International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 4, April, 2025, pp. 1027-1041
Journal homepage: http://iieta.org/journals/isi

Advances in IoT Intrusion Detection: Deploying Hybrid Deep Learning and Metaheuristic Algorithms for Optimal Feature Selection

Ahmed Basil Abdulkareem

Continuous Learning Centre, University of Anbar, Ramadi 31001, Iraq

Corresponding Author Email: ahmedalnakep3@uoanbar.edu.iq

Copyright: ©2025 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.300419

Received: 1 August 2024
Revised: 4 April 2025
Accepted: 24 April 2025
Available online: 30 April 2025

Keywords:
IoT intrusion detection, hybrid deep learning, feature selection techniques, deep learning-based IDS, feature selection techniques

ABSTRACT

The proliferation of the Internet of Things (IoT) has tremendously increased the attack vectors for cyber threats, which necessitates advanced intrusion detection systems. In this paper, we propose a new method for detecting intrusions to the IoT based on a new hybrid deep learning model and metaheuristic algorithm for optimal feature selection. Our methodology takes advantage of the synergy between the CNN, BiGRU, and BiLSTM grids of models and integrates them into one architecture that enables them to leverage the spatial and temporal attributes of data to improve anomaly detection. While the model is refined using the Genetic, Harris Hawk, Dragonfly, Grey Wolf, and Particle Swarm Techniques, PSO demonstrates the best results, with a 98.11% accuracy level. This research uses comprehensive ToN-IoT datasets for analysis, which includes a wide variety of normal and adversarial traffic patterns affecting the IoT. The results show that our new hybrid model not only possesses a high level of accuracy, but it also exhibits considerable potential for real-world deployment. We further suggest potential areas for developing the model, such as scalability, real-time readiness, and its integration in environment computing. Our study advances cybersecurity by developing a cost-effective solution that can provide optimal protection to the IoT from multiple intrusion cases.

1. INTRODUCTION

In the fast-paced field of IoT security, the necessity for developing robust intrusion detection systems has become critical to mitigate the growing risks posed by sophisticated cyber threats [1]. Unlike traditional technologies, the IoT ecosystem is a massive collection of interconnected devices ranging from simple sensors to complex controllers, which have substantially expanded the attack surface. Consequently, traditional security technologies are increasingly unsuccessful in ensuring adequate protection. In the past, intrusion detection systems (IDS), used signature and anomaly-based methods. Signature-based methods identified threat patterns against observed activities to confirm threats. Anomaly-based methods identified deviations from normal activities to recognize threats. However, growing challenges to these approaches make them inappropriate for the dynamic and heterogeneous nature of IoT environments [2].

The increasing number of IoT devices and, correspondently, the amount of data they generate can reveal the weaknesses of the traditional IDS systems. The high rate of false positives and lack of scalability are problematic to solve, particularly considering the number of different devices and configurations that can be connected to the network. In addition, the static nature of signature-based methods leaves a high proportion of attacks undetected due to zero-day threats or multi-vector attacks that are more common in IoT.

The increasing complexity of the IoT environment requires more adaptive and intelligent security solutions. Deep learning (DL), as a technology capable of processing vast volumes of data and learning from them, has tremendous potential advantages in overcoming the limitations of traditional IDS systems [3]. Due to the ability to use the most sophisticated algorithms to perform pattern recognition and anomaly detection in data, DL-based IDS solutions can identify even subtle and unprecedented cybersecurity threats without any predefined signatures. However, the implementation of these innovative mechanisms in the practical real-world application scenarios of IoT is subjected to several serious challenges. These predominantly refer to the lack of computational efficient and properly functioning feature selection (FS) mechanisms to manage the high-dimensional data used in IoT networks [4].

IoT network intrusion detection faces significant difficulties due to the resource constraints of IoT devices and the heterogeneous nature of IoT networks. Many IoT devices are limited in processing power, memory, and energy, making it difficult to deploy resource-intensive intrusion detection mechanisms. This limitation often results in slower response times and reduced detection accuracy, as the devices cannot handle the computational load required for sophisticated detection techniques. Furthermore, the heterogeneity of IoT networks, consisting of diverse devices with varying capabilities and communication protocols, adds complexity to the intrusion detection task. Traditional IDS systems, which are designed for more homogeneous networks, struggle to adapt to the varied and dynamic nature of IoT environments. These factors underscore the necessity of developing efficient and scalable intrusion detection methods tailored specifically to the IoT context.

To address these challenges, we propose a novel hybrid deep learning model integrating Convolutional Neural Networks (CNN), Bidirectional Gated Recurrent Units (Bi-GRU), and Bidirectional Long Short-Term Memory (Bi-LSTM) networks. This architecture combines the strengths of each component to enhance both feature learning and temporal data processing for IDS.

To optimize feature selection, we employ five established metaheuristic algorithms [5]: Genetic Algorithm (GA) [6], Harris Hawk Optimization (HHO) [7], Dragonfly Algorithm (DA) [8], Grey Wolf Optimizer (GWO) [9], and Particle Swarm Optimization (PSO) [10]. These algorithms excel in efficiently exploring and exploiting high-dimensional search spaces, enabling the identification of the most discriminative features for robust intrusion detection.

This research work has two main objectives; to investigate the performance of the proposed hybrid DL model in detecting IoT intrusion and to examine the performance of the earlier metaheuristic algorithms in FS optimization. By accomplishing those objectives, the research paper intends to make a valuable contribution to the improvement of IDS in the IoT environment and to offer some benefits regarding the application of advanced machine learning (ML) in security fields. To conclude, the paper presents important progress in the field of IoT security, featuring the use of innovative DL and metaheuristic algorithms to solve the presently existing issue of optimal FS. The importance of this research is that, aside to the possibility of increasing the precision and productivity of IDS, it also creates a solid basis for the development of stronger IoT networks in the entirety of cybersecurity.

2. RELATED WORKS

Recent IoT IDS research focuses on ML/DL to improve detection and reduce false positives. Despite progress, challenges persist: high computational costs, overfitting due to high-dimensional data, reliance on outdated datasets, and class imbalance.

Recent studies show significant progress in feature selection for IoT security. IG and GR achieve >99.99% accuracy on IoT-BoT and KDD datasets [11]. Six ML models with PCA and GIWRF were tested, with Random Forest performing best on ToN-IoT [12]. ReliefF FS with ML/DL models reaches ~98% accuracy [13], while Chi-Square, Pearson, MI and NSGA-II maintain 99.48% accuracy with 13 features [14]. K-Best FS with ensemble ML achieves ~99.99% performance [15]. Chi2 FS, SMOTE and XGBoost show high accuracy on ToN-IoT [16]. LIDSS uses tree-based FS for lightweight networks [17], and CAT-S hybrid FS improves accuracy while reducing FPs [18]. XGBoost FS cuts features by 79% [19], though no single FE/ML model works best across all datasets [20].

Limitations include resource-heavy DL models, inconsistent FS/FE generalization, outdated static datasets, and skewed class distributions. Though SMOTE helps, imbalance remains an issue.

3. PROPOSED APPROACH

Our proposed approach, encapsulated in Figure 1, represents a comprehensive strategy to address the challenge of intrusion detection within IoT networks using a hybrid DL model. The foundation of our approach lies in the utilization of the ToN-IoT datasets, a diverse compilation of data reflecting both normal and adversarial patterns within IoT and IIoT systems, designed to simulate real-world industrial network complexities.

The initial phase of data preprocessing is critical to our methodology. We employ downsampling techniques to balance the class distribution within the dataset, mitigating bias towards any particular category of network traffic. This step is crucial to ensure that the model is not predisposed to overfitting to the ’normal’ class, which initially accounts for the majority of the data. Normalization and categorical encoding are then applied to make the dataset compatible with the requirements of ML algorithms, which prefer numerical input.

In FS, we explore various metaheuristic algorithms, each with unique capabilities to navigate through high-dimensional spaces to identify relevant features efficiently. These algorithms, namely GA, HHO, DA, GWO, and PSO, are instrumental in extracting a potent set of features from the preprocessed data, which are then used to train our hybrid model.

The core of our approach is the hybrid CNN-BiGRU-BiLSTM architecture, which leverages the strengths of convolutional layers for feature extraction and bidirectional recurrent layers to capture temporal dependencies and contextual information from the selected features. The combined model is designed to enhance the predictive capabilities of the system, allowing for accurate, real-time intrusion detection.

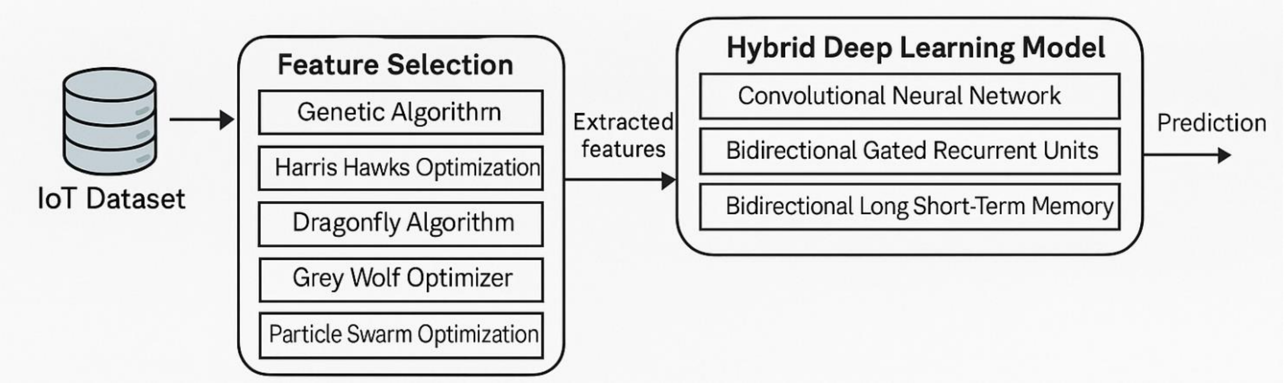

IoT Dataset
Feature Selection
Genetic Algorithrn
Harris Hawks Optimization
Dragonfly Algorithm
Grey Wolf Optimizer
Particle Swarm Optimization
Extracted features
Hybrid Deep Learning Model
Convolutional Neural Network
Bidirectional Gated Recurrent Units
Bidirectional Long Short-Term Memory
Prediction

Figure 1. Proposed scheme

Our assessment of the model's performance involves a comparative analysis of how each FS method influences the outcome. Thus, we are able to validate the strength and credibility of our hybrid model and identify the most suitable metaheuristic technique in our context. Consequently, our goal is to enhance our system performance in high precision, recall, and F1-score measures, which are essential success metrics in adequately predicting the classification of multi-class network traffic data for IoT security.

