computers

MDPI

Article

Enhancing IoT Network Security: A BPSO-Optimized Attention-GRU Deep Learning Framework for Intrusion Detection

Abdallah Elayan and Michel Kadoch *

Department of Electrical Engineering, École de Technologie Supérieure, Université du Québec, Montreal, QC H3C 1K3, Canada; abdallah.elayan.1@ens.etsmtl.ca
* Correspondence: michel.kadoch@etsmtl.ca

Abstract

The exponential expansion of computer networks, alongside the rapid development of the Internet of Things (IoT), has significantly increased the volume and complexity of transmitted data, emphasizing the need for robust network security measures to secure sensitive data and prevent unauthorized access or breaches. Intrusion Detection Systems (IDSs) have emerged as a vital tool for protecting networks and IoT environments from threats. Various IDSs have been proposed in the literature; however, the lack of optimal feature learning, computational efficiency, and reliance on obsolete datasets poses significant challenges, limiting their effectiveness against evolving cyber threats. Moreover, traditional IDSs struggle to efficiently manage the high-dimensional and imbalanced nature of IoT network traffic data. To address these challenges, this research proposes a hybrid deep learning (DL)-based IDS integrating Binary Particle Swarm Optimization (BPSO), MultiHead Attention mechanisms (MHA), and a deep Gated Recurrent Unit (GRU) architecture, improving detection effectiveness while reducing computational overhead. Our proposed approach also utilizes a Target Sampling strategy to balance class distributions, enhancing the model's ability to accurately identify minority attacks. The BPSO algorithm is employed to identify the most influential features from the high-dimensional network traffic datasets, enhancing model interpretability and supporting more efficient learning. This optimized feature subset is then fed into a GRU-based DL architecture augmented with MHA, which performs sequence processing and attention-based learning for intrusion detection. The performance of the proposed model is evaluated utilizing the BoT-IoT and the CIC-IDS2017 benchmark datasets, ensuring a comprehensive assessment of anomaly detection capabilities. Extensive experimental results demonstrate the superior performance of the proposed model, achieving a recall of 98.42% and 99.76%, with F1-score of 98.94% and 99.76% for binary classification and a recall of 99.79% and 98.69%, with F1-score of 99.89% and 98.04% for multiclass classification on the BoT-IoT and CIC-IDS2017 datasets, respectively, highlighting the effectiveness of our model in enhancing threat detection for computer networks and IoT environments in comparison to recent state-of-the-art IDSs.

Keywords: binary particle swarm optimization (BPSO); deep learning (DL); multihead attention (MHA); intrusion detection system (IDS); gated recurrent unit (GRU)

Check for updates

Academic Editor: Yazan Otoum

Received: 14 March 2026
Revised: 19 April 2026
Accepted: 21 April 2026
Published: 23 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Internet of Things (IoT) refers to an interconnected ecosystem of physical objects, such as household appliances, autonomous vehicles, wearable gadgets, and industrial

machines. These devices communicate and exchange data over the Internet through embedded software and advanced technologies. This rapidly evolving paradigm has enabled various innovative applications, including smart homes, self-driving cars, and healthcare applications. The IoT expansion brings it to the top of technological innovation and makes it an integral component of the future Internet. Despite their numerous benefits, IoT devices often suffer from inherent security vulnerabilities due to limited computational resources, insufficient built-in security, and large-scale dynamic network configurations, requiring the development of efficient network security solutions to ensure data integrity and prevent cyber threats [1].

In the last decade, IoT network security has emerged as a critical area of focus, driven by the rising complexity of communication technologies and the increasing frequency of cyber threats. The heterogeneous nature of interconnected devices, diverse communication protocols, and the enormous volume of generated data traffic exacerbate these security challenges. According to the 2019 Cyberthreat Report by the CyberEdge Group, cyberattacks have increased at an alarming rate, posing serious threats to data confidentiality, integrity, authentication, and availability [2,3]. Traditional security mechanisms, such as firewalls and antivirus software, primarily focus on monitoring network traffic to identify potential threats. These tools rely on predefined rules to detect malicious activity. However, the dynamic and evolving nature of cyberattacks enables new threats to often get around these conventional security measures, highlighting the need for intelligent detection systems.

To enhance network security, IDSs have emerged as a critical component in network security, actively monitoring and analyzing network behavior to detect potential intrusions [4]. The intrusion detection techniques are generally classified into two categories based on their detection procedures: signature-based and anomaly-based. Signature-based approaches rely on predefined patterns or signatures of known attacks acquired from network traffic analysis, security vendors, or other relevant data sources to identify intrusions. While highly effective in accurately detecting known attacks, this approach encounters limitations in detecting unknown threats and requires continual updates to its database. On the other hand, anomaly-based IDSs utilize statistical analysis and machine learning (ML) techniques to establish a baseline of normal network behavior, identifying deviations as potential intrusions. Despite anomaly-based IDS being capable of detecting previously unseen attacks, it often suffers from higher false alarm rates compared to signature-based techniques [5,6].

In recent years, artificial intelligence (AI) has revolutionized various fields in our modern lives, including natural language processing, healthcare, image recognition, and cybersecurity. Within IDSs, ML and DL techniques have significantly improved intrusion detection capabilities. Traditional ML models commonly used in IDSs, including Decision Trees, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machines (SVM), lack the capacity for hierarchical feature learning from raw data and require feature engineering, limiting their implementation adaptability and ability to generalize to unknown attack patterns. In contrast, DL-based models, including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Autoencoders, Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN), attracted researchers' attention due to their capabilities in processing and learning complex representations of data. Unlike ML approaches, DL models leverage multilayered artificial neural networks (ANNs) to extract complex features from raw data, improving the detection of both known and unknown threats. The hierarchical structure of DL models allows them to process complex attack patterns more effectively, making them particularly valuable for securing computer networks and IoT environments [6,7].

Despite the promising advantages achieved through employing DL-based models in IDSs, various challenges continue to constrain their practical effectiveness in computer networks and IoT environments. First, the extensive volumes of generated data by heterogeneous IoT devices contain significant noise, missing and uncertain data, creating a highly complex environment for network security. Therefore, robust preprocessing techniques are necessary to ensure accurate threat detection. Second, network datasets are typically high-dimensional and contain several redundant and irrelevant features, which increase computational overhead and can mask critical patterns. Effective feature selection and dimensionality reduction, such as BPSO, are essential for identifying relevant features, reducing training time, and enhancing detection capabilities. Third, severe class imbalance remains a challenge in IDS, leading to weak detection performance for critical underrepresented classes. A robust sampling strategy, therefore, is important for improving minority classes detection. Additionally, many existing IDS models have been developed utilizing insufficient obsolete datasets, limiting the model's relevance and generalization against evolving attacks. Therefore, comprehensive evaluation on diverse modern datasets is essential for developing a robust IDS to demonstrate robust effectiveness against modern threats. Finally, the evolving and dynamic nature of cyberattacks presents a continuous challenge to existing security techniques as traditional approaches struggle to adapt to the rising threats, highlighting the need for robust IDS architectures capable of learning complex intrusion patterns from network traffic while dynamically focusing on the most critical informative features for identifying complex intrusions from benign behaviors.

In light of these challenges, this research introduces a hybrid DL-based IDS that effectively integrates BPSO for feature selection, a Target Sampling strategy for class imbalance mitigation, and an MHA-stacked GRU architecture for sequence processing. The BPSO-based feature selection generates a compact and informative feature subset from network traffic, while Target Sampling selectively downscales majority instances without synthetic data generation, therefore preserving minority class representation. This optimization feeds into the MHA-GRU classifier for effective intrusion detection, which performs sequence processing and captures significant attack patterns in network traffic.

The primary objectives of this research are to develop a robust DL-based IDS framework utilizing feature optimization, intelligent sampling, and advanced classification for superior detection accuracy and efficiency; to design a preprocessing procedure to handle the complexity of heterogeneous imbalanced networks and IoT data; and to comprehensively evaluate the proposed model using modern benchmark datasets to validate its generalizability.

The key contributions of this paper are:

- A hybrid BPSO-MHA-GRU IDS framework that integrates optimal feature selection into an attention-stacked GRU architecture, embedding influential feature subsets within an attention mechanism enabling more selective representation of network traffic.
- An MHA-stacked GRU architecture that performs sequence processing and dynamically weights critical features, enabling robust distinction between complex intrusions and benign behavior.
- A Target Sampling strategy for handling severe class imbalance by selectively downsampling majority classes while preserving all minority records to avoid artificial data distortions and improve minority class sensitivity and classification stability.
- A BPSO-based feature selection mechanism that efficiently identifies non-redundant, informative features from high-dimensional network traffic, thereby reducing input dimensionality and training computational cost while enhancing model detection performance.

The rest of this paper is structured as follows: Section 2 provides a brief background on IDSs and DL architectures. Section 3 presents a review of key developments in IDS research, summarizing recent advancements in ML and DL-based intrusion detection approaches over the last few years. Section 4 details the proposed IDS model, while Section 5 discusses experimental results and comparative evaluations with benchmark models. Finally, Sections 6 and 7 provide a discussion and conclude this research paper by summarizing key findings and outlining future research directions.

2. Background

This section presents an overview of IDS architecture and the role of DL in enhancing network security.

Intrusion Detection System Architecture and Deep Learning

IDSs are essential components of modern cybersecurity frameworks, designed to monitor network traffic and system activities to detect and mitigate malicious activities. Intrusions refer to unauthorized attempts that compromise the confidentiality, integrity, or availability of a network or system. IDSs can be classified based on their deployment and detection methodology. From a deployment perspective, there are two main types: Network Intrusion Detection Systems (NIDSs) and Host Intrusion Detection Systems (HIDSs). HIDSs monitor individual hosts or servers, analyzing system activities for signs of potential security breaches. NIDSs, in contrast, focus on examining and analyzing the entire network traffic to identify threats, serving as a primary line of defense against attacks [8]. Considering detection methodology, IDSs are categorized as either signature-based or anomaly-based. Signature-based IDSs rely on predefined attack signatures to identify known threats; on the other hand, anomaly-based IDSs detect threats by identifying deviations from normal network behavior.

Modern cybersecurity monitoring settings, particularly in industrial and IoT environments, extend beyond the traditional NIDSs and HIDSs categories. Recent studies highlight a number of categories of monitoring techniques, including network monitoring such as NIDS and NIPS (network intrusion prevention system), endpoint protection such as AV (antivirus), EDR (endpoint detection and response), and XDR (extended detection and response), physics-based monitoring at the process level, and event management platforms such as SIEM (security information and event management) [9]. The model proposed in this paper is an anomaly-based network intrusion detection methodology that operates on network traffic data and performs intrusion detection through traffic-based feature learning and classification.

In recent years, the emergence of DL has significantly enhanced IDS capabilities. DL methods have proven effective in processing extensive and complex datasets by autonomously learning hierarchical representations from raw data. Each layer in a DL model builds upon the features learned by the previous layer, allowing for enhanced detection of complex cyber threats, unlike traditional ML approaches. Moreover, DL techniques reduce the need for manual feature engineering, making them highly suitable for the dynamic and evolving cybersecurity challenges [10].

3. Literature Review

The rapid expansion of AI, including ML and DL, has significantly transformed numerous domains, including cybersecurity. The development of advanced AI algorithms, the availability of powerful computational resources, and access to diverse datasets have contributed to important advancements in IDSs. Due to their exceptional processing

capabilities and capacity to learn features from raw data, DL models attracted considerable attention for IDS applications for enhancing the precision of intrusion detection.

Khan et al. [11] proposed a two-stage deep learning model utilizing a stacked autoencoder, DNN, and softmax classifier to address classification challenges in IDS. The deep-stacked autoencoder was utilized to filter out irrelevant features, and the model's performance was evaluated using the KDD99 and UNSW-NB15 datasets. While the model achieved nearly 99.9% detection on KDD99, its accuracy dropped to 89.13% on the more recent and complex UNSW-NB15 dataset. This performance drop on contemporary data highlights a generalization issue, indicating limitations in adapting to evolving attack patterns.

