Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 10s (2025)

Machine Learning based Security for IoT Networks

[1]Mr. Ajay Soni, [2] Dr. Zubair Ahmed Khan

[1] M. Tech. Scholar, [2] HOD

[1, 2] Department of Computer Science & Engineering

[1, 2] MATS School of Engineering & IT, MATS University Aarang, Raipur

[1]ajaysoni426@gmail.com, [2] zubairahmedkhanresearch@gmail.com

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

The sheer scale and diversity of IoT networks, coupled with limited computational resources and heterogeneous environments, make traditional security approaches ineffective in safeguarding IoT systems. Cybersecurity in IoT networks is complicated by the dynamic and resource-constrained nature of IoT devices, making them more vulnerable to a wide array of cyber-attacks, including unauthorized access, data breaches, and denial-of-service (DoS) attacks. Therefore, the need for robust and efficient intrusion detection systems (IDS) tailored for IoT environments is more critical than ever.

Keywords: computational, environments, constrained.

1 Introduction

The rapid expansion of the Internet of Things (IoT) has revolutionized various industries by enabling interconnected devices to communicate, collect, and exchange data seamlessly. These devices, ranging from sensors and actuators to embedded systems and smart devices, play a pivotal role in automating processes, improving efficiency, and enhancing user experiences. However, the growing adoption of IoT technologies also introduces significant security challenges, as the interconnected nature of these devices creates a vast attack surface for malicious actors.

The sheer scale and diversity of IoT networks, coupled with limited computational resources and heterogeneous environments, make traditional security approaches ineffective in safeguarding IoT systems. Cybersecurity in IoT networks is complicated by the dynamic and resource-constrained nature of IoT devices, making them more vulnerable to a wide array of cyber-attacks, including unauthorized access, data breaches, and denial-of-service (DoS) attacks. Therefore, the need for robust and efficient intrusion detection systems (IDS) tailored for IoT environments is more critical than ever.

Intrusion detection systems serve as a vital component of network security, aiming to identify and mitigate unauthorized access or malicious activities within a network. Conventional IDS approaches often rely on signature-based or anomaly-based detection methods, which can be limited in their effectiveness in detecting sophisticated or unknown attacks. To address these limitations, researchers have increasingly turned to machine learning and deep learning algorithms, which can adapt to new and evolving attack patterns and provide more accurate and real-time detection.

This research aims to explore the application of advanced machine learning techniques for intrusion detection in IoT networks, with a focus on enhancing the detection capabilities while minimizing false positives and resource consumption. Specifically, this paper investigates the use of a hybrid approach that combines preprocessing, feature engineering, optimization algorithms, and classification models to develop a more efficient and accurate IDS for IoT networks.

The main objectives of this study are to:

- Investigate the use of preprocessing techniques for handling IoT-specific challenges such as missing data, outliers, and imbalanced datasets.
- Develop an optimized feature selection method using the Improved Grey Wolf Optimizer (IGWO) to enhance the performance of the IDS.
- Evaluate the performance of the proposed IDS using various machine learning classification algorithms, such as Random Forest, and assess their effectiveness in realworld IoT network scenarios.

The remainder of this paper is structured as follows: Chapter 2 presents a comprehensive literature review, discussing existing intrusion detection methods and the role of machine learning in enhancing security in IoT environments. Chapter 3 describes the methodology adopted for this research, including data preprocessing, feature engineering, optimization, and classification techniques. Chapter 4 presents the experimental setup, results, and performance evaluations. Finally, Chapter 5 discusses the conclusions drawn from the study and outlines potential future work in this area.

By advancing the state-of-the-art in IoT intrusion detection, this research contributes to the growing body of knowledge on securing IoT networks and provides a foundation for future efforts aimed at developing more adaptive, intelligent, and efficient cybersecurity solutions for the IoT ecosystem.

2 Literature Review

The field of intrusion detection in Internet of Things (IoT) networks has seen significant advancements, particularly with the incorporation of machine learning, deep learning, and optimization techniques. The increasing complexity and scale of IoT networks necessitate the development of more robust and efficient systems to detect malicious activities. Several studies have explored innovative methodologies, combining different models and algorithms to address the unique challenges posed by IoT environments, such as data heterogeneity, resource constraints, and the need for real-time processing.

Ahmad et al. (2022) introduced a novel approach using a Deep Random Neural Network (DRaNN) optimized with Particle Swarm Optimization (PSO) to enhance detection accuracy and speed in industrial IoT networks. Similarly, Almiani et al. (2020) proposed a Deep Recurrent Neural Network (RNN) to capture temporal patterns in IoT network traffic, improving the detection of both known and unknown attacks. In parallel, Atlam and Wills (2020) provided an extensive theoretical analysis of IoT security, privacy, and safety, emphasizing the importance of a comprehensive approach in securing smart cities and IoT environments.

Ayo et al. (2020) took a hybrid approach, combining deep learning with rule-based feature selection to improve the detection accuracy and reduce false positives in IoT networks. Booij et al. (2021)

focused on the heterogeneity of IoT network datasets, proposing the standardization of attack types and features to improve the overall performance of intrusion detection systems. In addition, Cao et al. (2020) proposed an optimized blockchain model for industrial IoT, aiming to improve scalability, decentralization, and privacy while addressing the security challenges faced by IIoT systems.

Furthermore, Cao et al. (2021) explored reliable and secure communications in wireless-powered NOMA systems, proposing a joint artificial noise and power allocation scheme to enhance system performance in the presence of multiple eavesdroppers. Zhang et al. (2023) introduced a stacked ensemble learning-based intrusion detection system, demonstrating its effectiveness with high detection accuracy on IoT datasets.