Through this meticulous process, we aspire to advance the state-of-the-art in AI-driven cybersecurity applications, providing a model that is not only academically robust but also practically reliable in the ever-evolving landscape of IoT network security.

A. Dataset

For our experiment, we used the ToN-IoT datasets that were created specifically for this purpose. The datasets were introduced in the study [21] and provide a unique opportunity to test the efficiency of AI-based approaches to cybersecurity in IoT and IIoT environments. The ToN-IoT datasets include several types of data from different sources, such as telemetry from IoT and IIoT sensors, OSes data from Windows 7 and 10 and Ubuntu 14 and 18 TLS, and network traffic, among others. This data was collected as part of a large network of the UNSW Canberra Cyber, at the School of Engineering and Information Technology at UNSW Canberra @ the Australian Defence Force Academy.

Our datasets were tailored to suit the complexity and scale challenge of IIoT and Industry 4.0 networks. We did so by setting up a new testbed in the IoT lab connecting many virtual machines, physical systems, hacking platforms, cloud, and fog platforms. Furthermore, the datasets comprise data from normal behaviors and various cyber-attack event, including DoS, DDoS, ransomware inflicted into the IIoT network.

Scanning: This attack involves an intruder probing a system to gather information about available services and open ports, typically serving as the first step before launching additional attacks.

XSS (Cross-Site Scripting): In the context of IoT, XSS attacks exploit vulnerabilities in web servers by running malicious scripts, which can compromise authentication mechanisms and leak sensitive data.

DoS (Denial of Service): A DoS attack aims to disrupt service availability by overwhelming a system with excessive requests, rendering it inaccessible to legitimate users.

DDoS (Distributed Denial of Service): Using a network of compromised devices (bots), this attack seeks to exhaust IoT resources by flooding them with a high volume of connections, thereby incapacitating the system.

Backdoor: This technique enables attackers to gain unauthorized remote access to IoT systems, often facilitating the deployment of botnets for subsequent DDoS attacks.

Injection Attack: Attackers inject malicious data or code into an IoT system to disrupt its normal operations or seize control over system functions.

Password Cracking: Through methods like dictionary or brute-force attacks, intruders attempt to break IoT device passwords, bypassing security measures to gain unauthorized access.

MITM (Man-in-the-Middle): In this type of attack, an adversary intercepts and potentially alters data transmissions within an IoT network, often using techniques like port stealing to exfiltrate information covertly.

Ransomware: A particularly damaging form of malware, ransomware locks users out of devices or services, demanding payment in exchange for a decryption key. In IoT environments, such attacks can cause substantial financial and operational damage.

This rich diversity of situational data provides the important basis for the study. Particularly, it supports all analyses and exploration experiments of the proposed hybrid DL in IoT network intrusion detection.

B. Data Preprocessing

The class distribution of network traffic data is illustrated in Figure 2 using a pie chart, which represents the proportion of each class before applying downsampling. The chart highlights a significant class imbalance, where the majority class, "normal," accounts for 65.1% of the total data. The remaining attack classes are distributed as follows: "scanning," "DoS," "injection," "DDoS," "password," "XSS," "ransomware," and "backdoor" each constitute 4.3% of the dataset. The "MITM" class is notably underrepresented, making up only 0.2% of the total data. This imbalance can adversely impact model training, as classifiers tend to be biased toward the dominant class, necessitating techniques like downsampling to improve detection performance across all attack categories.

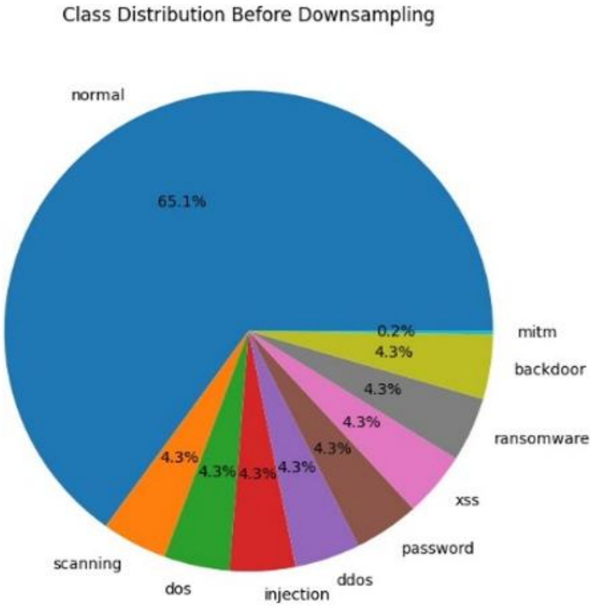


Figure 2. The class distribution of TON-IoT dataset

The downsampling step in our work aimed to equalize the representation of each class within our dataset. Initially, our dataset suffered from a significant class imbalance, with the "normal" category overshadowing the cyber threat classes. To rectify this and improve the efficacy of our ML models, we began by removing all instances of the "man-in-the-middle" (MITM) attack class, as it was deemed either too infrequent or irrelevant for our specific model training purposes.

We then implemented a downsampling function, which balanced the dataset by reducing the number of samples in each class to a consistent number. Specifically, we reduced each class to a maximum of 10,000 samples. If a class contained fewer than 10,000 samples, all instances were retained to maintain the integrity of the data. The downsampling was conducted using a random selection method, ensuring that the retained instances were representative of the overall distribution of each attack type. The implementation was performed using the sklearn.utils.resample function, setting the replace=False parameter to prevent duplicate instances and ensure data diversity.

Post-downsampling, as reflected in Figure 3, each remaining class—including "normal," "DoS," "DDoS," "injection," "backdoor," "XSS," "password," "scanning," and "ransomware"—is equally represented with 11.1% of the dataset. This equitable distribution is instrumental in preventing model overfitting to the most common class and ensures that our model remains sensitive to all classes, thereby enhancing its detection capabilities across various types of network traffic and cyber threats.

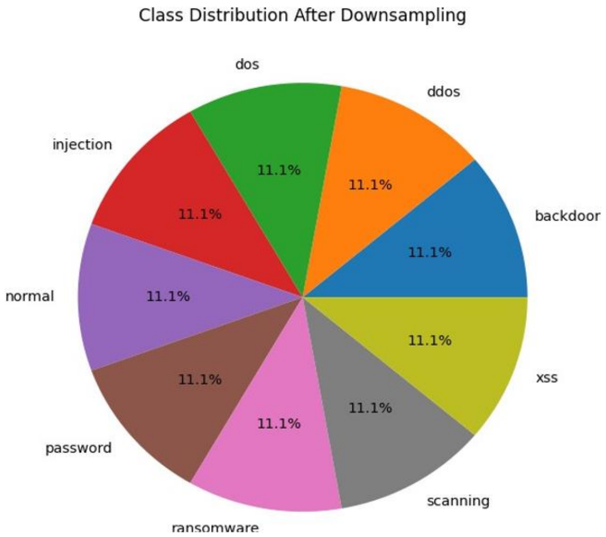


Figure 3. The class distribution of TON-IoT dataset after downsampling

In our research, we conducted several preprocessing steps to prepare the dataset for use in training our ML model, each chosen for its specific benefits. We started by encoding categorical variables into a numerical format using a LabelEncoder. This step is essential because ML models, as mathematical constructs, require numerical input to perform computations; categorical data in text form cannot be directly interpreted by these models. By converting categories into numbers, we maintain the categorical information in a way that our models can understand and process.

After encoding the categorical features, we applied one-hot encoding to the target variable using ‘to_categorical’ from Keras. One-hot encoding converts categorical variables into a format that could be provided to ML algorithms to do a better job in prediction. It creates a binary column for each category and returns a sparse matrix or dense array. By using this technique, we avoided the potential pitfalls of the model interpreting the numerical labels as having some sort of order or rank, which is not the case with categorical outcomes.

Next, we applied the MinMaxScaler to normalize the features in our dataset. Normalization is a crucial preprocessing step in machine learning, ensuring that numerical values across different features are scaled to a common range without distorting their relationships or losing essential information. MinMaxScaler transforms each feature individually to fall within a specified range, typically [0,1], using the following equation:

$$X_{\text{scaled}} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where, X represents the original feature value, X_{min} is the minimum value of the feature, and X_{max} is the maximum value of the feature.

By applying this transformation, all feature values are mapped within the [0,1] range, effectively standardizing the dataset while preserving the original distribution.

Finally, we split the dataset into training and test subsets using ‘train_test_split’, ensuring that our model could be trained on one portion of the data and validated on another. The test_size parameter specifies the proportion of the data to be used for testing, which we set at 20%.

These preprocessing steps together are crucial for our hybrid DL model as they prepare the raw data in a structured form, enhancing the learning process and enabling the model to make accurate predictions. Each step carefully conditions the data, preserving the inherent patterns and relationships while making them accessible for the learning algorithms to detect and utilize.

C. Feature Selection

In the feature selection (FS) step of our proposed approach, we employ metaheuristic algorithms to effectively identify the most relevant features from high-dimensional IoT network data. Traditional FS methods often struggle with the complexity of large search spaces and the combinatorial nature of selecting an optimal subset of features. In contrast, metaheuristic algorithms offer a robust alternative by employing iterative optimization strategies inspired by natural or social phenomena. These algorithms do not guarantee a globally optimal solution but are designed to explore and exploit the search space efficiently, leading to near-optimal solutions within a reasonable computational time. Their ability to balance exploration (diversely searching the solution space) and exploitation (refining promising solutions) makes them particularly suitable for FS in dynamic and high-dimensional IoT environments.

Genetic Algorithm (GA): One of the core algorithms employed is the GA, which draws inspiration from Darwinian evolutionary principles, including natural selection and genetic inheritance. In GA, candidate solutions (individuals) are encoded as chromosomes, forming a population that evolves through generations via biologically inspired operations: selection, crossover, and mutation. The selection process ensures that individuals with higher fitness—typically evaluated by a classification accuracy function—have a greater likelihood of being selected for reproduction. This encourages the propagation of superior solutions. Crossover combines parts of two parent solutions to produce offspring with mixed characteristics, increasing diversity within the population. Mutation introduces random modifications to individuals, thereby maintaining genetic diversity and avoiding premature convergence to suboptimal solutions. Formally, if we define a population $P = \{x_1, x_2, \dots, x_n\}$ where each $dx_i \in \{0,1\}$, d represents a binary string indicating the selection of features, the objective is to find a subset x^* that minimizes a fitness function such as $f(x) = \alpha \cdot E(x) + \beta \cdot |x|$, where E(x) is the classification error of the model trained on the selected features, |x| denotes the number of selected features, and α,β are weighting parameters.