More recent approaches have moved toward hybrid methods in IoT threat detection. Otoum et al. [12] integrated Spider Monkey Optimization, a swarm intelligence technique, with a deep polynomial network for IoT environments, improving performance on NSL-KDD. Furthermore, Shoab et al. [13] combined Cuckoo Search and PSO for feature selection, paired with an autoencoder-GRU ensemble, yielding strong results on NSL-KDD. However, reliance on non-IoT datasets leaves open questions about the real-world performance of these models in IoT environments. These studies highlight the importance of effective feature optimization techniques adapting to modern heterogeneous traffic patterns without reliance on legacy benchmarks or extremely complex architectures that affect deployment. Many early DL-based IDSs were evaluated on legacy benchmarks like KDD99 or NSL-KDD, which do not fully reflect the diversity of modern network traffic, especially in IoT environments. As a result, models often appeared highly accurate on paper, yet their applicability is questionable. Gao et al. [14] approach, incremental extreme learning machine, encountered an accuracy drop from 81% on NSL-KDD to 70% on UNSW-NB15, highlighting how models performed well on outdated datasets and failed to generalize to modern threats and evolving network behaviors. Varghese et al. [15] proposed a domain adaptive multi-modal approach that balances feature distributions and enhances generalization across different modern datasets such as CIC-IDS2017.

Imbalanced datasets, where some class instances greatly outnumber others, bias model learning toward majority classes and limit the detection of rare but critical attacks. Various strategies have been employed to mitigate this. Zeeshan et al. [16] merged and resampled IoT attack datasets for enhanced detection, but this resulted in massive datasets with excessive computational training costs. Fu et al. [17] combined Adaptive Synthetic Sampling (ADASYN) with a CNN and Bi-LSTM, achieving 90% accuracy, but still faced challenges in capturing rare attack patterns. Abdelkhalek et al. [18] combined ADASYN oversampling with Tomek links to mitigate the class imbalance problem on the NSL-KDD dataset, highlighting the importance of preprocessing in increasing detection rates just as effectively as model architecture. However, because NSL-KDD is an older benchmark, strong results on that dataset do not necessarily demonstrate robustness against modern and different attack behavior. Nemalikanti et al. [19] propose an intrusion detection system for IoT that employs an Autoencoder for feature extraction followed by a Ridge classifier, validated on multiple IoT datasets, achieving an F1-score of 93%. While this hybrid Autoencoder-Ridge approach effectively addresses feature extraction and classification, the linear nature of Ridge classification may limit detection of complex minority attack patterns compared to DL models, highlighting the importance of both architecture selection and dataset characteristics in IDS performance.

Dhirar et al. [20] benchmarked four DL-models, namely, CNN, LSTM, RNN/GRU, and DNN, against the BoT-IoT, ToN-IoT, and SDN-IoT datasets. The experimental results demonstrate that realistic, multiclass, and imbalanced environments are more challenging. On the BoT-IoT dataset, minority classes significantly reduced precision and recall across models such as LSTM and RNN, while an artificially balanced SDN-IoT dataset provided

better performance, illustrating how class imbalance can cause performance drops even in DL models. Recent work by Prasad et al. [21] introduced a Cosine Similarity-based Majority Class Reduction (CSMCR) explicitly targeting class imbalance without synthetic oversampling that eliminates redundant benign samples before training. On BoT-IoT, their best configuration reports an F1-score of 92%. The study shows how using various majority to minority class ratios to balance the data reduces overfitting and improves performance when compared to the unbalanced baseline.

Recent studies have also explored the integration of swarm techniques with advanced learning models. For example, an Adaptive Swarm Reinforcement Learning (ASRL) model combines adaptive reinforcement learning with Salp Swarm Optimization, incorporating dimensionality reduction using Uniform Manifold Approximation and Projection (UMAP), pattern recognition through LSTM, and deep feature extraction using Bidirectional Encoder Representations from Transformers (BERT) to enhance intrusion detection in dynamic online social networks (OSN) [22]. While ASRL demonstrates promising results, its design is specifically tailored to certain environments. In contrast, our proposed IDS model targets both IoT and computer network environments.

In summary, despite significant advancements, the literature highlights persistent challenges in developing efficient and effective IDSs for modern computer networks and IoT, including noisy high-dimensional data, class imbalance, outdated benchmarks, and computational complexity. These challenges motivate the development of an innovative IDS framework that combines intelligent feature selection, imbalance aware sampling strategies, and an efficient DL model. In this work, stacked GRU layers serve as the core sequence processing component, offering computational efficiency over LSTM through fewer gates and faster convergence [23]. The stacked GRU processes the feature subsets selected by BPSO, enabling the model to effectively learn attack patterns in network traffic. To further enhance the performance of the recurrent architecture, MHA is integrated to enable the model to dynamically focus on the most relevant parts of the sequence of hidden states [24]. MHA employs multiple attention heads to capture diverse patterns in network traffic, allowing the model to attend to different representation subspaces of the input and facilitating parallel feature extraction. This mechanism supports more stable learning and improves the model's ability to distinguish complex intrusion patterns [25].

The proposed framework synergistically integrates three key components to address core challenges in modern network and IoT threat detection: BPSO feature optimization, Target Sampling, and MHA-GRU classification. The proposed model is comprehensively evaluated on recent datasets, namely BoT-IoT and CIC-IDS2017, both of which include modern network behaviors and diverse attack scenarios. Experimental results demonstrate that the proposed approach achieves superior performance compared with several existing IDS methods in both binary and multiclass detection tasks.

4. Proposed DL-Based Intrusion Detection Approach

The development of an innovative IDS is a complex process consisting of several important phases, each contributing to the overall performance of the model. The proposed IDS methodology consists of four stages: dataset analysis and selection, advanced data preprocessing and Target Sampling, optimal feature selection, and intrusion detection and classification.

4.1. Dataset Analysis

The initial stage includes selecting and analyzing benchmark datasets to ensure robust detection of various cyber threats by identifying records of normal traffic, categorizing attack types, and understanding the features in the dataset. Various publicly available

datasets, such as KDD99, NSL-KDD, CIC-IDS2017, and BoT-IoT, have been used in IDS research for training and evaluation. Various IDS research relied mainly on datasets such as KDD99 and NSL-KDD, which do not properly capture modern network traffic characteristics and attack complexity. Therefore, in this research, we focus on two recent benchmark datasets, namely BoT-IoT and CIC-IDS2017, selected due to their comprehensive representation of modern attack scenarios and relevance for IoT networks covering real-world network traffic.

The BoT-IoT dataset [26], developed by the Australian Centre for Cyber Security, is a comprehensive benchmark tailored for IoT network security. It includes a wide range of attack types relevant to IoT environments, such as Denial of Service (DoS), Distributed Denial of Service (DDoS), reconnaissance, and information theft. The dataset consists of more than 72 million network traffic records, featuring simulated IoT traffic and attacks targeting IoT infrastructures. In this research, the 5% BoT-IoT subset was employed, comprising approximately 3.7 million records distributed across four CSV files. The dataset is designed specifically to provide IoT-specific attack scenarios, making it suitable for assessing IDS models designed for IoT environments.

The CIC-IDS2017 dataset [27], a recent IDS benchmark introduced by the Canadian Institute of Cybersecurity, offers a modern perspective on network traffic by including updated attack scenarios. This dataset includes eight files that capture five days of both normal and attack traffic, resulting in 2,830,743 records with 79 features. It consists of normal traffic (2,273,097 records) and 14 different attack types (557,646 records), such as DDoS, PortScan, Web attacks, DoS, Bruteforce and Botnet. For this study, the MachineLearningCSV data file was utilized to train and evaluate the proposed model. Additionally, the Monday file was used for normal traffic representation. The motivation for selecting the CIC-IDS2017 dataset is to ensure that the proposed IDS model is assessed against modern and realistic network attack scenarios. The use of this dataset enhances the model's generalizability, as it covers a broad range of attacks encountered in modern network environments.

By utilizing both the BoT-IoT dataset and the CIC-IDS2017 dataset in our evaluation, we ensure that the proposed IDS model is tested not only on traditional network attacks but also on evolving and large-scale IoT-specific threats, reflecting the growing need for robust IDS solutions. The utilization of these datasets provides a critical benchmark for evaluating the generalization ability and applicability of the proposed IDS framework.

4.2. Data Preprocessing Phase

The data preprocessing stage is a critical step in preparing the dataset for effective training and evaluation of the proposed DL-based intrusion detection model. The quality and structure of the input data significantly influence the model's performance, making comprehensive preprocessing important. The proposed framework employs a unified and systematic preprocessing strategy across both BoT-IoT and CIC-IDS2017 datasets to ensure consistency and effectiveness. This phase encompasses data cleaning, features filtrating, categorical encoding, and numerical scaling.

Missing Value Imputation: Missing numerical entries, including NaNs and infinite values, are replaced using mean substitution, ensuring consistency without distorting feature distributions while minimizing bias.

Duplicate Records: Duplicate data instances are removed to prevent overfitting and ensure data diversity.

Removal of Non-Informative and Constant Features: Features irrelevant to the detection task or constant are removed. In particular, in BoT-IoT, features including packet sequence identifiers, source/destination addresses, and ports are removed, including pkSeqID, seq, saddr, sport, daddr, dport, flgs, state, and proto features are removed. Similarly,

in CIC-IDS2017, features, namely bwd psh flags, fwd avg bytes/bulk, bwd urg flags, fwd avg packets/bulk, bwd avg bytes/bulk, fwd avg bulk rate, bwd avg bulk rate, cwe flag count, fwd urg flags, and bwd avg packets/bulk are discarded.

Categorical Feature Encoding: Class labels are converted to integer encodings through label encoding to enable efficient processing by the deep learning classifier. For BoT-IoT, categories are mapped as Normal:0, DoS:1, DDoS:2, Reconnaissance:3, Theft:4. In CIC-IDS2017, class labels are encoded as: BENIGN:0, DoS:1, DDoS:2, PortScan:3, BruteForce:4, WebAttack:5, Other:6.

Feature Scaling: Given the non-uniform distribution of numerical features, to balance the scale of diverse features and eliminate unwanted dominance by features with large numeric ranges. The min-max normalization is utilized to ensure all feature values are scaled within the range [0, 1], as depicted in Equation (1):

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{1}$$

where x_{norm} is the normalized value, x is the original value, and x_{min} and x_{max} represent the minimum and maximum values of the feature, respectively.

4.3. Target Sampling

Both BoT-IoT and CIC-IDS2017 datasets show considerable class imbalance where highly frequent classes representing normal traffic or common attacks outnumber rare but crucial classes. This imbalance influences the model learning by biasing predictions toward dominant classes and missing rare classes detection [28]. To address this challenge, we employed a target class aware sampling strategy aiming to preserve the semantic diversity of all classes without introducing synthetic data bias. For each class, Target Sampling is conducted using class-wise random sampling. Given the target count for a specific class, instances are randomly selected from all records corresponding to that class without replacement, ensuring that no record is chosen more than once. Minority classes with limited numbers of records are fully retained. The class-wise Target Sampling technique modifies only the relative class frequencies.

In the BoT-IoT dataset, attack labels were grouped into five main categories: Normal, DoS, DDoS, Reconnaissance, and Theft. The data distribution is highly imbalanced, DoS (1.65 million), DDoS (1.93 million), Reconnaissance (91k), greatly outnumber minority classes Normal (477), and Theft (79). Our strategy keeps all samples from the minority classes Normal and Theft intact to ensure exposure to critical minority events. We down-sampled Reconnaissance to 42,600 records and proportionally reduced DoS and DDoS to 561k and 666k, respectively, cutting majority class dominance by 50% while preserving realistic traffic characteristics. We intentionally maintain some imbalance to stress test the proposed IDS and its ability to detect rare but critical anomalies in distorted environments reflecting real world IoT environments. This design aligns with modern studies emphasizing the role of stress testing IDS frameworks to validate robustness and sensitivity to rare classes [29].

The CIC-IDS2017 dataset shows a moderately imbalanced distribution. The attacks were grouped into seven categories: BENIGN, DoS, DDoS, PortScan, BruteForce, WebAttack, and Other, which includes Bot, Infiltration, and Heartbleed attacks. Majority classes, such as BENIGN, DoS, DDoS, and PortScan, were downsampled from their original counts, about 500k, 193k, 128k, 90k, to 75k, 29k, 19k, and 13k samples, respectively, selecting 15% of the original count. This aligns them more closely with the minority classes, creating a balanced test where the number of normal records is close to the number of attacks. BruteForce (9152), WebAttack (2143), and Other (2000) were fully retained. This scaling

limits bias from dominant classes and facilitates efficient training and fair evaluation across all classes, especially for less frequent but critical attacks. The class distributions for both the BoT-IoT and CIC-IDS2017 datasets with the respective target sample counts after target sampling are summarized in Tables 1 and 2, the original count is the count after the preprocessing stage. For performance evaluation of the proposed IDS model, the datasets were partitioned into training 70%, validation 10%, and testing 20% sets using stratified split.

In the implementation, the GRU input does not correspond to ordered sequences across multiple traffic records. Each sample is represented by its feature vector and processed as an individual structured input. Therefore, the GRU operates on the feature representation of each sample rather than on ordered flows across different records. This structured resampling strategy avoids generating synthetic minority or majority samples, thereby reducing the risk of overfitting or creating artificial patterns that could influence the training process. The approach preserves the characteristics and distributions of real network traffic, ensuring that trained models are robust and generalizable, and sensitive to different attack types.

