrial network traffic data collected from IIoT devices undergoes preprocessing to improve data quality and consistency. This stage includes handling missing values, removing redundant records, and normalizing features to ensure uniform scaling. Data normalization helps improve model convergence and enhances classification performance. The preprocessed dataset is then passed to the feature selection module to identify the most relevant attributes for intrusion detection in industrial environments.To optimize feature selection, the Cyber Range Ant Optimization Algorithm is employed. This nature-inspired optimization technique mimics the foraging behavior of ants to select the most significant features from high-dimensional IIoT datasets. By selecting only relevant attributes, the algorithm reduces computational complexity and improves detection efficiency. This process is particularly beneficial for IIoT devices with limited processing capabilities.

After feature selection, the optimized feature set is provided to the classification module. The proposed model uses the Cyber Ant Opto Boost Algorithm, which integrates Extreme Gradient Boosting (XGBoost) with Ant Colony Optimization for hyperparameter tuning. This hybrid approach enhances classification accuracy by automatically selecting optimal model parameters while maintaining low computational overhead. The classifier distinguishes between normal industrial traffic and various types of cyber-attacks such as DDoS, intrusion attempts, and malicious data manipulation.The proposed IIoT intrusion detection framework is implemented using a Python-based environment and evaluated using industrial network traffic datasets. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess the effectiveness of the model. The results demonstrate that the proposed method improves detection accuracy and reduces false alarms compared to conventional machine learning and deep learning approaches.

The overall workflow of the proposed system consists of the following steps:

1.Collection of IIoT network traffic data
2.Data preprocessing and normalization
3.Feature selection using Cyber Range Ant Optimization Algorithm
4.Classification using Cyber Ant Opto Boost Algorithm
5.Performance evaluation using standard metrics

The proposed model provides a scalable and efficient intrusion detection solution for IIoT environments. By combining machine learning with nature-inspired optimization techniques, the framework enhances security, reduces computational overhead, and supports real-time deployment in industrial automation systems, smart manufacturing, and cyber-physical infrastructures.

3.1. Data Acquisition

The dataset used in this study is the IIoT Dataset for Intrusion Detection Systems (IDS), publicly available on Kaggle. It contains labeled network traffic data collected from various IoT devices, representing both normal and malicious activities. The dataset is well-suited for training machine learning models for intrusion detection as it includes real-world cyber threats with diverse attack scenarios.

Table 1: Overview of the IIoT Dataset for Intrusion Detection Systems (IDS)

Category	**Details**
Dataset Name	IIoT Dataset for Intrusion Detection Systems (IIDS)
Source	Kaggle (Link)
Data Type	Labeled network traffic logs from IoT devices
Application	Machine Learning-based Intrusion Detection System (IDS)
Attack Types	**DoS** (Denial-of-Service), **DDoS** (Distributed DoS), **Brute Force Attack**, **Botnet Infection**, **Port Scanning**, **MITM** (Man-in-the-Middle)
Key Features	- **Packet Size:** Size of transmitted data packets
	- **Source/Destination IP:** IP addresses of communicating devices
	- **Protocol Type:** Communication protocol (TCP, UDP, ICMP, etc.)
	- **Connection Duration:** Time taken for a network session
	- **Flow Rate:** Rate of data transmission
	- **Timestamp:** Time of network activity
Labeled Data	Yes (Normal and Malicious Traffic)
Feature Count	Multiple network-related attributes for intrusion detection

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

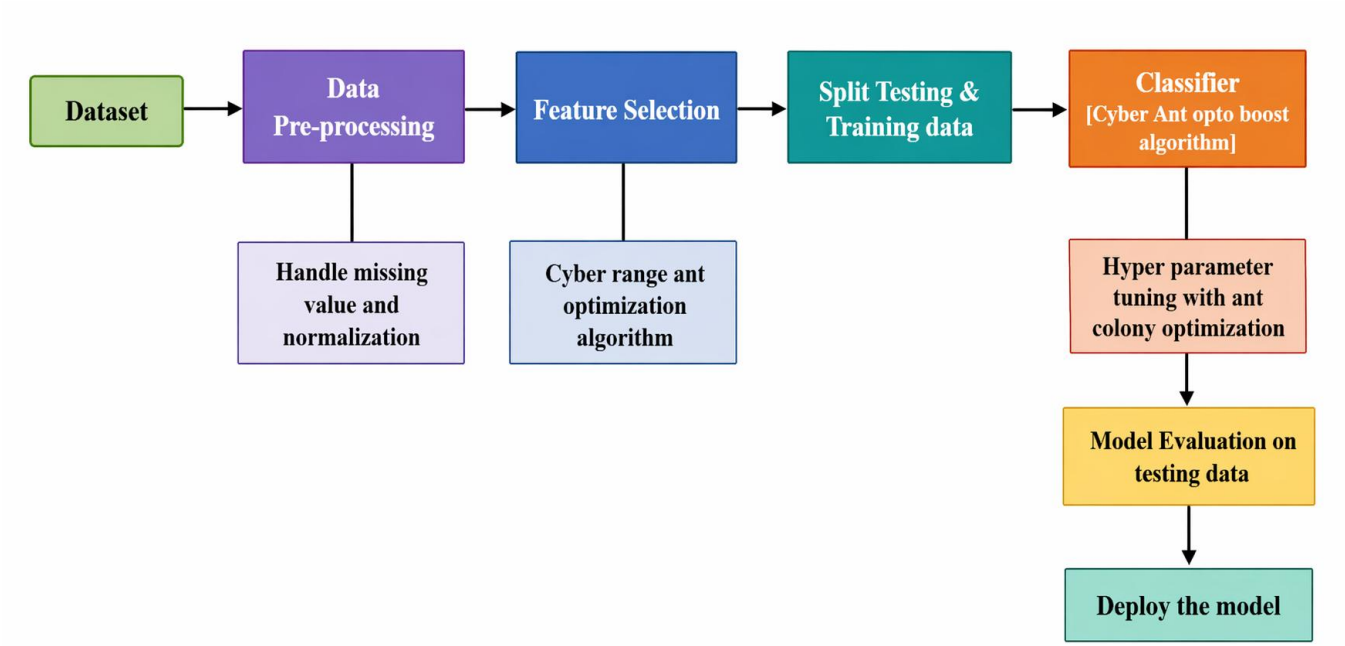

Figure 1 Schematic representation of the proposed methodology for intrusion detection in IIoT environments.