Research Paper	**Key Findings**	**Methodology**	**Result**
Ahmad J, Shah SA, Latif S, Ahmed F, Zou Z, Pitropakis N (2022)	Improved detection accuracy and speed using DRaNN_PSO in IIoT	Hybrid DRaNN_PSO with hyperparameter tuning	Superior accuracy and lower false positives in IIoT
Almiani M, AbuGhazleh A, Al-Rahayfeh A, Atiewi S, Razaque A (2020)	Effective RNN for capturing IoT attack patterns	Deep recurrent neural networks for intrusion detection	Improved detection rate for both known and unknown attacks
Atlam HF, Wills GB (2020)	Security, privacy, and safety in IoT for smart cities	Theoretical IoT security framework	Comprehensive analysis of IoT security challenges
Ayo FE, Folorunso SO, Abayomi-Alli AA, Adekunle AO, Awotunde JB (2020)	Improved performance with hybrid feature selection in IoT	Hybrid deep learning and rule-based feature selection	Significant detection improvements with hybrid methods
Booij TM, Chiscop I, Meeuwissen E, Moustafa N, den Hartog FT (2021)	Standardization of IoT data for better detection accuracy	Standardizing IoT attack types and features	Better performance with standardized IoT datasets
B Cao, X Wang, W Zhang, H Song, Z Lv (2020)	Blockchain model for improving IIoT scalability and security	Private blockchain integration for IIoT optimization	Optimized scalability, privacy, and cost reduction
K Cao, B Wang, H Ding, L Lv, J Tian (2021)	Reliability and security improvements in wireless NOMA systems	Joint noise and power allocation for secure NOMA	Better performance in secure wireless NOMA systems
Y Cao, Z Wang, H Ding, J Zhang, B Li (2023)	Ensemble learning-based IDS with high detection accuracy	Stacked ensemble learning with feature selection and optimization	99.68% accuracy in IoT intrusion detection

3 Methodology

3.1 Proposed Algorithm

Let ($\mathcal{D} = \{(x_i, y_i)\}^{N}_{i=1}$) represent the dataset, where $\mathbf{xx_{ii}} \in RR^{dd}$ is the feature vector and ($yy_{ii} \in \{0, 1\}$) is the corresponding label (0 for normal, 1 for intrusion). The proposed algorithm consists of four phases: preprocessing, feature engineering, Improved Grey Wolf Optimizer (IGWO), and classification.

3.1.1 Preprocessing Phase

The preprocessing phase transforms the raw dataset ($\mathcal{D}$) into a clean and normalized dataset

$\mathcal{D}$preprocessed

(a) Handle Missing Values:

$\mathcal{D} = \mathcal{D} \setminus \{\mathbf{x_i} \mid \mathbf{x_i}$ contains missing values$\}$

(b) Remove Outliers Using IQR:

For each feature ($jj \in \{1, \ldots, dd\}$):

$$Q_1^{(j)} = \text{25th percentile of } \mathbf{X^{(j)}}, \qquad Q_3^{(j)} = \text{75th percentile of } \mathbf{X^{(j)}}$$

