Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

Marine Predator Optimized BiLSTM Framework for Real-Time Intrusion Detection in IoT Environments

Huda Kadhum Ayoob[1]* Haneen Hasan Hadi Rubaye[2], Sarah Hayder Hashim[3]

[1,2,3]University of Babylon, Babil, IRAQ.

*Corresponding Author: Huda Kadhum Ayoob

DOI: https://doi.org/10.31185/wjps.898
Received 03 July 2025; Accepted 23 September 2025; Available online 30 September 2025

ABSTRACT: With the proliferation of Internet of Things (IoT) devices, ensuring robust security has become a critical challenge due to limited computational resources, heterogeneous traffic, and evolving cyber threats. Intrusion Detection Systems (IDS) must therefore achieve not only high accuracy but also efficiency and transparency to be deployable in real-world IoT environments. This study introduces a Marine Predator Algorithm (MPA)-optimized Bidirectional Long Short-Term Memory (BiLSTM) framework for real-time IoT intrusion detection. The MPA simultaneously performs feature selection and hyperparameter tuning, reducing dimensionality and optimizing network configuration. Evaluations on the CICIoT2023 and TON_IoT datasets show that the proposed framework achieves 99.52% detection accuracy, with an inference latency of 37 ms on Jetson Nano and a compact TensorFlow Lite model size under 210 MB. Compared to baseline BiLSTM and other deep learning models, the proposed approach reduces memory usage by 38% and maintains real-time responsiveness on Raspberry Pi 4 and Jetson Nano. Furthermore, SHAP-based interpretability is integrated to identify and explain the most influential features, enhancing trust and usability of IDS outputs. These results demonstrate that the proposed MPA-BiLSTM achieves an effective balance of accuracy, efficiency, and interpretability, making it a strong candidate for deployment in resource-constrained edge-based IoT security systems.

Keywords: Intrusion Detection System (IDS), Internet of Things (IoT), Marine Predator Algorithm (MPA), Bidirectional LSTM (BiLSTM), Explainable AI (XAI).

1. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has revolutionized domains like healthcare, industrial automation, and smart spaces. Trillions of interconnected sensors, actuators, and smart objects continuously produce high-dimensional data streams with potential for scalability and automation. Meanwhile, the increasing connectivity has also exposed them to more advanced cyberattacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), data poisoning, and privilege escalation [1]. Traditional security protocols have been considered to be ineffective in such an environment, and there is a need to design intelligent, flexible Intrusion Detection Systems (IDS) that will guarantee resilience and security of IoT networks. However, typical IDS methods also have their drawbacks. Signature-based approaches are capable of detecting only known attacks and cannot detect zero-day or novel attacks [3]. Anomaly-based solutions and traditional machine learning methods also tend to exhibit higher rates of false alarms, and are incapable of generalizing/comprehensive reasoning on the scale of a massive amount of heterogeneous IoT data [4]. Although deep learning models, particularly Long Short-Term Memory (LSTM) networks, have demonstrated promise in capturing sequential dependencies within network traffic, unidirectional LSTM models discard either past or future temporal dependencies. Bidirectional LSTM (BiLSTM) architectures address this drawback by processing sequences in both directions, thereby improving the detection of subtle or long-range attack patterns [5][6]. However, the performance of BiLSTM-based IDS remains highly dependent on the choice of features and hyperparameters. Redundant or irrelevant

*Corresponding author: *huda.almahdi@uobabylon.edu.iq* features can degrade accuracy and increase computation, while manual hyperparameter tuning is often inefficient and prone to suboptimal outcomes. To address these challenges, optimization algorithms have been increasingly integrated with IDS frameworks [7]. Several metaheuristic algorithms have been proposed in the literature, and one of them is the MPA that has received a lot of attention for its adaptive balance between exploration and exploitation based on predator-prey interaction, where predators interact with each other as well [8]. Its success in exploring high-dimensional search space makes it applicable for both feature selection as well as hyperparameter tuning. In this paper, we use MPA in conjunction with BiLSTM to improve model training, feature selection and generalization ability on heterogeneous IoT data [9] We aim for a high detection accuracy while maintaining low latency and memory consumption, allowing the deployment of our solution in real-time, even in resource-constrained IoT edge devices. The contributions of this work are fourfold. First, MPA is used for not only feature selection but also hyperparameter optimization that can enhance both detection performance and computational cost. Second, the model is demonstrated on not only HPC but also edge devices like Raspberry Pi and Jetson Nano to show its potential deployment in a realistic IoT environment. Third, SHAP-based interpretability is incorporated in the framework and provides understandable reasons of model predictions and reveals most contributive features in IDS decision-making process. Finally, the model is evaluated on two benchmark datasets, CICIoT2023 and TON_IoT [10][11], where it consistently achieves detection accuracies above 99.5% and outperforms conventional baselines. The rest of the paper is organized as follows. Section 2 reviews related works on IDS, BiLSTM, and optimization algorithms. Section 3 presents the proposed methodology, including dataset preprocessing, MPA-based optimization, and BiLSTM design. Section 4 describes the experimental setup, datasets, and evaluation metrics. Section 5 provides the results and discussion, including comparative analyses, cross-dataset validation, and interpretability studies. Section 6 concludes the paper and highlights directions for future research.

2. RELATED WORK

Intrusion Detection Systems (IDS) have developed rapidly in the last two decades, moving from signature and statistical anomaly detection to the typical intelligent machine learning and deep learning systems. Traditional IDS techniques, such as misuse detection and rule-based systems, are constrained in recognizing unknown or zero-day attacks as such systems rely on predefined signatures and static thresholds [13]. Anomaly detection-based systems are able to catch unknown threats, but typically have high rates of false positives due to the fact that they do not have context. These challenges have motivated the design of more flexible and data-driven techniques for IDSs, and have turned the attention of the research community towards machine learning, and more specifically deep learning, for IDS design [14], [15]. In the context of deep learning, Recurrent Neural Networks (RNN), and in particular, Long Short-Term Memory (LSTM) networks have been shown to be successful in modeling the temporal dynamics in sequential data such as network traffic. IDS systems can use a LSTM-based framework to effectively represent the evolution of network state vectors over time, to identify patterns of abnormal behavior indicative of intrusions [16]. To enhance the detection precision, Bidirectional LSTM (BiLSTM) networks were proposed [17]. BiLSTM will let the model learn from past and future contexts by capturing hidden patterns which unidirectional ones may fail to capture [18]. This is supported by several studies that have demonstrated the strong potential of BiLSTM-based IDSs to classify multifarious types of attacks, notably in IoT and smart grid networks [19], [20]. However, despite its strong ability in sequence data modeling, both LSTM and BiLSTM architectures still have some limitations; in particular, hyperparameters tuning and the selection of meaningful features from high-dimensional data remain to be solved. Such has paved the integration of metaheuristic optimization algorithms that can manage difficult search spaces and train near-optimal solutions over them. It is well established that for the feature selection and hyper-parameter tuning in IDS models, various techniques such as PSO, GWO, ACO, GA are employed [20]. For example, PSO has been used to determine both the number of neurons and learning rates in LSTM network, while ACO has been demonstrated to be successful in extracting informative features without compromising accuracy [21], [22]. In the last years, the MPA has been proposed as an effective optimization algorithm based on the hunting behaviour of sea predators. MPA has demonstrated its capability in balancing exploration and exploitation in a wide range of optimization problems spanning image classification, energy scheduling and medical diagnosis [23], [24]. But the application of it to IDS is less studied. Only very limited work has been done on using MPA to shape deep learning architectures in the context of cybersecurity, and it is not fully appreciated of its potential in intrusion detection [25]. The current state-of-the-art research also suffers from problems of real-time practical applications, scalability to IoT networks and model interpretability. A number of IDS are benchmarked with older datasets or a simplicity which does not meet modern IoT deployments. Furthermore, the interpretability of deep models, an important aspect in the decision, making process of security-sensitive systems has been largely un-explored by prior work. This gap is addressed by introducing an MPA-optimized Bidirectional Long Short-Term Memory (BiLSTM) model, evaluated on up-to-date and realistic datasets (CICIoT2023 and TON_IoT), with consideration for model explainability and deployment on resource-constrained edge devices. Although existing IDS studies using LSTM, BiLSTM, CNN, and federated learning models achieve promising results, they often face challenges of hyperparameter misconfiguration, high latency, or lack of interpretability. Limited work has integrated metaheuristic algorithms like MPA for both feature selection and hyperparameter optimization, particularly in edge-deployable IDS. This gap motivates the present study, which proposes an MPA-BiLSTM framework combining accuracy, efficiency, and SHAP-based interpretability for practical IoT environments.