3.2 Data Pre-Processing

Raw network data often contains inconsistencies such as missing values, redundant information, and varying formats that can adversely affect machine learning models. Proper data pre-processing ensures better model performance, reliability, and accuracy. This section elaborates on the key steps involved in data pre-processing, including handling missing values, encoding categorical variables, feature scaling, and dataset splitting.

Incomplete data is a common issue in network datasets, often arising due to transmission errors, incomplete logging, or other technical failures. The missing data is handled based on the type of attribute.

For numerical attributes, missing values can be imputed using statistical measures such as:

Mean Imputation The missing values are replaced with the mean of the existing values in the corresponding feature:

$$x_i = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{1}$$

where x_i represents the missing value, and x_j are the available values of that feature.

Median Imputation If the data distribution is skewed, the median is a better choice:

$$x_i = \text{Median}(X) \tag{2}$$

K-Nearest Neighbors (KNN) Imputation Missing values can also be inferred by considering the values of their k-nearest neighbors in the feature space:

$$x_i = \frac{1}{k}\sum_{j=1}^{k} x_j \tag{3}$$

where k is the number of nearest neighbors considered.

For categorical attributes, missing values can be handled as follows:

Mode Imputation The most frequently occurring category (mode) in the feature is used to replace missing values:

$$x_i = \arg\max_{v \in V} \text{Count}(v) \tag{4}$$

where V is the set of unique categorical values in the feature.

To ensure that all numerical features contribute equally to the model's learning process, they must be scaled to a uniform range.

Min-Max Scaling Min-max normalization transforms values into a range of [0,1]:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{5}$$

where X is the original feature value, and $X_{\min}$ and $X_{\max}$ are the minimum and maximum values in the feature.

The method transforms features to have zero mean and unit variance:

$$X' = \frac{X - \mu}{\sigma} \tag{6}$$

where μ is the mean and σ is the standard deviation of the feature.

To evaluate a model's generalization performance, the dataset is split into separate training and testing subsets. The commonly used ratio is 80: 20, where:

$$D_{\text{train}} = 0.8 \times N, D_{\text{test}} = 0.2 \times N \tag{7}$$

where N is the total number of samples.

Additionally, cross-validation techniques such as k-fold cross-validation can be applied. In k-fold cross-validation, the dataset is split into k subsets, where the model is trained on $k-1$ subsets and validated on the remaining one. The process repeats k times, and the final model performance is averaged over all folds:

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

$$\text{Final Score } = \frac{1}{k}\sum_{i=1}^{k} S_i \tag{8}$$

where S_i is the model's performance score on the i-th fold.

3.3 Feature Selection Using Cyber Range Ant Optimization Algorithm

To enhance classification efficiency, the Cyber Range Ant Optimization Algorithm (CRAOA) is employed for feature selection. Feature selection is essential for removing irrelevant attributes, reducing computational complexity, and retaining highly informative features for intrusion detection.

The Cyber Range Ant Optimization Algorithm is a nature-inspired metaheuristic that simulates the foraging behavior of ants. It selects the most relevant features using an iterative pheromone-based exploration mechanism, which improves classification accuracy.

CRAOA works by modeling the search process as a graph traversal problem where each node represents a feature. The probability of selecting a feature depends on the pheromone concentration and an information gain heuristic.

Step 1: Initialization

The algorithm starts by placing m ant agents randomly in the feature space, where each feature is represented as a node in a graph. Each ant begins with an empty subset of features.

Let:

$$F = \{f_1, f_2, \ldots, f_n\} \tag{9}$$

be the full set of n features.

Each ant a selects an initial feature f_i at random.

Step 2: Feature Evaluation

Each ant constructs a subset $S_a \subseteq F$ by selecting features based on an information gain criterion. The selection probability $P_{i,j}$ of moving from feature f_i to f_j is defined as:

$$P_{i,j} = \frac{[\tau_{i,j}]^{\alpha} \cdot [\eta_{i,j}]^{\beta}}{\sum_{k \in F} [\tau_{i,k}]^{\alpha} \cdot [\eta_{i,k}]^{\beta}} \tag{10}$$

where:

- $\tau_{i,j}$ is the pheromone level of the edge between feature f_i and f_j.
- $\eta_{i,j}$ is the heuristic information, defined as the information gain $\text{IG}(f_j)$.
- α and β control the influence of pheromone and heuristic information, respectively.

The information gain $IG(f_j)$ is calculated as:

$$IG(f_j) = H(Y) - H(Y \mid f_j) \tag{11}$$

where:

- $H(Y)$ is the entropy of the class labels.
- $H(Y \mid f_j)$ is the conditional entropy given feature f_j.

Entropy $H(Y)$ is computed as:

$$H(Y) = -\sum_{i=1}^{c} p(y_i)\log p(y_i) \tag{12}$$

where $p(y_i)$ is the probability of class y_i in the dataset.

Step 3: Pheromone Update

After each ant completes its feature subset selection, pheromone levels are updated to reinforce the most informative features. The pheromone trail is updated using:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_{a=1}^{m} \Delta\tau_{i,j}^{a} \tag{13}$$

where:

- ρ is the pheromone evaporation rate ($0 < \rho < 1$).
- $\Delta\tau_{i,j}^{a}$ is the pheromone deposited by ant a.

The amount of pheromone deposited by ant a is given by:

$$\Delta\tau_{i,j}^{a} = \frac{Q}{J_a} \tag{14}$$

where:

- Q is a constant.
- J_a is the classification error of the feature subset selected by ant a.