Table 1. BoT-IoT Class Grouping and Sampling Overview.

Group	Original Count	Target Count
DDoS	1,332,371	666,000
DoS	1,123,555	561,000
Reconnaissance	85,261	42,600
Normal	477	477
Theft	79	79
Total	2,541,743	1,270,156

Table 2. CIC-IDS2017 Class Grouping and Sampling Overview.

Group	Original Count	Target Count	Included Attacks
BENIGN	502,983	75,000	Normal Traffic
DoS	193,748	29,000	DoS Hulk (172,849); DoS Slowloris (5385); DoS Slowhttptest (5228); DoS GoldenEye (10,286)
DDoS	128,016	19,000	DDoS
PortScan	90,819	13,000	PortScan
BruteForce	9152	9152	FTP-Patator (5933); SSH-Patator (3219)
WebAttack	2143	2143	Web-Brute Force (1470); Web-XSS (652); Web-SQL Injection (21)
Other	2000	2000	Bot (1953); Infiltration (36); Heartbleed (11)
Total	928,861	149,295	–

4.4. Optimal Feature Selection Stage

This stage is essential for identifying the most influential features in the datasets. The importance of this stage lies in its ability to enhance the model's performance while reducing its computational complexity and reducing the risk of overfitting by eliminating redundant features. Originally introduced by Kennedy and Eberhart in 1995, PSO has been widely utilized for various optimization problems inspired by the social behavior of bird flocking [30]. In this work, we employ the BPSO algorithm for optimal feature selection given its rapid convergence, simplified implementation with few control hyperparameters, and effective feature exploration, enabling the identification of the most relevant feature subsets [31]. BPSO is utilized for the proposed IDS framework because network traffic is

high-dimensional, which includes redundancy and irrelevant features for classification. BPSO selects a reduced subset of informative features, reducing the input dimensionality and limiting the influence of irrelevant features, providing the stacked GRU model with a more discriminative input representation.

BPSO aims to identify the optimal subset through information sharing among the particles in a swarm. Each particle represents a potential solution characterized by its position and velocity; several particles form a swarm, and each particle tracks its attributes. The velocity and position of each attribute within the particle continuously update according to its current fitness value and the best found across all particles within the swarm during each iteration of the search for the optimal solution. For the purpose of feature selection, the BPSO variant of PSO is adopted to handle binary variables that represent the inclusion or exclusion of features. The swarm size selected to balance exploration and computational cost, and 50 iterations allows the swarm enough search steps to converge without unnecessary computations. The BPSO algorithm approach is explained as follows.

System Initialization: The algorithm begins by initializing a swarm in the D-dimensional space, with each particle representing a potential solution defined by a position vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and a velocity vector $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{id})$, where d is the dimensionality of the search space, i.e., the number of features, after preprocessing 68 for CIC-IDS2017 and 34 for BoT-IoT. The number of particles N is set to 20, and the maximum number of iterations is set at 50, with a fixed random seed ensures reproducibility.

Fitness Evaluation: The fitness function evaluates the quality of each particle based on the validation accuracy achieved by training the attention-based GRU classification model using only the features selected by that particle. A higher accuracy indicates a better feature subset. Only the training and validation sets are used during this stage to avoid data leakage.

$$\text{Fitness}\ (\mathbf{x}_i) = \text{ValidationAccuracy(MHA-GRU)}_{\mathbf{x}_i} \tag{2}$$

Personal and Global Particle Update: Particles are assigned the updated fitness value if their new score surpasses the current one. Through the search process of the BPSO iterations, each particle maintains a record of its optimal position found so far, denoted as its personal best. The personal best for the i-th particle is defined as $\mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. Concurrently, the one achieving the highest fitness across the swarm is designated as the global best, defined as $\mathbf{p}_g = (p_{g1}, p_{g2}, \ldots, p_{gd})$.

Velocity and Position Update: In each iteration, the velocity and position of each particle are updated according to current velocity, personal best positions, and global best position utilizing the equations:

$$v_{id}^{(t+1)} = w \cdot v_{id}^{(t)} + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}^{(t)}) + c_2 \cdot rand_2 \cdot (p_{gd} - x_{id}^{(t)}) \tag{3}$$

$$x_{id}^{(t+1)} = \begin{cases} 1 & \text{if } rand < sigmoid(v_{id}^{(t+1)}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $i = 1, 2, \ldots, N$, with N being the number of particles; $d = 1, 2, \ldots, D$, where D is the dimensionality of the search space; w represents the inertia weight, providing a balance between exploration and exploitation. c_1 and c_2 are the cognitive and social scaling parameters. $rand_1$ and $rand_2$ are random numbers uniformly distributed in $[0, 1]$. The function $sigmoid(v_{id}^{(t+1)})$ is the sigmoid function applied to the velocity, defined as:

$$sigmoid(v_{id}^{(t+1)}) = \frac{1}{1 + e^{-v_{id}^{(t+1)}}} \tag{5}$$

The sigmoid function transforms the velocity into a value in the range [0, 1], leading to the BPSO update of each particle's position determining whether a feature is included (1) or excluded (0) in the next iteration.

Termination: The algorithm terminates upon completing the specified iterations or reaching an adequate fitness level. The global best position is then returned, representing the optimal feature subset.

The datasets are partitioned using a stratified split; the first 80% of the data is assigned to training and validation, while 20% is reserved as a held-out test set. The 80% training/validation portion is then split into 70% training and 10% validation, resulting in an overall allocation of 70% training, 10% validation, and 20% testing. During the BPSO feature selection stage, the fitness function trains the model on the training subset and evaluates candidate feature subsets using the validation subset only. The test set is never used during BPSO optimization or model training and is used only for the final performance evaluation, preventing data leakage during the process.

In our implementation, w is set to 0.7, c_1 and c_2 coefficients are set to 2, which are commonly used BPSO parameters in the literature for feature selection applications, offering a balance between exploration and exploitation [31]. Additionally, the number of particles N is set to 20 and the maximum number of iterations to 50, chosen to maintain sufficient search diversity without incurring a high computational cost during the search for the optimal subset of features. The particle dimension d corresponds to the total number of features, which is equal to 68 features in the CIC-IDS2017 dataset and 34 in the BoT-IoT dataset. We set an early stopping mechanism in case of no significant improvement; then the BPSO will stop, and the global best set of optimal features will be returned. This step helps with reducing unnecessary computational overhead. During initialization, particles are assigned binary values; 1 indicates the inclusion of a feature, and 0 denotes its exclusion. Validation accuracy is used to evaluate the fitness of each particle, ensuring that each particle's position directly corresponds to a specific subset of features. For illustration, BPSO is utilized for feature selection on a dataset with 10 features $x_1 \ldots x_{10}$. In the swarm initialization, a particle is represented as a set of 0's and 1's (0, 1, 1, 0, 1, 0, 1, 1, 0, 1). Here, 1 means a selected feature, and 0 means a discarded feature. As iterations proceed, the influence of personal and global best positions may lead to changes in this binary representation, resulting in the selection of different feature subsets.

BPSO effectively reduces the dimensionality of feature space by selecting the most relevant features, thereby lowering computational complexity and improving IDS accuracy through the elimination of redundant and irrelevant data. In addition, compared to other feature selection algorithms, BPSO delivers rapid computing speed and robust global search capabilities with few parameters, making it both efficient and easy to implement. This balance makes BPSO particularly well-suited for optimal feature selection in IDS. Tables 3 and 4 illustrate the optimal features selected for BoT-IoT and CIC-IDS2017 datasets by BPSO, giving the best accuracy through the binary classification run, respectively. Algorithm 1 and Figure 1 illustrates the execution steps of the BPSO algorithm.

Table 3. BoT-IoT Selected Features by BPSO.

No.	Selected	Feature Name	No.	Selected	Feature Name
1	No	stime	18	Yes	rate
2	No	flgs_number	19	No	srate
3	Yes	proto_number	20	Yes	drate
4	Yes	pkts	21	Yes	TnBPSrcIP
5	No	bytes	22	No	TnBPDstIP
6	Yes	state_number	23	Yes	TnP_PSrcIP

Table 3. *Cont.*

No.	Selected	Feature Name	No.	Selected	Feature Name
7	No	ltime	24	No	TnP_PDstIP
8	No	dur	25	Yes	TnP_PerProto
9	No	mean	26	No	TnP_Per_Dport
10	No	stddev	27	No	AR_P_Proto_P_SrcIP
11	Yes	sum	28	Yes	AR_P_Proto_P_DstIP
12	No	min	29	Yes	N_IN_Conn_P_DstIP
13	No	max	30	Yes	N_IN_Conn_P_SrcIP
14	Yes	spkts	31	Yes	AR_P_Proto_P_Sport
15	Yes	dpkts	32	No	AR_P_Proto_P_Dport
16	Yes	sbytes	33	No	Pkts_P_State_P_DestIP
17	Yes	dbytes	34	Yes	Pkts_P_State_P_SrcIP

Algorithm 1 BPSO for Feature Selection

Input: Number of particles N, number of features D, number of iterations T, cognitive coefficient c_1, social coefficient c_2, inertia weight w
Output: Optimal features subset $\mathbf{p}_g$
Initialization:
for $i = 1$ to N **do**
    Initialize position $\mathbf{x}_i$ randomly
    Initialize velocity $\mathbf{v}_i$ randomly
    Evaluate fitness of $\mathbf{x}_i$
    Initialize personal best $\mathbf{p}_i = \mathbf{x}_i$
end for
$\mathbf{p}_g \leftarrow$ particle with best $\mathbf{p}_i$
Optimization:
for $t = 1$ to T **do**
    for $i = 1$ to N **do**
        for $d = 1$ to D **do**
            Generate random numbers $rand_1, rand_2 \sim U(0,1)$
            Update velocity $v_{id}^{(t+1)} = w \cdot v_{id}^{(t)} + c_1 \cdot rand_1 \cdot (p_{id} - x_{id}^{(t)}) + c_2 \cdot rand_2 \cdot (p_{gd} - x_{id}^{(t)})$
            Update position $x_{id}^{(t+1)} = \begin{cases} 1 & \text{if rand} < sigmoid(v_{id}^{(t+1)}) \\ 0 & \text{otherwise} \end{cases}$
        end for
        Evaluate fitness of $\mathbf{x}_i^{(t+1)}$
        if fitness of $\mathbf{x}_i^{(t+1)}$ is better than fitness of $\mathbf{p}_i$ **then**
            Update personal best $\mathbf{p}_i \leftarrow \mathbf{x}_i^{(t+1)}$
        end if
    end for
    Update global best $\mathbf{p}_g$ if any $\mathbf{p}_i$ is better
end for
Return Optimal feature subset $\mathbf{p}_g$

Table 4. CIC-IDS2017 Selected Features by BPSO.