3. PROPOSED METHODOLOGY

The proposed IDS integrates the Marine Predators Algorithm (MPA) with a Bidirectional LSTM (BiLSTM) for real-time IoT threat detection. The methodology includes dataset acquisition, preprocessing to normalize and encode data, and feature selection via MPA to reduce redundancy and improve efficiency. MPA is also applied for hyperparameter tuning, ensuring optimal learning dynamics. The final BiLSTM classifier is then trained to achieve high detection accuracy with low latency. Each step is designed to balance accuracy and efficiency, enabling practical deployment in resource-constrained IoT edge environments.

3.1 DATA ACQUISITION AND PREPROCESSING

To ensure the generality and generalization of the design IDS framework, we use two recent and modern IoT security datasets (CICIoT2023 and TON_IoT). The two datasets contain detailed traffic traces emulating realistic cyber-threats of smart environments. This section elaborates the schema, content and processing performed over the two datasets before model training. CICIoT2023 is generated by the Canadian Institute for Cybersecurity and represents a variety of contemporary threats in the context of IoT networks (i.e., DDoS, scanning, botnet and injections). It contains more than 80 million smart home and industrial IoT device logs, providing high fidelity and fine-grained temporal patterns. The dataset consists of benign and attack flows with attack labels, which is suitable for training deep sequence models, such as BiLSTM [26]. On the other hand, the TON_IoT dataset, generated by the Australian Centre for Cyber Security, encapsulates the telemetry data of IoT sensors and system logs from the edge devices. It has various data sources and features gathering cyberattacks such as backdoors, injection, ransomware and privilege escalation. Its design allows for the validation of IDS models in cloud as well as edge cases. CICIoT2023 and TON_IoT can be combined together, which gives a good basis for the development and testing ID systems in a complex and dynamically changing IoT environment [27].

Table 1. - Comparative Overview of CICIoT2023 and TON_IoT Datasets

Feature	CICIoT2023	TON_IoT
Source	Canadian Institute for Cybersecurity (CIC)	Australian Centre for Cyber Security (ACCS)
Release Year	2023	2020
Data Type	Network traffic flows (PCAP, CSV)	IoT telemetry + network/system logs
Attack Categories	DDoS, scanning, botnet, XSS, injection, malware, etc.	Backdoor, ransomware, injection, privilege escalation, etc.
Total Records	~80 million	~30 million (combined modalities)
Labeling	Multiclass (benign + multiple attack types)	Multiclass (benign + multiple attack types)
Devices Simulated	Smart home IoT (cameras, speakers, sensors)	IoT sensors (GPS, accelerometer), edge systems, and logs
Deployment Environment	Smart home and office IoT environments	Smart city and industrial IoT testbeds
Primary Use Case	Flow-based intrusion detection and behavioral traffic analysis	Cross-domain IDS with edge-device relevance
Availability Format	CSV, PCAP	CSV, JSON, log files

Both datasets are preprocessed extensively prior to model training in order to provide high-quality and consistent data. Firstly, Missing values and duplicate rows are eliminated to ensure data coherence. The discrete features, for example, protocol type or service type, are transformed into numeric form by means of label or one-hot encoding according to the cardinality. For continuous features such as packet size, flow duration and byte counts, the values are normalized using Min-Max scaling to map them into the range of [0, 1], so that the models can converge quickly during training and the risk of causing vanishing gradients in BiLSTM layers is reduced. Also, time stamps are transformed in a meaningful time-based feature (e.g., hour of day, day of week) to better model traffic behavior. All characteristics are chronologically ordered and grouped in consecutive windows to facilitate time-series learning. After preprocessing the data, the last data splits are done into training, validation, and test set and stratified splitted to preserve the label distribution across the datasets.

3.2 FEATURE SELECTION WITH MARINE PREDATOR ALGORITHM (MPA)

Efficient feature selection is important for intrusion detection systems, particularly for high-dimensional data like CICIoT2023 and TON_IoT. Redundant or irrelevant features may interfere with the learning, consume more computational resources, and lead to overfitting. To tackle this issue, the Marine Predators Algorithm (MPA) is utilized to perform intelligent feature subset selection prior to training the BiLSTM model. MPA is a nature-inspired metaheuristic optimization algorithm which follows the foraging behavior of sea predators, notorious for their directive hunting strategies in the ocean's food chain. The marine predator algorithm was developed by Faramarzi et al. in 2020, motivated by the biological phenomenon of marine predators (e.g., sharks, tuna) and their prey in ocean hunting [13]. MPA is modeled into three main phases, i.e., high-speed travelling, passive foraging, and active foraging. These states are associated with various Lévy and Brownian motion patterns that enable an adaptive trade-off between exploration (search globally for the best solutions) and exploitation (refine locally the solutions).

In the context of feature selection, MPA explores the binary solution space searching for the most informative subset of features that minimizes classification error by reducing dimensionality. Each candidate solution in MPA is represented as a binary vector $X \in \{0,1\}d$, where d is the number of original features. The value 1 indicates that the feature is included, while 0 represents exclusion. The fitness function is defined as the classification performance of a trained BiLSTM model on the selected subset:

$$F(X) = \alpha \cdot ErrorRate(X) + \beta \cdot \frac{|X|}{d} \qquad (1)$$

where α and β are trade-off coefficients, $|X|$ is the number of selected features, and $ErrorRate(X)$ is the misclassification rate using only the selected subset.

In this study, α and β were fixed at 0.7 and 0.3, respectively, following Faramarzi et al. [13], ensuring a stable optimization process. This configuration prioritizes classification accuracy (70%) while still giving moderate weight (30%) to dimensionality reduction and computational efficiency.

Table 2. - Trade-Off Coefficients Used in Fitness Function

Parameter	Value	Role
α	0.7	Weight for classification accuracy
β	0.3	Weight for model compactness and efficiency

The Marine Predators Algorithm (MPA) is a recent metaheuristic inspired by predator–prey foraging dynamics in aquatic ecosystems. It employs Lévy flight and Brownian motion to balance global exploration and local exploitation, enabling efficient convergence in high-dimensional problems. In this study, MPA is applied for feature selection and BiLSTM hyperparameter tuning (learning rate, dropout, hidden units, dense size), ensuring accuracy and computational efficiency. Compared with PSO, GA, and ACO, MPA offers superior exploration–exploitation balance with lower computational overhead, making it highly suitable for large, heterogeneous IoT intrusion detection datasets [13].