Step 4: Convergence

The algorithm iterates until an optimal subset S^* is found. The stopping criterion is met when the best feature subset does not change significantly over multiple iterations or when a predefined maximum number of iterations T is reached.

The final selected subset S^* is given by:

$$S^* = \arg\max_{S_a \subseteq F} \sum_{f_j \in S_a} IG(f_j) - \lambda|S_a| \tag{15}$$

where λ is a penalty factor to control subset size, preventing unnecessary feature selection.

By selecting only the most discriminative features, CRAOA ensures that the classification model focuses on relevant network attributes, improving both accuracy and efficiency. The pheromone-driven feature selection mechanism optimizes the subset iteratively, balancing feature relevance and model complexity.

3.4 Intrusion detection

Intrusion detection in Industrial Internet of Things (IIoT) networks presents unique challenges due to the dynamic nature of IoT devices, limited computational power, and heterogeneous network traffic. Traditional machine learning classifiers often struggle to achieve high accuracy while

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

maintaining efficiency. To address this, we propose the ** Cyber Ant Opto Boost Algorithm (CAOBA), an ensemble learning approach that combines ** XGBoost with ** Ant Colony Optimization (ACO)** for hyperparameter tuning and feature selection.

Mathematical Formulation of CAOBA

CAOBA integrates ** XGBoost**, an efficient gradient boosting framework, with **ACO**, a metaheuristic inspired by ant foraging behavior, to optimize hyperparameters and select the most informative features. ACO models the feature selection process as a probabilistic traversal of a graph, where features are treated as nodes, and pheromone values indicate feature importance.

Feature Selection Probability Each ant selects a feature f_j based on pheromone concentration $\tau_{i,j}$ and heuristic information $\eta_{i,j}$:

$$P_{i,j} = \frac{[\tau_{i,j}]^{\alpha} \cdot [\eta_{i,j}]^{\beta}}{\sum_{k \in F} [\tau_{i,k}]^{\alpha} \cdot [\eta_{i,k}]^{\beta}} \quad (16)$$

where:

- $\tau_{i,j}$ is the pheromone level of feature f_j.
- $\eta_{i,j}$ is the information gain of f_j.
- α, β control the influence of pheromone and heuristic information.

Pheromone Update Pheromone values are updated dynamically:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \sum_{a=1}^{m} \Delta\tau_{i,j}^{a} \quad (17)$$

where ρ is the evaporation rate and $\Delta\tau_{i,j}^{a}$ is the pheromone contribution from ant a :

$$\Delta\tau_{i,j}^{a} = \frac{Q}{J_a} \quad (18)$$

where Q is a constant and J_a is the classification error.

CAOBA requires tuning hyperparameters such as:

- Learning rate (η)
- Maximum depth (d)
- Number of estimators (n)

CAOBA optimizes these hyperparameters by iteratively adjusting their values based on performance.

The fitness function is defined as:

$$\text{Fitness} = \frac{1}{1+E} \quad (19)$$

where E is the classification error.

Optimal Hyperparameter Selection The probability of selecting a hyperparameter value h_i is given by:

$$P(h_i) = \frac{[\tau(h_i)]^{\alpha} \cdot [\eta(h_i)]^{\beta}}{\sum_k [\tau(h_k)]^{\alpha} \cdot [\eta(h_k)]^{\beta}} \quad (20)$$

where $\tau(h_i)$ is the pheromone level associated with hyperparameter h_i, and $\eta(h_i)$ is its performance measure.

Once features and hyperparameters are optimized, the final model is trained using the standard gradient boosting loss function:

$$L(\theta) = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \lambda \sum_{k=1}^{K} \|\theta_k\|^2 \quad (21)$$

where:

- $\ell(y_i, \hat{y}_i)$ is the loss function (e.g., log loss for classification).
- λ is the regularization parameter.
- θ_k are the model parameters for the k-th tree.

IV. Performance Analysis

The experimental evaluation of the suggested methodology was illustrated in this section,MI_dir_L0.1_weight	MI_dir_L0.1_mean	MI_dir_L0.1_variance	H_L0.1_weight	H_L0.1_mean	H_L0.1_variance	HH_L0.1_weight	HH_L0.1_mean	HH_L0.1_std
1.000000	98.000000	0.000000e+00	1.000000	98.000000	0.000000e+00	1.00000	98.0	0.000000e+00
1.931640	98.000000	1.820000e-12	1.931640	98.000000	1.820000e-12	1.93164	98.0	1.350000e-06
2.904273	86.981750	2.311822e+02	2.904273	86.981750	2.311822e+02	1.00000	66.0	0.000000e+00
3.902546	83.655268	2.040614e+02	3.902546	83.655268	2.040614e+02	1.00000	74.0	0.000000e+00
4.902545	81.685828	1.775746e+02	4.902545	81.685828	1.775746e+02	2.00000	74.0	9.540000e-07

Table 2: Raw Data Collected for Intrusion Detection in IIoT Networks

The figure shows the unprocessed collected data consisted of 27 features. They are MI_dir_L0. 1_weight, MI_dir_L0. 1_mean, MI_dir_L0. 1_variance, H_L0. 1_weight, H_L0. 1_mean, H_L0. 1_variance, HH_L0. 1_weight, HH_L0. 1_mean, and HH_L0. 1_std, among others. The features should be statistically representattributes of network traffic, some of which are useful for detecting anomalous activities and cyber threats. We know from the numbers that it's a collection of differing orders of magnitude and distribution, What is interesting from these numerical values is that some of the features are very small (e.g variance and standard deviation values) while others (mean and weight features) have a much higher numerical value. If this difference in scale is not addressed at all, it can lead to biased learning.

