International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I7P107

Volume 74 Issue 7, 96-117, July 2026

Original Article

BBFS-LSTM-AE: An Optimized Intrusion Detection Model for Securing IoT Networks

Nancy Thomas[1], R. Gunasundari[2]

[1,2]*Department of Computer Science, Karpagam Academy of Higher Education (Deemed to be University), Coimbatore, India.*

[1]*Corresponding Author : nancyjismon@gmail.com*

Received: 29 August 2025 Revised: 30 May 2026 Accepted: 18 June 2026 Published: 28 July 2026

Abstract - *Rapid growth in Internet of Things (IoT) devices has expanded the attack surface of modern-day networks. Therefore, it is imperative to deploy an IoT Intrusion Detection System (IDS) for secure communication and reliable operation. However, traditional IDS systems are often inefficient when dealing with high-dimensional data, new attack strategies, and severe imbalanced data problems, leading to low detection performance and high false alarms. In order to solve these problems, this paper introduces a novel intelligent IDS system using the Feature Selection technique inspired by Bowerbird Courtship (BBFS) and Long Short-Term Memory Autoencoder (LSTM-AE) using Seagull Optimizer (SGO). The most important objective here is to build an effective and scalable IDS that can perform efficient feature selection, learn deep temporal dependencies, and tune its hyperparameters to classify malicious and benign traffic more effectively. In this way, we consider all aspects of data preprocessing, feature optimization, balancing, and classification, overcoming the limitations of existing techniques. The evaluation of the CIC IoT 2023 intrusion dataset proves the efficiency of the model, which is shown by high scores: 99.63% of accuracy, 99.55% of detection rate, 99.71% of precision, and 99.59% of F1 score. As seen from the comparison with other models, the BBFS-LSTM-AE-SGO model is better than the compared model in terms of all metrics. It means that the proposed IDS can detect various types of attacks with minimum errors. All through this research has made way for the design of a novel, optimized IoT-IDS model that thereby strengthens cybersecurity resilience, supports real-time monitoring, and hence advances the intrusion detection for IoT-enabled environments.*

Keywords - *BBFS, CIC-IoT-2023, Deep learning, IoT, Intrusion detection, LSTM-AE, SGO.*

1. Introduction

Cyber-attack refers to a targeted attempt to breach the security of computer systems or networks. A cyber-attack can include actions like hacking, spreading malware, or phishing for the purpose of gaining information, disrupting operations, or financial gains. The current state of cyber-attacks indicates a major and increasing threat in various industries, such as health care, industrial networking, and organizations [1]. The fast growth of IoT and distributed systems has made the network environment more complex than ever before. Therefore, the IDS, which is responsible for ensuring the security of the network environment, is currently facing unprecedented challenges. This is because conventional IDSs have been found to be unable to adapt to different network environments because of feature extraction challenges and data imbalance.

IoT consists of sensors, machines, and cameras that continuously add to the number of devices that are connected to the Internet. According to the International Data Corporation, by 2025, there will be 41.6 billion IoT devices, which will produce 79.4 zettabytes of data [3]. IDS is vital for securing the IoT [4]. IDS stands out as an important tool for cybersecurity and is used for surveillance purposes on systems or networks in order to detect misuse, any unauthorized access, or violations of policies.

The primary role of an IDS is to alert the administrator of the systems or security professionals about any potential threat so as to ensure data integrity and availability. With increased complexity in the networks, the importance of IDS becomes even more pronounced.

The block diagram of an IDS data flow pipeline has been presented in Figure 1, where all the steps in sequence have been included to show how one converts network traffic into a detection measure.

Initially, network traffic is taken as input and passed through feature selection/feature extraction so that only the features that improve the process of learning are considered, and the rest are disregarded. Data balancing or handling is performed because intrusion detection datasets face the issue of data imbalance.