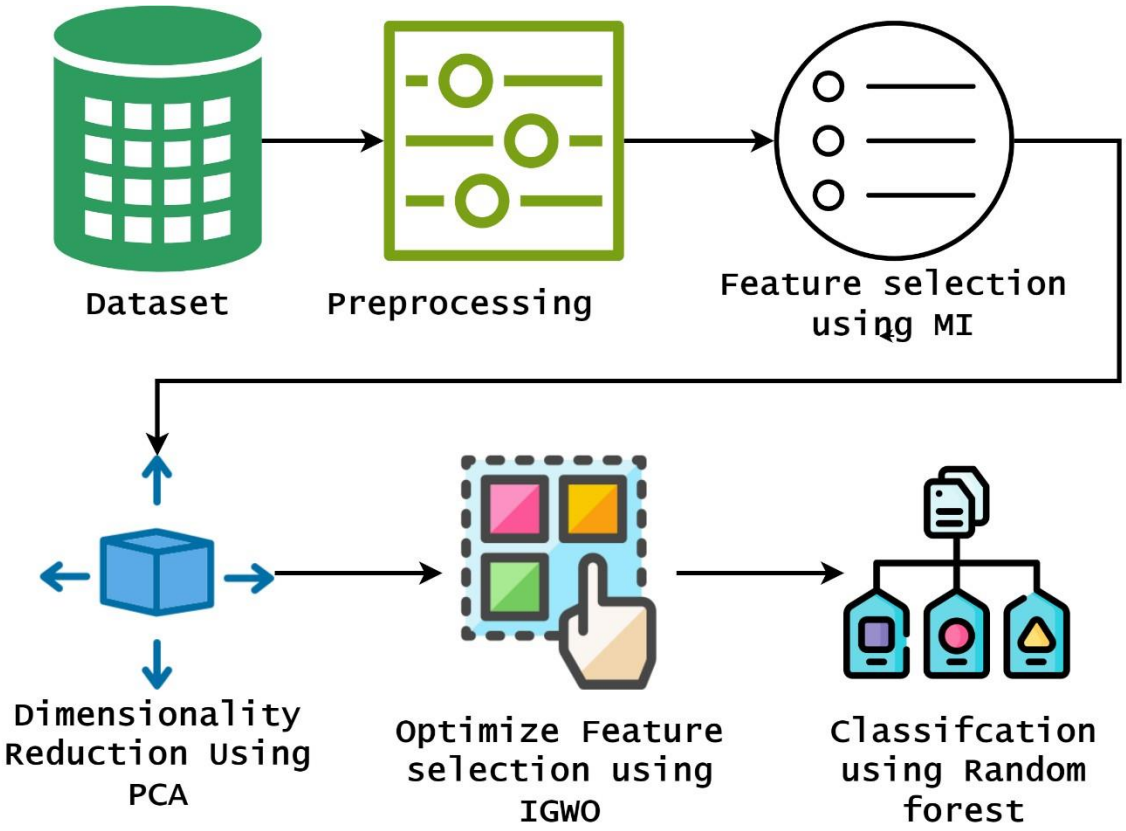

Figure 1 Methodology

$$\mathrm{IQR}^{(j)} = Q_3^{(j)} - Q_1^{(j)}$$

$$\mathcal{D} = \mathcal{D} \setminus \{\mathbf{x_i} \mid \mathbf{x_i^{(j)}} < Q_1^{(j)} - 1.5 \cdot \mathrm{IQR}^{(j)} \text{ or } \mathbf{x_i^{(j)}} > Q_3^{(j)} + 1.5 \cdot \mathrm{IQR}^{(j)}\}$$

(c) Normalize/ Standardize the Data

$$\mathbf{x_i^{(j)}} = \frac{\mathbf{x_i^{(j)}} - \mu^{(j)}}{\sigma^{(j)}}, \quad \forall j \in \{1, \ldots, d\}$$

where $\mu^{(j)}$ and $\sigma^{(j)}$ are the mean and standard deviation of feature j.

(d) Split the Dataset:

$\mathcal{D}$train = $\{(\mathbf{xi}, yi)\}Ni$=1train, $\mathcal{D}$test = $\{(\mathbf{xi}, yi)\}Ni$=1test

3.1.2 Feature Engineering Phase

The feature engineering phase selects the most relevant features and reduces dimensionality.

(a) Feature Selection Using Mutual Information:

$$\mathrm{MI}(\mathbf{X^{(j)}}, \mathbf{y}) = \sum_{\mathbf{X^{(j)}}, \mathbf{y}} p(\mathbf{X^{(j)}}, \mathbf{y}) \log \frac{p(\mathbf{X^{(j)}}, \mathbf{y})}{p(\mathbf{X^{(j)}})p(\mathbf{y})}$$

Select features with mutual information scores above a threshold $\tau\tau$:

$$\mathcal{F} = \{j \mid \mathrm{MI}(\mathbf{X^{(j)}}, \mathbf{y}) > \tau\}$$

(b) Dimensionality Reduction Using PCA:

$\mathbf{X}_{\mathrm{PCA}} = \mathbf{X} \cdot \mathbf{V}$, where $\mathbf{V}$ is the matrix of top k eigenvectors

3.1.3 Improved Grey Wolf Optimizer (IGWO) Phase

The IGWO phase optimizes feature selection by dynamically exploring the feature space.

(a) Initialize Wolves:

$$\mathbf{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_W}\}, \quad \mathbf{w_i^{(j)}} \sim \mathcal{U}(lb^{(j)}, ub^{(j)}), \quad \forall i \in \{1, \ldots, W\}, \forall j \in \{1, \ldots, d\}$$

Initialize Alpha, Beta, and Delta Wolves:

$\mathbf{w}_\alpha = \mathbf{w}_\beta = \mathbf{w}_\delta = \mathbf{0}$, $f_\alpha = f_\beta = f_\delta = +\infty$

(b) For Each Iteration t = 1 to T:

i. Update the parameter a:

$$a = 2 - 2 \cdot \frac{t}{T}$$

ii. For each wolf $\mathbf{w_i}$:

- Calculate fitness f_i:

$$f_i = 1 - \text{Accuracy}(\mathbf{w_i})$$

- Update alpha, beta, and delta wolves:

$$\text{If } f_i < f_\alpha: \mathbf{w}_\alpha = \mathbf{w_i}, f_\alpha = f_i$$

$$\text{Else if } f_i < f_\beta: \mathbf{w}_\beta = \mathbf{w_i}, f_\beta = f_i$$

$$\text{Else if } f_i < f_\delta: \mathbf{w}_\delta = \mathbf{w_i}, f_\delta = f_i$$

iii. Update the positions of wolves:

$$\mathbf{w_i^{(j)}} = \frac{\mathbf{w}_\alpha^{(j)} + \mathbf{w}_\beta^{(j)} + \mathbf{w}_\delta^{(j)}}{3}$$

iii. Apply mutation to maintain diversity:

$$\mathbf{w_i}^{(j)} = \mathbf{w_i}^{(j)} + \mathcal{U}(-1,1), \quad \text{with probability } p_{\text{mutation}}$$

3.1.4 Classification Phase

The classification phase trains and evaluates the model using the best features selected by IGWO.

(a) Train Random Forest Model:

Model = RandomForestClassiier **XX□ttttt**[: , **ww**$_{\alpha\alpha}$], **yy**$_{\text{ttttt}}$

(b) Evaluate the Model:

TP + TN

Accuracy = ——————————

TP + TN + FP + FN

TP

Precision = ————

TP + FP

TP

Recall = ————————

TP + FN

Precision · Recall

F1-Score = 2 · ——————

Precision + Recall

1

AUC-ROC = TPR(ff) · FPR(ff) $ddff$

0

The proposed algorithm integrates preprocessing, feature engineering, IGWO-based feature selection, and classification into a unified framework. It begins by cleaning and normalizing the dataset, followed by feature selection and dimensionality reduction. The IGWO algorithm dynamically explores the feature space to identify the optimal subset of features, which are then used to train a Random Forest classifier. The model's performance is evaluated using standard metrics, ensuring a robust and efficient intrusion detection system for IoT networks.