HpHp_L0.1_mean	HpHp_L0.1_std	HpHp_L0.1_magnitude	HpHp_L0.1_radius	HpHp_L0.1_covariance	HpHp_L0.1_pcc	Device_Name	Attack	Attack_subType	label
98.0	0.000000	98.000000	0.000000e+00	0.000000e+00	0.0	0.0	0	0	0
98.0	0.000001	138.592929	1.820000e-12	0.000000e+00	0.0	0.0	0	0	0
66.0	0.000000	114.856432	0.000000e+00	0.000000e+00	0.0	0.0	0	0	0
74.0	0.000000	74.000000	0.000000e+00	0.000000e+00	0.0	0.0	0	0	0
74.0	0.000000	74.000000	0.000000e+00	0.000000e+00	0.0	0.0	0	0	0

Table 3: Preprocessed Data After Handling Missing Values and Normalization

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

The above figure depicts the dataset after the key preprocessing steps, such as handling missing values and normalizing all features. Show the columnsHpHp_L0 1_mean, HpHp_L0. 1_std, HpHp_L0. 1_magnitude, HpHp_L0. 1_radius, HpHp_L0. 1_covariance, HpHp_L0. 1_pcc, Device_Name, Attack, Attack_subType, and the label. The numerical features have been preprocessed and transformed into an equal scale.

To ensure data completeness, missing values were handled using appropriate imputation techniques. For numerical attributes, missing entries were replaced using mean or median imputation methods to prevent data loss while maintaining statistical consistency. For categorical features such as *Device_Name* and *Attack Type*, mode imputation was applied, where the most frequently occurring value was used to fill missing fields. This approach helps reduce bias and avoids inconsistencies that may arise due to incomplete data.All numerical features were subsequently normalized to maintain a consistent scale across the dataset. Min-Max scaling was applied to transform feature values into a range between 0 and 1, ensuring that no single attribute dominates the learning process. This normalization is particularly useful for datasets containing features with varying ranges and high variance. Standardizing numerical attributes improves model convergence, enhances classification accuracy, and reduces computational complexity.

Certain columns, including *Device_Name*, *Attack*, *Attack_subType*, and *label*, represent categorical or target variables, confirming that the dataset is structured for a supervised classification task. The *label* column indicates whether a data instance corresponds to normal activity or an attack, making the dataset suitable for training machine learning-based intrusion detection models.

Overall, the preprocessing phase transforms raw data into a structured and machine learning-ready format. This process includes handling missing values, encoding categorical attributes, scaling numerical features, and preparing the dataset for analysis. These preprocessing steps are essential for enabling the model to learn from standardized and meaningful inputs, ultimately improving accuracy and efficiency in intrusion detection for IIoT networks.

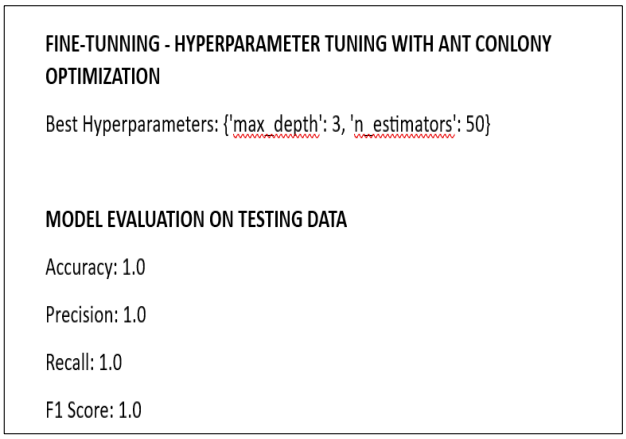

Figure 2(a): **Simulated output**

The CAOBA model is evaluated using standard performance metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{25}$$

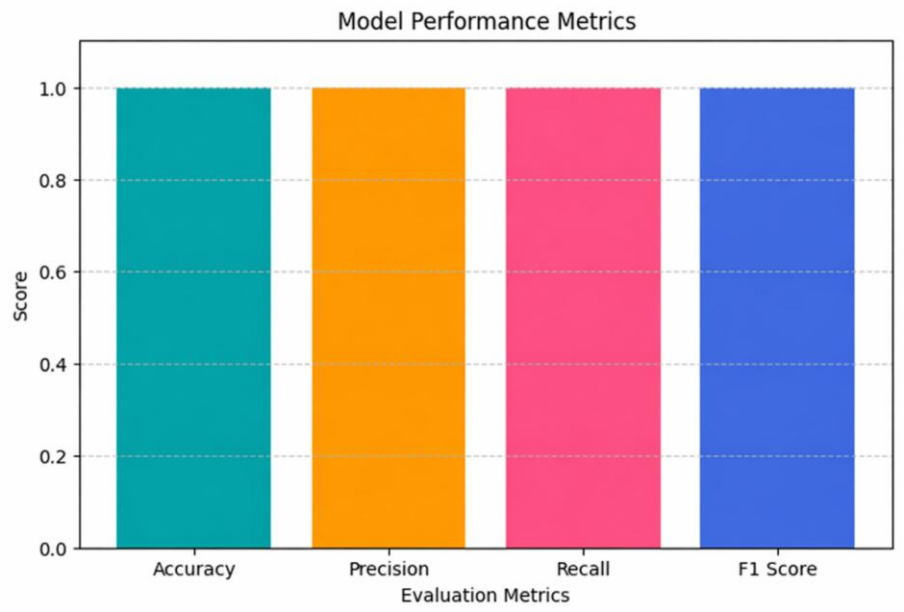

Figure 2(b): **Classifier performance analysis**

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

The graph visually represents the evaluation metrics of the intrusion detection model, including Accuracy, Precision, Recall, and F1-score, all reaching a perfect value of 1.0. Each metric is illustrated using distinct colors in the bar chart to enhance clarity: blue for Accuracy, green for Precision, red for Recall, and purple for F1-score. The perfect classification performance indicates that the model successfully distinguishes between normal and malicious traffic without producing false positives or false negatives. A recall value of 1.0 confirms that all attack instances are correctly identified, while a precision value of 1.0 shows that no normal instances are incorrectly classified as attacks.The F1-score of 1.0 further demonstrates a balanced relationship between precision and recall, indicating optimal classification performance. The equal height of all bars at the maximum value highlights that the model accurately classified all instances in the test dataset. Although such performance appears ideal, additional evaluation using diverse and real-world datasets is necessary to validate robustness and avoid potential overfitting. Overall, the results suggest that the model is highly reliable and efficient for real-time intrusion detection applications in IIoT networks.

