ITC 1/55
Information Technology and Control
Vol. 55 / No. 1/ 2026
pp. 1-19
DOI 10.5755/j01.itc.55.1.42179

Hierarchical Deep Learning with Nature-Inspired Optimization for Robust Network Intrusion Detection

Received 2025/07/09 | Accepted after revision 2025/10/09

HOW TO CITE: Li, H. (2026). Hierarchical Deep Learning with Nature-Inspired Optimization for Robust Network Intrusion Detection. *Information Technology and Control*, 55(1), 1-19. https://doi.org/10.5755/j01.itc.55.1.42179

Hierarchical Deep Learning with Nature-Inspired Optimization for Robust Network Intrusion Detection

He Li
CETC Potevio Science & Technology Co.,Ltd., Haizhu District, Guangzhou City, Guangdong, 510300, China

Corresponding author: lhaqzjsci@126.com

This study proposes a Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) that combines hierarchical deep learning and swarm intelligence-based feature optimization to enhance network security. Our framework integrates Convolutional Neural Networks (CNNs) for packet-level anomaly detection, Long Short-Term Memory (LSTM) networks for session-level behavior analysis, and fuzzy logic-based context analyzers to minimize false alarms. Feature selection is optimized through bio-inspired Elephant Herding Optimization (EHO), improving classification accuracy while reducing computational overhead. Evaluation on CIC-IDS2017 dataset shows that SML-IDS outperforms conventional IDS models, achieving superior detection accuracy, false positive rate reduction, and real-time feasibility.

KEYWORDS: Network Intrusion Detection, Cybersecurity, Deep Learning, Feature Optimization, Hierarchical Deep Learning.

1. Introduction

The increasing interconnectivity of devices in modern computing environments has facilitated the rapid adoption of the Internet of Things (IoT), enabling seamless communication across various domains, including healthcare, manufacturing, smart cities, and critical infrastructure [21]. IoT networks have transformed the digital landscape by enabling real-time data acquisition, automation, and remote control of devices across various applications [3]. However, their deployment in mission-critical domains such

as healthcare (Internet of Medical Things – IoMT), industrial control systems, and smart grids introduces significant security and privacy concerns [31]. Unlike traditional computing infrastructures, IoT devices often have limited computational resources, making them vulnerable to lightweight yet highly effective attacks such as distributed denial-of-service (DDoS), packet injection, eavesdropping, and adversarial manipulations [29]. The heterogeneity of IoT devices, coupled with diverse communication protocols, complicates the deployment of standardized security solutions [40].

Recent advancements in artificial intelligence (AI) and deep learning (DL) have shown promise in addressing IoT security challenges [25]. However, conventional deep learning-based Intrusion Detection System (IDS) models often suffer from high false-positive rates, adversarial vulnerabilities, and inefficient feature selection mechanisms [2]. Existing feature extraction techniques rely heavily on domain knowledge, limiting the adaptability of IDS frameworks to emerging attack patterns [8]. Bioinspired optimization algorithms, particularly Elephant Herding Optimization (EHO), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), offer a promising alternative by dynamically selecting and optimizing feature subsets for improved classification accuracy [3].

Despite advancements in IoT security research, several challenges persist, preventing the deployment of highly reliable and scalable IDS solutions [16]. One of the primary challenges is the dynamic nature of cyber threats, where traditional signature-based detection methods fail to identify novel and zero-day attacks [24]. Attackers increasingly use polymorphic malware, adversarial perturbations, and sophisticated evasion techniques to bypass conventional security mechanisms [14]. Polymorphic malware dynamically alters its code to avoid signature-based detection, and adversarial perturbations use carefully crafted inputs exploit the vulnerabilities of deep learning models to evade anomaly detection.

Another challenge lies in the decentralized and heterogeneous nature of IoT environments, where vendors use proprietary communication protocols and security mechanisms [12]. This fragmentation creates interoperability issues, making it difficult to establish unified security policies.

Traditional security mechanisms such as rule-based intrusion detection systems (IDS) and conventional firewalls have proven inadequate in handling the dynamic and evolving nature of cyber threats. The necessity for an advanced, adaptive, and intelligent security framework has driven researchers to explore machine learning and bio-inspired algorithms for real-time intrusion detection. The evolving nature of cyber threats, particularly zero-day attacks, poses significant challenges to traditional signature-based Intrusion Detection Systems (IDS). These systems rely on predefined attack signatures and therefore struggle to detect previously unseen or rapidly mutating threats that do not match known patterns. As attackers increasingly employ techniques such as polymorphism, obfuscation, and automated exploit generation, signature-based IDS become less effective due to their limited adaptability. To address these challenges, we propose a Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) that integrates bio-inspired optimization techniques and hierarchical deep learning. The key contributions of this research are summarized as follows:

- We introduce a feature selection mechanism that uses Elephant Herding Optimization (EHO) [39] to enhance the accuracy of IDS models. This approach ensures optimal feature selection, reducing computational overhead while maintaining detection robustness.
- A multi-layered deep learning framework that integrates Convolutional Neural Networks (CNNs) for packet-level detection, Long Short-Term Memory (LSTM) for session-level behavior analysis, and temporal modeling for sequential attack pattern identification.
- To improve resilience against adversarial attacks, we incorporate Generative Adversarial Networks (GANs) for crafting adversarial samples, thereby enhancing the robustness of the IDS against evasion techniques.
- A fuzzy logic-based context analyzer to prioritize intrusion alerts based on severity, impact on network segments, and historical attack patterns. This mechanism significantly reduces false positive rates and improves decision-making in automated security responses.

2. Related Work

With the growth of IoT networks, security mechanisms have struggled to keep pace with increasingly sophisticated cyber threats. Traditional security solutions, such as firewalls and rule-based intrusion detection systems (IDS), are often inadequate in protecting resource-constrained IoT devices from evolving attack patterns [25]. Recent research has focused on enhancing IDS through machine learning (ML), deep learning (DL), and bio-inspired optimization techniques to improve accuracy, efficiency, and adaptability [3]. This section provides an overview of existing IDS models in IoT environments, explores the role of ML and DL in enhancing detection capabilities, examines bio-inspired optimization strategies for intrusion detection, and discusses the limitations of existing approaches.

Intrusion Detection Systems (IDS) serve as a critical layer of defense in network security, aiming to detect, analyze, and respond to malicious activities within IoT ecosystems [8]. IDS mechanisms are broadly classified into two categories: signature-based detection and anomaly-based detection [40]. Signature-based IDS rely on predefined attack signatures to detect known threats, offering high accuracy for previously encountered attacks but failing to identify zero-day exploits [24]. In contrast, anomaly-based IDS monitor deviations from normal network behavior, making them more effective in detecting novel attacks [16]. However, traditional anomaly detection systems often generate a high number of false positives due to their reliance on predefined normal behavior patterns.

IoT environments introduce unique challenges for IDS deployment due to device heterogeneity, limited computational resources, and dynamic network topologies [9]. Researchers have proposed lightweight IDS frameworks specifically designed for IoT networks, incorporating techniques such as feature reduction, hybrid detection models, and edge computing integration [12]. Cloud-assisted IDS solutions have also emerged, leveraging centralized processing to analyze network traffic from multiple IoT devices [18]. However, these approaches introduce latency and privacy concerns, necessitating more efficient, decentralized, and real-time intrusion detection mechanisms.

The integration of machine learning (ML) and deep learning (DL) techniques has significantly enhanced the effectiveness of IDS in identifying cyber threats [7]. ML-based IDS models utilize classification algorithms such as Support Vector Machines (SVM), Decision Trees (DT), Random Forest (RF), and k-Nearest Neighbors (KNN) to detect anomalous patterns in network traffic [13]. These models have demonstrated notable improvements in detection accuracy compared to traditional rule-based IDS, however, their reliance on handcrafted features and predefined decision boundaries limits adaptability to new attack vectors [17].

Deep learning approaches, particularly Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), LSTM networks, and Autoencoders, have shown promise in addressing the limitations of conventional ML models [23]. CNNs have been successfully applied for packet-level intrusion detection by extracting spatial features from network traffic data [33]. RNNs and LSTMs, on the other hand, are well-suited for sequential attack detection, leveraging temporal dependencies to identify malicious activities over time [19]. Variants such as GAN-augmented training and hybrid DL architectures have further improved robustness by mitigating adversarial attacks and enhancing generalization [32].