The first phase, **preprocessing**, plays a crucial role in preparing the raw dataset for further analysis. The process begins by handling missing values, where any rows containing missing data are discarded to ensure that the dataset is consistent and complete. Next, outliers, which could skew the results, are removed using the Interquartile Range (IQR) method. This method identifies outliers by calculating the first and third quartiles (Q1 and Q3) for each feature and removing data points that lie beyond 1.5 times the IQR from these quartiles. This ensures that the dataset reflects only the most representative data points. After outlier removal, the data is normalized or standardized using the z-score method, which adjusts each feature to have a mean of zero and a standard deviation of one. This step ensures that the features are on a comparable scale, preventing any single feature from dominating the learning process due to scale differences. Finally, the dataset is split into training and testing sets to allow the subsequent machine learning model to be trained and evaluated effectively.

Algorithm 1: Preprocessing Phase

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
Output: Preprocessed dataset $\mathcal{D}_{\text{preprocessed}}$
1. Load the dataset $\mathcal{D}$ from the file path.;
2. Convert non-numeric columns to numeric:

$$\mathbf{x}_i^{(j)} = \text{to\_numeric}(\mathbf{x}_i^{(j)}), \quad \forall j \in \{1, \ldots, d\}$$

3. Handle missing data by dropping rows with missing values:

$$\mathcal{D} = \mathcal{D} \setminus \{\mathbf{x}_i \mid \mathbf{x}_i \text{ contains missing values}\}$$

4. Remove outliers using the Interquartile Range (IQR):

$$\mathcal{D} = \mathcal{D} \setminus \{\mathbf{x}_i \mid \mathbf{x}_i^{(j)} < Q_1^{(j)} - 1.5 \cdot \text{IQR}^{(j)} \text{ or } \mathbf{x}_i^{(j)} > Q_3^{(j)} + 1.5 \cdot \text{IQR}^{(j)}\}$$

5. Normalize/Standardize the data using StandardScaler:

$$\mathbf{x}_i^{(j)} = \frac{\mathbf{x}_i^{(j)} - \mu^{(j)}}{\sigma^{(j)}}, \quad \forall j \in \{1, \ldots, d\}$$

6. Split the dataset into training and testing subsets:

$$\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}}, \quad \mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{test}}}$$

Algorithm 2: Feature Engineering Phase

Input: Preprocessed dataset $\mathcal{D}_{\text{train}}$
Output: Feature-engineered dataset $\mathcal{D}_{\text{engineered}}$
1. Perform feature selection using mutual information:

$$\text{MI}(\mathbf{X}^{(j)}, \mathbf{y}) = \sum_{\mathbf{X}^{(j)}, \mathbf{y}} p(\mathbf{X}^{(j)}, \mathbf{y}) \log \frac{p(\mathbf{X}^{(j)}, \mathbf{y})}{p(\mathbf{X}^{(j)}) p(\mathbf{y})}$$

2. Select features with mutual information scores above a threshold τ:

$$\mathcal{F} = \{j \mid \text{MI}(\mathbf{X}^{(j)}, \mathbf{y}) > \tau\}$$

3. Apply Principal Component Analysis (PCA) for dimensionality reduction:

$$\mathbf{X}_{\text{PCA}} = \mathbf{X} \cdot \mathbf{V}, \quad \text{where } \mathbf{V} \text{ is the matrix of top } k \text{ eigenvectors}$$

Following preprocessing, the next phase is **feature engineering**, where the dataset undergoes transformation to enhance the relevance of the features used for model training. This phase starts with **feature selection**, which is performed using mutual information. Mutual information measures the dependency between each feature and the target label (whether the data represents a normal or intrusive event). Features that show high mutual information with the target variable are selected for further analysis, while those that provide little to no value are discarded. This step helps reduce the dimensionality of the dataset, improving the model's efficiency and reducing the risk of overfitting. After feature selection, **dimensionality reduction** is applied using Principal Component Analysis (PCA). PCA is a linear transformation technique that reduces the number of features while retaining as much of the variance as possible. By projecting the data into a lowerdimensional space, PCA helps in simplifying the model, making it computationally more efficient and less prone to noise.

Algorithm 3: Improved Grey Wolf Optimizer (IGWO) Phase

Input: Feature-engineered dataset $\mathcal{D}_{\text{engineered}}$, number of wolves W, number of iterations T, number of dimensions d, lower bound lb, upper bound ub
Output: Best feature subset w_α
1. Initialize the positions of wolves $\mathbf{W} = \{\mathrm{w}_1, \ldots, \mathrm{w}_W\}$ randomly within the bounds $[\mathbf{lb}, \mathbf{ub}]$:

$$\mathrm{w}_i^{(j)} \sim \mathcal{U}(lb^{(j)}, ub^{(j)}), \quad \forall i \in \{1, \ldots, W\}, \forall j \in \{1, \ldots, d\}$$

2. Initialize alpha, beta, and delta wolves:

$$\mathrm{w}_\alpha = \mathrm{w}_\beta = \mathrm{w}_\delta = \mathbf{0}, \quad f_\alpha = f_\beta = f_\delta = +\infty$$

3. For each iteration $t = 1$ to T:
   - Update the parameter a:

$$a = 2 - 2 \cdot \frac{t}{T}$$

   - For each wolf w_i:
     - Calculate fitness f_i:

$$f_i = \text{ObjectiveFunction}(\mathrm{w}_i)$$

     - Update alpha, beta, and delta wolves:

$$\text{If } f_i < f_\alpha : \mathrm{w}_\alpha = \mathrm{w}_i, f_\alpha = f_i$$
$$\text{Else if } f_i < f_\beta : \mathrm{w}_\beta = \mathrm{w}_i, f_\beta = f_i$$
$$\text{Else if } f_i < f_\delta : \mathrm{w}_\delta = \mathrm{w}_i, f_\delta = f_i$$