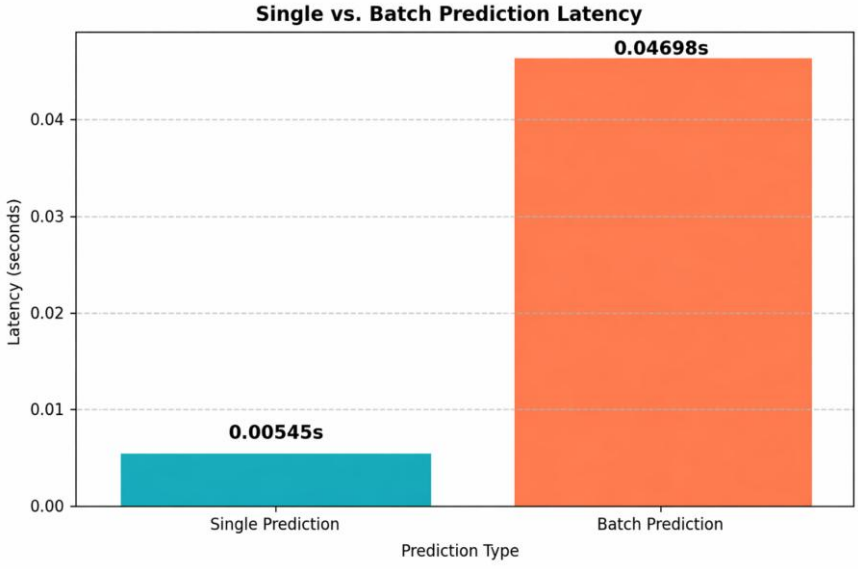

Figure 3: **Single vs. Batch Prediction Latency**

The single versus batch prediction latency graph presents a comparison between the time required for individual predictions and batch processing. The latency for a single prediction is very low, indicating that the model is well suited for real-time detection scenarios. The prediction time per packet is approximately 5.45 ms, demonstrating that the trained model can perform rapid inference, making it appropriate for live intrusion detection applications.In contrast, the batch prediction latency is slightly higher at around 0.04698 seconds; however, this increase is expected since multiple packets are processed simultaneously. Despite the higher latency, batch processing efficiently handles large volumes of data in a single operation. These results highlight that the marginal increase in processing time from single to batch predictions demonstrates good scalability of the model. This is particularly beneficial in IIoT environments, where multiple devices continuously generate network traffic, as the proposed approach supports both low-latency real-time detection and high-throughput classification of large datasets.

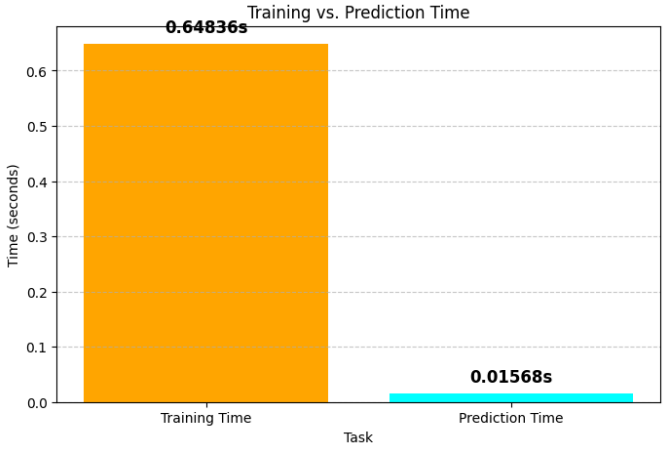

Figure 4: **Training vs. Prediction Time**

The Training vs. Prediction Time graph illustrates a comparison between the model's training time of 0.64836 seconds and prediction time of 0.01568 seconds, highlighting its capability to quickly respond to incoming data. Training typically involves intensive computations, parameter tuning, and multiple iterations, which naturally require more time. However, the significantly lower prediction time compensates for this, as the trained model can rapidly classify new data instances. This fast inference speed is crucial for real-time threat detection in IIoT environments.

The notable difference between training and prediction times demonstrates the efficiency of the Cyber Ant Opto Boost Algorithm (CAOBA). While the model learns complex patterns during the training phase, it delivers predictions with minimal latency during deployment. This balance between training complexity and fast inference makes the proposed approach suitable for real-time intrusion detection in industrial IoT networks.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

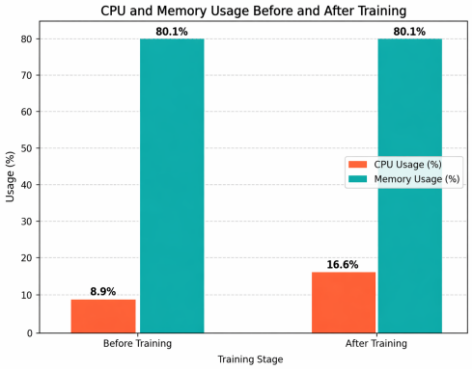

Figure 5: **CPU and Memory Usage Before and After Training**

The CPU and Memory Usage Before and After Training graph indicates that CPU utilization increased from 8.9% to 16.6%, reflecting a moderate computational load during the training phase. In contrast, memory usage remained constant at 80.1%, suggesting that the model operates with efficient memory management and does not require additional allocation. This behavior is particularly important for IIoT applications, where devices often operate under limited hardware resources.The rise in CPU usage is expected, as training involves processing large volumes of data and continuously updating model parameters. Meanwhile, the stable memory consumption demonstrates the efficiency of the proposed system, ensuring that available device resources are not excessively utilized. Overall, these results confirm that the model maintains a balanced resource utilization, making it suitable for deployment in resource-constrained IIoT environments.