Deep learning models have several challenges, including high computational complexity, data dependency, and susceptibility to adversarial manipulation [5]. Training deep learning-based IDS requires extensive labeled datasets, which are often scarce, imbalanced, or outdated. Adversarial attack techniques can exploit model vulnerabilities, leading to misclassification and evasion. To address these issues, researchers have explored the integration of bio-inspired optimization algorithms to improve feature selection, enhance model efficiency, and reduce false positive rates [36].

Bio-inspired optimization algorithms gained increasing attention in IDS research due to their ability to efficiently select relevant features, optimize hyperparameters, and improve accuracy [38]. These algorithms mimic natural evolutionary and swarm behaviors to explore the solution space and refine IDS models [20]. Several bio-inspired approaches have been explored in intrusion detection, including:

- Genetic Algorithms (GA) is used for feature selection and rule optimization in IDS models, enhancing attack classification efficiency [15].
- Ant Colony Optimization (ACO) is used to optimize routing and intrusion detection in IoT by modeling attack path probabilities [41].
- Particle Swarm Optimization (PSO) is applied for feature selection and hyperparameter tuning in ML-based IDS, improving detection accuracy while reducing model complexity [4].
- Elephant Herding Optimization (EHO) is inspired by elephant social behaviors, used to select optimal feature subsets and improve IDS performance [39]. EHO has already been successfully used in intrusion detection tasks [26, 27, 30].
- Artificial Bee Colony (ABC) is used in IDS for attack classification by improving convergence speed and model adaptability [28].

Bio-inspired optimization techniques have demonstrated significant improvements in IDS performance by reducing dimensionality, enhancing feature selection, and accelerating convergence in ML and DL models [11]. However, their effectiveness depends on careful parameter tuning and adaptation to the dynamic nature of IoT security environments.

Despite notable advancements in IDS research, several limitations persist, hindering the deployment of fully reliable and scalable intrusion detection solutions in IoT networks. The challenges include:

- Many anomaly-based IDS models generate false alarms due to their sensitivity to minor deviations in network traffic. This leads to alert fatigue and reduces operational efficiency.
- Traditional IDS frameworks often struggle to provide real-time intrusion detection, especially in resource-constrained IoT environments. Computationally expensive deep learning models introduce latency, making them unsuitable for low-power IoT devices.
- Deep learning-based IDS models are susceptible to adversarial attacks, where small perturbations in network traffic can deceive the model into misclassifying threats. Existing models lack robust adversarial training mechanisms to counter evasion techniques.
- The growing volume of IoT devices generates massive network traffic, overwhelming conventional IDS solutions. Cloud-based IDS architectures introduce privacy risks and dependence on centralized processing, making them less suitable for large-scale IoT deployments.
- Traditional IDS models rely on manually engineered features, which may not generalize well to evolving attack patterns. Feature extraction techniques need to be automated, adaptive, and computationally efficient.
- Many IDS frameworks focus on real-time detection but lack post-incident analysis mechanisms. The absence of secure forensic logging and traceability makes it difficult to investigate security breaches and implement proactive mitigation strategies.

3. Proposed Framework

The increasing complexity and scale of cyber threats in IoT environments, as stated in recent Cisco report [6], necessitate the development of adaptive, scalable, and intelligent intrusion detection mechanisms. To address these challenges, we propose a Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) that integrates bio-inspired optimization, deep learning, and hierarchical security analysis to enhance detection accuracy while minimizing false positives. The proposed framework consists of three core components: (i) an optimized feature selection mechanism using EHO to identify the most relevant intrusion patterns, (ii) a multilayer deep learning model that employs CNN for packet-level analysis, LSTM networks to capture sequential dependencies in network traffic, enabling it to monitor long-term patterns and detect anomalies that unfold over time, (iii) sliding time window mechanism to adapt to dynamic threat patterns by continuously updating detection parameters based on recent activity, and (iv) a context-aware detection and alert mechanism to reduce false alarms and prioritize security responses.

3.1. System Architecture Overview

The proposed SML-IDS architecture is designed to enhance the efficiency of intrusion detection while ensuring real-time processing capabilities for IoT networks. The architecture is structured into four main layers, as depicted in Figure 1:

- Data Collection Layer captures network traffic data from IoT devices, including packet headers, payload information, and communication metadata. The collected data undergoes preprocessing to remove redundant features and normalize numerical attributes.
- Feature Optimization Layer uses the Elephant Herding Optimization (EHO) algorithm to the preprocessed data to perform dimensionality reduction while preserving high-informative features. This optimization enhances detection efficiency by reducing model complexity and computational overhead.
- Deep Learning-Based Intrusion Detection Layer is a hierarchical deep learning model employed for multi-layer intrusion analysis:
  - CNN for Packet-Level Detection extracts spatial features from network traffic. By applying convolutional filters across the input, CNNs can identify distinctive signatures and correlations associated with various types of malicious traffic, such as unusual port combinations, packet sizes, or protocol usage patterns. These learned spatial features provide a rich and compact representation of the packet data, which significantly enhances the accuracy and efficiency of the subsequent session-level and temporal analysis layers.
  - LSTM for Session-Level Analysis captures sequential dependencies in intrusion patterns. This enables the system to recognize complex attack sequences that involve stateful interactions or delayed malicious actions, and detect evolving and stealthy threats such as zero-day attacks.
  - Temporal Behavior Analysis monitors long-term attack behaviors to detect evolving threats.
- Alert and Response Layer includes a context-aware fuzzy logic mechanism to prioritize threats based on severity, impact, and confidence scores, thereby reducing false positives and facilitating automated incident response.

To assist readers in understanding the layered design of our proposed Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS), we provide a summarized view of the key architectural components in Table 1. This table outlines the function of each layer, the techniques employed, and its specific contribution to the overall IDS framework.

3.2. Feature Selection Using Elephant Herding Optimization (EHO)

Feature selection is a crucial step in enhancing the accuracy and efficiency of IDS models. Traditional feature selection techniques often suffer from high dimensionality, leading to overfitting and increased computational complexity.

To address these issues, we employ EHO, a bioinspired swarm intelligence algorithm that mimics the social behavior of elephant herds in selecting optimal feature subsets. EHO operates by partitioning the dataset into groups representing elephant clans, where each feature subset is treated as an individual elephant. The optimization process follows two key mechanisms: (i) clan updating, where feature subsets with higher classification accuracy influence the selection, and (ii) separating male elephants, which removes redundant features to enhance generalization.

EHO enables the dynamic selection of the most relevant and non-redundant features from network traffic data, thereby reducing the input dimensionality, lowering computational overhead, and minimizing the risk of overfitting.

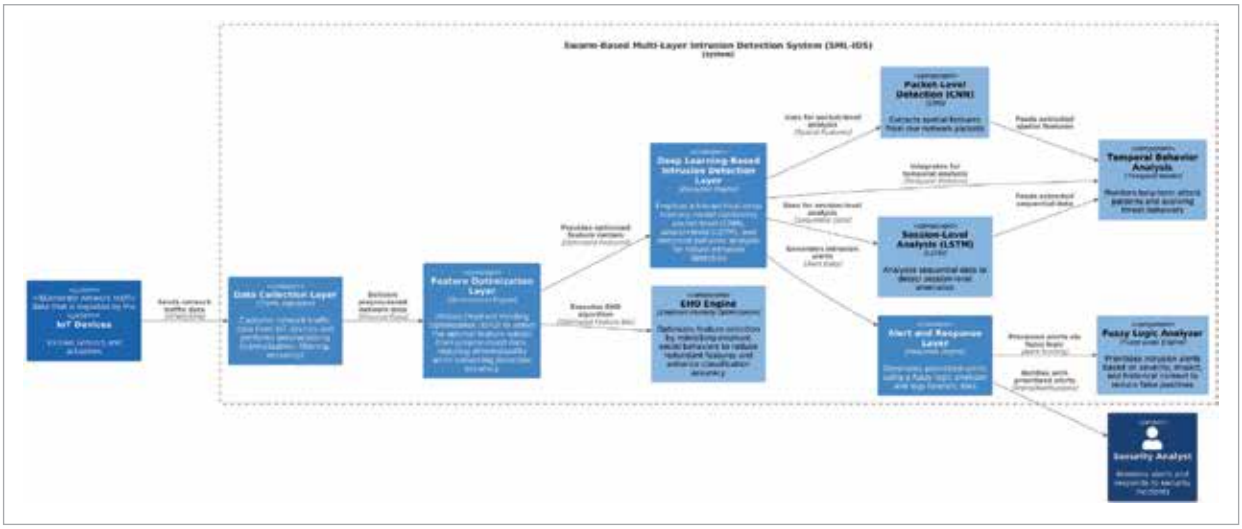
3.3. Multi-Layer Intrusion Detection with Deep Learning

The first level of our hierarchical detection system uses Convolutional Neural Networks (CNNs) to extract spatial features from raw network packets. CNNs are well-suited for this task due to their ability to capture local dependencies and hierarchical structures in input data. The CNN layer operates at the packet level, transforming raw network traffic data into structured feature maps that encapsulate spatial characteristics such as protocol patterns, byte sequences, and packet header distributions. Each network packet is transformed into a feature matrix, where convolutional filters detect attackspecific patterns. The extracted features are then flattened and temporally ordered to form a time-series input sequence, which is fed into the LSTM layer. The LSTM leverages this sequential representation to model the temporal dynamics of network sessions, enabling the system to detect anomalies that evolve over time. This hierarchical integration ensures that both spa-

Table 1
Summary of Architectural Components in the Proposed SML-IDS Framework.