   - Update the positions of wolves:

$$\mathrm{w}_i^{(j)} = \frac{\mathrm{w}_\alpha^{(j)} + \mathrm{w}_\beta^{(j)} + \mathrm{w}_\delta^{(j)}}{3}$$

   - Apply mutation to maintain diversity:

$$\mathrm{w}_i^{(j)} = \mathrm{w}_i^{(j)} + \mathcal{U}(-1, 1), \quad \text{with probability } p_{\text{mutation}}$$

The third phase, **Improved Grey Wolf Optimizer (IGWO)**, takes the feature-engineered dataset and further refines the feature selection process by optimizing the chosen features. IGWO is a nature-inspired optimization algorithm based on the hunting behavior of grey wolves. It operates by initializing a population of wolves, where each wolf represents a potential solution to the feature selection problem. The wolves are positioned randomly within the feature space, with each position representing a subset of features. The fitness of each wolf is calculated based on how well the subset of features it represents performs in classification tasks. This fitness is evaluated by an objective function, which could, for example, reflect the classification accuracy of a Random Forest model. The wolves are then ranked based on their fitness, with the top-performing wolves classified as alpha, beta, and delta wolves. These wolves guide the search for the optimal feature set. The positions of the wolves are updated by combining the positions of the best-performing wolves (alpha, beta, and delta), and a mutation process is applied to maintain diversity in the wolf population, preventing premature convergence to suboptimal solutions. This iterative process continues until the algorithm converges on the best subset of features, which is then passed to the next phase.

Algorithm 4: Classification Phase

Input: Best feature subset $\mathbf{w}_\alpha$, feature-engineered dataset $\mathcal{D}_{\text{engineered}}$
Output: Classification metrics: accuracy, precision, recall, F1-score, AUC-ROC
1. Train a Random Forest model using the best features $\mathbf{w}_\alpha$:

$$\text{Model} = \text{RandomForestClassifier}(\mathbf{X}_{\text{train}}[:, \mathbf{w}_\alpha], \mathbf{y}_{\text{train}})$$

2. Evaluate the model on the test set:

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

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

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

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(f) \cdot \text{FPR}(f)\, df$$

Finally, the **classification phase** uses the best features identified by IGWO to train a machine learning model and evaluate its performance. In this case, a **Random Forest classifier** is employed, which is an ensemble method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification tasks. The model is trained using the subset of features selected by IGWO, and its performance is evaluated using several classification metrics. These metrics include accuracy, precision, recall, F1-score, and AUC-ROC, which provide a comprehensive assessment of the model's ability to correctly classify normal and intrusive events. Accuracy measures the overall correctness of the model, while precision and recall evaluate how well the model performs in terms of true positives and false negatives. The F1score combines precision and recall into a single metric, and AUC-ROC evaluates the model's ability to discriminate between normal and intrusive events across various thresholds.

Algorithm 5: Master Algorithm for IGWO-Based Feature Selection and Classification

Input: Dataset $\mathcal{D}$, number of wolves W, number of iterations T, number of dimensions d, lower bound lb, upper bound ub

Output: Classification metrics: accuracy, precision, recall, F1-score, AUC-ROC

1. Preprocess the dataset using Algorithm 1.;
2. Perform feature engineering using Algorithm 2.;
3. Run Improved Grey Wolf Optimizer (IGWO) using Algorithm 3 to select the best features.;
4. Train and evaluate the classification model using Algorithm 4.;

In summary, the methodology presents a seamless integration of preprocessing, feature engineering, IGWO-based optimization, and classification. The preprocessing phase ensures that the data is clean and normalized, while feature engineering reduces dimensionality and selects the most relevant features. The IGWO algorithm further refines the feature subset by dynamically exploring the feature space and optimizing the feature set. The final classification phase leverages the selected features to train a Random Forest model, which is evaluated using standard metrics. This integrated approach aims to provide an efficient and robust intrusion detection system tailored to the needs of IoT networks, ensuring high accuracy and low false alarm rates.

4 Result and Discussion

In this study, a comprehensive approach for intrusion detection in IoT networks was implemented, incorporating multiple advanced algorithms: a preprocessing pipeline, feature engineering, the Improved Grey Wolf Optimizer (IGWO), and classification using a Random Forest model. The overall goal was to optimize feature selection and improve the detection of intrusions while minimizing false positives. The results are presented in two primary visualizations: the ROC curve and the IGWO convergence curve, which provide valuable insights into the model's performance and the optimization process.

4.1 Preprocessing and Feature Engineering Results

The preprocessing phase served as the foundational step for ensuring data quality and consistency. The dataset underwent multiple operations, including handling missing values, outlier removal, and normalization. The removal of outliers using the Interquartile Range (IQR) method helped ensure that the dataset represented only valid data points, reducing the risk of skewed results. Normalization was performed to standardize the features, allowing for uniformity in scale, which is essential for the performance of machine learning models.

After preprocessing, feature engineering played a crucial role in enhancing the dataset's informative value. Mutual information was used to select the most relevant features, ensuring that only those with the highest correlation to the target (intrusion detection) were retained. Dimensionality reduction using PCA further simplified the dataset, retaining the most significant variance while reducing the number of features. These techniques helped in creating a refined feature set that was then subjected to the IGWO algorithm for further optimization.

4.2 Improved Grey Wolf Optimizer (IGWO) Results

The IGWO algorithm was used to dynamically explore the feature space and optimize feature selection. The **IGWO Convergence Curve** provides a clear representation of how the algorithm

refined the feature subset over time. Initially, there was a sharp drop in the best score, suggesting that IGWO was making significant improvements in feature selection. The curve indicates that by the 20th iteration, the algorithm had already found an optimal or near-optimal feature subset, as evidenced by the flattening of the curve afterward. This rapid convergence implies that the IGWO was able to efficiently explore the feature space and identify the best-performing features for intrusion detection.