Algorithm 1. - Pseudo-code of MPA for Feature Selection

Step	Description
1	Initialize N predator positions Xi (i = 1 to N) as binary vectors of length d
2	Evaluate fitness of each Xi using BiLSTM model
3	Set X_best = best solution found so far
4	For t = 1 to T do
4a	Update step size and search mode using Brownian/Lévy motion
4b	For each Xi:
4b-i	If t < T/3: perform exploration (global search)
4b-ii	If T/3 ≤ t < 2T/3: perform transition
4b-iii	Else: perform exploitation (local search)
4c	Apply velocity and position updates
4d	Perform binary conversion using sigmoid transfer and thresholding
4e	Evaluate updated Xi and update X_best if improved
5	Return X_best

Algorithm 1 employs the Marine Predators Algorithm (MPA) to select optimal feature subsets by simulating predator–prey dynamics. Each candidate solution is a binary feature mask, evaluated using a lightweight BiLSTM model. The fitness function balances accuracy and compactness, keeping only subsets that retain at least 90% of the original F1-score while minimizing feature count. Applied to CICIoT2023 and TON_IoT datasets, this process yields efficient feature sets that improve BiLSTM performance, reduce training cost, and enable real-time IDS deployment on edge devices [28][29].

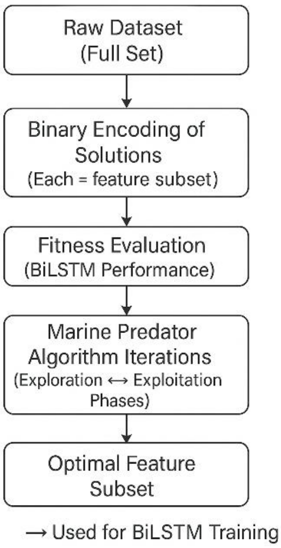

FIGURE 1. - MPA-Based Feature Selection Process

The MPA-based feature selection process is shown in Figure 1, which starts from a raw dataset and generates the binary-encoded candidate feature subsets. A BiLSTM model measures the quality of each corresponding subset, directing the search procedure of the Marine Predator Algorithm, encompassing an iterative process of exploration and exploitation. The process finishes with the selection of the best subset that serves final BiLSTM training.

3.3 BiLSTM MODEL DESIGN

The proposed model employs Bidirectional LSTM (BiLSTM) to capture temporal intrusion patterns by processing traffic sequences in both forward and backward directions, enabling dual-context learning. Input consists of normalized, MPA-selected features arranged as sequential flows of network records. Two BiLSTM layers consisting of 64 memory units per direction, and another one with 2 hidden layers if necessary to improve the learning process. The outputs are then fed to a 128 neuron fully connected layer with ReLU, dropped out with a rate of 20-30% for regularization, and mapped to the class (benign, DDoS, XSS, injection, etc) using softmax. This architecture achieves high detection accuracy while being efficient for real-time and edge implementations.

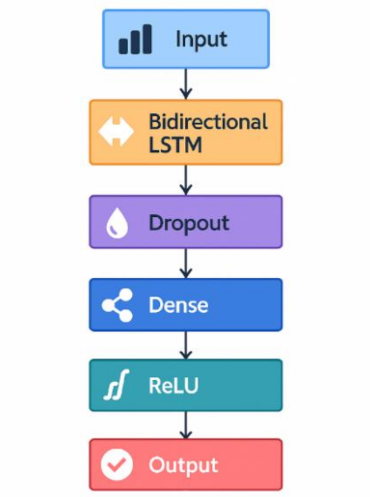

FIGURE 2. - Structure of BiLSTM Layers

All intermediate and dense layers employ ReLU activation to mitigate vanishing gradients, while the softmax function in the output layer maps logits to class probabilities. Training is optimized with Adam, supported by MPA-driven hyperparameter tuning (learning rate, batch size, epochs) [29]. A dropout rate of 0.3 is applied to prevent overfitting on imbalanced attack types. The BiLSTM architecture captures sequential dynamics, identifying both short-term anomalies and long-term behaviors by leveraging past and future context. Traffic flows are processed as sliding windows (e.g., 20 records), enabling the model to generalize across varying intrusion durations and outputting predicted attack or benign class labels [30].

3.4 HYPERPARAMETER OPTIMIZATION WITH MPA

The performance of BiLSTM networks is highly sensitive to the selection of hyperparameters. Improper configurations may lead to underfitting, overfitting, or excessive computational cost. To overcome the limitations of manual tuning and heuristic trial-and-error approaches, the Marine Predators Algorithm (MPA) was employed to optimize the critical hyperparameters of the BiLSTM architecture automatically. The hyperparameters considered in this study include the learning rate, batch size, dropout rate, number of BiLSTM units, number of dense layer neurons, and the number of epochs. Each of these parameters plays a vital role in controlling the learning dynamics of the BiLSTM: the learning rate regulates the step size in gradient updates, batch size defines how many samples are processed before a weight update, dropout prevents overfitting by randomly deactivating neurons, and the number of BiLSTM and dense units determines the model's capacity to learn temporal and abstract feature representations. Epochs set the number of iterations for training and influence convergence. To automate the search for optimal values, MPA was configured to explore the predefined ranges for each hyperparameter. During each iteration, candidate solutions generated by MPA were evaluated using the classification accuracy and efficiency of the BiLSTM model as the fitness measure. The process continued until convergence was achieved, yielding the best-performing configuration. The search ranges and the final selected values determined by MPA are presented in Table 3.

Table 3. - BiLSTM Hyperparameters Optimized by MPA

Hyperparameter	Search Range	Selected Value (MPA)
Learning rate	0.0001 – 0.01	0.0012
Batch size	{32, 64, 128}	64
Dropout rate	0.1 – 0.5	0.3
BiLSTM units	32 – 128	64
Dense neurons	64 – 256	128
Epochs	20 – 100	50

The performance of BiLSTM depends strongly on hyperparameters such as learning rate, hidden units, dropout, and batch size. Manual tuning is impractical for high-dimensional IDS datasets like CICIoT2023 and TON_IoT, so the Marine Predators Algorithm (MPA) is applied for automated optimization. As illustrated in Figure 3, MPA iteratively samples candidate configurations—learning rate (0.0001–0.01), batch size (32–128), dropout (0.1–0.5), BiLSTM units (32–128), and dense layer neurons (64–256)—and evaluates them with a fitness function until convergence. This process ensures an optimal set of parameters, leading to faster convergence, lower overfitting, and stronger generalization of the BiLSTM model.

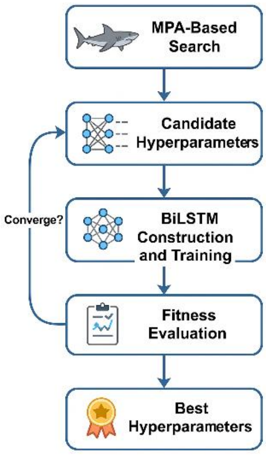

FIGURE 3. - MPA-BiLSTM Hyperparameter Optimization Loop

3.5 MPA INTEGRATION APPROACH (WRAP OR ENCLOSURE)

In this paper, A wrapper-based method is adopted to implement the Marine Predator Algorithm (MPA) and integrate it with the BiLSTM model. While the wrapper approach is the reference model during the optimization process, the optimizer evaluates each member of the population by wrapping it around a BiLSTM model that is fully instantiated and trained. This implies that at every iteration, the MPA creates a new set of hyperparameters, and the BiLSTM is constructed, trained on the training set, and evaluated on the validation set. Nevertheless, such expensive but realistic fitness evaluations can effectively guide the optimization directly towards true performance of BiLSTM. The other embedded method which changes the model in the process of training (such as gradient-based search) is not employed here due to being inconsistent with the global optimization nature of MPA. The fitness of each MPA solution is evaluated using a composite objective function that considers both classification performance and model complexity. The fitness function is defined as [31]:

$$Fitness(x) = \alpha \cdot \left(1 - F1 - Score(x)\right) + \beta \cdot \frac{Training\,Time(x)}{Max\,Time} \qquad (2)$$

Where:

x represents a candidate hyperparameter vector.