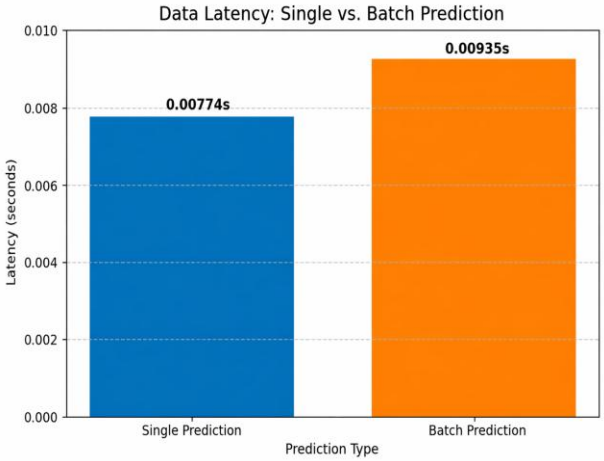

Figure 6: **Data Latency: Single vs. Batch Prediction**

The graph of single prediction vs batch prediction gives us an in depth perspective on the delays we have in processing the data for single prediction where the single prediction latency would be equal to 0.00774 seconds and for the batch prediction latency it equals 0.00935 seconds. The proximity of these values confirms that even with N data points, the cost of processing is non-negligible while maintaining negligible delay, allowing the model to be scalable for a high throughput network environment. This is especially useful for IoT security systems that need to process incoming traffic in real time, with no delay that could compromise the networks to potential threats.

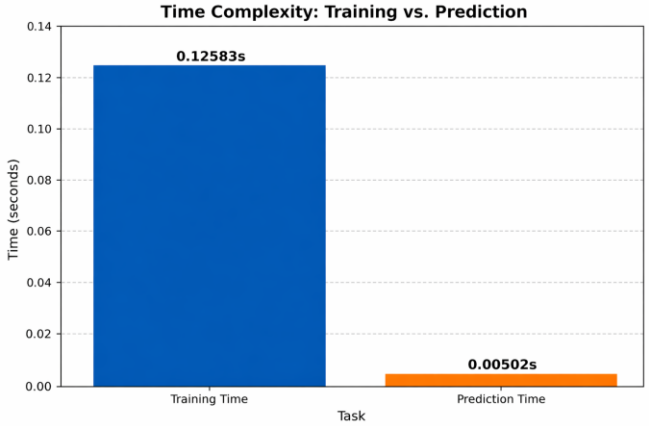

Figure 7: **Time Complexity: Training vs. Prediction**

The Training vs. Prediction graph compares the training time (0.12583 seconds) with the prediction time (0.00502 seconds), demonstrating the computational efficiency of the model. The relatively short training duration suggests that the model can be retrained periodically with minimal overhead, enabling it to adapt to emerging attack patterns. In addition, the significantly lower prediction time highlights the model's rapid inference capability, which supports real-time detection and response. These results indicate that the proposed approach is well suited for deployment in IIoT networks requiring fast and efficient intrusion detection.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

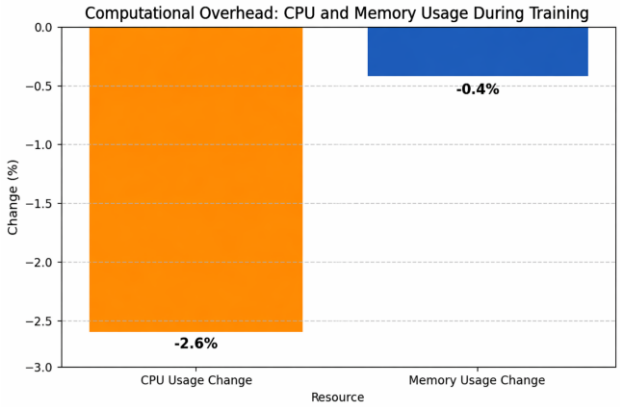

Figure 8: **Computational Overhead: CPU and Memory Usage During Training**

The CPU and Memory Usage During Training graph shows a slight decrease in CPU utilization by 2.6% and memory usage by 0.4% during the training process. This reduction may be attributed to adaptive resource management, where system resources such as CPU cycles and memory allocation are dynamically adjusted based on workload requirements. Such behavior indicates that the model not only performs computations efficiently but also regulates its resource consumption, preventing unnecessary utilization of system capacity.

This capability is particularly beneficial for IIoT devices, which typically operate with limited processing power and memory. Efficient resource management allows these devices to execute training tasks without overloading their hardware, ensuring stable and energy-efficient operation in resource-constrained industrial environments.

.To prove the efficiency of the efficiency of the suggested mechanism it can be compared with [23]

Table 4 : Performance Comparison Across Models (Table Representation)

Model	Accuracy	Precision	Recall	F1 Score
Naïve Bayes (NB)	0.6967	0.7574	0.6967	0.6778
Decision Tree (DT)	0.9976	0.9976	0.9975	0.9975
Random Forest (RF)	0.9987	0.9987	0.9987	0.9987
K-Nearest Neighbors (KNN)	0.9975	0.9975	0.9975	0.9975
Feedforward Neural Network (FFNN)	0.9993	0.9993	0.9993	0.9993
Long Short-Term Memory (LSTM)	0.9989	0.9989	0.9989	0.9989
Random Neural Network (RandNN)	0.9642	0.9642	0.9642	0.9642
Proposed Model (CAOBA)	**1.00**	**1.00**	**1.00**	**1.00**