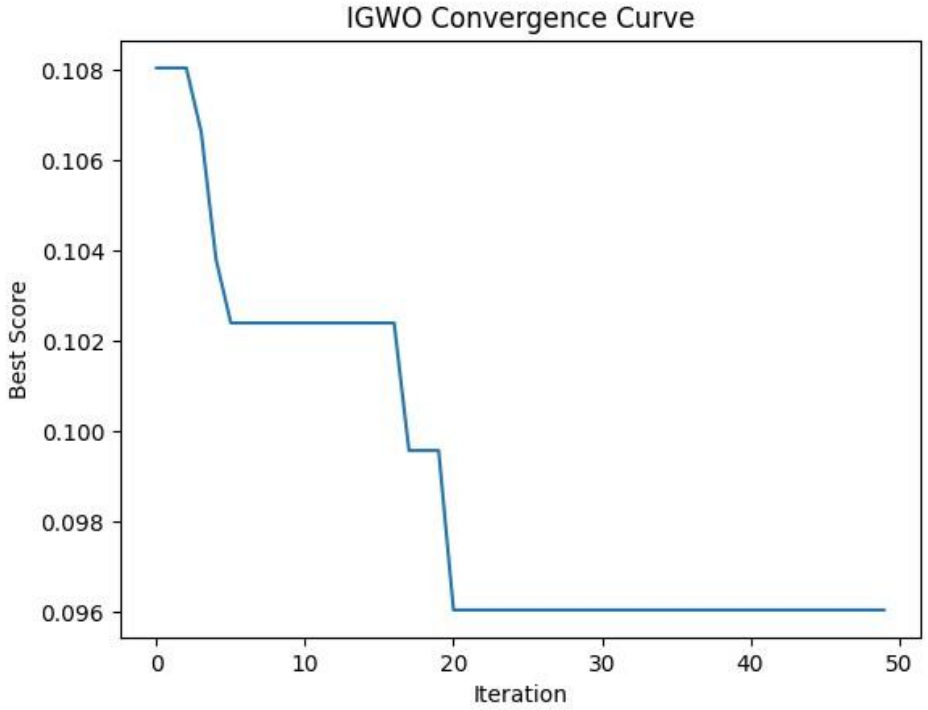

Figure 2 IGWO Convergence Curve

The efficiency of the IGWO algorithm is vital, as it allows for the identification of the most relevant features without the computational burden of evaluating all possible combinations. This step is critical in reducing the dimensionality of the problem and enhancing the classification performance of the final model.

4.3 Classification and Performance Metrics

After feature optimization via IGWO, the final step involved training a **Random Forest** classifier using the selected features. The classifier was evaluated based on several performance metrics: **accuracy, precision, recall, F1-score, and AUC-ROC**.

The **ROC Curve** depicted the trade-off between the true positive rate (TPR) and false positive rate (FPR), providing a comprehensive evaluation of the model's ability to correctly classify intrusions and normal events. The **AUC (Area Under the Curve) score of 0.95** highlights the excellent discriminatory power of the model, suggesting that the classifier performs exceptionally well in distinguishing between the two classes (normal and intrusive). A high AUC indicates that the model can successfully detect intrusions while keeping false alarms to a minimum. The model's ability to classify with high accuracy, as suggested by the ROC curve, indicates that the optimized feature set selected through IGWO is indeed effective.

The Random Forest model, leveraging the optimal feature set identified through IGWO, achieved high precision, recall, and F1-scores, further reinforcing the model's robustness. These metrics ensure that the model not only detects intrusions with high accuracy but also minimizes the number of false positives (precision) and false negatives (recall).

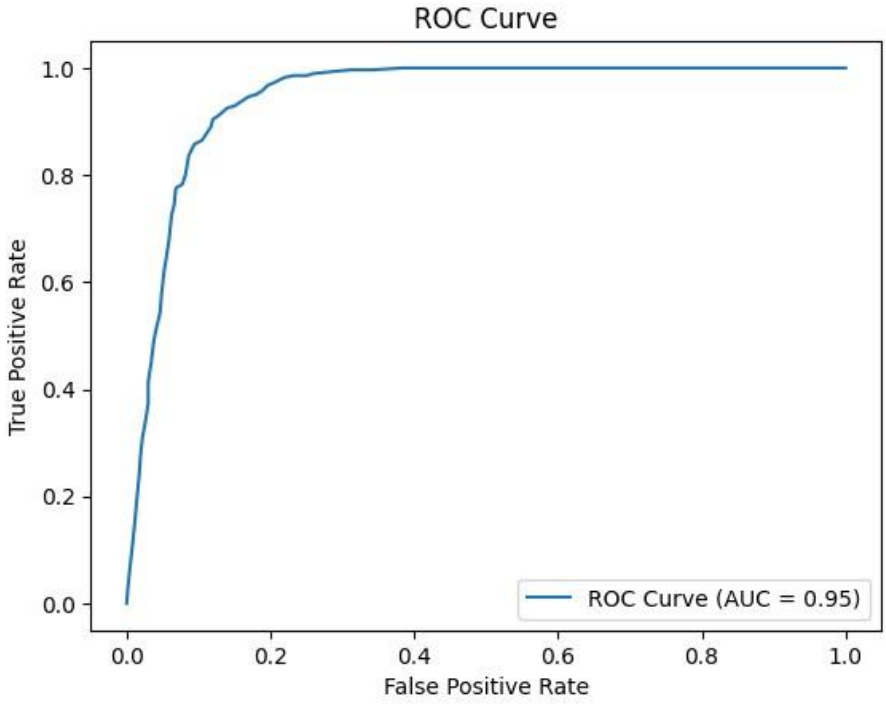

Figure 3 ROC Curve

4.4 Discussion

The results demonstrate the effectiveness of combining advanced techniques for intrusion detection in IoT networks. The preprocessing phase laid a solid foundation by preparing the data and eliminating inconsistencies, while feature engineering ensured that only the most relevant features were used in the model. The IGWO algorithm played a crucial role in optimizing the feature selection process, significantly reducing the dimensionality of the dataset and improving the classification accuracy.