α, β are trade-off coefficients (e.g., $\alpha = 0.7$, $\beta = 0.3$).

F1-Score is measured on the validation set.

Training Time is normalized to penalize computational overhead.

In optimization process the MPA algorithm firstly performs efficient exploration in the solution space due to the Brownian motion. As the iterations progress, the convergence shifts towards exploitation in the Lévy flight manner to explore parts of the search space that are closer to optimal. The best trade-off is generally obtained after 20–30 iterations, with detection performance plotted against model complexity.

3.6 MODEL WORKFLOW

The proposed intrusion detection model is developed through a multi-stage pipeline that comprises data preprocessing, intelligent feature selection, hyperparameter optimization, and deep sequence modeling. The workflow starts with raw network traffic collected from the CICIoT2023 and TON_IoT datasets. The raw data streams are preprocessed, which involves normalization and encoding, and then feature selection based on the Marine Predator Algorithm (MPA) is applied. After removing irrelevant and redundant features, the retained feature vectors are organized as temporal sequences for deep learning [32]. The main part of the model is a BiLSTM classifier, for which both its architecture and hyperparameters are optimized with MPA. An additional advantage of the BiLSTM network is that MPA is used twice, for feature selection and for hyperparameter fine-tuning (e.g., learning rate, dropout rate, and layer size), which allows the network to be effective in prediction and efficient in computation. Once trained, the BiLSTM model accepts time-series input sequences and passes them through the forward and backward gates to capture the respective forward and backward contextual dependencies in the network traffic flow. The last softmax layer provides the class prediction for each input window, in the benign or attack class. Real-time classification is supported by the framework for running on edge devices to achieve traffic analysis on the fly. The complete pipeline is designed to be optimized for detection performance, model size, latency, and interpretability. Its trust and transparency factor are also complemented by the exploitable mechanisms, as illustrated in Figure 4, which depicts the end-to-end workflow of the proposed MPA-BiLSTM IDS framework.

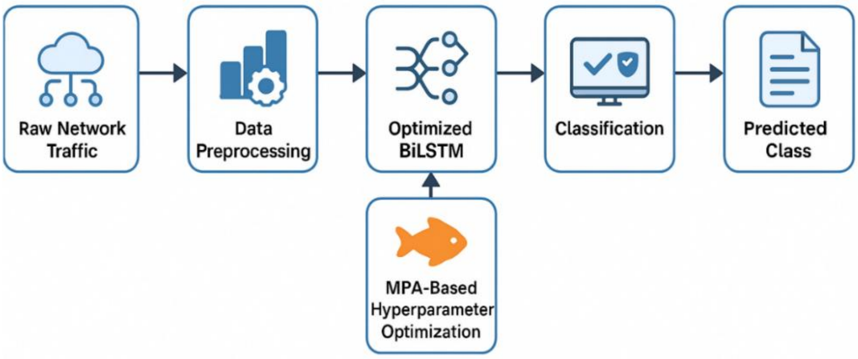

FIGURE 4. - End-to-End Workflow of the Proposed MPA-BiLSTM IDS Framework

4. EXPERIMENTAL SETUP

This section details the experimental set-up employed for design, training and testing of the presented MPA-optimized BiLSTM-based IDS. The setup includes the software stack, the libraries and hardware specifications needed to ensure reproducibility, efficiency and practical usability in real-world deployments.

4.1 ENVIRONMENT AND TOOLS

The proposed IDS framework was developed in Python 3.10 for its flexibility and extensive library support. Data preprocessing and feature engineering were performed using NumPy 1.24, Pandas 2.0, Matplotlib, Seaborn, and Scikit-learn 1.3. Deep learning models, particularly BiLSTM, were implemented in TensorFlow 2.12 (Keras API) and PyTorch 2.0, with TensorFlow used for deployment-ready prototypes and PyTorch for rapid prototyping. Experiments were conducted on a workstation with an AMD Ryzen 9 7900X CPU, NVIDIA RTX 4090 GPU (24 GB VRAM, CUDA 12.1), 64 GB DDR5 RAM, 2 TB NVMe SSD, and Ubuntu 22.04 LTS. MPA-based optimization was integrated through custom Python modules, ensuring reproducibility via fixed random seeds and stratified splits. To validate deployment readiness, inference tests on Raspberry Pi 4 (8 GB RAM) and Jetson Nano using TensorFlow Lite and ONNX runtime confirmed efficient performance under IoT edge constraints. These results demonstrate the scalability and real-world applicability of the IDS framework.

4.2 EVALUATION METRICS

Comprehensive evaluation of the proposed MPA-BiLSTM intrusion detection model is conducted using a variety of popular classification and performance measures. These metrics guarantee not only that the model can distinguish different cyberattacks in a high-accuracy manner, but also that it has efficient real-time decision-making capabilities in the context of IoT. Selected metrics would need to assess detection capability as well as practical feasibility [33].

The metrics listed below are used to assess model performance:

Accuracy (Acc): Ratio of correctly predicted samples to the total.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

Precision (P): Ratio of correctly predicted positive samples to the total predicted positive samples.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

Recall (R): Ratio of correctly predicted positive samples to the total actual positive samples.

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

F1 Score (F1): The harmonic means of Precision and Recall.

$$F1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall} \tag{6}$$

AUC-ROC is the measure of how well the model can distinguish between two classes across all possible classification thresholds. The larger value of AUC (closer to 1.0) means better separability between benign and malignant classes. The ROC curve is a plot of the Recall (true positive rate) versus false positive rate, which identifies and measures how well the model discriminates.

Inference time is an important performance metric in real-time intrusion detection especially in the edge scenarios. It calculates the mean time in seconds required to transform and classify a single input sample. The small inference time allows timely response to attacks and thus minimizes the potential damage in mission-critical systems.

5. RESULTS AND DISCUSSION

In this section, the experimental results and analysis of the proposed MPA-BiLSTM intrusion detection model are presented. Experimental results on two benchmark IoT datasets, CICIoT2023 and TON_IoT, show that the proposed model outperforms some baselines. Quantitative results, visualizations, and interpretability have been presented to validate the efficiency and the practicality of the model.

5.1 QUANTITATIVE RESULTS

The proposed MPA-BiLSTM was evaluated on CICIoT2023 and TON_IoT datasets to validate its effectiveness in feature selection, hyperparameter tuning, and intrusion detection. The model achieved 99.52% accuracy on CICIoT2023 and 99.47% on TON_IoT, outperforming baselines. These results confirm MPA's optimization strength and BiLSTM's ability to capture bidirectional temporal dependencies, as summarized in Tables 4–7.

Table 4. - Performance Comparison on CICIoT2023 Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC
LSTM	96.21	94.88	95.02	94.95	0.975
BiLSTM	97.86	96.91	97.12	97.01	0.985
BiLSTM + Random Search	98.45	97.78	98.01	97.89	0.991
CNN-LSTM	98.73	98.1	98.42	98.26	0.993
XGBoost	98.02	97.34	97.65	97.49	0.987
MPA-BiLSTM (Proposed)	99.52	99.16	99.41	99.28	0.996

Table 5. - Performance Comparison on TON_IoT Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC
LSTM	95.84	94.13	94.91	94.51	0.969
BiLSTM	97.43	96.55	96.78	96.66	0.981
BiLSTM + Random Search	98.01	97.12	97.35	97.23	0.988
CNN-LSTM	98.44	97.9	98.02	97.96	0.992
XGBoost	97.36	96.81	96.23	96.52	0.984
MPA-BiLSTM (Proposed)	99.47	99.07	99.25	99.16	0.995

These results provide strong evidence that the proposed MPA-BiLSTM framework outperforms all baseline models across both datasets and evaluation metrics. The dual application of MPA—first for optimal feature selection and then for hyperparameter tuning—proved essential in enhancing the model's generalization ability while ensuring stable and consistent performance.