No.	Selected	Feature Name	No.	Selected	Feature Name
1	Yes	Destination Port	35	Yes	Bwd Packets/s
2	Yes	Flow Duration	36	No	Min Packet Length
3	No	Total Fwd Packets	37	No	Max Packet Length
4	No	Total Backward Packets	38	No	Packet Length Mean
5	Yes	Total Length of Fwd Packets	39	Yes	Packet Length Std

Table 4. *Cont.*

No.	Selected	Feature Name	No.	Selected	Feature Name
6	No	Total Length of Bwd Packets	40	No	Packet Length Variance
7	No	Fwd Packet Length Max	41	No	FIN Flag Count
8	No	Fwd Packet Length Min	42	No	SYN Flag Count
9	No	Fwd Packet Length Mean	43	Yes	RST Flag Count
10	No	Fwd Packet Length Std	44	Yes	PSH Flag Count
11	Yes	Bwd Packet Length Max	45	No	ACK Flag Count
12	Yes	Bwd Packet Length Min	46	No	URG Flag Count
13	No	Bwd Packet Length Mean	47	No	ECE Flag Count
14	No	Bwd Packet Length Std	48	Yes	Down/Up Ratio
15	No	Flow Bytes/s	49	Yes	Average Packet Size
16	Yes	Flow Packets/s	50	No	Avg Fwd Segment Size
17	No	Flow IAT Mean	51	Yes	Avg Bwd Segment Size
18	No	Flow IAT Std	52	No	Fwd Header Length.1
19	Yes	Flow IAT Max	53	No	Subflow Fwd Packets
20	No	Flow IAT Min	54	No	Subflow Fwd Bytes
21	No	Fwd IAT Total	55	No	Subflow Bwd Packets
22	Yes	Fwd IAT Mean	56	No	Subflow Bwd Bytes
23	No	Fwd IAT Std	57	Yes	Init_Win_bytes_forward
24	No	Fwd IAT Max	58	No	Init_Win_bytes_backward
25	Yes	Fwd IAT Min	59	No	act_data_pkt_fwd
26	No	Bwd IAT Total	60	No	min_seg_size_forward
27	No	Bwd IAT Mean	61	No	Active Mean
28	No	Bwd IAT Std	62	No	Active Std
29	Yes	Bwd IAT Max	63	No	Active Max
30	No	Bwd IAT Min	64	Yes	Active Min
31	No	Fwd PSH Flags	65	No	Idle Mean
32	No	Fwd Header Length	66	No	Idle Std
33	No	Bwd Header Length	67	Yes	Idle Max
34	No	Fwd Packets/s	68	No	Idle Min

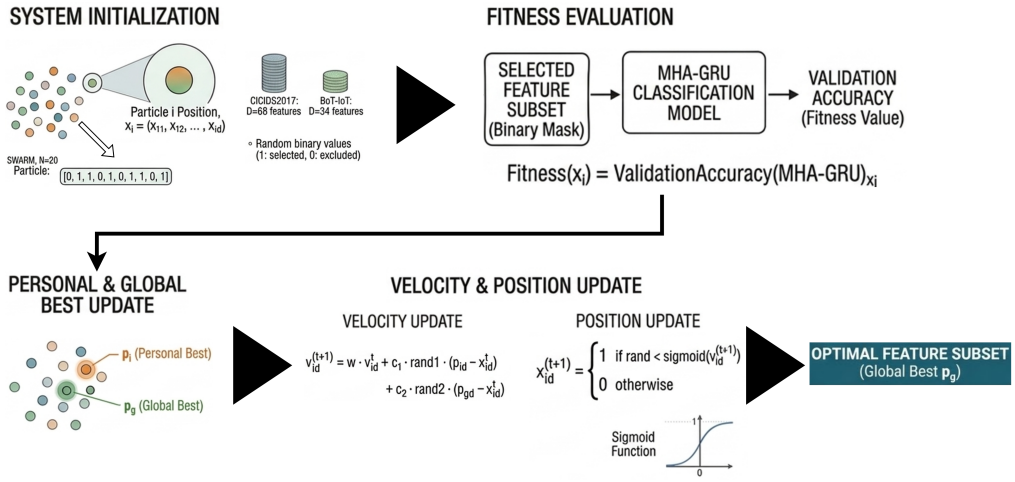


Figure 1. BPSO Algorithm.

4.5. Classification Process

This section presents the architecture of the DL-based classifier employed in this research. The classification architecture leverages a stacked GRU design, combining the sequence processing capability of GRUs with an MHA mechanism that dynamically weights sequence elements to focus on the most informative features. Layer Normalization is integrated to stabilize training and accelerate convergence, while dropout layers are utilized to mitigate overfitting. This multilayered setup is specifically suitable for complex, high-dimensional data characteristic of IoT and computer network traffic. This hybrid

design enhances the model's ability to detect intrusions by learning informative patterns through sequence processing and focusing on relevant patterns within network traffic.

GRU is the simplified and improved variant of the LSTM proposed by Cho et al. in 2014 [32], addressing the vanishing gradient problem inherent in traditional RNNs. By leveraging two gating mechanisms, the update gate and the reset gate, GRUs effectively regulate the flow of information without relying on a separate memory cell. This simplified architecture allows GRUs to maintain performance while reducing computational complexity. Figure 2 illustrates the GRU neural network structure.

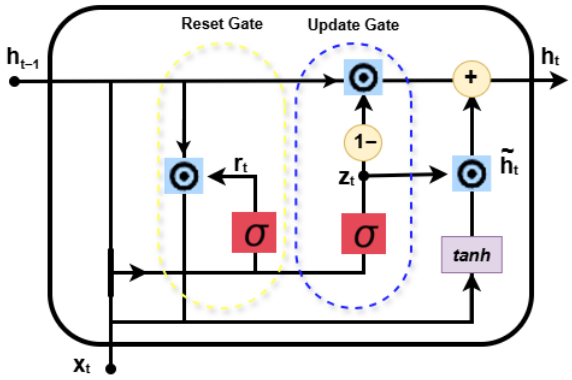


Figure 2. Structure Diagram of GRU.

The GRU unit at each time step t processes an input vector x_t and updates its hidden state h_t based on its previous state h_{t-1}. The key operations within the GRU cell are as follows:

The update gate controls how much of the previous information needs to be carried over to the future and it is defined as:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{6}$$

where x_t denotes the input at time step t, $W^{(z)}$ and $U^{(z)}$ are the weights corresponding to the input and the previous hidden state h_{t-1}, respectively, and σ denote the sigmoid function applied to suppress the result between 0 and 1.

Meanwhile, the reset gate controls how much to forget from the past information and is computed as:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{7}$$

Following the reset gate operation, the candidate hidden state is calculated ($\tilde{h}_t$). $\tilde{h}_t$ representing the new information to be added to the hidden state. It is computed as:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \tag{8}$$

where $\odot$ denotes element-wise multiplication, and tanh is the hyperbolic tangent activation function, which outputs a value between -1 and 1. As the last step, the final hidden state h_t is calculated which carries information for the current unit and passes it through to the network.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{9}$$

This mechanism enables the GRU at each phase to effectively determine which data is relevant to retain and which to discard, enhancing its learning capability and computational efficiency compared to more complex LSTM architectures.

The proposed GRU-based deep learning architecture employs a stacked configuration to deepen sequence processing and exploit the representational capacity of GRUs. Specif-

ically, the first GRU layer comprises 128 hidden units, followed by a second GRU layer with 64 hidden units, a design that is used for both binary and multiclass classification tasks. This structure helps maintain training stability while providing sufficient capacity to learn informative patterns within the input representation, with the hidden size decreasing toward the output layer. To mitigate overfitting and improve generalization, a dropout rate of 0.2 is applied after each GRU layer.

To further enhance the representational power of the GRU backbone, a MHA mechanism is integrated into the architecture [25]. MHA enables the model to assign adaptive importance weights to different elements of the GRU outputs, improving the detection of subtle and complex patterns in network traffic. Let $\mathbf{X}$ denote the sequence of hidden states output from the final GRU layer, where d is the feature dimensionality equal to 64 units. The attention mechanism projects $\mathbf{X}$ into three learnable matrices:

$$Q = XW_Q, K = XW_K, V = XW_V, \tag{10}$$

where W_Q, W_K, W_V are the query, key, and value matrices, respectively.

The attention output for a single head is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{11}$$

where d_k is the dimensionality of the key vectors. The softmax function assigns attention weights based on their relevance to the current query.

In the MHA setup with H heads, the attention process is performed in parallel across different parameter matrices for each head. We set four parallel attention heads $H = 4$, each head with a dimensionality $d_k = 16$. Since the last GRU layer output is of dimension 64, dividing it into 4 heads with 16 dimensions each ensures the total attention embedding dimension remains consistent. The outputs of all heads are concatenated and linearly transformed:

$$\text{MHA}(\mathbf{X}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)W_O, \tag{12}$$

where W_O is the output projection matrix. Dropout with a rate of 0.1 is applied within the attention module to promote regularization. This mechanism allows the model to pay attention to information from multiple representation subspaces at different positions, enabling the model to simultaneously detect diverse patterns such as sudden attacks or stealthy scanning. Following the attention layer, residual addition is performed by summing the attention output with the original GRU output, and the result is normalized using LN ($\varepsilon = 10^{-3}$) to stabilize training. The subsequent global average pooling layer reduces feature dimensionality by averaging each feature across the sequence, producing a compact and informative representation that facilitates efficient classification, as illustrated in Figure 3. For classification, a fully connected dense output layer equipped with softmax activation is used for multi-class classification. For the binary classification case, sigmoid activation is employed. The proposed model employs sparse categorical cross-entropy loss for multi-class classification and binary cross-entropy loss for binary classification, optimized with the Adam algorithm using an initial learning rate of 0.01. A learning rate scheduler is employed to reduce the learning rate by a factor of 0.5 upon stagnation of validation loss allowing stable learning. Key hyperparameters influencing the GRU model's learning process include the batch size, the learning rate, and the number of epochs. In our experiments using the BoT-IoT and the CIC-IDS2017 datasets, we set the batch size to 64 matching the GRU layer, the dropout rate to 0.2 between the stacked GRU layers, and

the training conducted over 50 epochs, providing enough iterations to ensure convergence while avoiding overfitting. The batch size and the epoch count determine the training sample size per iteration and the total dataset passes during training, respectively.

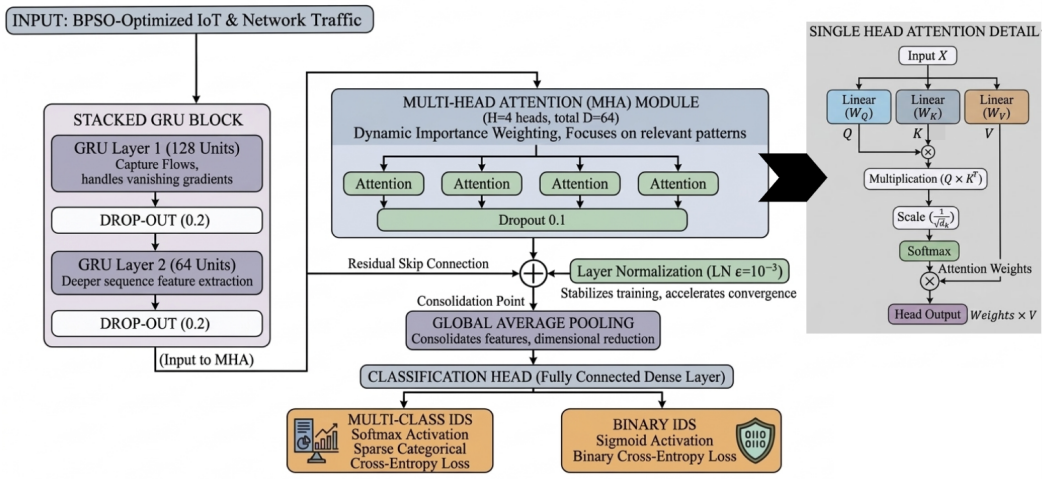


Figure 3. The Proposed MHA-stacked GRU IDS Model.

The proposed architecture handles challenges inherent in IDSs. BPSO serves as the feature selection mechanism due to its proven efficacy in navigating high-dimensional search spaces to identify compact, informative feature subsets that minimize redundancy while improving training efficiency. Stacked GRU layers perform sequence processing over the input representation. MHA enables selective focus on the most relevant features within the sequence of hidden states. Target Sampling strategy addresses dataset class imbalance without synthetic sample generation by preserving minority classes.

5. Performance Evaluation

This section presents a comprehensive analysis of the performance of the proposed IDS model, along with a comparative evaluation against other recently developed algorithms.

5.1. Benchmark Dataset

To evaluate the effectiveness of our proposed model, we utilize two widely recognized recent benchmark datasets, namely BoT-IoT and CIC-IDS2017. The BoT-IoT dataset includes four main types of attacks, While the CIC-IDS2017 dataset features fourteen attack types. Initially, both datasets undergo an extensive preprocessing stage, preparing them for effective feature selection and classification. Subsequently, BPSO is applied to select the most relevant features in both datasets, associated with the MHA-stacked GRU for feature learning and classification. The experiments were conducted on a Windows 11 i7-computer with 16 GB of RAM in a Python 3 environment.

5.2. Evaluation Metrics

The effectiveness of the proposed IDS model is assessed using several standard evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics provide a detailed understanding of the model's ability to correctly detect intrusions while minimizing false detections. We report macro-averaged precision, recall, and F1-score across classes in both binary and multiclass evaluations to reflect minority-class detection performance, treating all classes equally regardless of their frequency [28].

Accuracy represents the overall model performance in correctly detecting threats. A high precision means that real attacks are accurately identified by the algorithm, decreasing the possibility of false alarms. High recall guarantees that the majority of attacks are

detected by IDS, decreasing the possibility of missing any. F1-score is the harmonic mean of precision and recall, providing a balance between the two. These evaluation metrics used for evaluating the performance of the proposed DL-based IDS are defined as [33]:

Accuracy: The ratio of correctly predicted instances to the total number.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Precision: The ratio of true positive predictions (correctly classified as intrusion) to all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

Recall: Also known as Detection rate, the ratio of positively predicted attacks to all instances of actual intrusions.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

F1-score: The harmonic mean of precision and recall.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{16}$$

Matthews Correlation Coefficient (MCC): A correlation based metric that accounts for all four confusion matrix entries, provides a reliable performance measure under class imbalance.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{17}$$

where: True negative (TN): The model correctly predicts an instance as normal. False negative (FN): The model fails to predict an attack, incorrectly classifying it as normal. True positive (TP): The model correctly predicts an instance as an attack. False positive (FP): The model incorrectly predicts a normal instance as an attack.