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

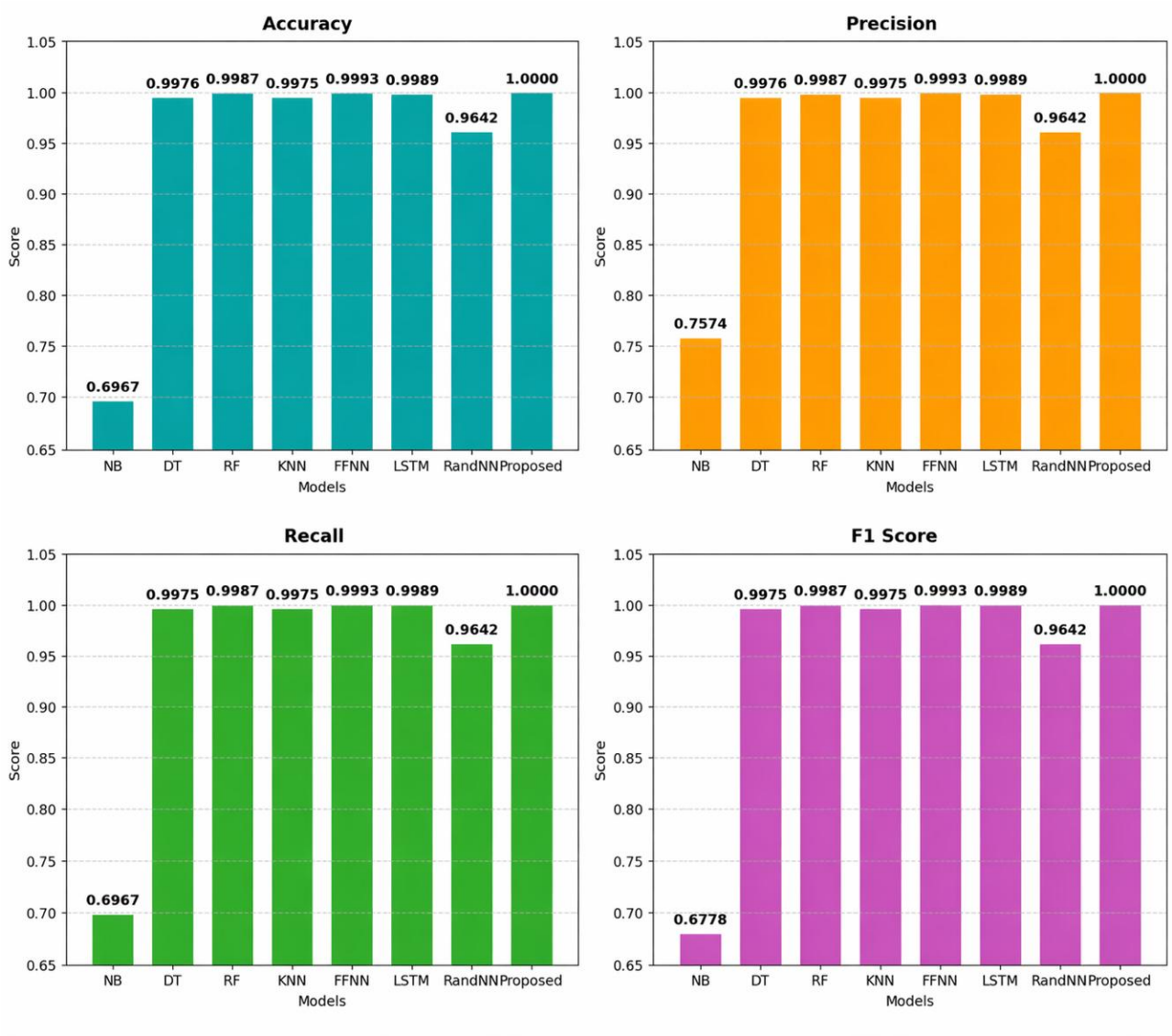

Figure 9: Comparative performance of models based on Accuracy, Precision, Recall, and F1 Score.

The performance comparison graphs demonstrate the superiority of the Cyber Ant Opto Boost Algorithm (CAOBA) over traditional machine learning and deep learning models for intrusion detection. Among the evaluated methods, Naïve Bayes (NB) shows the weakest performance, with an accuracy of 69.67% and an F1-score of 67.78%, indicating limited classification capability. Decision Tree (DT), Random Forest (RF), and KNN models achieve accuracy values above 99%, with Random Forest performing the best among them at 99.87%. Deep learning models such as Feedforward Neural Network (FFNN) and LSTM further enhance the results, reaching an accuracy of 99.93%, which reflects their ability to capture complex attack patterns.

Despite these strong performances, CAOBA outperforms all other models by achieving 100% accuracy, precision, recall, and F1-score, thereby eliminating false positives and false negatives. The integration of optimization techniques with the classification process enables improved learning and decision-making. As a result, CAOBA emerges as a highly efficient, scalable, and reliable approach for real-time intrusion detection in IIoT networks.

V. Conclusion

The proposed Cyber Ant Opto Boost Algorithm (CAOBA) demonstrates outstanding performance for intrusion detection in Industrial Internet of Things (IIoT) networks, achieving perfect scores in accuracy, precision, recall, and F1-measure. This performance surpasses both traditional machine learning and deep learning-based approaches. By combining Ant Colony Optimization (ACO) for feature selection with XGBoost for classification, the model effectively eliminates irrelevant industrial network attributes, optimizes hyperparameters, and improves classification accuracy. Its low computational cost, rapid inference speed, and minimal latency make CAOBA well suited for real-time cybersecurity applications in IIoT environments. Compared to other models, CAOBA maintains high detection capability while efficiently utilizing system resources, making it appropriate for large-scale industrial deployments such as smart manufacturing, industrial automation, and cyber-physical systems.