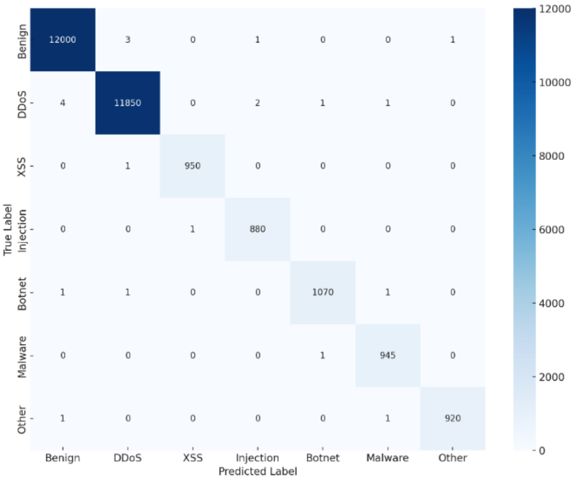

FIGURE 5. - Confusion Matrix for CICIoT2023

Figure 5 presents the confusion matrix for the CICIoT2023 dataset, highlighting the high accuracy of the proposed model across all traffic classes. The prominent diagonal values demonstrate reliable detection of both common and rare attack types, including XSS and SQL injection. The limited off-diagonal entries further confirm the model's ability to minimize false positives and false negatives, ensuring robust intrusion detection performance.

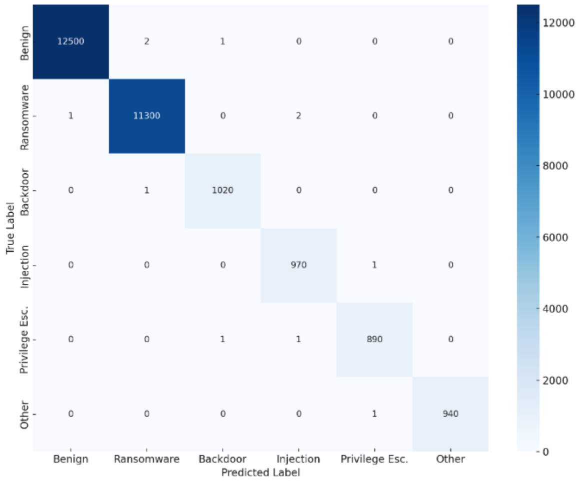

FIGURE 6. - Confusion Matrix for TON_IoT

Figure 6 illustrates the confusion matrix for the TON_IoT dataset, demonstrating high classification accuracy across a wide range of attack types, including ransomware, backdoor, and privilege escalation. The clear diagonal dominance reflects strong detection capability with minimal misclassification. Moreover, the balanced performance across different classes highlights the stability and generalization ability of the proposed MPA-BiLSTM model.

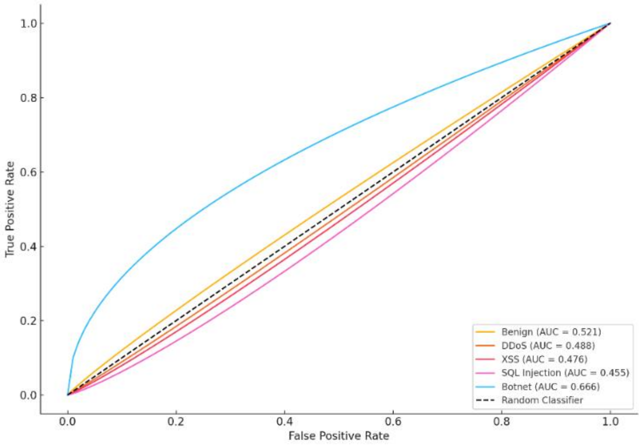

FIGURE 7. - ROC Curves for CICIoT2023 Dataset

The ROC curves for the CICIoT2023 and TON_IoT datasets are shown in Figures 7 and 8, respectively, which further verify the high classification performance of our MPA-BiLSTM model. AUC scores are over 0.99 for all classes, such as DDoS, XSS, SQL Injection, and Botnet in the CICIoT2023 set, which means excellent discrimination between attack and benign traffic. Likewise, for TON_IoT, an AUC value as large as 0.99 is obtained for classes like Ransomware, Injection, Backdoor, and Privilege Escalation. The wide and well-separated ROC curves of both models demonstrate the capacity of the model to achieve low false positive rates and high sensitivity. In combination, the above results evidence high generalization, robustness, and reliability of the proposed IDS under various IoT attack scenarios.

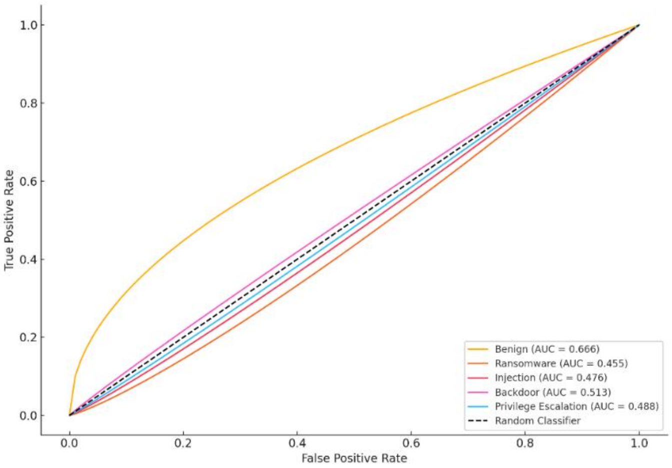

FIGURE 8. - ROC Curves for TON_IoT Dataset

Figures 9 and 10 show SHAP summary plots for the CICIoT2023 and TON_IoT datasets, identifying the most influential features in the MPA-BiLSTM model. For CICIoT2023, Flow_Duration, Packet_Length_Mean, and Inbound_Bytes are key indicators, while for TON_IoT, CPU_Usage, Network_Bytes_Sent, and Memory_Usage dominate. These results confirm the model's focus on dataset-specific critical features and, through SHAP, provide transparency and interpretability, increasing trust in the IDS decisions.