For the evaluation conducted in this study, macro-averaged metrics are computed by averaging the per-class values of each metric across all classes. Additionally, we employ the confusion matrix to analyze the model's classification performance. The confusion matrix provides a detailed breakdown of correctly and incorrectly classified instances, highlighting areas where the model excels or needs improvement.

5.3. Experimental Results

The development of the proposed model begins with the initialization of a swarm of twenty particles, each representing a candidate feature subset from the BoT-IoT and CIC-IDS2017 datasets. This initialization ensures a diverse search space for optimal feature selection. The fitness of each particle is assessed by training the MHA-stacked GRU model on its respective feature subset and calculating the validation accuracy. Throughout the iterative optimization process, the BPSO algorithm continuously updates particle velocities and positions based on personal and global best fitness values. Eventually, the optimal feature subset, with the highest validation accuracy, is identified as the global best. The selected features for the BoT-IoT and CIC-IDS2017 datasets by the proposed approach are provided in Tables 3 and 4. Once training is completed, the model's robustness is assessed using the unseen test dataset to evaluate its effectiveness in detecting unseen threats.

Table 5 presents the binary classification performance of the proposed model, where traffic instances are categorized as either normal or attack. For the BoT-IoT dataset, the

binary experiment was intentionally conducted under a highly imbalanced environment by retaining all normal records and sampling majority attack records from the available attack traffic. This design was utilized as a stress test scenario to examine model behavior under highly class imbalanced environments common in IoT networks, evaluating the robustness of the proposed model under such conditions. The goal was not to artificially simplify the problem but to reduce the overwhelming majority class dominance while preserving the minority class. The resulting accuracy should not be considered alone since it is influenced by the dominance of the majority class. Therefore, in addition to accuracy, we report macro precision, recall, F1-score and the confusion matrix to provide more informative evaluation of the model's performance across classes. For the BoT-IoT dataset, we selected all the normal traffic records in the dataset as is at 477, and around 20% of the attack class was utilized selecting 500,000 records from the 2,541,266 attack records. For the CIC-IDS2017 dataset, we selected all records after preprocessing, 502,983 Benign and 425,878 attack records. Additionally, to provide a more realistic assessment of binary detection performance under class imbalance, the MCC defined in Equation (17), is also reported for both the BoT-IoT and CIC-IDS2017 datasets in Table 5. The results demonstrate that the proposed model achieves an F1-score of 98.94% with 99.46% precision employing the BoT-IoT and an F1-score of 99.76% with 99.76% precision on the CIC-IDS2017 dataset, highlighting the model's effectiveness in accurately detecting intrusions within diverse network traffic. Figure 4 provides a visual illustration of the proposed model's performance across the evaluation metrics, further demonstrating its effectiveness for binary intrusion detection.

Table 5. Performance Metrics for Binary Classification Using the Proposed Model.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
BoT-IoT	99.99	99.46	98.42	98.94	0.9788
CIC-IDS2017	99.76	99.76	99.76	99.76	0.9945

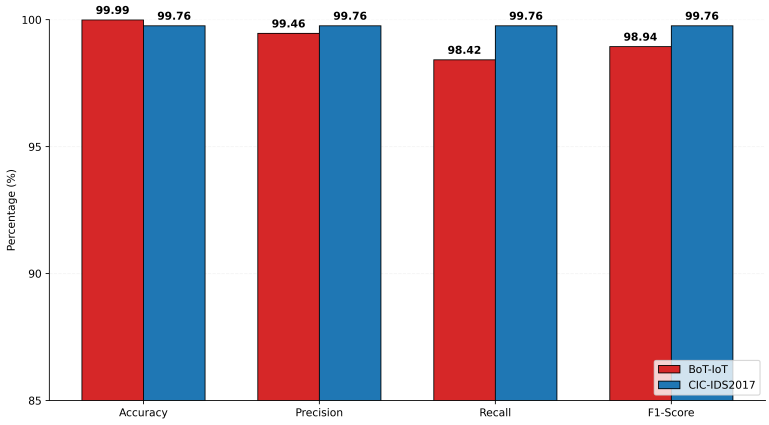


Figure 4. Performance Metrics of the Proposed Model for Binary Classification.

Figures 5 and 6 illustrate the model loss and accuracy throughout the training process for binary classification. The training accuracy curve reflects how well the model learns from the training dataset during each epoch. The validation accuracy curve shows how well the model generalizes on the unseen validation dataset. If both training and validation accuracy increase over time and converge, this indicates the model is learning and generalizing well. The training loss curve illustrates the error on the training dataset, it typically decreases as the model learns. The validation loss curve shows the error on the validation dataset, a small gap between training and validation loss indicates good generalization. In both datasets, the close alignment between training and validation accuracy, along with

steadily decreasing loss values, indicates the absence of overfitting and robust performance on unseen data. For the BoT-IoT dataset experiment in Figure 5, target class distribution was intentionally set to be highly imbalanced for stress testing purposes, with the number of instances being selected as Normal: 477 and Attack: 500,000, resulting in rapid accuracy convergence near 100.00 accuracy and minimal loss values after few epochs. However, this near perfect performance metrics primarily originate from the dominance of the majority class, so accuracy alone is insufficient to accurately represent effective minority class detection. Therefore, our results highlight additional focus on macro-averaged precision, recall, and F1-score to more comprehensively assess performance across all classes. In contrast, on the CIC-IDS2017 dataset, which is more balanced, with the number of instances being selected as Normal: 502,983 and Attack: 425,878, accuracy improves more gradually and loss decreases steadily across epochs, demonstrating effective generalization under a less extreme class distribution.

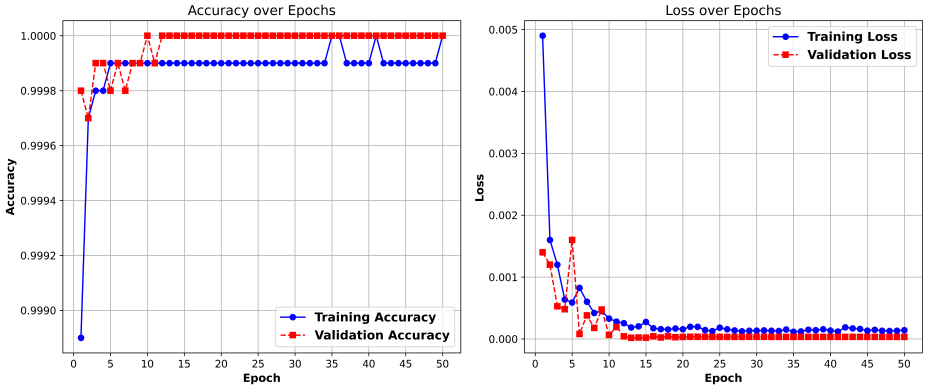


Figure 5. Model Accuracy and Loss using the BoT-IoT Dataset.

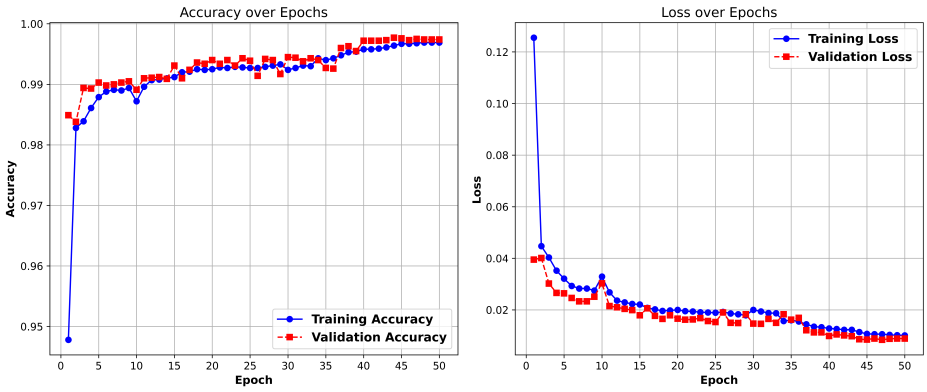


Figure 6. Model Accuracy and Loss using the CIC-IDS2017 Dataset.

Accurate intrusion detection is fundamental in cybersecurity. To further evaluate the classification effectiveness of the proposed model, we employed the confusion matrix, which provides a detailed breakdown of classification performance. This matrix enables a comprehensive evaluation of the model's ability to correctly distinguish between normal traffic and intrusion attempts. For the BoT-IoT dataset confusion matrix in Figure 7, the model correctly classified 92 normal traffic instances out of 95 records and successfully detected 100,000 intrusion attempts within the BoT-IoT dataset out of 100,001. Similarly, for the CIC-IDS2017 dataset Figure 8, the model accurately classified 100,378 normal traffic instances out of 100,597 records and correctly identified 84,948 intrusions out of 85,176 attacks.

In these confusion matrices, the x-axis represents the predicted labels "Predicted Normal" and "Predicted Attack", while the y-axis represents the actual labels "Actual Normal"

and "Actual Attack". The diagonal elements correspond to correctly classified instances. In contrast, the off-diagonal elements indicate misclassifications; false positives, normal traffic misclassified as an attack in the top-right, and false negatives, attacks misclassified as normal traffic in the bottom-left. The confusion matrix represents an important tool for analyzing classification performance, as high values along the diagonal highlight the proposed model's robustness and its strong ability to distinguish between benign and malicious traffic, even under highly skewed class distributions. These results further confirm the reliability of the proposed model in intrusion detection, demonstrating its capability to enhance cybersecurity by minimizing classification errors. To ensure reproducibility, our experiments were conducted with a fixed random seed of 40.

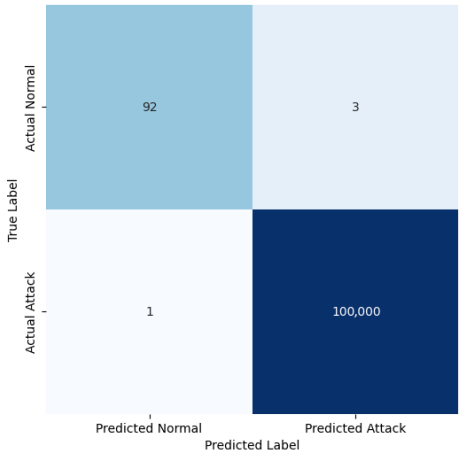


Figure 7. Confusion Matrix Utilizing the BoT-IoT Dataset.

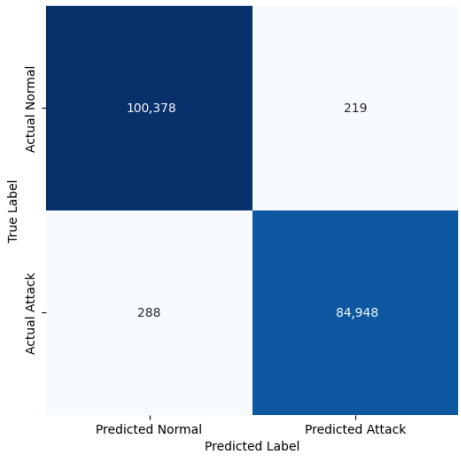


Figure 8. Confusion Matrix Utilizing the CIC-IDS2017 Dataset.

Following the evaluation of the proposed BPSO-MHA-GRU IDS approach on the binary classification task, we extended the analysis to evaluate its performance in the multiclass intrusion detection setting. For the proposed model multiclass experiments, Target Sampling was applied to both the BoT-IoT and CIC-IDS2017 datasets before training and evaluation to reduce the dominance of majority classes while preserving minority classes, Tables 1 and 2. The model achieved outstanding results, an accuracy of 99.99% and 99.89% F1-score on the BoT-IoT dataset and 99.34% accuracy with a 98.04% F1-score on the CIC-IDS2017 dataset, as provided in Table 6. Macro-averaged metrics are reported throughout all experiments to ensure a fair representation of classification performance, particularly for minority classes, therefore presenting a comprehensive and unbiased evaluation of the model across all class categories.