Component	Function	Technique(s)	Contribution to IDS
Data Collection Layer	Captures and preprocesses raw network traffic	Packet sniffing, normalization, one-hot encoding	Provides structured input for downstream analysis
Feature Optimization Layer	Selects most informative features, reduces dimensionality	EHO	Enhances model efficiency and detection accuracy
CNN Layer (Packet-level)	Extracts spatial features from packet data	CNN	Detects low-level patterns and anomalies
LSTM Layer (Session-level)	Captures temporal patterns across network sessions	LSTM	Detects slow, evolving, or stealthy attacks
Temporal Behavior Module	Tracks anomaly trends over sliding time window	Sequence modeling, trend analysis	Adapts to dynamic and novel attack behaviors
Alert and Response Layer	Prioritizes threats and reduces false positives	Fuzzy logic-based context analysis	Enables context-aware, humanlike intrusion alerting

Figure 1
Proposed Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) Architecture.



tial and temporal aspects of network behavior are captured, allowing for a more comprehensive and context-aware intrusion detection process.

$$F = ReLU(W \cdot X + b), \tag{1}$$

where F is the feature map, W represents convolutional weights, X is the input data, and b is the bias.

To analyze time-dependent attack patterns, we employ LSTM networks, which are designed to process sequential data by maintaining long-term dependencies. This enables the IDS to detect slow-moving and persistent threats that span multiple network sessions. The LSTM update equations are defined as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$

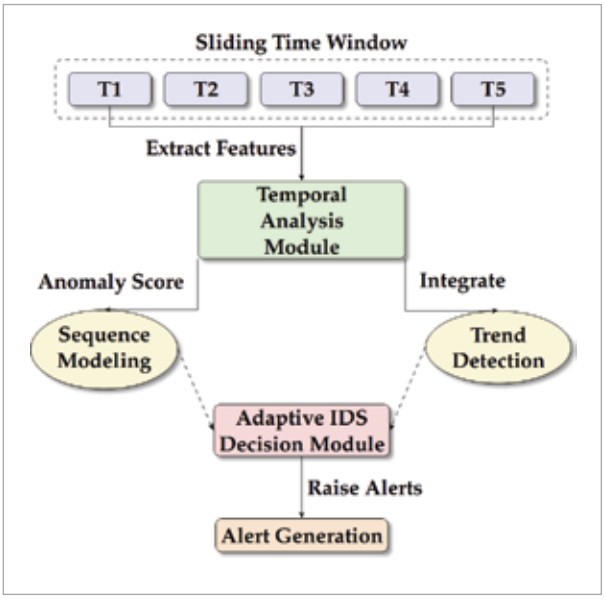
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (5)$$

$$h_t = o_t \odot tanh(c_t)\,. \quad (6)$$

To detect evolving attack patterns, the system includes temporal behavior analysis, where anomalies detected by CNN and LSTM are tracked over time. using a sliding time window mechanism, the IDS adapts to new attack strategies and dynamically updates its detection parameters as demonstrated visually in Figure 2. This module continuously monitors the sequence of network events over a sliding time window and analyzes evolving traffic patterns using the LSTM layer. As new data arrives, the system incrementally updates its internal states and adjusts thresholds, anomaly scores, and alert priorities based on recent trends and deviations from established behavioral baselines. This adaptive mechanism allows the IDS to respond to shifts in attack strategies or previously unseen behaviors without relying on static rules or fixed detection thresholds. As a result, the system remains responsive and context-aware, improving its ability to detect slow, stealthy, or multi-stage attacks in real-time.

Figure 2
Temporal Behavior Analysis for Adaptive IDS.



3.4. Sliding Window Configuration and Adaptation Mechanism

The sliding time window is configured with a fixed window size of 60 seconds and an update interval of 10 seconds. Every 10 seconds, the window shifts forward and incorporates the most recent 60 seconds of network traffic data for sequential pattern analysis using the LSTM layer. Within each window, anomaly scores are calculated for session sequences. To support real-time adaptivity, the system maintains a rolling average and standard deviation of recent anomaly scores.

A dynamic threshold θ_t is computed as:

$$\theta_t = \mu_t + k \cdot \sigma_t\,, \quad (7)$$

where μ_t and σ_t are the mean and standard deviation of anomaly scores over the past five windows, and k is a tunable sensitivity factor (set to 1.5 in our experiments). This adaptive threshold allows the IDS to adjust its sensitivity based on observed behavioral shifts and reduces both false positives and detection lag for slow-evolving or stealthy attacks. Anomalies are flagged when the score in the current window exceeds θ_t, and the alerting priority is further refined by the fuzzy logic-based context module.

4. Implementation Details

The implementation of the proposed Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) involves multiple stages, including dataset selection, preprocessing, feature engineering, model training, and evaluation. This section provides a detailed account of the dataset used, the preprocessing techniques applied, the feature optimization methodology, and the training strategies adopted for deep learning-based intrusion detection. We also discuss the evaluation metrics used to benchmark the system's performance.

4.1. Dataset and Preprocessing

For training and evaluating the proposed intrusion detection system, we utilize the CIC-IDS2017 dataset [35], a widely adopted benchmark dataset for network intrusion detection. Released by the Canadian Institute for Cybersecurity (CIC), the dataset contains

realistic attack scenarios generated in a simulated environment. It has various categories of attacks, including Distributed Denial of Service (DDoS), brute force, botnet, infiltration, and web-based exploits, along with normal network traffic. The dataset includes 80 network flow features, extracted using tools such as Bro-IDS and Wireshark. These features capture crucial packet-level and session-level network attributes such as flow duration, packet length, source and destination ports, TCP window size, and interarrival times. The dataset is highly imbalanced, with a significantly larger proportion of normal traffic compared to attack instances.

To prepare the dataset for training, we apply the following preprocessing steps:

- **Data Cleaning:** Removal of missing values and duplicate entries.
- **Normalization:** Min-max scaling is applied to bring feature values to a range of [0,1], ensuring uniformity.
- **Categorical Encoding:** Protocol types and other categorical variables are converted into numerical values using one-hot encoding.
- **Balancing the Dataset:** To address class imbalance, we apply Synthetic Minority Oversampling Technique (SMOTE) to oversample underrepresented attack classes.

4.2. Feature Optimization

To enhance the efficiency of the IDS, we employ a two-step feature engineering strategy consisting of feature selection followed by bio-inspired optimization using EHO.

We first perform Pearson correlation analysis to eliminate redundant and low-variance features. Features with a correlation coefficient above 0.9 are considered redundant and removed from the dataset.

After correlation-based feature reduction, the dataset still contains a high-dimensional feature space. We apply EHO to further refine the feature subset. The algorithm mimics the social herding behavior of elephants, iteratively selecting the most discriminative features that contribute to attack detection.

EHO was employed to identify and eliminate noncontributing or redundant features from the CICIDS2017 dataset. Through iterative evaluation of feature subsets based on classification performance, EHO was able to reduce the dimensionality of the dataset by discarding features that had minimal impact on detection accuracy. Features such as Bwd URG Flags, Fwd URG Flags, and RST Flag Count were frequently assigned low fitness scores due to their limited variability or low relevance to the learning task. Similarly, statistical attributes like Active Max, Active Min, and Active Std were found to be highly correlated and thus redundant in the presence of other temporal features. Sparse or underutilized attributes such as Bwd Avg Bulk Rate, Fwd Avg Bytes/Bulk, and Flow Bytes/sO1 were also removed, as they offered negligible discriminative value for distinguishing between normal and malicious traffic. Table 2 presents the complete list of features eliminated by EHO, along with brief descriptions and possible explanations for their removal. This optimization step was crucial in enhancing the computational efficiency of the proposed SML-IDS framework, reducing training time, and improving generalization without sacrificing detection performance. The resulting feature set retained the most informative and relevant attributes necessary for robust intrusion classification.