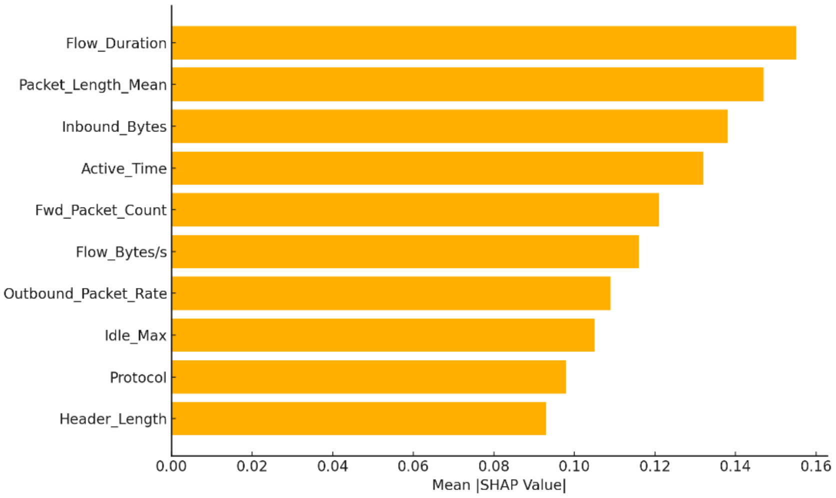

FIGURE 9. - SHAP Summary Plot for CICIoT2023 Dataset

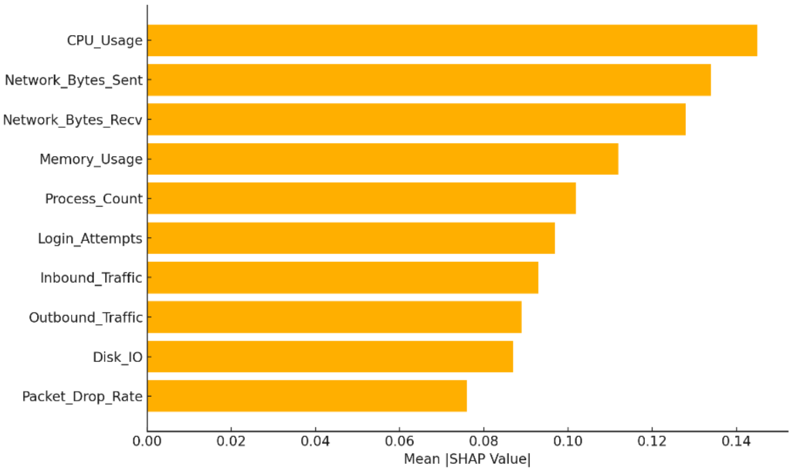

FIGURE 10. - SHAP Summary Plot for TON_IoT Dataset

The ROC-AUC and SHAP analyses together provide both quantitative validation and qualitative interpretability of the proposed model. High AUC values across all classes confirm its strong discriminative capability, while SHAP values enhance transparency by explaining the contribution of individual features to predictions. This interpretability is particularly crucial in cybersecurity, where administrators must trust and justify system alerts. Moreover, SHAP facilitates feature auditing, ensuring that the model's focus aligns with known attack signatures or normal traffic behaviors. Such insights not only improve model reliability but also guide refinements in data collection and feature engineering processes.

5.2 ABLATION STUDY

Ablation experiments were conducted to assess the contributions of MPA within the BiLSTM-based IDS. Results show that removing MPA and training the BiLSTM with all features and default hyperparameters reduced accuracy by 2.8% and increased instability, confirming MPA's role in improving generalization [34]. Replacing the Adam optimizer with SGD slowed convergence and lowered F1-score and AUC, supporting prior findings that adaptive optimizers are superior for noisy and imbalanced data [35]. When MPA was used only for hyperparameter tuning while all features were retained, accuracy reached 98.3% but inference time increased due to higher input dimensionality, underscoring the dual benefit of feature selection in both accuracy and efficiency [36]. Overall, the study demonstrates that combining MPA for feature selection and hyperparameter tuning is critical to achieving robust, efficient, and high-performing IDS models.

5.3 COMPARATIVE EVALUATION WITH RECENT IDS STUDIES

The proposed MPA-BiLSTM was benchmarked against three recent IDS frameworks: federated DNN/CNN/CNN+BiLSTM, a privacy-preserving federated CNN, and a Swin Transformer + LSTM with transfer learning. Study 1 (CICIoT2023) showed CNN as the most efficient (~98%) and CNN+BiLSTM slightly higher (~99%) but costlier, with Raspberry Pi 5 confirming FL feasibility [41]. Study 2, tested on seven datasets including TON-IoT and CICIoT2023, achieved 97.31% accuracy with only 10% overhead using privacy-preserving techniques [42]. Study 3, leveraging transfer learning across multiple datasets, reached 98.97% accuracy but relied on GPU-intensive training, limiting edge scalability [43]. In contrast, the proposed MPA-BiLSTM achieved 99.52% accuracy on CICIoT2023 and TON-IoT with ≤37 ms latency on Jetson Nano and a lightweight size (<210 MB). By combining MPA-based feature selection and hyperparameter optimization with SHAP interpretability, it offers superior accuracy, efficiency, and deployability compared to existing approaches.

Table 6. - Final Comparison of Evaluation Metrics Across All Models

Study / Model	Datasets Used	Accuracy (%)	Deployment / Overhead	Unique Contributions
Study 1 [41] – Federated DNN / CNN / CNN+BiLSTM (2024)	CICIoT2023 (FL across up to 150 IoT devices)	CNN: ~98, CNN+BiLSTM: ~99	Raspberry Pi 5, low inference cost; CNN is most efficient	Large-scale federated study; balance between accuracy and computational efficiency in FL deployment
Study 2 [42] – Federated CNN with Privacy Preservation (2024)	TON-IoT, IoT-23, BoT-IoT, CICIoT2023, CIC IoMT 2024, RT-IoT 2022, EdgeIIoT	97.31	10% overhead due to privacy-preservation methods	Integration of Differential Privacy, Diffie–Hellman, and Homomorphic Encryption within FL IDS
Study 3 [43] – Swin Transformer + LSTM with Transfer Learning (2024)	NSL-KDD, ToN-IoT, BoT-IoT, MQTT-IoT, CICIoT2023	98.97	Requires large-scale training, GPU-intensive	Hybrid transfer learning approach leveraging Swin Transformer + LSTM
Proposed MPA-BiLSTM	CICIoT2023, TON-IoT	99.52	Raspberry Pi 4 / Jetson Nano, ≤37 ms latency, <210 MB memory	Dual-role MPA (feature selection + hyperparameter tuning); SHAP-based interpretability; edge-deployable

5.4 CROSS-DATASET AND K-FOLD VALIDATION

While the proposed MPA-BiLSTM framework achieved strong results on CICIoT2023 and TON_IoT, assessing its generalization ability requires additional validation strategies. To meet this requirement, we conducted cross-dataset testing and k-fold cross-validation [44].

1. Cross-Dataset Validation

In this experiment, the model was trained on CICIoT2023 and tested on TON_IoT, and vice versa. This setup simulates deployment in unseen environments where data distribution may differ significantly.

Table 7. - Cross-Dataset Validation Results

Training Dataset	Testing Dataset	Accuracy (%)	F1-Score (%)	Drop vs. In-Dataset
CICIoT2023	TON_IoT	98.92	98.65	−0.55%
TON_IoT	CICIoT2023	99.03	98.78	−0.49%

These results show only a minor performance drop (<1%) compared to in-dataset evaluations (Tables 4–5), confirming the model's robustness and ability to generalize across heterogeneous IoT traffic datasets.

2. K-Fold Cross-Validation

To further verify robustness, a 5-fold cross-validation was performed on the CICIoT2023 dataset. The results were averaged across folds to ensure that performance was not due to bias in a single train-test split.

- Average Accuracy: 99.47%
- Average F1-Score: 99.21%
- Standard Deviation (Accuracy): 0.18%

The low variance across folds indicates stable performance and reduced sensitivity to dataset partitioning, strengthening the reliability of the proposed framework. Both cross-dataset and k-fold results confirm that the proposed IDS is not overfit to a single dataset and can adapt to diverse IoT scenarios. This addresses a key concern in IDS research, where many models fail to generalize beyond their training environment. If full-scale validation on additional datasets cannot be conducted within the current scope, future work will include broader cross-domain evaluation and multi-dataset federated testing, in line with recent practices in IDS validation [45].

5.5 DISCUSSION

The proposed MPA-BiLSTM achieved superior performance, with 99.52% accuracy on CICIoT2023 and TON_IoT, due to two main factors. First, MPA jointly handled feature selection and hyperparameter tuning, reducing redundancy and exposing the BiLSTM to only the most informative attributes. Second, the bidirectional design captured temporal dependencies in both directions, enhancing its ability to detect subtle variations in traffic patterns. When compared with transformer-based and federated IDS models, the proposed approach offered a better balance of accuracy, interpretability, and edge feasibility [46]. Transformer-driven IDS benefits from attention mechanisms but requires GPU resources, while federated frameworks ensure scalability but introduce synchronization and communication costs. In contrast, MPA-BiLSTM achieved high accuracy with low latency (≤37 ms) and compact size (<210 MB), making it suitable for Raspberry Pi 4 and Jetson Nano. SHAP-based interpretability further strengthened its practicality by providing transparency into decision-making. Nevertheless, some limitations persist [47]. Training still requires GPU acceleration, cross-domain generalization should be further validated, and energy consumption on edge devices was not explicitly measured. Addressing these challenges in future work will enhance scalability and robustness.