Table 6. Performance Metrics for Multiclass Classification Utilizing the Proposed Model.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BoT-IoT	99.99	99.99	99.79	99.89
CIC-IDS2017	99.34	97.42	98.69	98.04

Additionally, we employed the confusion matrix to provide a detailed breakdown of the classification performance of the proposed model utilizing the BoT-IoT and CIC-IDS2017 datasets in the multiclass classification task, as depicted in Figures 9 and 10. The matrix for the BoT-IoT dataset demonstrates high true positive rates across all classes, particularly both DoS and DDoS attacks, with minimal misclassifications among the remaining attack categories and normal traffic. Most off-diagonal misclassified records are zero, indicating strong classification capability despite class deviation. This reflects the model's effectiveness in capturing unique patterns for each intrusion category even in the presence of class imbalance. For CIC-IDS2017, the confusion matrix demonstrates robust classification, with high successful classification rates along the diagonal for both benign traffic and various types of attacks. The overall distribution confirms that the model achieves reliable multiclass classification, even for minority classes, supporting its generalizability and robustness across complex modern network and IoT environments.

In addition to the confusion matrices, Tables 7 and 8 provide the detailed classification reports for the BoT-IoT and the CIC-IDS2017 datasets, respectively. These tables provide precision, recall, and F1-score for each class, offering a comprehensive evaluation of the per-class detection performance of the proposed model. The consistently high metrics across all classes, including both majority and minority categories, highlight the robust effectiveness and generalizability of the proposed methodology in multiclass intrusion detection scenarios.

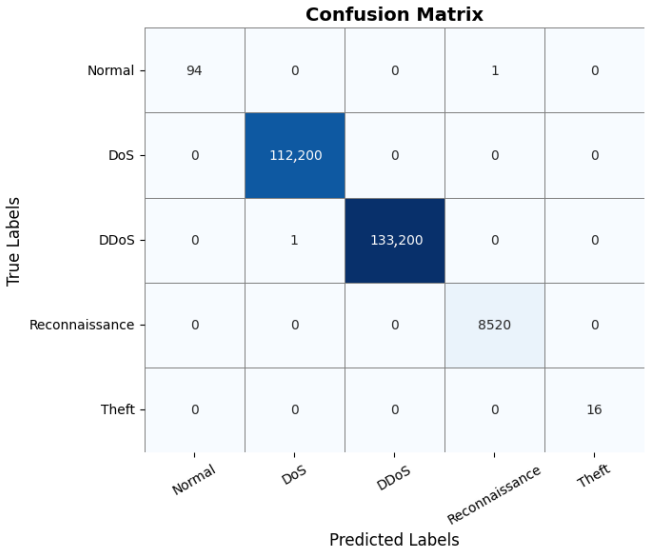


Figure 9. Confusion Matrix Utilizing the BoT-IoT Dataset (Multiclass Classification).

To ensure a fair comparison, the baseline models, LSTM and GRU, were trained using the same preprocessing steps and training hyperparameters as the proposed model, including data cleaning, recurrent layers, data partitioning, batch size, learning rate, dropout, and epochs. Also, for the ablation experiments, the same training settings were maintained, while the architectural components within the study were selectively removed according to the comparison conducted.

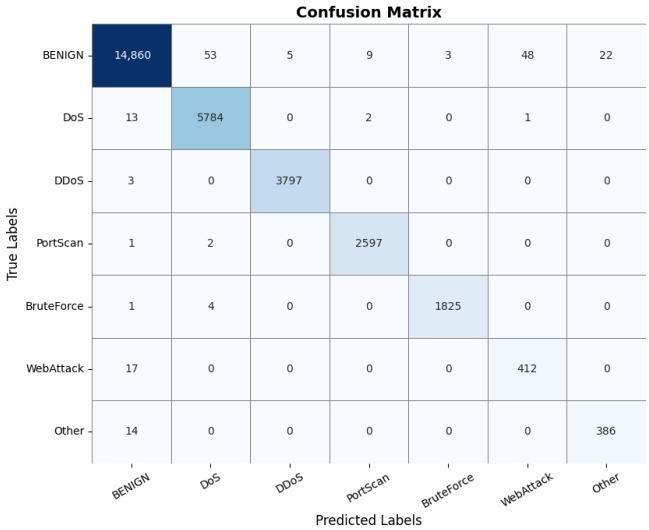


Figure 10. Confusion Matrix Utilizing the CIC-IDS2017 Dataset (Multiclass Classification).

Table 7. Classification Report for the BoT-IoT Dataset (Multiclass).

Class	Precision	Recall	F1-Score
Normal	1.0000	0.9895	0.9947
DoS	1.0000	1.0000	1.0000
DDoS	1.0000	1.0000	1.0000
Reconnaissance	0.9999	1.0000	0.9999
Theft	1.0000	1.0000	1.0000

Table 8. Classification Report for the CIC-IDS2017 Dataset (Multiclass).

Class	Precision	Recall	F1-Score
BENIGN	0.9967	0.9907	0.9937
DoS	0.9899	0.9972	0.9936
DDoS	0.9987	0.9992	0.9989
PortScan	0.9958	0.9988	0.9973
BruteForce	0.9984	0.9973	0.9978
WebAttack	0.8937	0.9604	0.9258
Other	0.9461	0.9650	0.9554

To further analyze the contribution of MHA, BPSO, and Target Sampling within the proposed BPSO-MHA-GRU Target Sampling framework, we conducted a comparison by benchmarking the proposed model against baseline LSTM and GRU models using the same evaluation metrics. Both baseline models consisted of two recurrent layers with 128 neurons in the first layer and 64 neurons in the second, each followed by a dropout rate of 0.2, corresponding exactly to the recurrent depth of the proposed model. The baseline models used the full datasets, excluding BPSO and the Target Sampling stages, with identical preprocessing steps, data partitioning, and the same hyperparameters of the proposed model, including a batch size of 64, 50 training epochs, and a learning rate of 0.01. This comparison was conducted to evaluate the effectiveness of the proposed framework compared to recurrent baseline models under identical preprocessing and training settings.

The results of this analysis are outlined in Table 9, reporting the mean $\pm$ standard deviation over three independent runs, to evaluate the robustness and stability of the proposed model. On both datasets, the proposed model achieved the best overall performance. On the BoT-IoT dataset, the proposed model achieved 99.72 ± 0.38 precision, 99.65 ± 0.20 recall, and

99.68 ± 0.17 F1-score, outperforming both LSTM, which achieved 98.99 ± 0.82, 96.24 ± 0.45, and 97.48 ± 0.12, respectively, and GRU, which achieved 98.72 ± 0.62, 96.72 ± 0.55, and 97.61 ± 0.08, respectively. Furthermore, on the CIC-IDS2017 dataset, the proposed model achieved 97.13 ± 0.22 precision, 98.63 ± 0.12 recall, and 97.86 ± 0.13 F1-score, outperforming the LSTM and GRU baselines under the same preprocessing and training settings. In addition to the performance improvement, the proposed model reduced the average training time over three runs by approximately 18% compared to both the LSTM and GRU baseline models on the BoT-IoT dataset and approximately 71% reduction on the CIC-IDS2017 dataset. The improved performance of the proposed model is attributed to the effective integration of the BPSO optimization algorithm and the MHA-stacked GRU architecture, which together enable high classification performance with lower computational overhead. Moreover, the Target Sampling strategy rebalances imbalanced class distributions, mitigating majority class bias without introducing synthetic records.

Table 9. Comparative Performance of the Proposed, LSTM, and GRU Models on the BoT-IoT and CIC-IDS2017 Datasets (Multiclass Classification).

Metric	Proposed	LSTM	GRU
BoT-IoT			
Precision	99.72 ± 0.38	98.99 ± 0.82	98.72 ± 0.62
Recall	99.65 ± 0.20	96.24 ± 0.45	96.72 ± 0.55
F1-Score	99.68 ± 0.17	97.48 ± 0.12	97.61 ± 0.08
Average Training Time (s)	9490	11,603	11,695
CIC-IDS2017			
Precision	97.13 ± 0.22	95.29 ± 0.85	95.04 ± 0.80
Recall	98.63 ± 0.12	94.66 ± 0.11	94.87 ± 0.13
F1-Score	97.86 ± 0.13	94.24 ± 0.60	94.09 ± 0.57
Average Training Time (s)	1134	3937	4293

Figure 11 provides a visual illustration of the multiclass classification results reported in Table 9 for the proposed, LSTM, and GRU models on the BoT-IoT dataset and the CIC-IDS2017 dataset. For each evaluation and model, the bar height represents the mean performance over three independent runs, while the black error bars represent the standard deviation. The three runs were conducted using fixed random seeds of 40, 32, and 20; the same seeds were also used in the subsequent ablation studies. The proposed model achieves the highest mean performance with low deviation across runs, indicating robust and stable intrusion detection performance under the class imbalance of the BoT-IoT dataset and under more diverse attack scenarios of the CIC-IDS2017 dataset. In contrast, LSTM and GRU models show performance changes throughout the runs, especially in recall and F1-score, indicating less stability and increased risk of missing intrusions. These findings highlight the advantage of combining optimized feature selection methods with attention mechanisms within recurrent networks for reliable IDS.

To further investigate the contribution of the MHA and the BPSO components within the proposed BPSO-MHA-GRU Target Sampling framework, an ablation study was conducted by removing both the MHA layer and BPSO feature selection while retaining the stacked GRU layers. The baseline model without MHA and BPSO, referred to as GRU-Target Sampling, utilized the same deep learning architecture, preprocessing steps, dropout, and training hyperparameters of the proposed model, consisting of two stacked GRU layers and including the Target Sampling strategy. Table 10 presents the multiclass classification performance comparison for both datasets, reported as mean ± standard deviation over three independent runs. The proposed IDS outperformed the baseline

GRU-Target Sampling in terms of recall and F1-score on both datasets. For the BoT-IoT dataset, the proposed model achieved a recall of 99.65 ± 0.20 and an F1-score of 99.68 ± 0.17, compared to 95.94 ± 0.20 recall and 96.90 ± 0.55 F1-score for the baseline model. On the CIC-IDS2017 dataset, the proposed model achieved a recall of 98.63 ± 0.12 and an F1-score of 97.86 ± 0.13, outperforming the GRU-Target Sampling baseline model, which achieved 95.86 ± 1.68 recall and 96.04 ± 0.61 F1-score, respectively. Figure 12 provides a visual illustration of the results reported in Table 10. The proposed model achieves higher mean performance with lower deviation across recall and F1-score, confirming the effectiveness of the proposed framework.

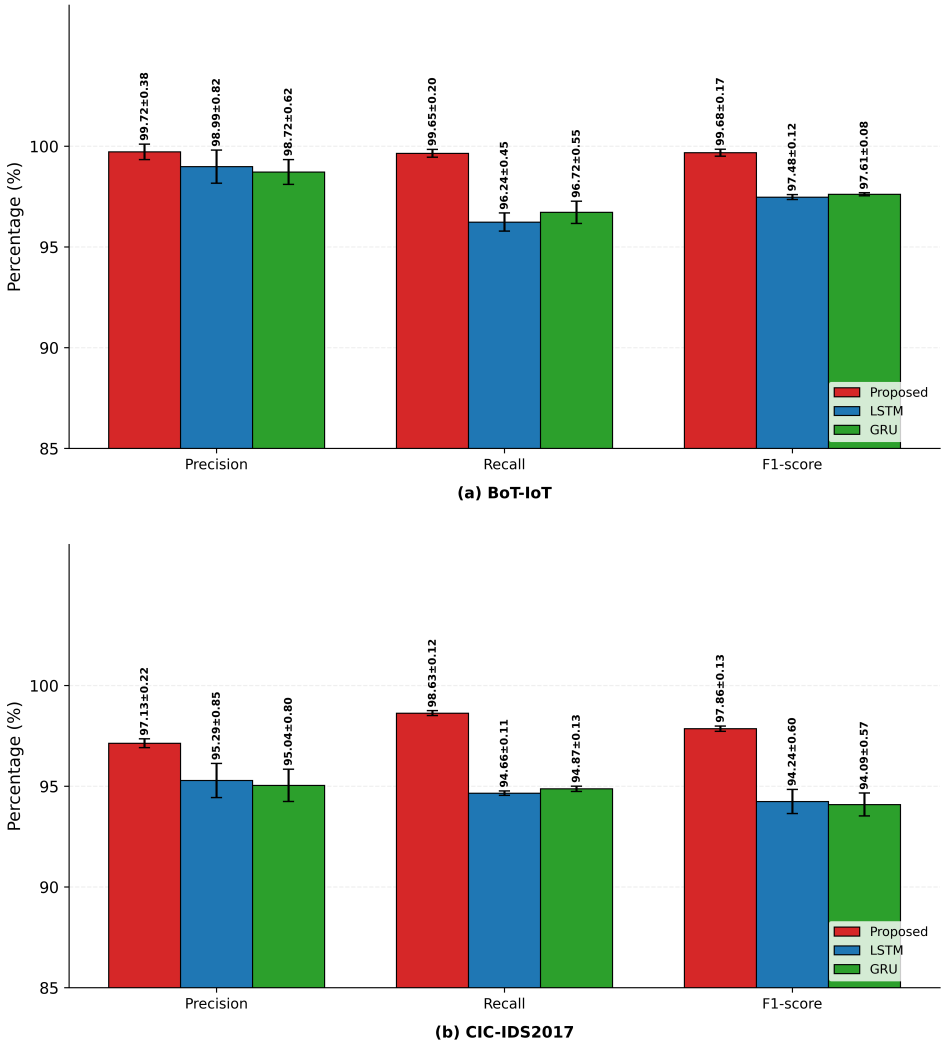


Figure 11. Multiclass Classification Performance Comparison of the Proposed, LSTM, and GRU models on (**a**) the BoT-IoT dataset and (**b**) the CIC-IDS2017 dataset.