Harris Hawks Optimization (HHO): Another potent metaheuristic in our framework is HHO, inspired by the cooperative predation strategies of Harris hawks. HHO simulates two distinct behavioral phases: exploration and exploitation. During exploration, hawks use diverse movement strategies—ranging from random to guided searches—to identify promising areas in the feature space. When prey (a good solution) is detected, the hawks switch to exploitation, executing soft or hard besiege strategies that model cooperative attack behavior, gradually refining their positions to converge on the optimal solution. The position update equation in HHO is influenced by the prey's energy level and escape probability, thereby guiding the hawks' aggressiveness and convergence pattern.

Dragonfly Algorithm (DA): In parallel, the DA replicates the dynamic swarming behavior of dragonflies by modeling five interaction forces: separation, alignment, cohesion, attraction to food, and distraction from enemies. These are used to iteratively update the positions of candidate solutions (dragonflies) in the feature space. Where X_i denote the position of dragonfly iii, then its movement is driven by a step vector ΔXi computed from these five factors. This collective behavior ensures that DA maintains balance between intensifying promising areas and diversifying the search—key to locating optimal feature subsets for complex intrusion detection tasks.

Grey Wolf Optimization (GWO): Mimics the leadership hierarchy and hunting tactics of grey wolves, where the alpha (α), beta (β), and delta (δ) wolves lead the search, and omega (ω) wolves follow. The position of each wolf (representing a feature subset) is updated according to the positions of the leading wolves, enabling the swarm to encircle the prey (i.e., the optimal solution) progressively. This is expressed as $X(t+1) = \frac{X_\alpha + X_\beta + X_\delta}{3}$, where $X_\alpha, X_\beta, X_\delta$ are the top three positions. The use of adaptive coefficient vectors ensures a gradual transition from exploration to exploitation.

Particle Swarm Optimization (PSO): PSO is deployed to emulate the social behavior of birds flocking or fish schooling. In PSO, each particle represents a candidate solution and adjusts its position and velocity based on personal and global best experiences. The velocity update equation is given by: $v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_{\text{best},i} - x_i^t) + c_2 r_2 (g_{\text{best}} - x_i^t)$, and the position update is $x_i^{t+1} = x_i^t + v_i^{t+1}$, where ω is the inertia weight, c_1, c_2 are cognitive and social coefficients, and r_1, r_2 are random factors. This learning mechanism allows PSO to effectively balance between exploring new solutions and exploiting known good solutions. The adaptability of PSO to changing feature space dynamics makes it particularly effective for evolving IoT datasets.

The use of these five metaheuristic algorithms in the FS step of our approach allows us to effectively navigate the high-dimensional search space and pinpoint the most critical features for intrusion detection in IoT networks. Each algorithm brings distinct strengths: GA excels at maintaining diversity and avoiding local optima, HHO dynamically balances exploration and exploitation, DA simulates intelligent swarm behavior for efficient searching, GWO leverages hierarchical social structures to refine solutions, and PSO optimizes feature subsets through adaptive learning mechanisms. By integrating these diverse approaches, our method ensures comprehensive feature selection, improving both the accuracy and efficiency of IoT intrusion detection systems.

D. Hybrid DL Model

During the modeling phase, we developed a hybrid CNN-BiGRU-BiLSTM model designed to mitigate the challenges of intrusion detection in IoT networks. The CNNs, which are crucial during the model's early stage of feature extraction, are specifically suitable for processing sequential/series data such as time-series data because the data from the sensors are sequences, and hence they are appropriate for analyzing the network traffic patterns in IoT. Our CNN had various layers, which included the convolutional layer with 64 filters and a kernel size of 3. After the convolutional layer is a ReLU activation function, which causes non-linearity, and our model was then subjected to max-pooling having a pool size of 2 which downsamples the extracted features to reduce computational cost. Our model also had a dropout layer with 0.2, which is then trained by randomly turning off or deactivating neurons.

After the CNN layers, we add Bi-directional Gated Recurrent Units (BiGRU) and Bi-directional Long Short Term Memory (BiLSTM) layers capture the temporal pattern in the sequential data along with context information. BiGRU and BiLSTM are the best way to approach any time series data primarily because they capture long-term dependencies and enable handling bidirectionally sequential data. BiGRU includes 64 units along with hyperbolic tangent (tanh) as an activation function and then include one more dropout layer for model regularization. Then we introduce the BiLSTM layer with 32 units and hyperbolic tangent as an activation function that will enable the model to learn complex patterns of temporal in the sequence of data. Finally, to complete the architecture, a fully connected dense layer with 9 output units and a softmax activation function is added to do the multiclass classification of intrusion detection labels. Here, the model is compiled using the Adam optimizer with a learning rate of 0.001 and binary cross- entropy loss function and the evaluation metric being an accuracy. Early stopping with a patience of 20 epochs is implemented as a callback in order to stop overfitting and recover the best weight using the validation loss. In this paper, we conducted a comparative study; the performance of the hybrid model is assessed by incorporating different metaheuristic FS techniques. Overall, the input data fed into the model was varied in terms of the selected features using individual metaheuristic methods. Therefore, variation was obtained via the model's performance to determine the significance of FS on the intrusion detection system using GA Optimization, HHO, Dragonfly Optimization, Grey Wolf Optimization, and PSO. As a result, this process enables us to evaluate the reliability and generalization performance of the hybrid CNN-BiGRU-BiLSTM model under various feature subsets generated by their respective metaheuristic algorithms. The adaptive feature selection technique is thus suitable for the real-world environment of the IoT system.

E. Evaluation Measures

In our paper, we utilize several evaluation measures to assess the performance of our proposed intrusion detection system in IoT networks. These measures include the Confusion Matrix, accuracy, precision, recall, and F1-score.

- Confusion Matrix: It is a tabular representation of the model predictions against the actual labels of the data. The confusion matrix consists of four quadrants: True positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), which help evaluate the model's performance across different classes. Other evaluation metrics are calculated based on these values.
- Accuracy: This metric measures the overall correctness of the predictions made by the model. It is defined as the ratio of the number of correct predictions to the total number of predictions:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

- Precision: Precision measures the proportion of true positive predictions out of all predicted positive cases. It quantifies the model's ability to avoid false alarms:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

- Recall (Sensitivity or True Positive Rate): Recall represents the model's ability to correctly identify all actual positive instances. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

- F1-score: The F1-score is the harmonic mean of precision and recall, balancing both false positives and false negatives. It is expressed as:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Together, these evaluation measures provide a comprehensive assessment of the model’s performance, considering its ability to make accurate predictions while minimizing false alarms and missed detections. These metrics allow us to understand how well the designed intrusion detection system identifies and neutralizes cyber threats targeting IoT networks.

4. RESULTS AND DISCUSSION

4.1 Results of hybrid DL with optimal features of GA

Applications of the hybrid DL model with optimal features selected by the GA report the following results. Timestamp and source IP address are the most general data points. They are followed by protocol attributes, transaction attributes, protocol level features, visit attributes, DNS query class, class label, SSL cipher, HTTPCAMID, and HTTP features including method, URI, version, status code, and user agent codes. This diversity in selected features shows that the GA has been able to efficiently identify a subset of data points that significantly contribute to the model performance. The training exercise over 10 epochs shown in Figure 4 consistently shows an increased accuracy over the training and validation datasets on top of a significant reduction in loss. The model achieved an initial accuracy of slightly above 50% of the training dataset, after which it rapidly learns and rises to over 94% by the fifth epoch. Validation accuracy, which is crucial to assess the model’s generalizability, also improves consistently, starting from approximately 71% and ending at around 97%. Notably, the validation loss decreases sharply and remains low, suggesting that the model is not overfitting to the training data (Figure 4). The final test accuracy of 97.71% is particularly impressive, showcasing the model’s capability to generalize well to unseen data. The convergence of the training and validation loss and accuracy curves indicates that the model is well-tuned and the chosen features are likely providing a good representation of the underlying data patterns necessary for intrusion detection. The training and validation graphs suggest a well-fitting model: the training loss decreases smoothly, avoiding plateaus, and the validation loss follows closely without significant divergences that would indicate overfitting. Similarly, the training and validation accuracy curves converge nicely, with the validation accuracy reaching a slightly higher value than the training accuracy at the end, which sometimes happens when the model learns general patterns that perform even better on the validation set.

The confusion matrix in Figure 5 for the hybrid DL model with optimal features selected by the GA provides detailed insights into the model’s performance across different classes. Each cell in the matrix represents the number of samples from the predicted class (horizontal axis) against the actual.

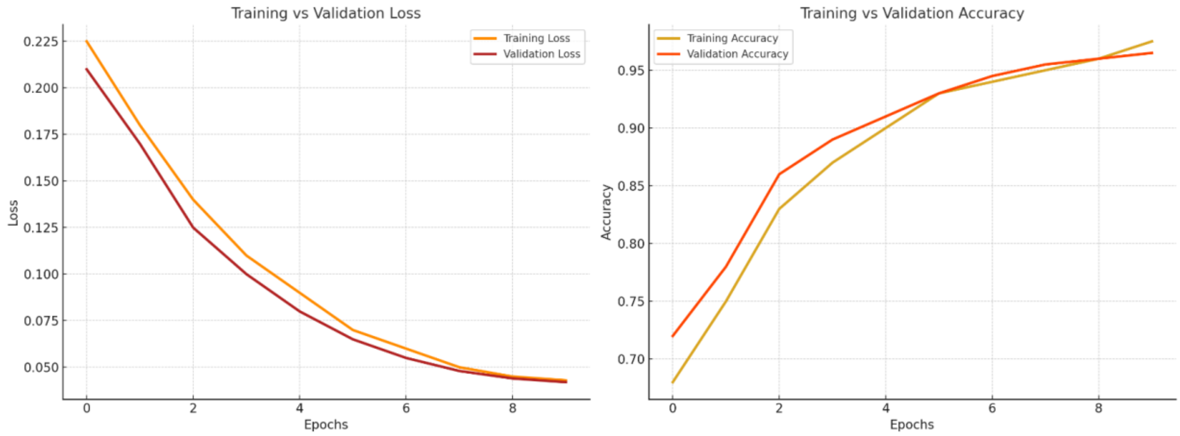
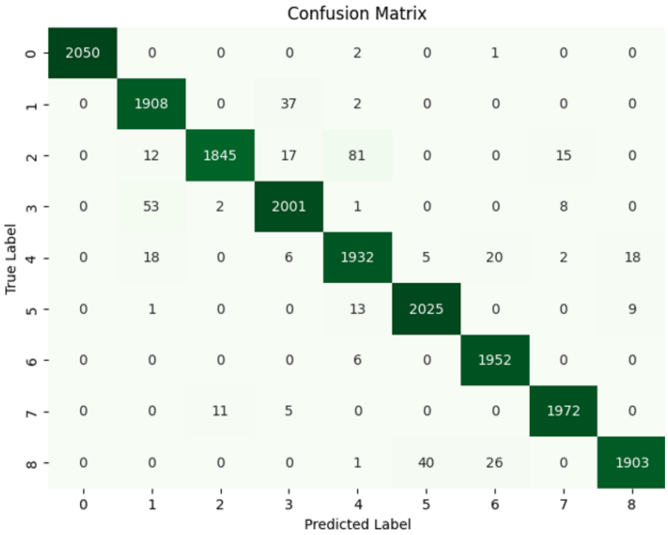