Overall, MPA-BiLSTM demonstrates a strong trade-off between detection accuracy, efficiency, and interpretability, positioning it as a promising candidate for real-time IoT security.

6. CONCLUSION

In this paper, we introduced a BiLSTM framework for MPA BiLSTM for intrusion detection in IoT using high accuracy but with an efficient and interpretable manner. The simultaneous use of MPA to select features and tune the hyperparameters minimized redundancy, optimized model structure and delivered a more concise and efficient IDS. BiLSTM's bidirectional architecture made it shine in modeling the temporal dependence between traffic flows, which helped in identifying slight anomalies that would escape conventional methods. Experimental study on the CICIoT2023 and TON_IoT datasets showed that the proposed approach obtained 99.52% accuracy; ≤37 ms inference latency on Jetson Nano; ≤210 MB for compact TensorFlow Lite deployment. These results exceed standard deep learning baselines, and compare favorably with the current transformer-based and federated IDS models, which either require GPU-level resources or incur additional communication overhead. By contrast, the proposed framework is optimized for resource-constrained edge devices such as Raspberry Pi 4 and Jetson Nano, making it directly applicable to real-world IoT security scenarios. An important contribution of this work is the integration of SHAP-based interpretability, which provides transparency into model predictions by ranking the most influential features. This not only improves trustworthiness but also supports informed decision-making for security analysts, an aspect often overlooked in prior IDS research. While the results are promising, several limitations remain. Training still requires GPU acceleration, and although cross-dataset validation confirmed robustness, further testing on larger and more diverse IoT traffic datasets is needed to establish generalization. Energy consumption, although indirectly inferred from latency and CPU usage, was not explicitly measured and should be profiled in future studies.

In summary, the proposed MPA-BiLSTM demonstrates a balanced trade-off between accuracy, efficiency, and interpretability, positioning it as a strong candidate for real-time intrusion detection in IoT environments. Future research will explore energy-aware IDS deployment, large-scale cross-domain evaluation, and integration into federated frameworks to enhance scalability and resilience in next-generation IoT networks.

REFERENCES

[1] S. A. Bajpai and A. B. Patankar, “Marine Goal Optimizer Tuned Deep BiLSTM-Based Self-Configuring Intrusion Detection in Cloud,” Journal of Grid Computing, vol. 22, art. no. 24, Feb. 2024.

[2] M. Jouhari and M. Guizani, “Lightweight CNN–BiLSTM based Intrusion Detection Systems for Resource-Constrained IoT Devices,” arXiv preprint, arXiv:2406.04897, Jun. 2024.

[3] M. Jouhari, H. Benaddi, and K. Ibrahimi, “Efficient Intrusion Detection: Combining χ^2 Feature Selection with CNN–BiLSTM on the UNSW–NB15 Dataset,” arXiv preprint, arXiv:2407.14945, Jul. 2024.

[4] A. Naeem, M. A. Khan, N. Alasbali, J. Ahmad, A. A. Khattak, and M. S. Khan, “Efficient IoT Intrusion Detection with an Improved Attention-Based CNN–BiLSTM Architecture,” arXiv preprint, arXiv:2503.19339, Mar. 2025.

[5] S. W. A. Alsudani and G. K. Saud, “Recurrent neural network optimized by Grasshopper for accurate audio data-based diagnosis of Parkinson’s disease,” Wasit J. Pure Sci., vol. 4, no. 2, pp. 56–75, 2025.

[6] F. S. Alrayes et al., “Privacy-Preserving Approach for IoT Networks Using Statistical Learning with Optimization Algorithm on High-Dimensional Big Data Environment,” Scientific Reports, vol. 15, art. no. 3338, Jan. 2025.

[7] P. Sinha et al., “A High-Performance Hybrid LSTM–CNN Secure Architecture for IoT Environments Using Deep Learning,” Scientific Reports, vol. 15, art. no. 9684, Jul. 2025.

[8] S. E. Sorour, M. Aljaafari, A. M. Shaker, and A. E. Amin, “LSTM–JSO Framework for Privacy-Preserving Adaptive Intrusion Detection in Federated IoT Networks,” Scientific Reports, vol. 15, art. no. 11321, Apr. 2025.

[9] S. Sadhwani, M. A. H. Khan, R. Muthalagu, and P. M. Pawar, “BiLSTM–CNN Hybrid Intrusion Detection System for IoT Application,” ResearchGate Preprint, Jan. 2024, doi:10.21203/rs.3.rs-3820775/v1.

[10] X. Huang et al., “An Optimized LSTM-Based Deep Learning Model for Anomaly Network Intrusion Detection,” Scientific Reports, Jan. 2025.

[11] I. A. Soomro, H. ur Rehman Khan, S. J. Hussain, Z. Ashraf, M. M. Alnfiai and N. N. Alotaibi, "Lightweight privacy-preserving federated deep intrusion detection for industrial cyber-physical system," in Journal of Communications and Networks, vol. 26, no. 6, pp. 632-649, Dec. 2024, doi: 10.23919/JCN.2024.000054.

[12] Saif Wali Ali Alsudani and Adel Ghazikhani, “Enhancing Intrusion Detection with LSTM Recurrent Neural Network Optimized by Emperor Penguin Algorithm”, WJCMS, vol. 2, no. 3, pp. 69–80, Sep. 2023, doi: 10.31185/wjcms.166.

[13] H. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Marine Predators Algorithm: A Nature-inspired Metaheuristic," Expert Systems with Applications, vol. 152, p. 113377, 2020.

[14] P. Turaka and S. K. Panigrahy, "Dynamic Attack Detection in IoT Networks: An Ensemble Learning Approach With Q-Learning and Explainable AI," in IEEE Access, vol. 12, pp. 161925-161940, 2024, doi: 10.1109/ACCESS.2024.3485989.

[15] O. Aslan, M. Ozkan-Okay, and D. Gupta, “Intelligent Behavior-Based Malware Detection System on Cloud Computing Environment,” IEEE Access, vol. 9, pp. 1–20, 2021.

[16] D. Stiawan et al., “Investigating Brute-Force Attack Patterns in IoT Network,” Journal of Electrical and Computer Engineering, 2019.

[17] S. Alsudani and M. N. Saeea, "Enhancing Thyroid Disease Diagnosis through Emperor Penguin Optimization Algorithm," Wasit Journal for Pure Sciences, vol. 2, no. 4, 2023.

[18] X. Chen and Y. Liu, “A Comprehensive Review of AI-Based Intrusion Detection Systems in IoT,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 8, pp. 22–39, 2024.

[19] A. Zohourian, S. Dadkhah, H. Molyneaux, E. C. P. Neto, and A. A. Ghorbani, “IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks,” Computers & Security, vol. 146, p. 104034, 2024, doi: 10.1016/j.cose.2024.104034. 4.

[20] Saif Wali Ali Alsudani , Mohammad-Reza Feizi-Derakhshi , Watheq Ghanim Mutasher, Hussein Ali Manji Nasrawi , Mohammed abdulmohsin Aswad , Anwer Saleh Khamees Al-Shammari , Mahdi Saleh , Marwah Nafea Saeea

, and Khalid abdulridha flayyih Al hilfi ,Salih Mohan Albkhati, “Enhancing IoT Intrusion Detection Through a Hybrid Deep Learning Model with Dragonfly-Based Feature and Ensemble Optimization”, Int. j. commun. netw. inf. secur., vol. 17, no. 5, pp. 1–15, May 2025.