Table 10. Comparative Performance of the Proposed Model and the GRU-Target Sampling Model (Multiclass Classification).

	BoT-IoT		**CIC-IDS2017**	
Metric	**Proposed**	**GRU-Target Sampling**	**Proposed**	**GRU-Target Sampling**
Precision	99.72 ± 0.38	98.15 ± 1.17	97.13 ± 0.22	96.60 ± 0.65
Recall	99.65 ± 0.20	95.94 ± 0.20	98.63 ± 0.12	95.86 ± 1.68
F1-Score	99.68 ± 0.17	96.90 ± 0.55	97.86 ± 0.13	96.04 ± 0.61

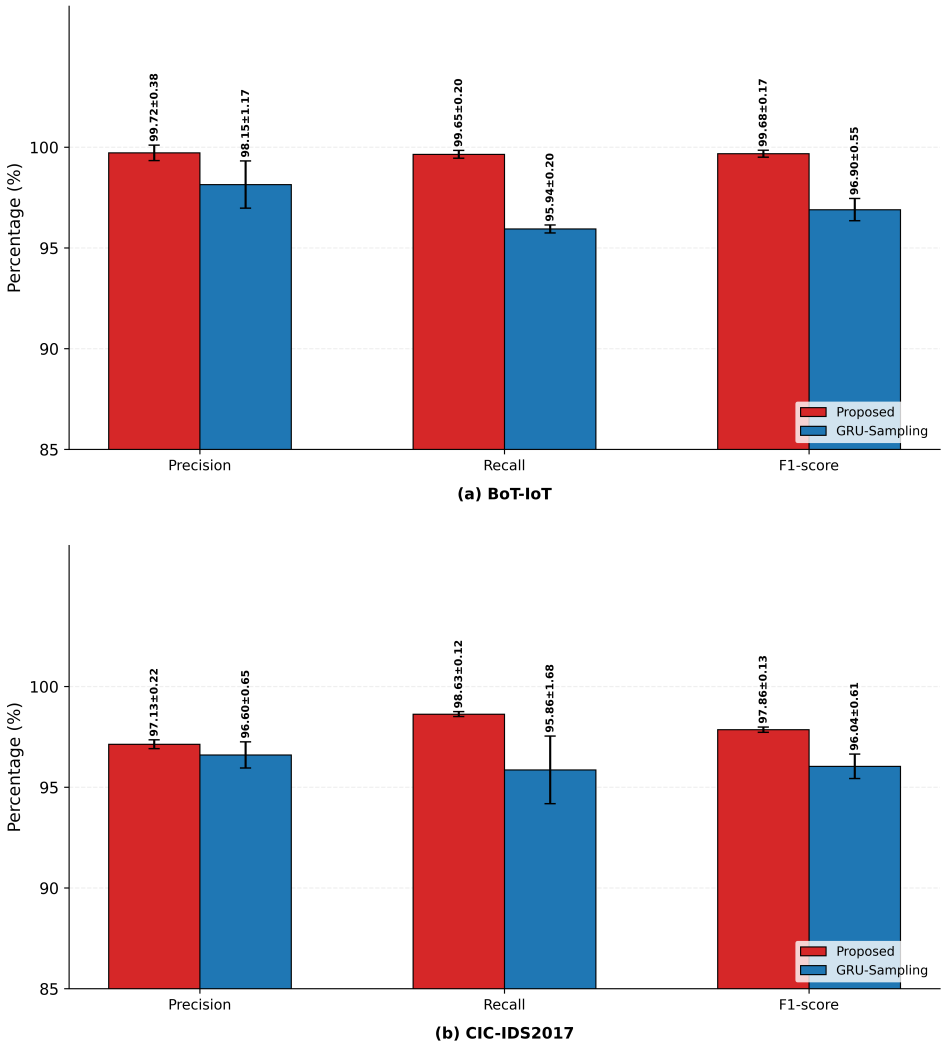


Figure 12. Multiclass Classification Performance Comparison of the Proposed and GRU-Target Sampling models on (**a**) the BoT-IoT dataset and (**b**) the CIC-IDS2017 dataset.

Additionally, we conducted an ablation study by removing the MHA layer only while keeping BPSO-GRU and Target Sampling to evaluate the contribution of the MHA component in the proposed model, comparing it to the proposed BPSO-MHA-GRU Target Sampling model under the same preprocessing steps, training hyperparameters, and experimental process for both. Table 11 demonstrates the multiclass classification performance comparison for both datasets, reported as mean ± standard deviation over three independent runs. The proposed model outperformed the baseline BPSO-GRU with Target Sampling model, demonstrating the effectiveness of incorporating the MHA mechanism in the proposed model. For the BoT-IoT dataset, the proposed model achieved a recall of 99.65 ± 0.20 and an F1-score of 99.68 ± 0.17, compared to 97.47 ± 0.45 recall and 96.99 ± 0.48 F1-score achieved by the baseline BPSO-GRU with Target Sampling model. For the CIC-IDS2017 dataset, the proposed model achieved a recall of 98.63 ± 0.12 and an F1-score of 97.86 ± 0.13, compared to 95.72 ± 1.04 recall and 95.88 ± 0.61 F1-score for the baseline model. Figure 13 provides a visual illustration of the results reported in Table 11. These results highlight the advantage of integrating the MHA mechanism with BPSO-based feature selection, improving detection capability and robustness of the proposed model in computer networks and IoT environments.

Table 11. Comparative Performance of the Proposed Model and the BPSO-GRU-Target Sampling Model (Multiclass Classification).

Metric	Proposed	BPSO-GRU-Target Sampling
BoT-IoT		
Precision	99.72 ± 0.38	96.62 ± 0.54
Recall	99.65 ± 0.20	97.47 ± 0.45
F1-Score	99.68 ± 0.17	96.99 ± 0.48
CIC-IDS2017		
Precision	97.13 ± 0.22	96.47 ± 0.68
Recall	98.63 ± 0.12	95.72 ± 1.04
F1-Score	97.86 ± 0.13	95.88 ± 0.61

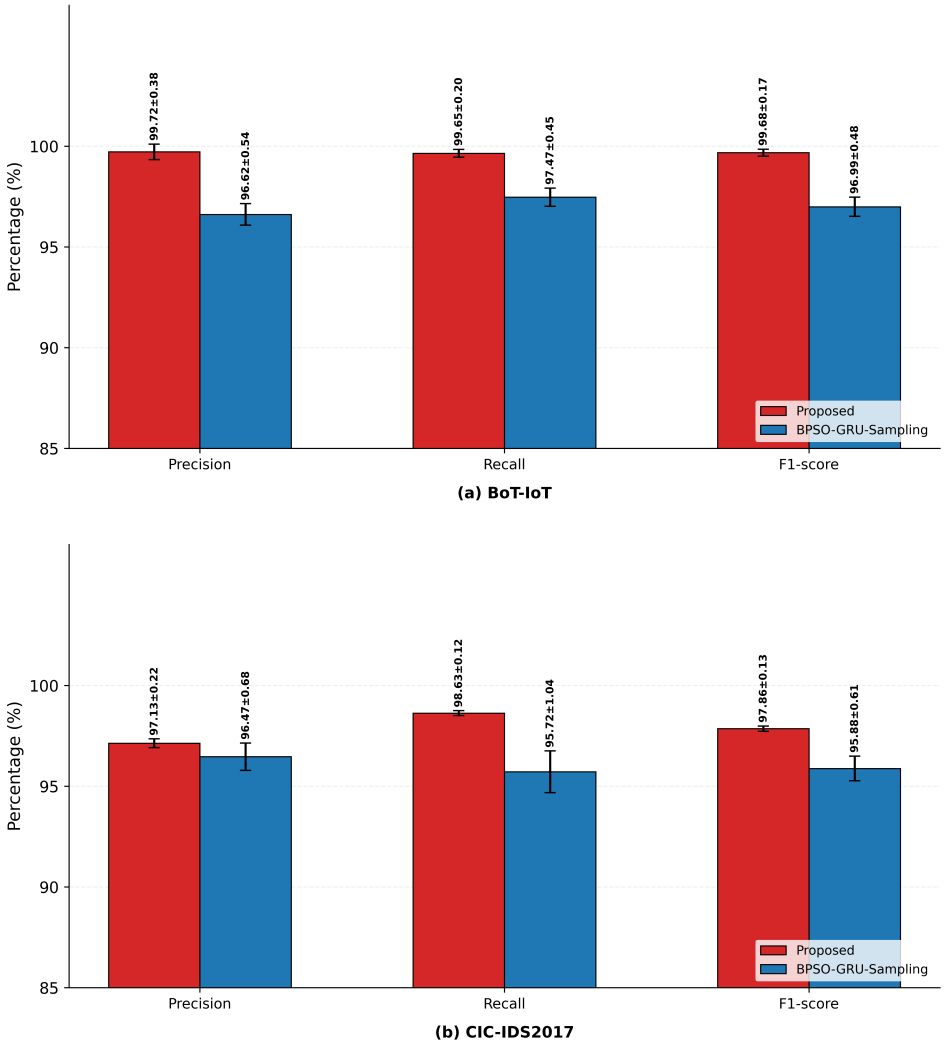


Figure 13. Multiclass Classification Performance Comparison of the Proposed and BPSO-GRU-Target Sampling models on (**a**) the BoT-IoT dataset and (**b**) the CIC-IDS2017 dataset.

The comparisons reported in Tables 12–15 are literature based comparisons using performance values reported in the respective original studies. To further demonstrate the effectiveness of our proposed model, we conducted a performance comparison against recent state-of-the-art IDS models. The comparison covered both binary and multiclass classification tasks utilizing the BoT-IoT and CIC-IDS2017 datasets. Tables 12 and 13 present the binary classification performance analysis on both datasets, respectively. Our

model achieved a precision of 99.46% and F1-score of 98.94% on the BoT-IoT dataset, outperforming existing models such as LSTM-GRU F1-score of 98.68% and CNN-LSTM with 97.50%. Similarly, on the CIC-IDS2017 dataset, our approach achieved a precision of 99.76% and F1-score of 99.76%, outperforming other existing models, including, GAN-CNN-BiLSTM with F1-score of 96.04%, ODODL-IDS with 94.17%.

Table 12. Binary Classification Performance Comparison on the BoT-IoT Dataset.

Model	Year	F1-Score	Recall	Precision	Accuracy
Proposed Model	2025	98.94	98.42	99.46	99.99
Deep LSTM+GRU [34]	2025	98.68	98.25	97.81	98.92
CNN-LSTM (PSO+GTO) [35]	2025	97.50	97.50	97.50	97.50

Table 13. Binary Classification Performance Comparison on the CIC-IDS2017 Dataset.

Model	Year	F1-Score	Recall	Precision	Accuracy
Proposed Model	2025	99.76	99.76	99.76	99.76
GAN-CNN-BiLSTM [36]	2023	96.04	95.38	96.55	96.32
ODODL-IDS [37]	2022	94.17	98.92	99.85	97.09

For multiclass classification, Tables 14 and 15 illustrate the proposed model's superior capability in handling multiclass classification tasks, achieving 99.79% recall and 99.89% F1-score on the BoT-IoT dataset and 98.69% recall and 98.04% F1-score on the CIC-IDS2017 dataset. These results are higher than those reported by other existing models, including RIDGE 93.00%, CSMCR 92.75%, Attention-RNN 98.94%, CNN-IoT 72.60% on the BoT-IoT dataset, and compared to DAMML 89.33%, S2CGAN 92.00% and CNN 1D-BLSTM 88.00% on the CIC-IDS2017 dataset. These results validate the robustness and generalization capacity of the proposed IDS across diverse network environments and attack scenarios.

Table 14. Multiclass Classification Performance Comparison on the BoT-IoT Dataset.