4.3. Model Training and Hyperparameter Tuning

For intrusion detection, we train a hybrid deep learning model consisting of Convolutional Neural

Algorithm 1: Feature Optimization using Elephant Herding Optimization (EHO)

Require: Dataset D with n features, classification model M, population size P, maximum iterations T, convergence threshold ϵ, learning factor α

Ensure: Optimal feature subset S^*

1: **Initialize** a population of P candidate feature subsets $\{S_1, S_2, ..., S_P\}$, where each $S_i \subseteq D$

2: **for** each candidate feature subset S_i **do**

3: Evaluate fitness $f(S_i)$ using model M (e.g., via cross-validation accuracy)

4: **end for**

5: Set iteration counter $t \leftarrow 0$

6: Let $S^* \leftarrow argmax_{S_i} f(S_i)$ be the best candidate in the initial population

7: Set previous best fitness $f_{prev} \leftarrow f(S^*)$

8: **while** $t < T$ **and** improvement $> \epsilon$ **do**

9: **for** each clan (subgroup) in the population **do**

10: Identify the best candidate S_{best} within the clan based on fitness
11: **for** each candidate S_i in the clan **do**
12: Update candidate via clan movement:

$$S_i \leftarrow S_i + \alpha \cdot (S_{\text{best}} - S_i) \quad (8)$$

13: Remove redundant features from S_i (using a predefined redundancy criterion)
14: **end for**
15: **end for**
16: **for** each candidate S_i in the population **do**
17: Re-evaluate fitness $f(S_i)$ using model M
18: **end for**
19: Update $S^* \leftarrow argmax_{S_i} f(S_i)$
20: Compute current best fitness $f_{curr} \leftarrow f(S^*)$
21: Compute improvement: $\Delta f \leftarrow |f_{curr} - f_{prev}|$
22: Set $f_{prev} \leftarrow f_{curr}$
23: Increment iteration counter $t \leftarrow t+1$
24: **end while**
25: **Return** the optimal feature subset S^*

Networks (CNNs) for packet-level feature extraction and LSTM networks for session-level temporal anomaly detection.

To fine-tune the performance of the proposed deep learning model, we employ a two-stage hyperparameter optimization strategy that combines grid search with Bayesian optimization. This hybrid approach allows us to systematically explore the hyper-parameter space while leveraging probabilistic modeling to reduce computational overhead.

In the first stage, a coarse grid search is applied to identify promising regions in the hyperparameter space. Let $H=\{h_1,h_2,...,h_n\}$ denote the set of all possible hyperparameter configurations. Grid search evaluates each $h_i \in H$ across a predefined parameter grid to select the top k configurations with the best validation performance. In this study, the grid search explored the following ranges:

- Learning rate: $\eta \in \{0.001,0.0005,0.0001\}$
- Batch size: $\in \{32,64,128\}$
- CNN filters: $F \in \{32,64,128\}$
- LSTM units: $U \in \{64,128,256\}$
- Dropout rate: $d \in \{0.2,0.3,0.5\}$

The objective of this stage is to reduce the size of the search space H to a manageable subspace $H' \subset H$ for more efficient fine-tuning.

In the second stage, we apply Bayesian optimization over the refined subspace H'. Unlike grid search, which is exhaustive and deterministic, Bayesian optimization models the objective loss function $f(h)$ as a black-box function using a surrogate Gaussian Process (GP) model. The core idea is to maximize an acquisition function $A(h)$ that balances exploration and exploitation:

$$h^* = \arg\max_{h \in H'} A(h). \quad (9)$$

As the acquisition function we use the Expected Improvement (EI), defined as:

$$A_{EI}(h) = E\left[\max\left(f(h) - f\left(h^+\right), 0\right)\right]. \quad (10)$$

where (h^+) is the best observed value so far. Assuming $f(h) \sim N(\mu(h),\sigma^2(h))$ under the GP, EI can be expressed as:

$$A_{EI}(h) = (\mu(h) - f(h^+))\Phi(Z) + \sigma(h)\phi(Z) \quad (11)$$

$$Z = \frac{\mu(h) - f(h^+)}{\sigma(h)}, \quad (12)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are cumulative distribution function (CDF) and probability density function (PDF) of the normal distribution, respectively. Bayesian optimization proceeds iteratively by:

1. Fitting the surrogate model to previously evaluated hyperparameter configurations.
2. Selecting the next configuration h_t by maximizing $A(h)$.
3. Evaluating $f(h_t)$ (e.g., model accuracy on validation set).
4. Updating the surrogate model with the new data point $(h_t, f(h_t))$.

This process continues until a stopping criterion is met (a maximum number of iterations or convergence of the acquisition function). The optimal hyperparameters identified through this process were:

- Learning rate: 0.0005
- Batch size: 64
- CNN filters: 64
- LSTM units: 128
- Dropout rate: 0.3

Table 2
Eliminated Features from the Dataset with Descriptions and Justifications.

Feature	Description	Possible Explanation for Removal
Active Max	Maximum duration of active period in flow	Low variance, minimal contribution to classification
Active Min	Minimum duration of active period in flow	Redundant with Active Max and Active Std
Active Std	Standard deviation of active period durations	High correlation with Active Max/Min
Bwd Avg Bulk Rate	Avg. bulk data rate in backward direction	Mostly zero or constant across sessions
Bwd Avg Bytes/Bulk	Avg. bytes per bulk segment in backward direction	Sparse data, low informativeness
Bwd Avg Packets/Bulk	Avg. packets per bulk segment in backward direction	Largely unused in actual traffic
Bwd IAT Min	Minimum inter-arrival time of packets (backward)	High correlation with other IAT metrics
Bwd Packets/s	Number of packets sent per second (backward)	Redundant with total packet counts
Bwd PSH Flags	Count of PSH flags in backward direction	Low occurrence, limited variability
Bwd URG Flags	Count of URG flags in backward direction	Rarely triggered in modern protocols
CWE Flag Count	Count of CWE (Congestion Window) flags	Very low entropy across flows
ECE Flag Count	Count of ECN-Echo flags	Low variability, high sparsity
Flow Bytes/sO1	Bytes per second in the flow (possibly typo for Flow Bytes/s)	Duplicate or malformed feature
Fwd Avg Bulk Rate	Avg. bulk data rate in forward direction	Similar to Bwd Avg Bulk Rate, low variation
Fwd Avg Bytes/Bulk	Avg. bytes per bulk segment in forward direction	Sparse and redundant
Fwd Avg Packets/Bulk	Avg. packets per bulk segment in forward direction	Not useful in detecting anomalies
Fwd URG Flags	Count of URG flags in forward direction	Infrequent in modern traffic
Idle Std	Standard deviation of idle times in flow	High correlation with Idle Min/Max
RST Flag Count	Count of TCP reset flags	Skewed distribution, dominated by zero values
Subflow Bwd Bytes	Bytes in the backward subflow	Overlapping with total backward bytes
Total Length of Fwd Packets	Total length of forward packets	Strongly correlated with packet count xand size

These settings yielded the highest classification accuracy on the validation dataset while minimizing overfitting and training time. The hybrid strategy provided a balance between exhaustive exploration (grid search) and efficient exploitation (Bayesian optimization), leading to a robust and well-tuned model suitable for real-time intrusion detection in IoT environments.

The final optimized model is trained using the Adam optimizer with a learning rate decay strategy to prevent overfitting.

4.4. Fuzzy Logic-Based Context-Aware Intrusion Prioritization

To enhance decision-making and reduce false positives, the proposed SML-IDS incorporates a fuzzy logic-based context analyzer in the alert and response layer. This module prioritizes detected intrusion events by considering multiple contextual factors such as severity of the attack, frequency of occurrence, historical relevance, and impact on network segments. Fuzzy inference allows the system

to mimic human-like reasoning in the presence of uncertainty and imprecision, which is particularly suitable for dynamic IoT environments where rigid decision boundaries may fail.

We define the input linguistic variables, each mapped to fuzzy sets using membership functions:

- Severity (S): {Low, Medium, High}
- Frequency (F): {Rare, Occasional, Frequent}
- Historical Match (H): {No Match, Partial Match, Exact Match}
- Impact (I): {Minor, Moderate, Critical}

Each fuzzy set is represented using triangular or trapezoidal membership functions. For instance, the membership function $\mu_{High}(x)$ for the severity variable is defined as:

$$\mu_{High}(x) = \{0,\ x \leq 0.6\ \frac{x-0.6}{0.4},\ 0 \quad (13)$$

These fuzzy values are derived from normalized inputs in the range [0,1] using min-max scaling.