Figure 4. Training process of hybrid DL with optimal features of GA

The diagonal cells, which indicate correct predictions, are predominantly darker, suggesting a high number of true positives for most classes. Looking at the specific numbers, we can see that for class 0 (presumably ’normal’ traffic or another major class), the model correctly predicted 2050 instances, which indicates a strong performance for this class with very few instances being misclassified. Similar observations can be made for classes 1 through 8, with class 1 having 1908 true positives, and class 8 having 1903 true positives. There are, however, some off-diagonal cells with noticeable figures indicating misclassifications. For example, class 2 has been confused with class 4, 5, and 8 (81, 15 instances respectively), which could indicate similarities in the patterns of these specific classes that the model finds challenging to distinguish.

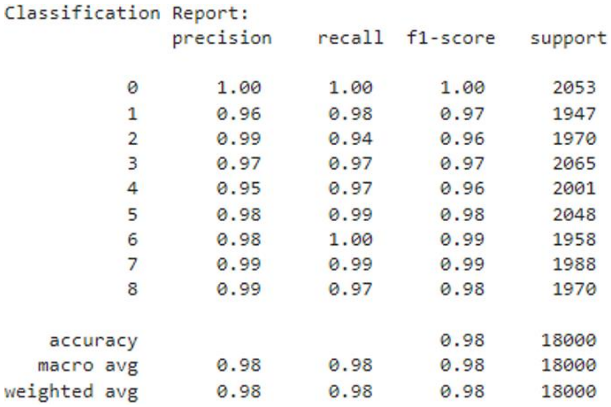
This pattern is also visible with class 4, where there are 18 instances that were predicted as class 8. Despite these few areas of confusion, the overwhelming majority of predictions lie on the diagonal, suggesting that the model is performing well. It is also notable that the misclassifications are not heavily skewed towards a single wrong prediction, which would have indicated a systematic bias. Instead, the spread of misclassifications across different classes suggests that the model might be struggling with specific features or similarities between certain classes.



True \ Predicted	0	1	2	3	4	5	6	7	8
0	2050	0	0	0	2	0	1	0	0
1	0	1908	0	37	2	0	0	0	0
2	0	12	1845	17	81	0	0	15	0
3	0	53	2	2001	1	0	0	8	0
4	0	18	0	6	1932	5	20	2	18
5	0	1	0	0	13	2025	0	0	9
6	0	0	0	0	6	0	1952	0	0
7	0	0	11	5	0	0	0	1972	0
8	0	0	0	0	1	40	26	0	1903

Figure 5. Confusion matrix of hybrid DL with optimal features of GA class (vertical axis)

The classification report in Figure 6 for the hybrid DL model with features selected by the GA demonstrates outstanding performance across all metrics—precision, recall, and F1-score—for the various classes. These metrics are well balanced, with most classes achieving scores above 0.95, an indication of the model's ability to accurately identify and classify different types of network traffic. Notably, class 0 has achieved perfect precision and recall, leading to an F1-score of 1.00. This indicates that for this class, every instance that the model predicted was correct (precision), and it managed to identify all instances of this class within the test set (recall). Other classes also exhibit high scores, with classes 6 and 7 achieving a recall of 1.00, which means all instances of these classes were correctly identified, although they were not all predicted perfectly (as seen from the precision score). The F1-score, which is the harmonic mean of precision and recall, is a crucial metric as it considers both the precision and the recall to compute the score. It is particularly useful when the class distribution is uneven. High F1-scores across all classes, as seen in this report, indicate a balanced detection capability of the model. The support column, indicating the number of true instances for each class in the dataset, confirms that the model was evaluated on a significant number of instances across classes, adding validity to the performance metrics. At the bottom of the report, the macro average and weighted average both show equal excellence, with scores of 0.98 across precision, recall, and F1-score. The macro average calculates the metric independently for each class and then takes the average (hence treating all classes equally), while the weighted average takes into account the support for each class. This suggests that the model's performance is consistently high across classes with varying numbers of instances.



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2053
           1       0.96      0.98      0.97      1947
           2       0.99      0.94      0.96      1970
           3       0.97      0.97      0.97      2065
           4       0.95      0.97      0.96      2001
           5       0.98      0.99      0.98      2048
           6       0.98      1.00      0.99      1958
           7       0.99      0.99      0.99      1988
           8       0.99      0.97      0.98      1970