Model	Year	F1-Score	Recall	Precision	Accuracy
Proposed Model	2025	99.89	99.79	99.99	99.99
RIDGE [19]	2025	93.00	93.00	93.00	93.36
CSMCR [21]	2025	92.75	86.91	99.44	91.10
Caps+Attention-RNN [38]	2025	98.94	98.63	98.50	98.60
CNN 1D-BLSTM [39]	2023	79.00	78.00	80.00	97.00
CNN-IoT [40]	2022	72.60	72.80	72.60	95.55

Table 15. Multiclass Classification Performance Comparison on the CIC-IDS2017 Dataset.

Model	Year	F1-Score	Recall	Precision	Accuracy
Proposed Model	2025	98.04	98.69	97.42	99.34
DAMML [15]	2025	89.33	90.00	90.00	91.33
SMOTE-Tomek [41]	2024	96.33	96.33	93.00	96.37
S2CGAN-IDS [28]	2024	92.00	92.00	93.00	Not Rep.
CNN 1D-BLSTM [39]	2023	88.00	84.00	86.00	98.00

6. Discussion

The comprehensive comparative analysis conducted in this research highlights the outstanding performance of the proposed BPSO-MHA-GRU model. Its superior performance in both binary and multiclass classification tasks confirms its enhanced ability to detect intrusions more effectively than existing state-of-the-art techniques. Consistent

improvements in macro F1-score and recall across both the BoT-IoT and the CIC-IDS2017 indicate fewer missed attacks and stronger minority class detection, which are critical for effective and reliable IDS development in imbalanced environments.

This high detection performance is achieved through a combination of efficient feature selection utilizing the BPSO, selecting the most informative features and removing irrelevant ones, and through the MHA mechanism enabling the model to dynamically focus on important components of the sequence of hidden states, enhancing identification of complex attack patterns. Demonstrating that integrating feature selection with a recurrent model enhanced by MHA not only reduces input dimensionality but also strengthens the model's capability to capture behavior variations within network traffic data. Additionally, a Target Sampling strategy was employed to address dataset class imbalance by selectively downsampling majority classes while retaining minority instances without synthetic data generation that further enhances the proposed model's ability to detect important rare intrusions. These results set the proposed model as a promising solution for intrusion detection in both computer networks and IoT environments. To clarify the evaluation process, after preprocessing and Target Sampling, the class distributions are adjusted, as shown in Tables 1 and 2, having the majority classes downsampled while retaining all minority records. Both datasets are then split into separate training, validation, and testing subsets. The test subset is never used during the optimization and training process and kept unseen until the final performance evaluation. The BPSO feature selection is conducted using only the training and validation subsets.

Despite the strong performance, this study faces certain limitations. The evaluation relies primarily on two benchmark datasets, which may limit generalizability to other environments. The detection of zero-day attacks is still an ongoing challenge in anomaly-based IDS. The model relies on labeled data for supervised learning, which poses challenges due to the high cost and difficulty of obtaining such datasets.

For future work, we aim to investigate the integration of different feature selection techniques, including mutual information, random forest, and genetic algorithms, as well as hybrid combinations to optimize model performance further. A sensitivity analysis of the BPSO parameters will be explored in future work to better assess their effect. We will also extend our investigation into different types of attention-based transformer architectures to evaluate their effectiveness within imbalanced datasets. Additionally, future work will evaluate inference latency and resource usage to fully characterize deployment-time computational performance. Furthermore, expanding the evaluation of the proposed IDS framework utilizing different IoT and network security datasets, such as ToN-IoT and CSE-IDS2018.

7. Conclusions

The growing occurrence of cyberattacks and the expansion of interconnected devices emphasize the urgent need for advanced IDS mechanisms. This research presents a hybrid DL-based IDS framework integrating the BPSO optimal feature selection with an attention-stacked GRU architecture for effective feature representation and classification. The proposed BPSO-MHA-GRU framework employs BPSO to identify the most discriminative features from high-dimensional network traffic, thereby reducing the training computational overhead while preserving critical information. The MHA-stacked GRU architecture efficiently performs sequence processing for network traffic classification, ensuring accurate and reliable intrusion detection in both binary and multiclass classification scenarios. Additionally, a target class-aware sampling strategy is implemented to mitigate class imbalance, enhancing the model's sensitivity to rare but critical attack types without relying on synthetic data generation. Extensive experimental evaluations on the BoT-IoT

and CIC-IDS2017 benchmark datasets demonstrate the effectiveness of the proposed model. In binary classification, the proposed model achieved high F1-score of 98.94% and 99.76% on BoT-IoT and CIC-IDS2017, respectively. In multiclass classification tasks, the proposed approach achieved a recall of 99.79% and 98.69% with high macro F1-scores of 99.89% and 98.04% on the BoT-IoT and CIC-IDS2017 datasets, respectively, confirming its robustness and adaptability across various attack scenarios. Comparative analysis further demonstrates that the proposed model outperforms traditional LSTM and GRU models, as well as other recently developed state-of-the-art IDS techniques, in terms of detection accuracy, precision, recall, and F1-score. This hybrid approach delivers state-of-the-art detection performance while maintaining computational efficiency, positioning it as a promising solution for addressing modern cybersecurity challenges in both computer networks and IoT environments.

Author Contributions: Conceptualization, A.E. and M.K.; methodology, A.E.; software, A.E.; validation, A.E. and M.K.; formal analysis, A.E.; investigation, A.E.; resources, A.E.; data curation, A.E.; writing—original draft preparation, A.E.; writing—review and editing, A.E. and M.K.; visualization, A.E.; supervision, M.K.; project administration, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: BoT-IoT dataset https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 1 February 2025), CIC-IDS2017 dataset https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 1 February 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376.
2. CyberEdge. 2019 Cyberthreat Defense Report. 2019. Available online: https://go.illusive.com/2019-cyberthreat-defense-report (accessed on 1 January 2025).
3. Otoum, Y.; Nayak, A. On securing IoT from deep learning perspective. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
4. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 1–27.
5. Khan, K.; Mehmood, A.; Khan, S.; Khan, M.A.; Iqbal, Z.; Mashwani, W.K. A survey on intrusion detection and prevention in wireless ad-hoc networks. *J. Syst. Archit.* **2020**, *105*, 101701.
6. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150.
7. Li, H.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Netw.* **2018**, *32*, 96–101.
8. Thakkar, A.; Lohiya, R. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artif. Intell. Rev.* **2022**, *55*, 453–563.
9. Armellin, A.; Caviglia, R.; Gaggero, G.; Marchese, M. A framework for the deployment of cybersecurity monitoring tools in the industrial environment. *IT Prof.* **2024**, *26*, 62–70.
10. Gamage, S.; Samarabandu, J. Deep learning methods in network intrusion detection: A survey and an objective comparison. *J. Netw. Comput. Appl.* **2020**, *169*, 102767.
11. Khan, F.A.; Gumaei, A.; Derhab, A.; Hussain, A. A novel two-stage deep learning model for efficient network intrusion detection. *IEEE Access* **2019**, *7*, 30373–30385.
12. Otoum, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning-based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3803.
13. Shoab, M.; Alsbatin, L. GRU enabled intrusion detection system for IoT environment with swarm optimization and Gaussian random forest classification. *Comput. Mater. Contin.* **2024**, *81*, 1.

14. Gao, J.; Chai, S.; Zhang, B.; Xia, Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. *Energies* **2019**, *12*, 1223.
15. Varghese, M.U.; Taghiyarrenani, Z. Intrusion detection in heterogeneous networks with domain-adaptive multi-modal learning. *arXiv* **2025**, arXiv:2508.03517.
16. Zeeshan, M.; Riaz, Q.; Bilal, M.A.; Shahzad, M.K.; Jabeen, H.; Haider, S.A.; Rahim, A. Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and BoT-IoT data-sets. *IEEE Access* **2021**, *10*, 2269–2283.
17. Fu, Y.; Du, Y.; Cao, Z.; Li, Q.; Xiang, W. A deep learning model for network intrusion detection with imbalanced data. *Electronics* **2022**, *11*, 898.
18. Abdelkhalek, A.; Mashaly, M. Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. *J. Supercomput.* **2023**, *79*, 10.
19. Nemalikanti, A.; Kaki, S.; Ambati, R.R.; Ponnuru, R.B. Enhancing intrusion detection: Protocol-based security using a hybrid RIDGE classifier on InSDN, UNSW-NB15, BoT-IoT, and ToN-IoT datasets. *Clust. Comput.* **2025**, *28*, 663.
20. Dhirar, H.; Hamad, A. Comparative evaluation of a novel IDS dataset for SDN-IoT using deep learning models against InSDN, BoT-IoT, and ToN-IoT. *Meas. Digit.* **2025**, *4*, 100015.
21. Prasad, A.; Alenazy, W.M.; Ahmad, N.; Ali, G.; Abdallah, H.A.; Ahmad, S. Optimizing IoT intrusion detection with cosine similarity based dataset balancing and hybrid deep learning. *Sci. Rep.* **2025**, *15*, 30939.
22. Boahen, E.K.; Sosu, R.N.A.; Ocansey, S.K.; Xu, Q.; Wang, C. ASRL: Adaptive swarm reinforcement learning for enhanced OSN intrusion detection. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 10258–10272.
23. Okey, O.D.; Rodriguez, D.Z.; Kleinschmidt, J.H. Enhancing IoT intrusion detection with federated learning-based CNN-GRU and LSTM-GRU ensembles. In Proceedings of the 2024 19th International Symposium on Wireless Communication Systems (ISWCS), Rio de Janeiro, Brazil, 14–17 July 2024; pp. 1–6.
24. Chen, B.; Li, H.; Zhang, M.; Zhao, M.; Liang, Z.; Li, K.; An, X. Performance enhancement of deep learning model with attention mechanism and FCN model in flood forecasting. *J. Hydrol.* **2025**, *658*, 133221.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
26. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796.
27. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of ICISSp*; SciTePress: Madeira, Portugal, 2018; Volume 1, pp. 108–116.
28. Wang, C.; Xu, D.; Li, Z.; Niyato, D. Effective intrusion detection in highly imbalanced IoT networks with lightweight S2CGAN-IDS. *IEEE Internet Things J.* **2023**, *11*, 15140–15151.
29. Ennaji, S.; De Gaspari, F.; Hitaj, D.; Kbidi, A.; Mancini, L.V. Adversarial challenges in network intrusion detection systems: Research insights and future prospects. *IEEE Access* **2025**, *13*, 148613–148645.
30. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.
31. Liu, J.; Yang, D.; Lian, M.; Li, M. Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access* **2021**, *9*, 38254–38268. [CrossRef]
32. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078. [CrossRef]
33. Molina-Coronado, B.; Mori, U.; Mendiburu, A.; Miguel-Alonso, J. Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2451–2479. [CrossRef]
34. Luqman, M.; Zeeshan, M.; Riaz, Q.; Hussain, M.; Tahir, H.; Mazhar, N.; Khan, M.S. Intelligent parameter-based in-network IDS for IoT using UNSW-NB15 and BoT-IoT datasets. *J. Frankl. Inst.* **2025**, *362*, 107440. [CrossRef]
35. Al-Shurbaji, T.; Anbar, M.; Manickam, S.; Al-Amiedy, T.A.; Al Mukhaini, G.; Hashim, H.; Farsi, M.; Atlam, E.-S. BoT-EnsIDS: Approach for detecting IoT botnet attacks leveraging bio-inspired ensemble feature selection and a hybrid deep learning model. *Alex. Eng. J.* **2025**, *129*, 744–767. [CrossRef]
36. Li, S.; Li, Q.; Li, M. A method for network intrusion detection based on GAN-CNN-BiLSTM. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 5. [CrossRef]
37. Ragab, M.; Sabir, M.F.S. Outlier detection with optimal hybrid deep learning enabled intrusion detection system for ubiquitous and smart environment. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102311. [CrossRef]
38. Logeswari, G.; Roselind, J.D.; Tamilarasi, K.; Nivethitha, V. A comprehensive approach to intrusion detection in IoT environments using hybrid feature selection and multi-stage classification techniques. *IEEE Access* **2025**, *13*, 24970–24987. [CrossRef]

39. Bowen, B.; Chennamaneni, A.; Goulart, A.; Lin, D. Blocnet: A hybrid, dataset-independent intrusion detection system using deep learning. *Int. J. Inf. Secur.* **2023**, *22*, 893–917. [CrossRef]
40. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
41. Khan, F.A.; Shah, A.A.; Alshammry, N.; Saif, S.; Khan, W.; Malik, M.O.; Ullah, Z. Balanced multi-class network intrusion detection using machine learning. *IEEE Access* **2024**, *12*, 178222–178236. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.