A set of expert-defined fuzzy inference rules is used to model the prioritization logic. Each rule follows the standard Mamdani form:

IF S is High *AND* F is Frequent *AND* I is Critical *THEN* Priority is Urgent

Table 3 presents a subset of the fuzzy rule base:

The fuzzy inference engine applies the Mamdani min–max composition method to evaluate the rule base. The aggregated fuzzy output $P(x)$ (priority level) is computed as:

$$P(x) = \max_{i=1}^{N} \min\left(\mu_{A_1^i}(x_1), \mu_{A_2^i}(x_2), \ldots, \mu_{A_n^i}(x_n)\right), \quad (14)$$

where $\mu_{A_j^i}(x_j)$ denotes the membership function for the j-th antecedent in the i-th rule.

The crisp priority score P^* is obtained using the centroid defuzzification method:

$$P^* = \frac{\int_a^b x \cdot P(x)\, dx}{\int_a^b P(x)\, dx}, \quad (15)$$

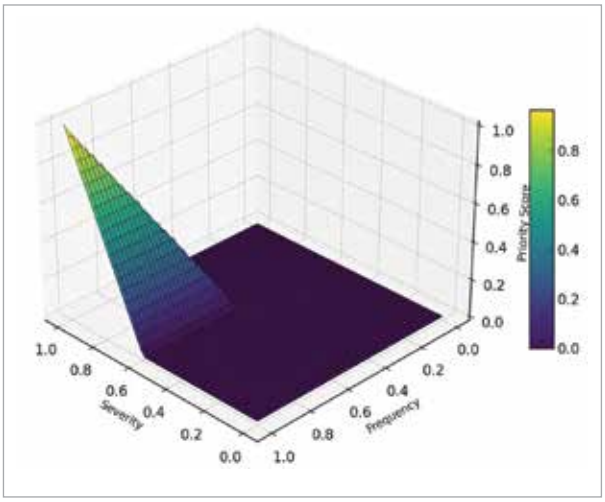
where $[a,b]$ is the support of the output fuzzy set.

Figure 3 illustrates the fuzzy output for various combinations of input variables using a surface plot generated from the rule base. By incorporating fuzzy reasoning, the system improves its ability to differentiate between critical and non-critical alerts, thereby reducing false positives and prioritizing high-risk intrusions for immediate response. This enhances operational efficiency and allows security analysts to allocate attention to the most relevant threats.

Table 3
Example Fuzzy Inference Rules for Intrusion Prioritization

Severity	Frequency	Impact	Match	Priority
High	Frequent	Critical	Exact Match	Urgent
Medium	Occasional	Moderate	Partial Match	High
Low	Rare	Minor	No Match	Low
High	Rare	Critical	Partial Match	Medium
Medium	Frequent	Moderate	Exact Match	High

Figure 3
Fuzzy Output Surface for Priority Scoring based on Severity and Frequency.



4.5. Evaluation Metrics and Performance Benchmarks

To evaluate the effectiveness of our model, we use the following key performance metrics:

- **Accuracy (ACC)** measures the proportion of correctly classified instances. It is defined as:

$$\mathrm{ACC} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{16}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

- **Precision (P)** evaluates the percentage of actual attacks correctly identified as:

$$\mathrm{Precision} = \frac{TP}{TP+FP} \tag{17}$$

- **Recall (R)** measures the IDS's ability to detect true intrusions, calculated as:

$$\mathrm{Recall} = \frac{TP}{TP+FN}. \tag{18}$$

- **F1-Score** is the harmonic mean of precision and recall, expressed as:

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}. \tag{19}$$

- **False Positive Rate (FPR)** evaluates the proportion of normal traffic instances that are incorrectly classified as attacks:

$$\mathrm{FPR} = \frac{FP}{FP+TN}. \tag{20}$$

- **Detection Time (DT)** measures the time taken to classify a network packet as:

$$\mathrm{DT} = \frac{T_{\mathrm{total}}}{N}, \tag{21}$$

where T_{total} is the total time for classification, and N is the number of packets classified.

5. Results and Analysis

5.1. Experimental Environment

The development and evaluation of the proposed Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) were carried out using Python 3.10.12 language. The implementation leveraged TensorFlow 2.13.0 library, which provided GPU-accelerated support for training the deep learning models. Keras 2.13.1, integrated within TensorFlow, was used as a high-level API to define and manage the CNN and LSTM architectures that form the core of the hierarchical intrusion detection system.

For data manipulation and numerical analysis, we employed NumPy 1.24.4 and Pandas 2.1.1. Scikitlearn 1.3.1 was used to implement traditional machine learning baselines, preprocess features, and compute performance metrics such as accuracy, precision, recall, and F1-score.

Bayesian hyperparameter optimization was conducted using Optuna 3.3.0, which uses a Treestructured Parzen Estimator (TPE) to efficiently explore the hyperparameter space while balancing exploration and exploitation. This was combined with a preliminary grid search to narrow down the search domain before Bayesian optimization was applied to fine-tune model parameters.

The fuzzy logic-based context analyzer, which prioritizes alerts based on severity, frequency, and historical patterns, was implemented as a custom module using native Python classes and NumPy-based vectorization to evaluate fuzzy rules and perform defuzzification using the centroid method.

All experiments were conducted on a high-performance workstation running Ubuntu 22.04 LTS (64-bit) equipped with an AMD Ryzen 9 7950X 16core processor clocked at 4.5 GHz, 64 GB of DDR5 RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM. GPU acceleration was facilitated via the CUDA 11.8 toolkit and cuDNN 8.6.0 library.

The CIC-IDS2017 dataset was obtained from the Canadian Institute for Cybersecurity's official repository and stored on high-speed NVMe M.2 PCIe Gen4 SSDs to allow for fast access during preprocessing and training. All preprocessing steps, including data cleaning, normalization, one-hot encoding, and oversampling, were performed in adwvance and saved in serialized formats using HDF5 and Pickle to support reproducibility. To maintain version control and ensure consistent execution across different runs, the entire software stack was managed using Anaconda 2023.09. A dedicated virtual environment named sml-ids-env was created to encapsulate all dependencies, and a YAML configuration file was exported to facilitate replication of the experimental environment on other machines. All sliding window operations were

implemented using time-indexed session logs and synchronized with the packet timestamps in the CIC-IDS2017 dataset. The fixed window size of 60 seconds was selected based on empirical testing, balancing temporal resolution and detection responsiveness. A 10second hop interval ensured near-continuous monitoring without significant computational overhead.

5.2. Performance Comparison with Traditional IDS Models

To benchmark the performance of the proposed SML-IDS, we compare it against conventional Machine Learning-based IDS models such as Support Vector Machines (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP). The models are trained on the same optimized feature set and evaluated using standard classification metrics, including Accuracy, Precision, Recall, and F1-Score. Table 4 summarizes the performance comparison between traditional models and the proposed SML-IDS. The results indicate that the SML-IDS model significantly outperforms traditional IDS approaches, achieving a detection accuracy of 97.4%, which is notably higher than Random Forest (93.8%) and SVM (91.3%). The results demonstrate that the proposed SML-IDS model outperforms conventional methods, achieving higher accuracy, lower false positives, and improved attack detection efficiency, making it a robust and scalable solution for intrusion detection in IoT networks.

Table 4
Performance Comparison of IDS Models

Model	Accuracy (%)	Precision	Recall	F1-Score
SVM	91.3	0.89	0.85	0.87
Random Forest	93.8	0.92	0.90	0.91
Multi-Layer Perceptron	95.1	0.94	0.91	0.93
Proposed SML-IDS	**97.4**	**0.96**	**0.94**	**0.95**

The improved accuracy of SML-IDS can be attributed to the combination of EHO algorithm for feature selection and the hierarchical deep learning architecture, which enables precise anomaly detection with lower false positives.