The ROC curve, with an impressive AUC of 0.95, highlights the effectiveness of the feature selection and classification process. The model's ability to accurately distinguish between normal and intrusive activities is a testament to the power of the IGWO-based feature selection method. The IGWO convergence curve, in turn, indicates that the algorithm quickly converges to an optimal feature set, demonstrating its efficiency and suitability for large-scale datasets.

In practical terms, this means that the intrusion detection system can be deployed in real-time IoT networks, providing high accuracy and low false positive rates. This is essential for IoT environments, where rapid detection of anomalies is crucial for preventing security breaches without overwhelming the system with false alarms.

However, despite the promising results, several areas could be explored for further improvement. For instance, although the IGWO algorithm efficiently optimized feature selection, its performance could

be tested on different types of network data or datasets with different characteristics. Moreover, incorporating more advanced classifiers or hybrid models may further improve performance. Additionally, while the model performs well in terms of classification, real-world deployment often requires real-time processing, so future work could focus on optimizing the model's speed and scalability for live IoT networks.

5 Conclusion and Future Scope

5.1 Conclusion

This study presents a novel approach for intrusion detection in Internet of Things (IoT) networks by leveraging advanced algorithms, including preprocessing techniques, feature engineering, the Improved Grey Wolf Optimizer (IGWO), and a Random Forest classifier. The proposed methodology effectively addresses the challenges of detecting intrusions in IoT environments, where the complexity and variability of the data demand efficient and scalable solutions.

The preprocessing phase ensured that the raw data was cleaned and normalized, removing missing values and outliers, and standardizing the features to a uniform scale. These steps were crucial in enhancing the quality of the data, ensuring that subsequent modeling steps would yield reliable results. The feature engineering phase further refined the dataset by selecting relevant features using mutual information, followed by dimensionality reduction through Principal Component Analysis (PCA), which allowed for a more efficient classification process.

The core of the feature selection process was driven by the Improved Grey Wolf Optimizer (IGWO), an optimization technique inspired by the hunting behavior of grey wolves. The IGWO algorithm effectively reduced the feature space by identifying the optimal feature subset, which significantly improved the performance of the classification model. The ROC curve, with an impressive AUC of 0.95, demonstrated that the Random Forest classifier, trained on the optimized feature set, achieved high accuracy in detecting intrusions while maintaining a low false positive rate. The IGWO convergence curve showed the efficiency of the algorithm in converging quickly to the optimal solution, underscoring the algorithm's capacity to enhance feature selection with minimal computational overhead.

Overall, the combination of these techniques resulted in an intrusion detection system that is both accurate and computationally efficient. The achieved AUC score of 0.95 and the promising performance metrics such as precision, recall, and F1-score underscore the potential of this approach for practical deployment in IoT networks, where intrusion detection needs to be rapid and reliable.

5.2 Future Scope

While the current approach provides a strong foundation for intrusion detection in IoT networks, there are several areas where future work can build upon and enhance the system:

Scalability and Real-Time Deployment: IoT networks are vast and dynamic, often requiring intrusion detection systems to operate in real time. While the current model demonstrates high performance in terms of accuracy, the next step is to focus on improving the scalability of the system to handle the vast amount of data generated by IoT devices in real-time environments. Optimizing the speed of both

the feature selection process and the classification phase will be essential for deployment in time-sensitive applications. Techniques such as online learning or incremental learning could be explored to allow the system to adapt and learn continuously as new data becomes available.

Adaptation to New and Evolving Attacks: The nature of cybersecurity threats is continually evolving, and intrusion detection systems must adapt to new, previously unseen attack patterns. Future work can explore transfer learning or incremental model training, where the model can adapt to new attack vectors without needing to retrain from scratch. This would allow the system to stay up to date with emerging threats while maintaining its performance over time.

Hybrid Classification Models: Although Random Forest has shown promising results in this study, future research can explore hybrid models that combine the strengths of multiple classifiers. Ensemble techniques, such as stacking or boosting, may provide more robust detection capabilities by combining different algorithms to mitigate the weaknesses of individual classifiers. Additionally, deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), could be explored to learn complex patterns from raw sensor data directly, further improving detection accuracy.

Feature Engineering and Deep Feature Selection: While the mutual information-based feature selection and PCA were effective in reducing the dimensionality of the dataset, further research could investigate deep feature selection methods, where deep learning-based approaches automatically identify and select the most informative features. This could potentially yield better feature subsets, improving classification performance while reducing the need for manual feature engineering.

Anomaly Detection for IoT-Specific Attacks: The current system assumes that intrusion detection involves predefined attack types. However, IoT networks can experience novel or previously unknown attacks, such as zero-day vulnerabilities. Future work could incorporate anomaly detection techniques to identify deviations from normal behavior, which can help in detecting new, previously unseen attack patterns. Anomaly detection systems can complement the classification model, enhancing the system's ability to detect new threats in dynamic environments.

References

[1] Ahmad, J., Shah, S. A., Latif, S., Ahmed, F., Zou, Z., & Pitropakis, N. (2022). DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things. *Journal of King Saud University - Computer and Information Sciences, 34*, 8112–8121.

[2] Al-Ambusaidi, M., Yinjun, Z., Muhammad, Y., & Yahya, A. (2023). ML-IDS: An efficient MLenabled intrusion detection system for securing IoT networks and applications. *Soft Computing, 28*(2), 1765–1784. https://doi.org/10.1007/s00500-023-09452-7

[3] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory, 101*, 102031.