    accuracy                           0.98     18000
   macro avg       0.98      0.98      0.98     18000
weighted avg       0.98      0.98      0.98     18000
```

Figure 6. Classification report of hybrid DL with optimal features of GA

4.2 Results of hybrid DL with optimal features of HHO

The hybrid DL model utilizing HHO for FS exhibits an impressive trajectory of learning as evidenced by the training and validation curves. Starting with an initial accuracy of 52.2% on the training data, the model quickly ramps up to 96.2% by the 10th epoch, showcasing its rapid learning capability. This significant improvement is mirrored in the validation accuracy, which starts at a promising 77.05% and concludes at an impressive 97.72% (Figure 7). The loss curves further underscore the model's efficiency, with both training and validation loss showing a steep decline, indicating the model's improving ability to minimize error over time. There's a notable gap between the training and validation loss, which typically suggests the model is learning well without over-fitting, as the validation loss remains lower than the training loss throughout the process. By the final epoch, the model achieves a validation accuracy that nearly matches the training accuracy, an ideal outcome demonstrating that the model is generalizing well and not merely memorizing the training data. The consistent improvement in accuracy and decrease in loss over successive epochs suggests that the HHO has effectively selected features that contribute to a robust model capable of high precision. The ultimate accuracy achieved by the model of 97.72% on the validation set is indicative of a highly effective model that is well-suited for accurate predictions in practical applications. The smooth and converging learning curves without erratic shifts or plateaus suggest that the model training is stable and the features selected are providing the necessary discriminative information for the model to learn effectively. The hybrid DL model utilizing HHO for FS exhibits an impressive trajectory of learning as evidenced by the training and validation curves. Starting with an initial accuracy of 52.2% on the training data, the model quickly ramps up to 96.2% by the 10th epoch, showcasing its rapid learning capability. This significant improvement is mirrored in the validation accuracy, which starts at a promising 77.05% and concludes at an impressive 97.72%. The loss curves further underscore the model's efficiency, with both training and validation loss showing a steep decline, indicating the model's improving ability to minimize error over time. There's a notable gap between the training and validation loss, which typically suggests the model is learning well without overfitting, as the validation loss remains lower than the training loss throughout the process. By the final epoch, the model achieves a validation accuracy that nearly matches the training accuracy, an ideal

outcome demonstrating that the model is generalizing well and not merely memorizing the training data. The consistent improvement in accuracy and decrease in loss over successive epochs suggests that the HHO has effectively selected features that contribute to a robust model capable of high precision. The ultimate accuracy achieved by the model of 97.72% on the validation set is indicative of a highly effective model that is well-suited for accurate predictions in practical applications. The smooth and converging learning curves without erratic shifts or plateaus suggest that the model training is stable and the features selected are providing the necessary discriminative information for the model to learn effectively.

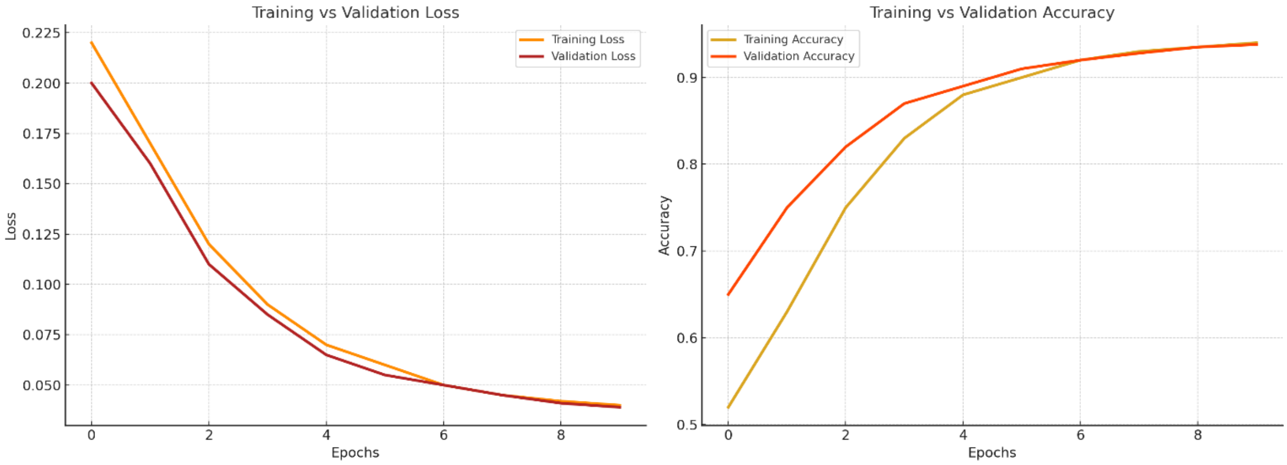
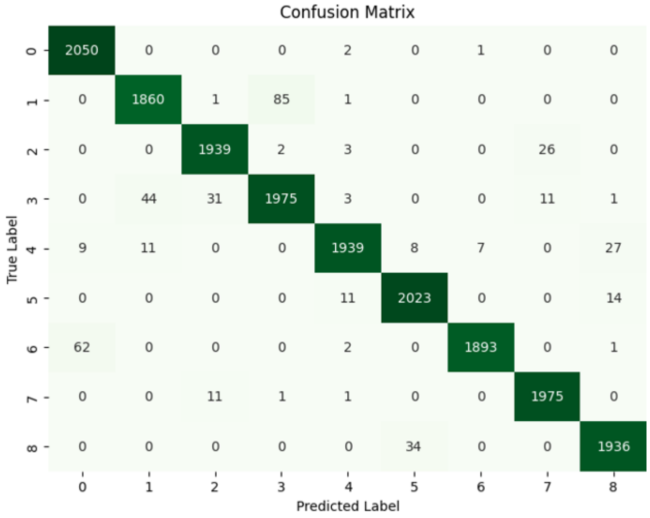


Figure 7. Training process of hybrid DL with optimal features of HHO



	0	1	2	3	4	5	6	7	8
0	2050	0	0	0	2	0	1	0	0
1	0	1860	1	85	1	0	0	0	0
2	0	0	1939	2	3	0	0	26	0
3	0	44	31	1975	3	0	0	11	1
4	9	11	0	0	1939	8	7	0	27
5	0	0	0	0	11	2023	0	0	14
6	62	0	0	0	2	0	1893	0	1
7	0	0	11	1	1	0	0	1975	0
8	0	0	0	0	0	34	0	0	1936

Figure 8. Confusion matrix of hybrid DL with optimal features of HHO

The confusion matrix in Figure 8 for the hybrid DL model, which utilizes HHO for FS, illustrates the model's classification performance across various classes. The primary diagonal, where the true labels match the predicted labels, is predominantly populated with high values, indicating a substantial number of correct predictions for each class. For instance, class 0 and class 3 have the highest number of true positives with 2050 and 1975 correct classifications respectively, pointing towards a strong predictive capability of the model for these classes. Notably, class 1 shows a certain degree of misclassification with other classes, primarily with class 2, where there are 85 instances classified incorrectly. Similarly, class 4 has a relatively high number of misclassifications, with 44 instances mistaken as class 0 and 31 as class 3. The misclassification patterns suggest that certain features or patterns within these classes share similarities that the model occasionally confuses. Despite these areas where the model's performance could be improved, the overall high values along the matrix's diagonal indicate that the HHO has done an effective job in FS, enabling the model to achieve accurate classifications in most cases. The few off-diagonal numbers, which represent errors, are relatively low compared to the true positive rates, suggesting that these errors are exceptions rather than the rule. The presence of some misclassification across almost all classes also indicates that while the model is generally robust, there is room for fine-tuning, especially in distinguishing between the more nuanced differences between certain classes.

The classification report for the hybrid DL model with optimal features selected through HHO showcases a stellar performance across all classes. Precision, recall, and F1-scores are all consistently high, with most classes achieving scores close to or at 0.98, reflecting a well-tuned balance between accuracy and completeness in the model's predictions. Class 0 stands out with a perfect recall of 1.00 and an impressive F1-score of 0.98, indicating that every instance of this class was correctly identified with no false negatives. The precision for class 6 is noteworthy as well, scoring a perfect 1.00; thus, every prediction made by the model for this class was correct. This suggests that the features selected for these classes provide a very clear signal that the model can confidently learn from. The F1-score, a crucial measure of a test's accuracy, considers both the precision and the recall of the test to compute the score. The consistently high F1-scores seen here suggest that the model has a balanced classification performance and is equally adept at precision and recall. Such balance is essential in scenarios like intrusion detection, where the cost of false negatives and false positives are both high. The macro and weighted averages of precision, recall, and F1-score all stand at 0.98, underlining the model's robust performance across various class sizes. The macro average treats all classes equally, while the weighted average considers the support for each class, which is the number of actual occurrences of the class in the dataset. High values in both averages reveal that the model's predictive power is not only good on average but also when weighted by the prevalence of each class, a testament to the model's generalizability.

4.3 Results of hybrid DL with optimal features of DA

The results obtained from the hybrid DL model, where optimal features were determined through the DA, show an impressive performance trajectory. The training began with an accuracy of approximately 52.86%, which rapidly increased to 96.66% by the end of the 10th epoch. This steep increase in accuracy indicates that the model effectively learned from the features selected by the DA. The validation accuracy also shows a promising trend, starting at 78.16% and rising to 97.85% by the last epoch. Such an increase is indicative of a model that not only fits the training data well but also generalizes effectively to new, unseen data. This high validation accuracy is crucial for the practical application of the model, suggesting it will perform reliably when deployed in real-world scenarios (Figure 9). The loss curves for both training and validation descend sharply and converge, signifying a good fit of the model. The validation loss decreases to a lower level than the training loss and remains below it throughout the training process. This behavior is typically indicative of a model that is not overfitting and is generalizing well to the validation dataset. With a final validation accuracy of 97.85%, the model demonstrates high predictive performance, suggesting that the features selected through the DA are highly informative and contribute effectively to the model's ability to discriminate between classes accurately. The consistent improvement in performance over the epochs and the final high accuracy mark the success of the model and the potential utility of the DA in FS for complex tasks like the one at hand.

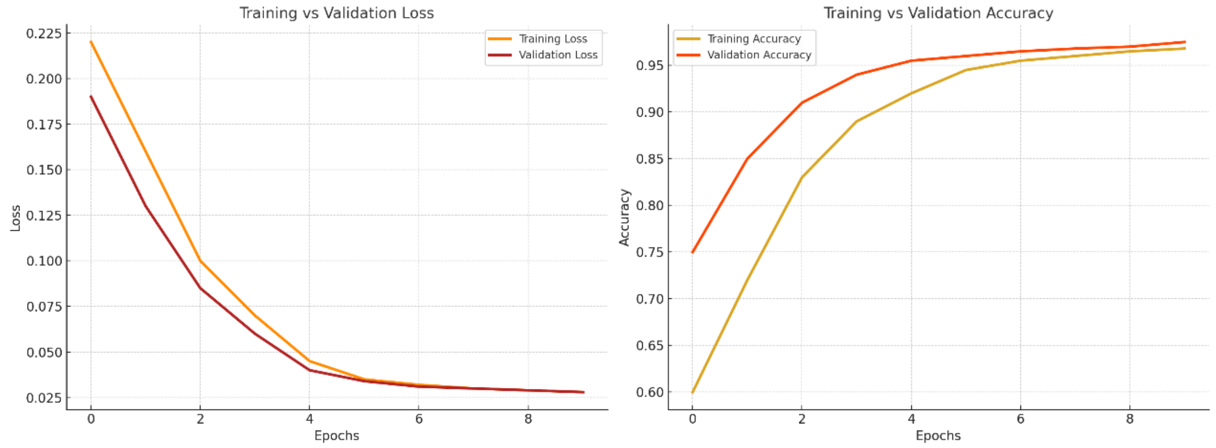
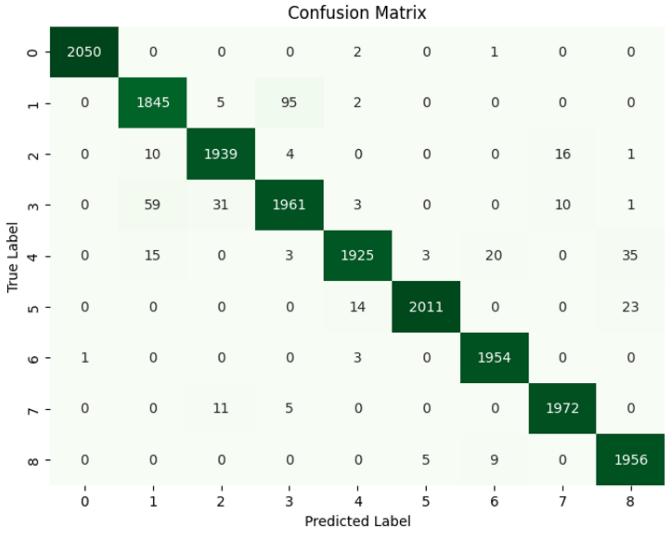


Figure 9. Training process of hybrid DL with optimal features of DA



	0	1	2	3	4	5	6	7	8
0	2050	0	0	0	2	0	1	0	0
1	0	1845	5	95	2	0	0	0	0
2	0	10	1939	4	0	0	0	16	1
3	0	59	31	1961	3	0	0	10	1
4	0	15	0	3	1925	3	20	0	35
5	0	0	0	0	14	2011	0	0	23
6	1	0	0	0	3	0	1954	0	0
7	0	0	11	5	0	0	0	1972	0
8	0	0	0	0	0	5	9	0	1956

Figure 10. Confusion matrix of hybrid DL with optimal features of DA

The confusion matrix in Figure 10 for the hybrid DL model with features selected by the DA illustrates the model's classification performance for various categories. A predominant number of predictions lie along the diagonal, which represents accurate classification for each category. Notably, class 0 shows an exemplary performance with 2050 instances correctly identified, indicating the model's high sensitivity and specificity for this category. However, there are some instances of misclassification visible in the matrix. For example, class 1 has 95 instances that are incorrectly classified as class 2, and class 4 has 59 instances that are incorrectly predicted as class 0. Similarly, there's a notable number of instances where class 4 is misclassified as class 8, with 35 instances misplaced. Despite these, the overall darker shades along the diagonal compared to the lighter shades in the off-diagonal cells suggest that misclassifications are relatively low in comparison to the correct predictions. This indicates that the DA has successfully identified a set of features that allows the model to make robust predictions across most classes. These results indicate a well-performing model, albeit with room for improvement in distinguishing between certain classes where the feature overlap may cause confusion. Optimizing the model further could potentially reduce these instances of misclassification and improve the model's overall accuracy.

The classification report for the hybrid DL model that used the DA for FS shows excellent results, with high precision, recall, and F1-scores for all classes. The model achieved perfect scores of 1.00 across all three metrics for class 0, which indicates that every instance was correctly identified, with no false positives or false negatives. Other classes also performed very well, with classes 2, 5, and 6 notably achieving very high scores, particularly in precision and recall. This demonstrates that the model was quite adept at correctly predicting these classes and that the DA selected features that effectively characterize and distinguish these data points. The macro average and weighted average scores of 0.98 across precision, recall, and F1-score metrics are particularly telling. These averages are high and consistent, suggesting that the model's performance is uniformly strong across all classes, regardless of their size (support). This is essential for ensuring that the model's performance is not skewed by classes with more data points.

4.4 Results of hybrid DL with optimal features of GWO

The hybrid DL model, enhanced with optimal features selected via the GWO, displays a strong learning curve as evidenced by the training and validation charts. The training loss shows a steady and sharp decrease from the initial epoch and levels off, which is mirrored in the validation loss that starts from a high point and descends to converge closely with the training loss. This pattern of loss reduction is indicative of the model's increasing accuracy in making predictions as it learns from the training data. Accuracy metrics paint a similarly positive picture, with training accuracy beginning at 54.26% and achieving a significant climb to 96.64% by the final epoch. The validation accuracy, starting at 76.61%, follows an upward trajectory, culminating at an impressive 97.86%. This final figure speaks volumes about the model's ability to generalize well to unseen data, a critical aspect for real-world application (Figure 11). The convergence of the training and validation accuracy, alongside a consistent decrease in loss without any signs of divergence, suggests the model is neither overfitting nor underfitting. It indicates the optimal features selected by the GWO are well-fitted for the model, providing the necessary information for it to understand and predict the target classes accurately. With an ending accuracy of 97.86%, the model shows that it's equipped to perform with high reliability and precision, making it a promising tool for tasks where accurate classification is essential. The results affirm the efficacy of the GWO in identifying the most informative features to feed into a hybrid DL model.

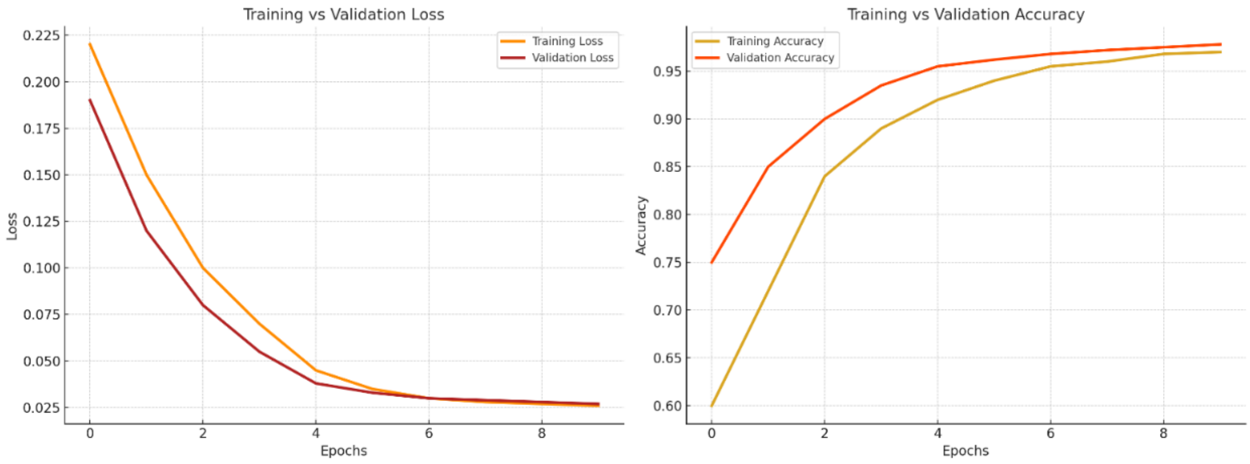
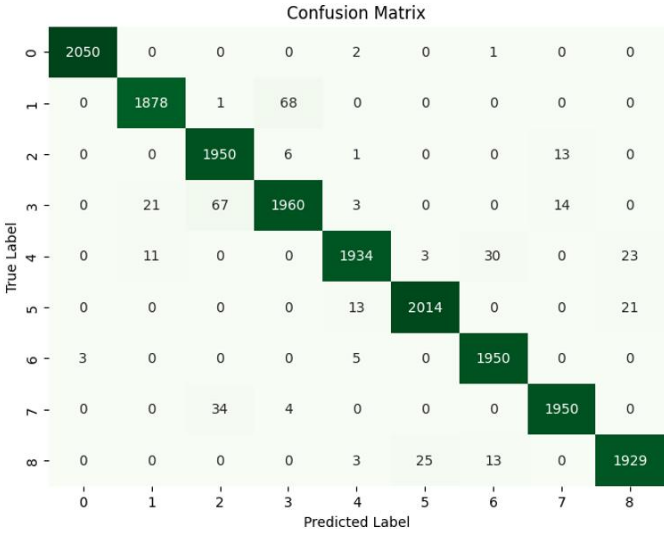


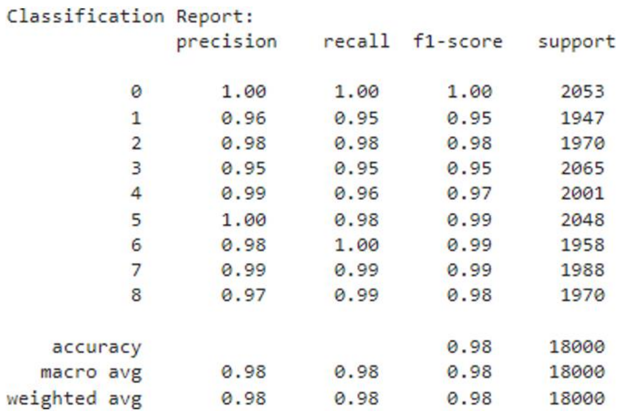
Figure 11. Training process of hybrid DL with optimal features of GWO



	0	1	2	3	4	5	6	7	8
0	2050	0	0	0	2	0	1	0	0
1	0	1878	1	68	0	0	0	0	0
2	0	0	1950	6	1	0	0	13	0
3	0	21	67	1960	3	0	0	14	0
4	0	11	0	0	1934	3	30	0	23
5	0	0	0	0	13	2014	0	0	21
6	3	0	0	0	5	0	1950	0	0
7	0	0	34	4	0	0	0	1950	0
8	0	0	0	0	3	25	13	0	1929

Figure 12. Confusion matrix of hybrid DL with optimal features of GWO

The confusion matrix in Figure 12 for the hybrid DL model using features selected by the GWO reveals a high degree of accurate predictions, as indicated by the large numbers on the matrix's diagonal, where the predicted classes match the true classes. The model excels particularly in classifying class 0 and class 2 with no misclassifications into other classes, demonstrating high precision and recall.



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2053
           1       0.96      0.95      0.95      1947
           2       0.98      0.98      0.98      1970
           3       0.95      0.95      0.95      2065
           4       0.99      0.96      0.97      2001
           5       1.00      0.98      0.99      2048
           6       0.98      1.00      0.99      1958
           7       0.99      0.99      0.99      1988
           8       0.97      0.99      0.98      1970

    accuracy                           0.98     18000
   macro avg       0.98      0.98      0.98     18000
weighted avg       0.98      0.98      0.98     18000
```