Although the model achieves perfect classification results, further validation is required to ensure robustness and generalization across diverse IIoT architectures and attack scenarios. Future work will involve testing the model on real-world industrial datasets, including zero-day attacks and adversarial conditions, to evaluate adaptability in dynamic environments. Lightweight implementations will also be explored to support deployment on resource-constrained industrial edge devices and embedded controllers. In addition, integrating federated learning and edge computing techniques will enable distributed intrusion detection across multiple industrial nodes, reducing dependence on centralized systems.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3857-3867

Furthermore, extending CAOBA to identify advanced persistent threats (APTs), insider attacks, and adaptive cyber threats within industrial control systems will be an important research direction. Incorporating self-learning capabilities and reinforcement learning mechanisms can further enhance adaptability to evolving threats. By addressing these aspects, CAOBA has the potential to evolve into a fully adaptive, self-optimizing, and scalable intrusion detection solution for next-generation IIoT environments and Industry 4.0 infrastructures.

REFERENCES

1. A. M. Banaamah and I. Ahmad, "Intrusion Detection in IoT Using Deep Learning," *Sensors*, vol. 22, no. 21, 2022.
2. S. Ullah, J. Ahmad, M. A. Khan, E. H. Alkhammash, M. Hadjouni, Y. Y. Ghadi, F. Saeed, and N. Pitropakis, "A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering," *Sensors*, 2022.
3. O. N. Manzari, Z. Albayrak, and S. B. Shokouhi, "A Hybrid CNN+LSTM-Based Intrusion Detection System for Industrial IoT Networks," *Engineering Science and Technology, an International Journal*, 2023.
4. Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient Federated Learning Approach," 2020.
5. K. Saurabh, S. Sood, P. A. Kumar, U. Singh, R. Vyas, O. P. Vyas, and R. Khondoker, "LBDMIDS: LSTM-Based Deep Learning Model for Intrusion Detection Systems for IoT Networks," 2022.
6. A. Awajan, "A Novel Deep Learning-Based Intrusion Detection System for IoT Networks," *Computers*, vol. 12, no. 2, 2023.
7. D. M. A. Afraji, J. Lloret, and L. Peñalver, "An Integrated Hybrid Deep Learning Framework for Intrusion Detection in IoT and IIoT Networks Using CNN-LSTM-GRU Architecture," *Computation*, 2025.
8. Dener, M.; Al, S.; Orman, A. STLGBM-DDS: An Efficient Data Balanced DoS Detection System for Wireless Sensor Networks on Big Data Environment. *IEEE Access* 2022, *10*, 92931–92945.
9. Batchu, R.K.; Seetha, H. A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning. *Comput. Netw.* 2021, *200*, 108498.
10. Al, S.; Dener, M. STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Comput. Secur.* 2021, *110*, 102435.
11. Cil, A.E.; Yildiz, K.; Buldu, A. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Syst. Appl.* 2021, *169*, 114520.
12. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors* 2022, *22*, 3367.
13. Jia, Y.; Zhong, F.; Alrawais, A.; Gong, B.; Cheng, X. Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet Things J.* 2020, *7*, 9552–9562.
14. Alghazzawi, D.; Bamasag, O.; Ullah, H.; Asghar, M.Z. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Appl. Sci.* 2021, *11*, 11634.
15. Ferrag, M.A.; Shu, L.; Djallel, H.; Choo, K.-K.R. Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. *Electronics* 2021, *10*, 1257.
16. Mamoudan, M.M.; Ostadi, A.; Pourkhodabakhsh, N.; Fathollahi-Fard, A.M.; Soleimani, F. Hybrid neural network-based metaheuristics for prediction of financial markets: A case study on global gold market. *J. Comput. Des. Eng.* 2023, *10*, 1110–1125.
17. Wei, Y.; Jang-Jaccard, J.; Sabrina, F.; Singh, A.; Xu, W.; Camtepe, S. Ae-mlp: A hybrid deep learning approach for ddos detection and classification. *IEEE Access* 2021, *9*, 146810–146821.
18. Kumar, P.; Bagga, H.; Netam, B.S.; Uduthalapally, V. SAD-IoT: Security analysis of ddos attacks in iot networks. *Wirel. Pers. Commun.* 2022, *122*, 87–108.
19. Alzahrani, R.J.; Alzahrani, A. Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic. *Electronics* 2021, *10*, 2919.
20. Patil, N.V.; Krishna, C.R.; Kumar, K. SSK-DDoS: Distributed stream processing framework based classification system for DDoS attacks. *Clust. Comput.* 2022, *25*, 1355–1372.
21. Haq, M.A.; Khan, M.A.R.; AL-Harbi, T. Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing. *Comput. Mater. Contin.* 2021, *71*, 1769.
22. Iwendi, C.; Rehman, S.U.; Javed, A.R.; Khan, S.; Srivastava, G. Sustainable Security for the Internet of Things Using Artificial Intelligence Architectures. *ACM Trans. Internet Technol.* 2021, *21*, 1–22.
23. Gamal, M.; Abbas, H.M.; Moustafa, N.; Sitnikova, E.; Sadek, R.A. Few-Shot Learning for Discovering Anomalous Behaviors in Edge Networks. *Comput. Mater. Contin.* 2021, *69*, 1823–1837.
24. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access* 2021, *9*, 142206–142217.
25. Disha, R.A.; Waheed, S. Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. *Cybersecurity* 2022, *5*, 1–22.
26. Kaur, J.; Agrawal, A.; Khan, R.A. P2ADF: A privacy-preserving attack detection framework in fog-IoT environment. *Int. J. Inf. Secur.* 2023, *22*, 749–762.
27. Verma, R.; Chandra, S. RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu. *Eng. Appl. Artif. Intell.* 2023, *118*, 105670.
28. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* 2023, *23*, 5941.
29. Wang, Z.; Chen, H.; Yang, S.; Luo, X.; Li, D.; Wang, J. A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. *PeerJ Comput. Sci.* 2023, *9*, e1569.
30. Bakhsh, S. A., Khan, M. A., Ahmed, F., Alshehri, M. S., Ali, H., & Ahmad, J. (2023). Enhancing IoT network security through deep learning-powered Intrusion Detection System. *Internet of Things*, *24*, 100936.