5.3. Impact of Swarm Optimization on Feature Selection

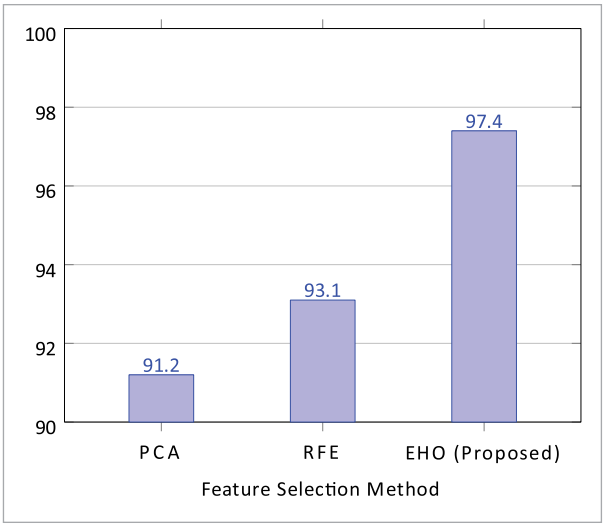
Feature selection is critical in improving the efficiency of IDS models by eliminating redundant and irrelevant features, thereby reducing dimensionality while retaining classification effectiveness. The EHO-based feature selection optimizes the feature subset by iteratively selecting the most relevant attributes. To evaluate the impact of EHO, we compare its performance against conventional Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE). The selected feature count and resulting model accuracy are shown in Table 5.

Table 5
Impact of Feature Selection on Model Performance

Feature Selection Method	Selected Features	Accuracy (%)	F1-Score
PCA	20	91.2	0.89
RFE	18	93.1	0.92
EHO (Proposed)	15	97.4	0.95

Figure 4 shows the impact of EHO feature selection on classification accuracy.

Figure 4
Comparison of intrusion detection accuracy achieved using different feature selection techniques: Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and the proposed Elephant Herding Optimization (EHO).



5.4. Effectiveness of Hierarchical Deep Learning

The multi-layer intrusion detection architecture integrates CNN for packet-level feature extraction and LSTM for temporal anomaly detection. This combination enables the system to identify low-level attack patterns in packets while capturing long-term sequential dependencies in network sessions. Figure 5 compares the detection accuracy of CNN, LSTM, and the hybrid CNN-LSTM approach.

The CNN-LSTM hybrid model outperforms standalone CNN and LSTM architectures, achieving the highest accuracy and recall values, as it captures both spatial and temporal features of network traffic.

5.5. Adversarial Attack Resistance

To assess the robustness of SML-IDS against adversarial attacks, we generate evasion samples using Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD). The detection rate before and after adversarial training is shown in Table 6.

Table 6
Adversarial Attack Resistance Before and After Adversarial Training.

Attack Type	Detection Rate (Before)	Detection Rate (After)
FGSM	75.2%	92.4%
PGD	68.7%	90.8%

After adversarial training, the IDS achieves a significant improvement in detection rates, demonstrating its ability to mitigate evasion attempts.

5.6. Context-Awareness and False Positive Reduction

A major challenge in IDS is the high false positive rate (FPR), where legitimate network activities are misclassified as attacks. By incorporating fuzzy logic-based context awareness, the system prioritizes alerts based on historical patterns, network behavior, and attack severity.

Figure 6 illustrates the reduction in false positive rate after integrating context-aware filtering.

Figure 5
Comparison of detection accuracy between individual deep learning models—Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)—and their combined hybrid architecture (CNN-LSTM).

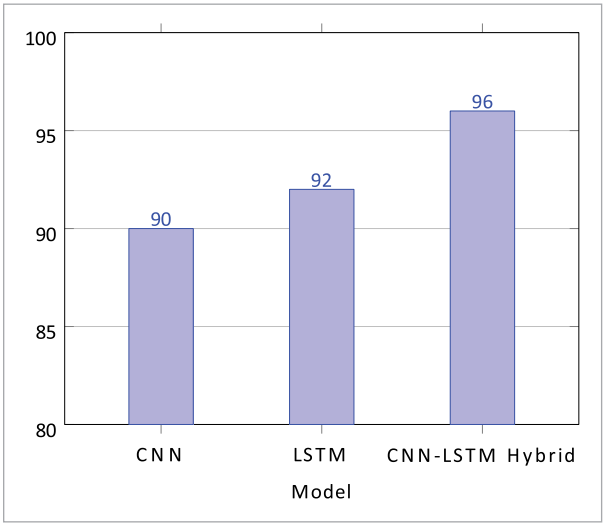
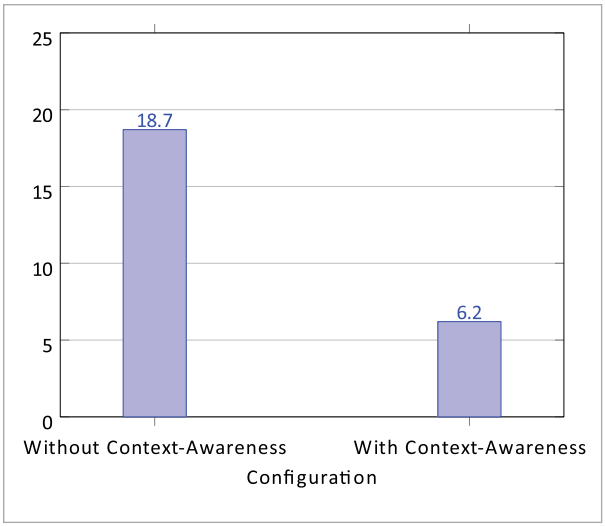


Figure 6
Comparison of false positive rates in intrusion detection with and without the integration of context-awareness.



5.7. Computational Overhead and RealTime Feasibility

Efficient real-time processing is essential for practical IDS deployment in IoT environments. Table 7 compares the average detection latency and computational overhead across different IDS models.

Table 7
Computational Overhead and Detection Latency Comparison.

Model	Detection Time (ms)	Computational Overhead
SVM	45.2	High
Random Forest	32.8	Moderate
CNN-LSTM (Proposed)	**18.4**	**Low**

The proposed CNN-LSTM model achieves the lowest latency, making it highly feasible for realtime intrusion detection in IoT networks.

5.8. Comparison with Other Approaches

To contextualize the performance of the proposed SML-IDS framework, we provide a comparative overview of its accuracy alongside several notable context-aware and hybrid intrusion detection models, as presented in Table 8. These models span a range of architectures, including deep learning ensembles, bio-inspired optimization strategies, and hybrid CNN–LSTM pipelines. However, it is important to emphasize that several of the compared methods were evaluated on different datasets, with varying class distributions, network environments, and attack types. The proposed SML-IDS achieved a competitive accuracy of 97.4%, aligning closely with or outperforming many deep learning-based approaches such as DeepCoin (98.23%) [10], DL-IDS (98.67%) [37], and Cu(LSTM–CNN) (98.60%) [22]. These models, while effective, do not incorporate fuzzy logic-based context reasoning for alert prioritization—an element that distinguishes our approach by reducing false positives and improving alert relevance. Although CNN+LSTM11 [42] reported a higher accuracy of 99.81%, it was evaluated on a different dataset and thus should not be interpreted as a directly comparable benchmark. The models marked with an asterisk (*) in Table 8 were tested under alternative experimental settings. To mitigate potential over-interpretation, we include this comparison solely for contextual insight into current trends in IDS research.

Although some of SOTA models such as CNN+LSTM11 and EBAO show slightly higher accuracy, these results were obtained on different datasets under varied conditions and often involve more complex architectures and a larger number of features. In contrast, SML-IDS achieves competitive performance using only 15 optimized features, thanks to the integration of EHO, which reduces input dimensionality without sacrificing detection capability. Moreover, the proposed model achieves high precision (0.96) and recall (0.94), ensuring that both true attacks are captured and false positives are minimized. Its lightweight architecture ensures a low computational footprint, making it suitable for real-time deployment in IoT environments. This trade-off of detection performance, interpretability, and efficiency underscores the core contribution of our work, even when benchmarked against models with slightly higher accuracy.

Table 8
Contextual Accuracy Comparison with State-of-the-Art IDS Approaches.

Model	Accuracy (%)	Reference / Notes
CNN+LSTM11*	99.81	[42] — Evaluated on different dataset with distinct class structure
Cu(LSTM–CNN)*	98.60	[22] — SDN-specific hybrid IDS
DL-IDS*	98.67	[37] — CNN-LSTM based feature extractor
DeepCoin*	98.23	[10] — Blockchain-based energy-aware IDS
EBAO*	99.37	[1] — Enhanced Binary Aquila Optimizer
CAFE-CNN*	99.29	[34] — Context-aware CNN architecture
Proposed SMLIDS	97.4	This study — Evaluated on CIC-IDS2017 dataset

Disclaimer: *Models marked with an asterisk were evaluated on datasets other than CIC-IDS2017. Due to differences in data structure, attack types, and class balance, these figures are not directly comparable and should be interpreted only as indicative references.