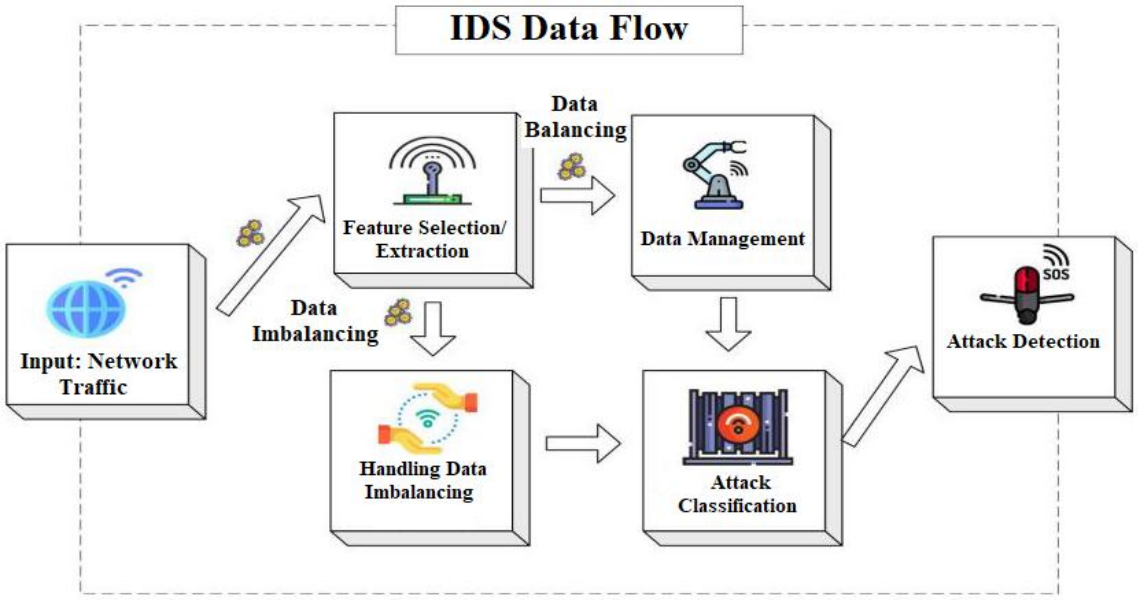

Fig. 1 Workflow of the IDS model

The data management module is responsible for organizing and preparing the processed traffic for analysis such that the input is consistent and accurate. The processed traffic data is then classified using either Machine Learning (ML) or Deep Learning (DL) techniques, depending on whether the traffic is malicious or not. Finally, after the process, there is an attack detection system that issues alerts to any suspicious activities for security monitoring purposes. In general, this workflow clearly indicates a systematic process of designing an IDS, starting from preprocessing to balancing, classification, and finally detection of attacks.

IDS is responsible for monitoring all the activities on the network and detecting any unusual or malicious activity that deviates from the norm. IDS can be categorized into two main categories, namely signature-based approach and anomaly-based approach [6].

The signature-based approach utilizes the rules or signatures to detect any attacks, while the anomaly-based approach uses machine learning or statistics to detect the features of legitimate and malicious information. Signature-based methods provide high accuracy, fewer false positives, and quick execution time when an attack is known; however, they cannot detect any new attacks and require updates on the signature database frequently. Anomaly-based approaches value their ability to detect new attacks; however, they have higher false positives and high computational complexity [7].

1.1. Problem Statement

The proliferation of IoT networks has led to increased vulnerability to cyber threats like DDoS, spoofing, brute force, and reconnaissance attacks. Existing intrusion detection mechanisms have been unable to detect these attacks due to factors like high-dimensional data, class imbalance, evolving nature of attacks, and temporal dependencies in the traffic of IoT networks [8]. Conventional Machine Learning (ML) methods generally require manual feature engineering and are characterized by poor flexibility, whereas many Deep Learning (DL) techniques have issues with high complexity and lack of interpretability. Moreover, some current techniques fail to incorporate proper feature selection and hyperparameter optimization, resulting in low detection accuracy and higher levels of false positives. Hence, there is a need for a robust IDS system that can efficiently detect attacks on the IoT [9].

Recent advancements in explainable AI were made to tackle this issue; however, the balance between interpretability and detection remains a critical factor in IDS [10]. Several studies have utilized multiple population-based metaheuristic optimization approaches to enhance the efficacy of IoT intrusion detection systems. In the majority of instances, this hybridization was employed to identify an optimal combination of features [11].