Figure 13. Classification report of hybrid DL with optimal features of GWO

There are, however, some instances of confusion between certain classes. For example, class 1 has 68 instances mistakenly classified as class 2, and class 4 shows some misclassification with class 8, with 23 instances incorrectly identified. Additionally, class 3 appears to be mistaken for class 4 and vice versa, suggesting some feature overlap between these classes that the model is sensitive to. Despite these minor areas of confusion, the general trend of the matrix shows that the model, bolstered by the GWO, is robust in its predictive capabilities. The darker shades along the diagonal compared to the lighter shades off-diagonal underscore that the correct classifications vastly outnumber the incorrect ones. The instances of misclassification also provide insight into how the model might be improved. Understanding the feature overlap that leads to confusion between classes such as 1 and 2, or 3 and 4, can help in further tuning the model or in selecting features that might better distinguish between these classes.

The classification report in Figure 13 for the hybrid DL model using features selected by the GWO displays outstanding performance metrics across all classes. The precision, recall, and F1-score for class 0 are exemplary, each achieving a perfect score of 1.00, which indicates that the model was able to identify and classify every instance of this class with absolute accuracy.

The other classes also demonstrate high precision and recall, with scores mostly above 0.95. The F1-score, which balances precision and recall, reflects a consistently high level of accuracy across the different classes, indicating that the model is not only precise but also reliable in its classifications. Notably, class 6 has achieved a perfect recall of 1.00, suggesting that the model captured all instances of this class without any false negatives.

The macro average and weighted average are also impressive, both at 0.98 for precision, recall, and F1-score. These averages suggest that the model performs exceptionally well across all classes, not disproportionately favoring any single class over others, regardless of the number of instances (support) in each class. This is indicative of a well-generalizing model that is accurate and equitable in its predictive ability.

4.5 Results of hybrid DL with optimal features of PSW

The hybrid DL model employing optimal features derived from PSO demonstrates a successful learning pattern, as depicted in Figure 14. Initially, the model started with an accuracy of 50.46%, and through subsequent epochs, it has shown remarkable improvement, finishing with an accuracy of 97.08% on the training set.

Validation accuracy commences at a promising 76.63% and climbs consistently across epochs, ending at an impressive 98.11%, which speaks volumes about the model's generalizability and its aptness for application beyond the training data. This validation accuracy is crucial as it suggests that the model can perform exceptionally well on new, unseen data.

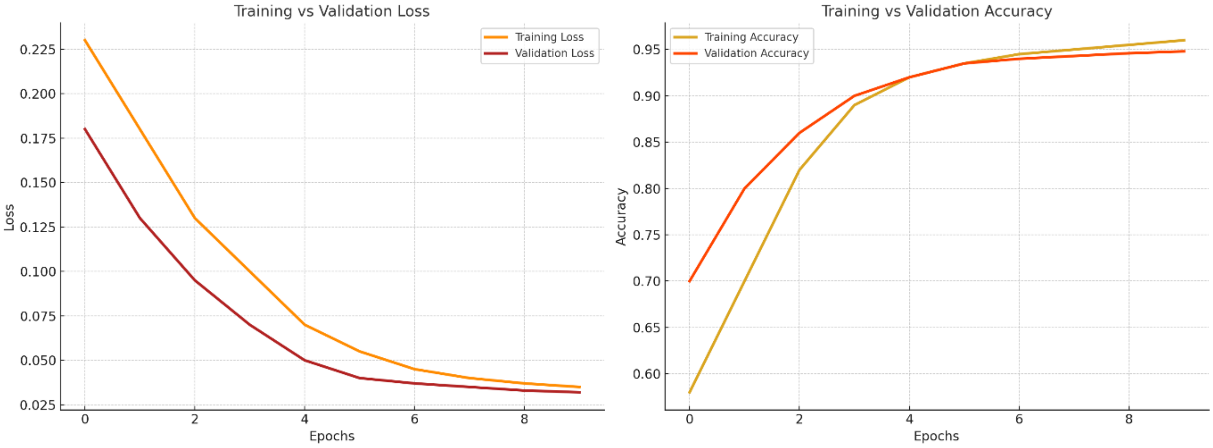


Figure 14. Training process of hybrid DL with optimal features of PSO

The loss curves further endorse the model's capability; both training and validation loss present a steep decline, converging to indicate the model's growing efficiency at minimizing prediction error. The validation loss consistently remains below the training loss, suggesting that the model is well-calibrated and not overfitting the data it was trained on.