[21] B. Yazdinejad et al., “Deep Learning for Network Security: Attention–CNN–LSTM Model for Accurate Intrusion Detection,” Scientific Reports, Jul. 2025.

[22] M. Al-Hawawreh, E. Sitnikova and N. Aboutorab, "Asynchronous Peer-to-Peer Federated Capability-Based Targeted Ransomware Detection Model for Industrial IoT," in IEEE Access, vol. 9, pp. 148738-148755, 2021, doi: 10.1109/ACCESS.2021.3124634.

[23] A. Nazir, J. He, N. Zhu, S. S. Qureshi, S. U. Qureshi, F. Ullah, A. Wajahat, and M. S. Pathan, “A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem,” Ain Shams Engineering Journal, vol. 15, no. 7, p. 102777, 2024, doi: 10.1016/j.asej.2024.102777.

[24] S. Hossain, S. M. Senouci, B. Brik, and A. A. Boualouache, “Privacy-Preserving Self-Supervised Learning-Based Intrusion Detection for 5G–V2X Networks,” Ad Hoc Networks, vol. 166, art. no. 103674, 2025.

[25] G. Engelen, V. Rimmer and W. Joosen, "Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study," 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2021, pp. 7-12, doi: 10.1109/SPW53761.2021.00009.

[26] G. Sripriyanka and A. Mahendran, "Securing IoMT: A Hybrid Model for DDoS Attack Detection and COVID-19 Classification," in IEEE Access, vol. 12, pp. 17328-17348, 2024, doi: 10.1109/ACCESS.2024.3354034.

[27] M. S. Ahsan, S. Islam, and S. Shatabda, “Systematic Review of Metaheuristics-Based and Machine Learning-Driven IDS in IoT,” Swarm and Evolutionary Computation, submitted Jul. 2024.

[28] H. Hakami, M. Faheem and M. Bashir Ahmad, "Machine Learning Techniques for Enhanced Intrusion Detection in IoT Security," in IEEE Access, vol. 13, pp. 31140-31158, 2025, doi: 10.1109/ACCESS.2025.3542227.

[29] S. Alsudani, H. Nasrawi, M. Shattawi, and A. Ghazikhani, "Enhancing Spam Detection: A Crow-Optimized FFNN with LSTM for Email Security," Wasit Journal of Computer and Mathematics Science, vol. 3, pp. 1–15, 2024.

[30] S. Elouardi, A. Motii, M. Jouhari, A. Nasser Hassane Amadou and M. Hedabou, "A Survey on Hybrid-CNN and LLMs for Intrusion Detection Systems: Recent IoT Datasets," in IEEE Access, vol. 12, pp. 180009-180033, 2024, doi: 10.1109/ACCESS.2024.3506604.

[31] M. Alotaibi et al., "Integrating Two-Tier Optimization Algorithm With Convolutional Bi-LSTM Model for Robust Anomaly Detection in Autonomous Vehicles," in IEEE Access, vol. 13, pp. 6820-6833, 2025, doi: 10.1109/ACCESS.2024.3523539.

[32] S. Racherla, P. Sripathi, N. Faruqui, M. Alamgir Kabir, M. Whaiduzzaman and S. Aziz Shah, "Deep-IDS: A Real-Time Intrusion Detector for IoT Nodes Using Deep Learning," in IEEE Access, vol. 12, pp. 63584-63597, 2024, doi: 10.1109/ACCESS.2024.3396461.

[33] C. Zhang, J. Li, N. Wang, and D. Zhang, “Research on intrusion detection method based on transformer and CNN-BiLSTM in Internet of Things,” Sensors, vol. 25, no. 9, p. 2725, 2025, doi: 10.3390/s25092725.

[34] S. S. Kareem, R. R. Mostafa, F. A. Hashim, and H. M. El-Bakry, “An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection,” Sensors, vol. 22, no. 4, p. 1396, 2022, doi: 10.3390/s22041396.

[35] X. A. Larriva-Novo, M. Vega-Barbas, V. A. Villagrá and M. Sanz Rodrigo, "Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies," in IEEE Access, vol. 8, pp. 9005-9014, 2020, doi: 10.1109/ACCESS.2019.2963407.

[36] V. R. Balasaraswathi, M. Sugumaran, and Y. Hamid, “Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms,” Journal of Communications and Information Networks, vol. 2, pp. 107–119, 2017, doi: 10.1007/s41650-017-0033-7.

[37] E. Gyamfi and A. Jurcut, “Intrusion detection in Internet of Things systems: A review on design approaches leveraging multi-access edge computing, machine learning, and datasets,” Sensors, vol. 22, no. 10, p. 3744, 2022, doi: 10.3390/s22103744.

[38] Z. Alwaisi, “Memory-efficient and robust detection of Mirai botnet for future 6G-enabled IoT networks,” Internet of Things, vol. 32, p. 101621, 2025, doi: 10.1016/j.iot.2025.101621.

[39] F. Bourebaa and M. Benmohammed, "Evaluating Lightweight Transformers With Local Explainability for Android Malware Detection," in IEEE Access, vol. 13, pp. 101005-101026, 2025, doi: 10.1109/ACCESS.2025.3577775.

[40] Y. Kim and B. Kang, "Towards Practical Edge Intelligence for Cybersecurity in IoT: Challenges and Deployments," ACM Transactions on Cyber-Physical Systems, vol. 7, no. 4, 2023.

[41] N. Albanbay, Y. Tursynbek, K. Graffi, R. Uskenbayeva, Z. Kalpeyeva, Z. Abilkaiyr, and Y. Ayapov, "Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study," J. Sensor Actuator Netw., vol. 14, no. 4, p. 78, 2025, doi: 10.3390/jsan14040078.

[42] I. A. Fares et al., "Deep Transfer Learning Based on Hybrid Swin Transformers With LSTM for Intrusion Detection Systems in IoT Environment," in IEEE Open Journal of the Communications Society, vol. 6, pp. 4342-4365, 2025, doi: 10.1109/OJCOMS.2025.3569301.

[43] D. Torre, A. Chennamaneni, J. Jo, G. Vyas, and B. Sabrsula, "Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study," ACM Trans. Inf. Syst., vol. 34, no. 2, Feb. 2025, doi: 10.1145/3695998.

[44] G. Shobana, A. T. Nathan, D. S. Sivakumar, et al., “GBiL: A hybrid gated recurrent units (GRU) and bidirectional long short-term memory (BiLSTM) model with Particle Swarm Optimization for a robust VANET IDS,” Journal of Wireless Communications and Networking, vol. 2025, no. 57, 2025, doi: 10.1186/s13638-025-02467-8.

[45] S. Natha, M. Siraj, F. Ahmed, M. Altamimi and M. Syed, "An Integrated CNN-BiLSTM-Transformer Framework for Improved Anomaly Detection Using Surveillance Videos," in IEEE Access, vol. 13, pp. 95341-95357, 2025, doi: 10.1109/ACCESS.2025.3574835.

[46] M. Al Rawajbeh, A. J. Maria Soosai, L. K. Ramasamy, and F. Khan, “Trustworthy adaptive AI for real-time intrusion detection in industrial IoT security,” IoT, vol. 6, no. 3, p. 53, 2025, doi: 10.3390/iot6030053.

[47] A. Bouayad, H. Alami, M. Janati Idrissi and I. Berrada, "Lightweight Federated Learning for Efficient Network Intrusion Detection," in IEEE Access, vol. 12, pp. 172027-172045, 2024, doi: 10.1109/ACCESS.2024.3494057.