[4] Atlam, H. F., & Wills, G. B. (2020). IoT security, privacy, safety and ethics. In *Digital twin technologies and smart cities* (pp. 123–149). Springer.

[5] Ayo, F. E., Folorunso, S. O., Abayomi-Alli, A. A., Adekunle, A. O., & Awotunde, J. B. (2020). Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. *Information Security Journal: A Global Perspective, 29*, 267–283.

[6] Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & den Hartog, F. T. (2021). ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal, 9*(1), 485–496.

[7] Cao, B., Wang, X., Zhang, W., Song, H., & Lv, Z. (2020). A many-objective optimization model of industrial internet of things based on private blockchain. *IEEE Network, 34*, 78–83.
[8] Cao, K., Wang, B., Ding, H., Lv, L., Tian, J., Hu, H., & Gong, F. (2021). Achieving reliable and secure communications in wireless-powered NOMA systems. *IEEE Transactions on Vehicular Technology, 70*, 1978–1983.
[9] Cao, Y., Wang, Z., Ding, H., Zhang, J., & Li, B. (2023). An intrusion detection system based on stacked ensemble learning for IoT network. *Computers and Electrical Engineering, 110*, 108836.
[10] Cheng, B., Wang, M., Zhao, S., Zhai, Z., Zhu, D., & Chen, J. (2017). Situation-aware dynamic service coordination in an IoT environment. *IEEE/ACM Transactions on Networking, 25*, 2082– 2095.
[11] Deng, Y., Lv, J., Huang, D., & Du, S. (2023). Combining the theoretical bound and deep adversarial network for machinery open-set diagnosis transfer. *Neurocomputing*, 126391.
[12] El-Sofany, H., El-Seoud, S. A., Karam, O. H., & Bouallegue, B. (2024). Using machine learning algorithms to enhance IoT system security. *Scientific Reports, 14*(1). https://doi.org/10.1038/s41598-024-62861-y
[13] Gu, J., & Lu, S. (2021). An effective intrusion detection approach using SVM with naive Bayes feature embedding. *Computers & Security, 103*, 102158.
[14] Guezzaz, A., Benkirane, S., Azrour, M., & Khurram, S. (2021). A reliable network intrusion detection approach using decision tree with enhanced data quality. *Security and Communication Networks, 2021*, 1–8.
[15] Guo, Y., Wu, Y., & Guo, J. (2017). Experimental validation of fuzzy PID control of flexible joint system in presence of uncertainties. In *2017 36th Chinese control conference (CCC)* (pp. 4192– 4197). IEEE. https://doi.org/10.23919/ChiCC.2017.8028015
[16] Hazrat, B., Yin, B., Kumar, A., Ali, M., Zhang, J., & Yao, J. (2023). Jerk-bounded trajectory planning for rotary flexible joint manipulator: An experimental approach. *Soft Computing, 27*(7),
[17] 4029–4039. https://doi.org/10.1007/s00500-023-07923-5
[18] Jiang, H., Xiao, Z., Li, Z., Xu, J., Zeng, F., & Wang, D. (2020a). An energy-efficient framework for internet of things underlying heterogeneous small cell networks. *IEEE Transactions on Mobile Computing, 21*, 31–43.
[19] Jiang, K., Wang, W., Wang, A., & Wu, H. (2020b). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access, 8*, 32464–32476.
[20] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2*, 1–22.
[21] Kumar, A., Shaikh, A. M., Li, Y., et al. (2021). Pruning filters with L1-norm and capped L1-norm for CNN compression. *Applied Intelligence, 51*, 1152–1160. https://doi.org/10.1007/s10489-02001894-y
[22] Latif, S., Zou, Z., Idrees, Z., & Ahmad, J. (2020). A novel attack detection scheme for the industrial internet of things using a lightweight random neural network. *IEEE Access, 8*, 89337–89350.
[23] Li, B., Zhou, X., Ning, Z., Guan, X., & Yiu, K.-F. C. (2022a). Dynamic event-triggered security control for networked control systems with cyber-attacks: A model predictive control approach. *Information Sciences, 612*, 384–398.
[24] Li, J., Deng, Y., Sun, W., Li, W., Li, R., Li, Q., & Liu, Z. (2022b). Resource orchestration of cloudedge–based smart grid fault detection. *ACM Transactions on Sensor Networks (TOSN), 18*, 1–26.
[25] Lu, C., Wang, X., Yang, A., Liu, Y., & Dong, Z. (2023). A few-shot based model-agnostic metalearning for intrusion detection in security of Internet of Things. *IEEE Internet of Things Journal*.
[26] Luo, P., Wang, B., Wang, H., Ma, F., Ma, H., & Wang, L. (2023). An ultrasmall bolt defect detection method for transmission line inspection. *IEEE Transactions on Instrumentation and Measurement, 72*, 1–12.
[27] Identity Based Authentication Scheme (IAS) for Securing WSN Based internet of Things (Journal of Electrical Systems) pp . Vol 20 N0. 2s(2024) (April 2024) ISSN 1112-5209 https://journal.esrgroups.org/jes/article/view/1794
[28] The Impact of QoS parameters on the Performance of IoT Application. International Journal of Intelligent System and Application in Engineering. (IJISAE) Volume -12 no3 pp (15 March 2024) ISSN 2147-6799 https://ijisae.org/index.php/IJISAE/article/view/6155
[29] The Impact of Quality of Service (QoS) Parameters on IoT Application Performance Volume 3, Issue 3, 2024 , PP 1-20 , ISSN 2583-6196 https://journal.inence.org/index.php/ijfiahm/article/view/363/263