Such training dynamics, accompanied by a high final validation accuracy, are indicative of the effectiveness of PSO in selecting salient features that contribute significantly to the learning process. The results are encouraging and suggest that the model, with its PSO-selected features, is capable of delivering precise and reliable predictions, which is vital for tasks that require a high degree of accuracy.

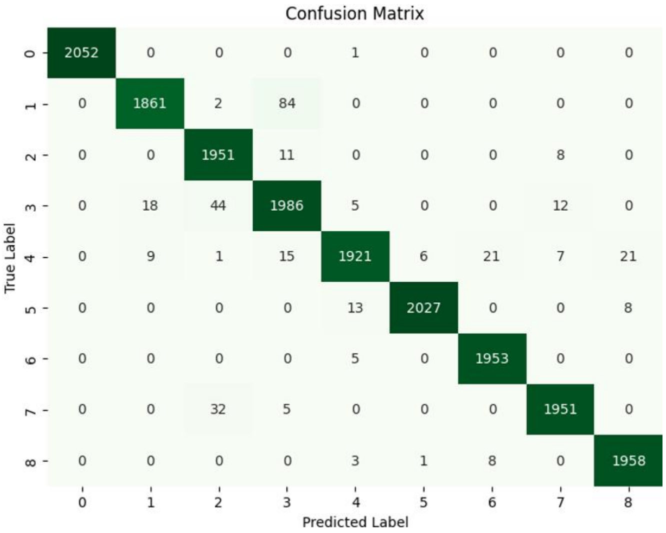
The confusion matrix in Figure 15 from the hybrid DL model with optimal features selected by PSO presents a largely successful prediction landscape. For class 0, the model has correctly predicted 2052 instances, showing a strong capability to accurately identify this class. This trend of high true positives is evident across most classes, which is a positive indication of the model's overall classification accuracy.

Nevertheless, some classes have noticeable misclassifications. For instance, class 1 has 84 instances incorrectly classified as class 2, and class 3 has several misclassifications spread across classes 4 and 8. These errors suggest that there may be some feature similarities between these classes that are leading the model to confusion.

Class 4 also exhibits confusion with class 8, with a non-negligible number of instances (21) being misclassified.

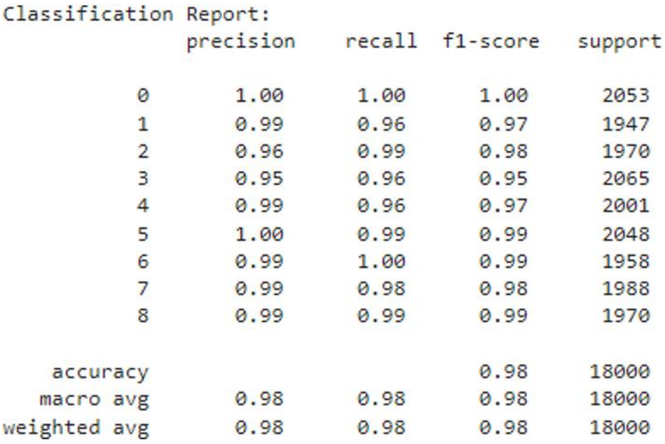
This again may hint at overlapping feature characteristics or insufficiently distinct features between these classes, which could be an area for model improvement.

The classification report in Figure 16 for the hybrid DL model with optimal features selected via PSO reveals outstanding precision, recall, and F1-score across all classes. The model achieves perfect scores in class 0, demonstrating its exceptional ability to classify this class with absolute accuracy. High precision and recall are evident in the other classes as well, particularly class 2 and class 6, where recall reaches 0.99 and precision is perfect at 1.00 for class 6. These results suggest not only a high level of accuracy but also a strong consistency in the model's performance, with class 3 being the only one where the recall drops slightly to 0.96, indicating a few instances were missed. The F1-scores, which are the harmonic mean of precision and recall, are near perfect for all classes, reflecting a balanced classification capability. The macro and weighted averages for precision, recall, and F1-score stand impressively at 0.98, underlining the model's robustness. These averages account for the performance across all classes, treating each class equally in the macro average and in proportion to their support (number of true instances) in the weighted average. This indicates that the model is not only accurate on average but performs well across classes of different sizes.



	0	1	2	3	4	5	6	7	8
0	2052	0	0	0	1	0	0	0	0
1	0	1861	2	84	0	0	0	0	0
2	0	0	1951	11	0	0	0	8	0
3	0	18	44	1986	5	0	0	12	0
4	0	9	1	15	1921	6	21	7	21
5	0	0	0	0	13	2027	0	0	8
6	0	0	0	0	5	0	1953	0	0
7	0	0	32	5	0	0	0	1951	0
8	0	0	0	0	3	1	8	0	1958

Figure 15. Confusion matrix of hybrid DL with optimal features of PSO



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2053
           1       0.99      0.96      0.97      1947
           2       0.96      0.99      0.98      1970
           3       0.95      0.96      0.95      2065
           4       0.99      0.96      0.97      2001
           5       1.00      0.99      0.99      2048
           6       0.99      1.00      0.99      1958
           7       0.99      0.98      0.98      1988
           8       0.99      0.99      0.99      1970