6. Discussion and Conclusion

The proposed Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) leverages a combination of bio-inspired feature selection, hierarchi-

cal deep learning, adversarial training, and context-aware alert prioritization to enhance intrusion detection in IoT environments. The results demonstrate improvements in detection accuracy, false positive reduction, and real-time feasibility compared to traditional IDS models. In this section, we discuss the key strengths and contributions of the proposed approach, acknowledge its limitations, explore potential applications beyond IoT security, and outline promising directions for future research.

6.1. Strengths of the Proposed Approach

The primary advantage of SML-IDS is its ability to achieve high detection accuracy while maintaining computational efficiency. The integration of EHO for feature selection significantly reduces the dimensionality of the dataset while preserving the most discriminative features. This results in a more efficient and scalable intrusion detection model.

Another major strength of SML-IDS is its hierarchical deep learning architecture, which combines CNN for packet-level intrusion detection and LSTM for session-level anomaly tracking. This enables the system to identify both isolated attack patterns and long-term sequential anomalies, leading to a lower false positive rate (FPR). Integration of adversarial training enhances robustness against evasion attacks, ensuring that the IDS remains effective even in adversarial scenarios. The context-aware alert prioritization mechanism reduces unnecessary alerts, allowing security analysts to focus on critical threats.

The sliding time window mechanism in our proposed framework plays a critical role in adapting to evolving attack patterns by enabling continuous monitoring and temporal correlation of network activities. Its key advantages include the ability to detect slow and stealthy attacks that span multiple sessions, identify trends or shifts in behavior over time, and maintain an up-to-date representation of network state for dynamic decision-making. Unlike static analysis approaches, the sliding window allows the system to learn from recent data while discarding outdated information, thereby enhancing adaptability to emerging threats. This mechanism ensures that intrusion detection remains context-aware and responsive to the temporal evolution of cyberattacks in real-time environments.

6.2. Limitations and Challenges

Despite its advantages, the proposed approach has certain limitations. One of the primary challenges is high computational complexity. Although EHObased feature selection reduces the number of features, the deep learning model, particularly the LSTM component, requires substantial processing power during both training and inference. This may pose challenges for real-time deployment in resource-constrained edge devices. However, adversarial training is not entirely foolproof against advanced adversarial attacks, such as adaptive adversarial examples, which continuously learn how to bypass detection systems. Enhancements in adversarial defense mechanisms are required to strengthen resilience against highly sophisticated attack strategies. Another challenge is data dependency. The effectiveness of SML-IDS is influenced by the availability of high-quality, up-to-date datasets. CIC-IDS2017 is a comprehensive dataset, but newer attack vectors that emerged post-dataset release may not be fully represented.

6.3. Future Research Directions

Although the proposed SML-IDS framework demonstrates notable improvements in intrusion detection accuracy, false positive reduction, and real-time feasibility, several avenues remain open for future enhancement. One significant limitation lies in the dependency on the CIC-IDS2017 dataset, which, while comprehensive, no longer reflects the latest trends in cyber threats, especially those targeting IoT and cloud-enabled infrastructures. Future research will focus on integrating more recent and diverse datasets, such as CIC-DDoS2019 and TON_IoT, to capture emerging attack patterns including botnets, ransomware, and adversarial exploits that are more representative of contemporary threat landscapes. This will enrich the training data and improve the generalizability and adaptability of the intrusion detection model across heterogeneous environments.

In cloud computing environments, where virtualized infrastructure and dynamic workloads introduce new security risks, SML-IDS can be extended to monitor inter-virtual machine communications, detect lateral movements, and identify unauthorized access patterns. The temporal behavior analysis module can be used to trace suspicious sequences of cloud API calls or abnormal inter-service interac-

tions, enhancing detection of persistent threats in cloud-native applications.

In Software-Defined Networking (SDN), the centralized control architecture and programmable data planes offer both opportunities and risks. The lightweight and layered design of SML-IDS can be deployed at both the controller and switch levels to inspect flow-level traffic in real time. The swarm-based feature selection process can be tailored to select protocol-specific or flow-level metrics for highspeed anomaly detection. Moreover, the contextual alerting module can assist in automated flow rule updates to isolate compromised network segments. Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) networks present another promising application domain. These systems are typically characterized by real-time, deterministic communication patterns and are highly sensitive to operational disruptions. The LSTM-based temporal module in SML-IDS can be customized to model control commands and sensor data streams, making it effective for detecting command injection, replay attacks, or other anomalies in control logic over time.

Another direction is the application of model compression techniques to enhance deployment efficiency on resource-constrained devices. Techniques such as model pruning, weight quantization, and knowledge distillation will be explored to reduce the model size and inference latency while maintaining comparable detection performance. These lightweight variants of the proposed model would facilitate real-time detection in distributed and bandwidth-limited networks, enabling broader deployment in practical IoT scenarios.

To improve the generalizability and applicability of our proposed method, future work also will focus on evaluating the SML-IDS framework using more recent and diverse datasets, including TON_IoT, NBaIoT, and NF-TON. These datasets capture contemporary IoT-specific threat landscapes, multiplatform network interactions, and a broader range of attack behaviors such as botnets, malware propagation, and low-rate stealth attacks. Evaluating on such datasets will allow to validate the robustness of the proposed model across heterogeneous environments, address evolving adversarial tactics, and ensure relevance to modern IoT and cloud infrastructures in real-world deployments.

6.4. Conclusion

This study introduces a Swarm-Based Multi-Layer Intrusion Detection System (SML-IDS) that integrates bio-inspired optimization, hierarchical deep learning, adversarial defense, and context-aware security analysis. The proposed framework significantly improves intrusion detection accuracy, reduces false positives, and enhances real-time feasibility compared to traditional IDS approaches.

Experimental results demonstrate that SML-IDS outperforms conventional machine learning-based IDS models by achieving a higher detection rate, lower false alarm rate, and improved adversarial robustness. The adoption of Elephant Herding Optimization (EHO) for feature selection ensures efficient dimensionality reduction, making the model computationally efficient.

Declarations

Funding

None.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

This study uses publicly available CIC-IDS2017 dataset (https://www.unb.ca/cic/datasets/ ids-2017.html).

References

1. Alawad, N. A., Abed-Alguni, B. H., Shakhatreh, A. M. Ebao: An Intrusion Detection Framework for Wireless Sensor Networks Using an Enhanced Binary Aquila Optimizer. Knowledge-Based Systems, 2025, 312, 113156. https://doi.org/10.1016/j.knosys.2025.113156
2. Al-Kadi, O., Moustafa, N., Turnbull, B., Choo, K. A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. IEEE Internet of Things Journal, 2021, 8, 9463-9472. https://doi.org/10.1109/JIOT.2020.2996590