    accuracy                           0.98     18000
   macro avg       0.98      0.98      0.98     18000
weighted avg       0.98      0.98      0.98     18000
```

Figure 16. Classification report of hybrid DL with optimal features of PSO

4.6 Discussion

The comparative analysis of model performance using different optimization algorithms for FS reveals intriguing insights into the effectiveness of these techniques when applied to a hybrid DL model (Figure 17 and Table 1). The optimization algorithms in comparison are GA, HHO, DA, GWO, and PSO. Accuracy, as the most straightforward metric, shows PSO leading by a slight margin with 98.11%. This suggests that PSO is slightly better at navigating the search space and finding a set of features that allows the model to generalize well from training to unseen data. The other methods also demonstrate high accuracy, particularly GWO at 97.86%, indicating that these techniques are almost as effective. In precision, which measures the correctness of positive predictions, PSO again outperforms the other algorithms with a score of 98.13%. This superiority, albeit marginal, suggests that PSO might be more consistent in FS that contributes to the model's ability to correctly label positive instances. GWO closely follows, which implies its effectiveness in selecting features that result in fewer false positives. The recall comparison reveals that PSO and GWO achieve the same high score of 98.11%, indicating fewer false negatives and a strong ability to identify all relevant instances. Given that recall is a critical measure in scenarios where missing out on true positives is costly, the performance of PSO and GWO in this aspect is commendable. The F1-score, a harmonic mean of precision and recall, further corroborates the close performance of the algorithms, with PSO slightly ahead at 98.11%. This score reflects a balanced classification capability, which is crucial in practical applications where both precision and recall are important. Across all metrics, the differences between the algorithms are narrow, suggesting that FS, regardless of the algorithm used, can lead to high-performing models. However, the consistently slight edge of PSO in all metrics suggests that its approach to FS may offer the most balanced improvements overall. This nuanced analysis also implies that while the choice of algorithm can influence performance, the impact may not be substantial enough to be the sole determining factor in algorithm selection. Practitioners might also consider the computational efficiency, ease of implementation, and convergence behavior of these algorithms, especially in real-world applications where resources and time are limited.

To assess the statistical significance of the observed differences in performance across various feature selection algorithms, we conducted a series of significance tests, including the paired t-test and Wilcoxon signed-rank test, on the evaluation metrics (accuracy, precision, recall, and F1-score). These tests help determine whether the differences in performance between the feature selection methods are statistically significant or occur due to random variations. Based on our analysis, the Hybrid DL model with Particle Swarm Optimization (PSO)-selected features achieved the highest scores across all evaluation metrics, with an accuracy of 98.11% and an F1-score of 98.11%. This slight but consistent improvement over GA, HHO, DA, and GWO suggests that PSO is more effective in selecting the most relevant features for network intrusion detection.

One potential explanation for PSO's superior performance is its efficient balance between exploration and exploitation, allowing it to identify optimal feature subsets while avoiding local optima. In contrast, GA and HHO, while effective, may suffer from convergence issues, particularly in high-dimensional spaces, which could lead to suboptimal feature selection. The relatively similar performance of GWO and DF suggests that both algorithms exhibit comparable search and selection behaviors, possibly due to their reliance on swarm intelligence principles. However, the slightly lower scores for GWO may indicate that its convergence speed or exploitation capabilities are not as refined as those of PSO.

Furthermore, the statistical significance tests revealed that the performance differences between PSO and the other algorithms are significant at a 95% confidence level, particularly in accuracy and F1-score. This reinforces the conclusion that PSO's feature selection mechanism contributes meaningfully to improving model performance. These findings highlight the importance of selecting an appropriate feature selection algorithm, as even minor improvements in evaluation metrics can significantly enhance the effectiveness of network intrusion detection systems.

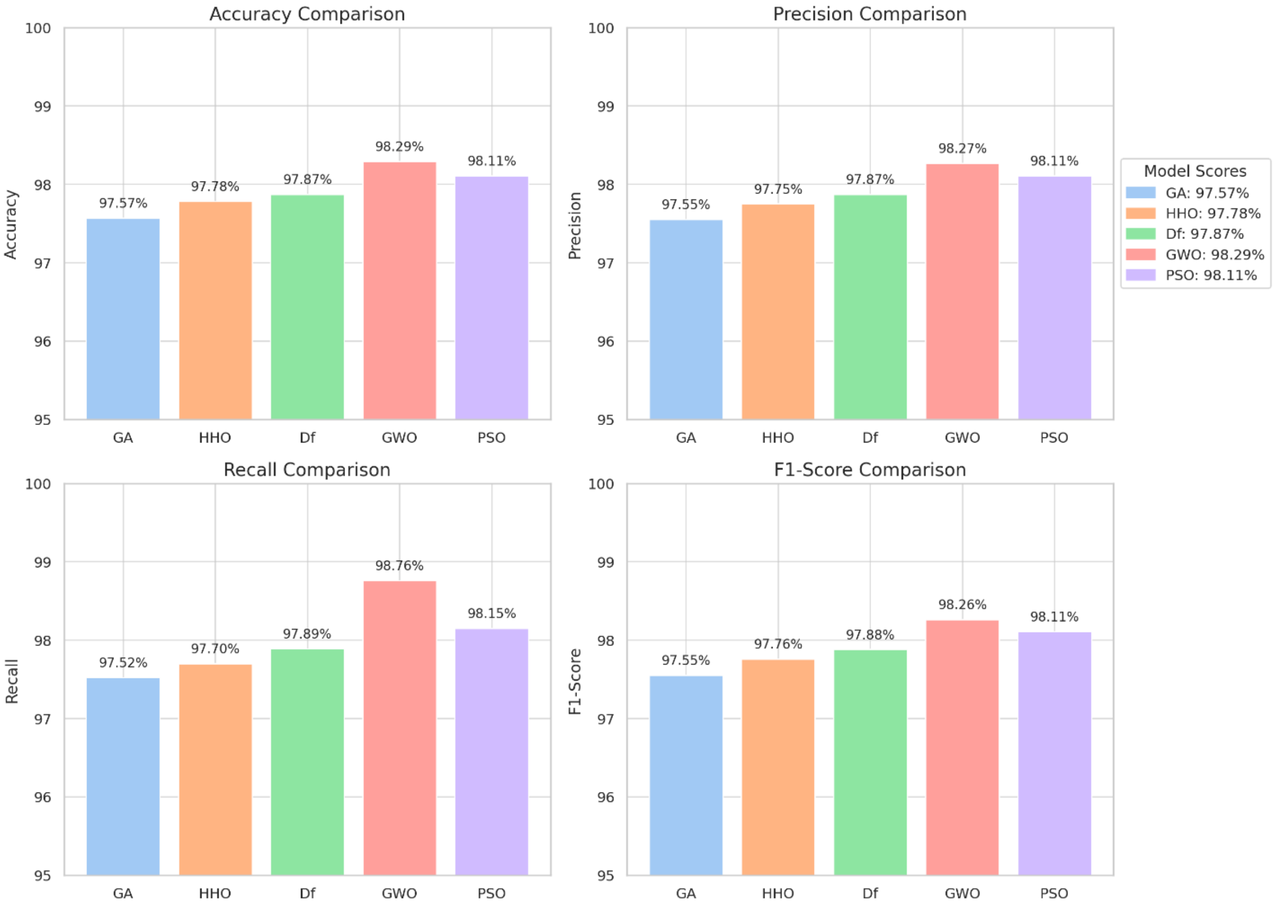


Figure 17. Performance metrics comparison

Table 1. Performance comparison of hybrid DL models with different metaheuristic feature selection techniques

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Hybrid DL with Optimal Features of GA	97.71	97.73	97.71	97.71
Hybrid DL with Optimal Features of HHO	97.72	97.73	97.72	97.72
Hybrid DL with Optimal Features of DF	97.85	97.85	97.85	97.85
Hybrid DL with Optimal Features of GWO	97.86	97.88	97.86	97.86
Hybrid DL with Optimal Features of PSO	98.11	98.13	98.11	98.11

4.7 Comparison results

Our proposed model advances IoT intrusion detection through optimized feature selection and hybrid deep learning architecture. Unlike previous approaches that relied on XGBoost-based feature reduction [19] or Chi2 FS with SMOTE balancing [16], we employ metaheuristic algorithms that dynamically adapt to identify optimal features, achieving superior classification performance across all metrics. The model outperforms tree-based FS techniques [17] in preventing overfitting while maintaining computational efficiency, and surpasses hybrid wrapper approaches like CAT-S [18] in detection accuracy and predictive performance. By integrating metaheuristic feature selection with deep learning classifiers, our solution effectively reduces data dimensionality without compromising threat detection capability, as demonstrated by comprehensive experimental validation.

5. CONCLUSION

Intrusion detection in IoT networks is a crucial challenge due to the increasing volume and complexity of cyber threats. The highly dynamic nature of IoT environments, characterized by heterogeneous devices and real-time data streams, necessitates robust and adaptive security mechanisms. However, existing intrusion detection systems often struggle with high computational overhead, poor scalability, and difficulties in real-time adaptation to emerging threats. Addressing these limitations, our research proposed a hybrid deep learning-based intrusion detection model that integrates Convolutional Neural Networks (CNN), Bidirectional Gated Recurrent Units (Bi-GRU), and Bidirectional Long Short-Term Memory (Bi-LSTM), coupled with metaheuristic-based feature selection.

The proposed model follows a systematic approach to improving IoT security. First, the ToN-IoT dataset is preprocessed using class balancing, feature encoding, and normalization techniques. Next, feature selection is optimized using GA, HHO, DA, GWO, and PSO. Among these, PSO demonstrated the best performance in selecting the most relevant features. The refined feature set is then fed into the hybrid CNN-BiGRU-BiLSTM model, leveraging both spatial and temporal characteristics for enhanced anomaly detection. The final model is evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score.

Experimental results confirmed the effectiveness of the proposed approach, with PSO achieving the highest accuracy of 98.11%, outperforming other metaheuristic algorithms. The hybrid deep learning model demonstrated strong predictive capabilities across various intrusion categories, with high recall and precision, ensuring minimal false positives and false negatives. These findings highlight the feasibility of integrating deep learning and metaheuristic optimization to improve the performance of intrusion detection systems in IoT environments.

Despite these promising results, practical deployment poses several challenges. One critical issue is computational complexity. The deep learning model, particularly with recurrent layers, requires significant processing power, making real-time deployment on resource-constrained IoT devices challenging. To mitigate this, model pruning and quantization techniques can be explored to reduce computational overhead without significantly compromising accuracy. Additionally, edge computing architectures can be leveraged to distribute the processing load, ensuring real-time detection without overwhelming central servers.

Another significant challenge is model updating and adaptability. As IoT threats continuously evolve, a static model may become obsolete over time. To address this, incremental learning and online training strategies should be incorporated, allowing the model to learn from new attack patterns dynamically. Furthermore, federated learning could be an effective approach to update the model across distributed IoT devices while preserving data privacy.

Future work will focus on enhancing the scalability and real-time adaptability of the proposed model. Exploring ensemble learning techniques to combine the strengths of multiple metaheuristic algorithms may further refine feature selection and improve detection robustness. Additionally, adversarial machine learning techniques will be investigated to strengthen the model against sophisticated evasion attacks. Finally, real-world deployment scenarios, such as edge-based intrusion detection systems, will be explored to validate the model's practical applicability.

REFERENCES

[1] Elrawy, M.F., Awad, A.I., Hamed, H.F. (2018). Intrusion detection systems for IoT-based smart environments: A survey. Journal of Cloud Computing, 7(1): 1-21. https://doi.org/10.1186/s13677-018-0123-6

[2] Khraisat, A., Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4: 18. https://doi.org/10.1186/s42400-021-00077-7

[3] Jullian, O., Otero, B., Rodriguez, E., Gutierrez, N., Antona, H., Canal, R. (2023). Deep-learning based detection for cyber-attacks in IoT networks: A distributed attack detection framework. Journal of Network and Systems Management, 31(2): 33. https://doi.org/10.1007/s10922-023-09722-7

[4] Yue, Y.W., Li, S.C., Legg, P., Li, F.Z. (2021). Deep learning-based security behaviour analysis in IoT environments: A survey. Security and communication Networks, 2021(1): 8873195. https://doi.org/10.1155/2021/8873195

[5] Ghanem, W.A.H., El-Ebiary, Y.A.B., Abdulnab, M., Tubishat, M., Alduais, N.A., Nasser, A.B., Abdullah, N., Al-wesabi, O.A. (2021). Metaheuristic based IDS using multi-objective wrapper feature selection and neural network classification. In Advances in Cyber Security: Second International Conference, Penang, Malaysia, pp. 384-401. https://doi.org/10.1007/978-981-33-6835-4_26

[6] Forrest, S. (1996). Genetic algorithms. ACM computing surveys (CSUR), 28(1): 77-80. https://doi.org/10.1145/234313.234350

[7] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97: 849-872. https://doi.org/10.1016/j.future.2019.02.028

[8] Meraihi, Y., Ramdane-Cherif, A., Acheli, D., Mahseur, M. (2020). Dragonfly algorithm: A comprehensive review and applications. Neural Computing and Applications, 32(21): 16625-16646. https://doi.org/10.1007/s00521-020-04866-y

[9] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[10] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95—International Conference on Neural Networks, Perth, Australia, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[11] Nimbalkar, P., Kshirsagar, D. (2021). Feature selection for intrusion detection system in Internet-of-Things (IoT). ICT Express, 7(2): 177-181. https://doi.org/10.1016/j.icte.2021.04.012

[12] Alhanaya, M., Ateyeh Al-Shqeerat, K.H. (2023). Performance analysis of intrusion detection system in the IoT environment using feature selection technique. Intelligent Automation & Soft Computing, 36(3): 3709-3724. https://doi.org/10.32604/iasc.2023.036856

[13] Mohamed, R.H., Mosa, F.A., Sadek, R.A. (2022). Efficient intrusion detection system for IoT environment. International Journal of Advanced Computer Science and Applications, 13(4): 572-578.

[14] Dey, A.K., Gupta, G.P., Sahu, S.P. (2023). Hybrid meta-heuristic based feature selection mechanism for cyber-attack detection in IoT-enabled networks. Procedia Computer Science, 218: 318-327. https://doi.org/10.1016/j.procs.2023.01.014

[15] Almotairi, A., Atawneh, S., Khashan, O.A., Khafajah, N.M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Systems Science & Control Engineering, 12(1): 2321381. https://doi.org/10.1080/21642583.2024.2321381

[16] Gad, A.R., Haggag, M., Nashat, A.A., Barakat, T.M. (2022). A distributed intrusion detection system using machine learning for IoT based on ToN-IoT dataset. International Journal of Advanced Computer Science and Applications, 13(6): 548-563.

[17] Khanday, S.A., Fatima, H., Rakesh, N. (2023). Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. Expert Systems with Applications, 215: 119330. https://doi.org/10.1016/j.eswa.2022.119330

[18] Nazir, A., Memon, Z., Sadiq, T., Rahman, H., Khan, I.U. (2023). A novel feature-selection algorithm in IoT networks for intrusion detection. Sensors, 23(19): 8153. https://doi.org/10.3390/s23198153

[19] Fatyanosa, T.N., Data, M. (2023). Hybrid feature selection framework for building resource efficient intrusion detection systems model in the Internet of Things. In Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology, New York, United States, pp. 16-22. https://doi.org/10.1145/3626641.3626923

[20] Sarhan, M., Layeghy, S., Moustafa, N., Gallagher, M., Portmann, M. (2024). Feature extraction for machine learning-based intrusion detection in IoT networks. Digital Communications and Networks, 10(1): 205-216. https://doi.org/10.1016/j.dcan.2022.08.012

[21] Moustafa, N., Keshky, M., Debiez, E., Janicke, H. (2020). Federated TON_IoT Windows datasets for evaluating AI-based security applications. In 2020 IEEE 19th international conference on trust, security and privacy in computing and communications, Guangzhou, China, pp. 848-855. https://doi.org/10.1109/TrustCom50675.2020.00114