3. Bouramoul, I. E., Zertal, S., Derdour, M., Zenbout, I. Enhancing IoT Security Through Deep Learning and Evolutionary Bio-Inspired Intrusion Detection in IoT Systems. 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), 2024, 1-8. https://doi.org/10.1109/PAIS62114.2024.10541160
4. Chaganti, R., Mourade, A., Ravi, V., Vemprala, N., Dua, A., Bhushan, B. A Particle Swarm Optimization and Deep Learning Approach for Intrusion Detection System in Internet of Medical Things. Sustainability, 2022, 14(19), 12828. https://doi.org/10.3390/su141912828
5. Chandu, B., Gadoo, S., Nidhi, A., Bhat, M. V. A Robust Intrusion Detection in Network Traffic Using Deep Learning. International Research Journal of Modernization in Engineering Technology and Science, 2024. https://doi.org/10.56726/IRJMETS51670
6. Cisco. Cyber Threat Trends Report: From Trojan Takeovers to Ransomware Roulette (Tech. Rep.). Cisco, 2024.
7. Das, A., Sobha, N., Natesh, M., Tiwary, G. An Enhanced Hybrid Deep Learning Model to Enhance Network Intrusion Detection Capabilities for Cybersecurity. Journal of Machine and Computing, 2024, 472-486. https://doi.org/10.53759/7669/jmc202404045
8. Dash, N., Chakravarty, S., Rath, A. Deep Learning Model for Elevating Internet of Things Intrusion Detection. International Journal of Electrical and Computer Engineering (IJECE), 2024. https://doi.org/10.11591/ijece.v14i5.pp5874-5883
9. Fatima, M., Rehman, O., Rahman, I. M. H., Ajmal, A., Park, S. Towards Ensemble Feature Selection for Lightweight Intrusion Detection in Resource-Constrained IoT Devices. Future Internet, 2024. https://doi.org/10.3390/fi16100368
10. Ferrag, M. A., Maglaras, L. DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids. IEEE Transactions on Engineering Management, 2020, 67(4), 1285-1297. https://doi.org/10.1109/TEM.2019.2922936
11. Ghanem, W., Jantan, A. Novel Multi-Objective Artificial Bee Colony Optimization for Wrapper-Based Feature Selection in Intrusion Detection. International Journal of Advances in Soft Computing & Its Applications, 2016, 8(1), 70.
12. Gungor, O., Li, E., Shang, Z., Guo, Y., Chen, J., Davis, J., Simunic, T. Rigorous Evaluation of Machine Learning-Based Intrusion Detection Against Adversarial Attacks. 2024 IEEE International Conference on Cyber Security and Resilience (CSR), 2024, 152-158. https://doi.org/10.1109/CSR61664.2024.10679443
13. Habib, B., Khursheed, F. Time-Based DDoS Attack Detection Through Hybrid LSTM-CNN Model Architectures: An Investigation of Many-to-One and Many-to-Many Approaches. Concurrency and Computation: Practice and Experience, 2024. https://doi.org/10.1002/cpe.7996
14. Hairab, B. I., Aslan, H., Elsayed, M. S., Jurcut, A., Azer, M. A. Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques. Electronics, 2023. https://doi.org/10.3390/electronics12030573
15. Huang, J.-C. Network Intrusion Detection Model Based on Genetic Ant Colony Algorithm. International Conference on Automation, Mechanical Control and Computational Engineering, 2015. https://doi.org/10.2991/amcce-15.2015.203
16. Isong, B., Kgote, O., Abu-Mahfouz, A. Insights Into Modern Intrusion Detection Strategies for Internet of Things Ecosystems. Electronics, 2024, 13(12), 2370. https://doi.org/10.3390/electronics13122370
17. Karanam, L., Pattanaik, K. K., Aldmour, R. Intrusion Detection Mechanism for Large-Scale Networks Using CNN-LSTM. 2020 13th International Conference on Developments in eSystems Engineering (DeSE), 2020, 323-328. https://doi.org/10.1109/DeSE51703.2020.9450732
18. Kumar, A., Das, T. K., Pandey, R. K. Sri: A Simple Rule Induction Method for Improving Resiliency of DNN-Based IDS Against Adversarial and Zero-Day Attacks. 10th ACM Cyber-Physical System Security Workshop, 2024. https://doi.org/10.1145/3626205.3659146
19. Laqtib, S., Yassini, K. E., Hasnaoui, M. L. Deep Learning Methods for Intrusion Detection Systems Based on Machine Learning in MANET. 4th International Conference on Smart City Applications, 2019. https://doi.org/10.1145/3368756.3369021
20. Li, P., Duan, H. Bio-Inspired Computation Algorithms. Bio-Inspired Computation in Unmanned Aerial Vehicles, 2013, 35-69. doi:10.1007/978-3-642-41196-0_2Liao, H., Murah, M.Z., Hasan, M.K., Aman, A., Fang, J., Hu, X., and Khan, A.U.R. A survey of deep learning technologies for intrusion detection in internet of things. IEEE Access, 2024, 12, 4745-4761.https://doi.org/10.1109/ACCESS.2023.3349287
21. Malik, J., Akhunzada, A., Bibi, I., Faheem, M., Amin, R. U., Shafiq, M. Hybrid Deep Learning: An Efficient Reconnaissance and Surveillance Detection Mechanism in SDN. IEEE Access, 2020, 8, 134695-134706. https://doi.org/10.1109/ACCESS.2020.3009849
22. Odeh, A., Taleb, A. A. Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection. Applied Sciences, 2023. https://doi.org/10.3390/app132111985
23. Ohtani, T., Yamamoto, R., Ohzahata, S. Detecting Zero-Day Attack with Federated Learning Using Autonomously Extracted Anomalies in IoT. 2024 IEEE 21st

Consumer Communications & Networking Conference (CCNC), 2024, 356-359. https://doi.org/10.1109/CCNC51664.2024.10454669

24. Oseni, A., Moustafa, N., Creech, G., Sohrabi, N., Strelzoff, A., Tari, Z., Linkov, I. An Explainable Deep Learning Framework for Resilient Intrusion Detection in IoT-Enabled Transportation Networks. IEEE Transactions on Intelligent Transportation Systems, 2023, 24, 1000-1014. https://doi.org/10.1109/TITS.2022.3188671
25. Pampapathi, B. M., Nageswara Guptha, M., Hema, M. S. Towards an Effective Deep Learning-Based Intrusion Detection System in the Internet of Things. Telematics and Informatics Reports, 2022, 7, 100009. https://doi.org/10.1016/j.teler.2022.100009
26. Praveena Anjelin, D., Ganesh Kumar, S. An Effective Classification Using Enhanced Elephant Herding Optimization With Convolution Neural Network for Intrusion Detection in IoMT Architecture. Cluster Computing, 2024, 27(9), 12341-12359. https://doi.org/10.1007/s10586-024-04512-5
27. Rani, M., Gagandeep. Employing Artificial Bee Colony Algorithm for Feature Selection in Intrusion Detection System. 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, 496-500. doi:10.1109/INDIACom51348.2021.00088
28. Roopak, M., Tian, G., Chambers, J. An Intrusion Detection System Against DDoS Attacks in IoT Networks. 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, 562-567. https://doi.org/10.1109/CCWC47524.2020.9031206
29. Sagu, A., Gill, N. S., Gulia, P., Priyadarshini, I., Chatterjee, J. M. Hybrid Optimization Algorithm for Detection of Security Attacks in IoT-Enabled Cyber-Physical Systems. IEEE Transactions on Big Data, 2025, 11(1), 35-46. https://doi.org/10.1109/TBDATA.2024.3372368
30. Saheed, Y., Arowolo, M. Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms. IEEE Access, 2021, 9, 161546-161554. https://doi.org/10.1109/ACCESS.2021.3128837
31. Sarker, I. H. Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective. SN Computer Science, 2021. doi:10.1007/s42979-02100535-6 https://doi.org/10.20944/preprints202102.0340.v1
32. Sayem, I. M., Sayed, M. I., Saha, S., Haque, A. ENIDS: A Deep Learning-Based Ensemble Framework for Network Intrusion Detection Systems. IEEE Transactions on Network and Service Management, 2024, 21, 5809-5825. https://doi.org/10.1109/TNSM.2024.3414305
33. Shams, E. A., Rizaner, A., Ulusoy, A. H. A Novel Context-Aware Feature Extraction Method for Convolutional Neural Network-Based Intrusion Detection Systems. Neural Computing and Applications, 2021, 33(20), 13647-13665. https://doi.org/10.1007/s00521-021-05994-9
34. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A. A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th International Conference on Information Systems Security and Privacy, 2018. https://doi.org/10.5220/0006639801080116
35. Sherin, R. J., Parkavi, K. Investigations on Bio-Inspired Algorithm for Network Intrusion Detection-A Review. International Journal of Computer Networks and Applications, 2022. https://doi.org/10.22247/ijcna/2022/214503
36. Sun, P., Liu, P., Li, Q., Zhu, Y., Song, H. DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. Security and Communication Networks, 2020, 1-11. https://doi.org/10.1155/2020/8890306
37. Thakkar, A., Lohiya, R. Role of Swarm and Evolutionary Algorithms for Intrusion Detection System: A Survey. Swarm and Evolutionary Computation, 2020, 53, 100631. https://doi.org/10.1016/j.swevo.2019.100631
38. Wang, G.-G., Deb, S., Coelho, L. d. S. Elephant Herding Optimization. 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), 2015, 1-5. https://doi.org/10.1109/ISCBI.2015.8
39. Xue, B., Zhao, H., Yao, W. Deep Transfer Learning for IoT Intrusion Detection. 3rd International Conference on Computing, Networks and Internet of Things (CNIOT), 2022, 88-94. https://doi.org/10.1109/CNIOT55862.2022.00023
40. Zhang, R., Wang, X. Network Intrusion Monitoring Based on Improved Ant Colony Algorithm. International Conference on Computer Network Security and Software Engineering (CNSSE 2022), 2022, 122900D-122900D-7. https://doi.org/10.1117/12.2640698
41. Zhang, Y., Chen, X., Jin, L., Wu, L., Li, Y. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. IEEE Access, 2019, 7, 37004-37016. https://doi.org/10.1109/ACCESS.2019.2905041