Nonetheless, IoT-based IDSs continue to face significant security challenges. Conventional static DL and ML methodologies are constrained in identifying low-frequency incursions, operate ineffectively with data sets that are imbalanced, and fail to detect previously undiscovered or zero-day threats. Nonetheless, using these strategies in low-power IoT environments necessitates further optimization calculations, leading to issues such as computational scalability and complexity. These constraints elucidate the necessity for improved IDS procedures, advanced IDS methodologies, and more suitable IoT network IDS frameworks [12].

1.2. Research Novelty and Objectives

The paper presents a new BBFS-LSTM-AE-SGO-based IDS framework that combines bio-inspired feature selection, deep sequential learning, and meta-heuristic optimization algorithms within a single framework to protect IoT systems from cyberattacks. It is noteworthy that the new model makes use of a novel BBFS algorithm. In this regard, LSTM-AE is used to extract non-linear and temporal attacks, while SGO tunes the hyperparameters for better convergence and classification performance. In contrast to the traditional IDSs that can only deal with either one or two issues at a time, the proposed scheme takes care of all the mentioned problems at once, thus outperforming other techniques.The objectives of this study are listed below to develop an optimized and intelligent IDS so that it can improve feature selection, manage data imbalance well, and better attack classification accuracy in the BBFS-LSTM-AE-SGO framework.

- To build an effective intrusion detection system that detects and classifies different types of cyberattacks in an IoT environment with high accuracy, using the CIC IoT 2023 dataset.
- To preprocess the dataset, SMOTE will be used to handle the data imbalance, while Min-Max normalization will be used so that the obtained classification would be highly reliable for both common and rare types of attacks.
- To implement BBFS for smart feature selection to remove redundant features and enhance the efficiency of the model.
- To apply LSTM-AE for representation learning, anomaly detection, and classification of normal and malicious traffic.
- To employ SGO to optimize LSTM-AE hyperparameters for detection accuracy and minimal false positives.
- To test the proposed BBFS-LSTM-AE-SGO model and then evaluate the performance based on the metrics such as Accuracy, Detection Rate, Precision, F1-score, and False Positive Rate (FAR).
- To compare the developed research model with the existing IDS framework in demonstrating the scalability, robustness, and adaptability to real-world IoT systems.

The main contributions of this study are highlighted as follows:

1) A hybrid IDS model called BBFS-LSTM-AE-SGO has been proposed to provide security in the IoT environment against various types of attacks.

2) A BBFS algorithm based on nature has been designed to remove irrelevant traffic features from the dataset, thus minimizing complexity.

3) An LSTM-AE model has been used to learn time-series patterns and nonlinearity in attacks.

4) The SGO method is used for hyperparameter optimization of the model to accelerate the process of convergence, improve generalization capabilities, and increase classification performance.

5) While current hybrid IDS models concentrate only on improving classification accuracy, the presented approach takes care of feature optimization, handling class imbalance, temporal learning, and false positive minimization at once.

6) The research model shows an outstanding result in terms of classification accuracy (99.63%) when using the CIC-IoT-2023 dataset, being better than several recently developed IDS models.

This paper will be structured as follows. In Section II, the current models related to the research work IoT-IDS will be discussed. In Section III, the implementation of the research methodology that has been proposed will be described. In Section IV, the experiment results of the research methodology and their comparison to current models will be presented. At last, the paper concludes with future works.

2. Literature Review

This part provides an analysis of recent research models aimed at enhancing IoT-based intrusion detection systems. All examined analyzed methodologies are extensively evaluated and displayed in Table 1. A detailed review of ML-based approaches was conducted in [13] for IoT anomaly detection.

ML techniques considered in the study included Random Forest (RF), Isolation Forest (IF), and AE methods, along with other popular techniques such as neural network (Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM)), Gradient Boosting (GB), extreme GB (XGBoost), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees (DT). Test results obtained based on benchmarking (CICIoT-2023 and IoT-23) have been analyzed and reviewed. Test results have shown that RF and AE approaches can detect anomalies efficiently. The results indicate that data preprocessing was essential for accurate anomaly detection.

The research in [14] presented a neural network-based model for IDS that effectively detected and classified the different types of attacks on IoT devices used in intelligent applications. This technique adopted a feature reduction method and hyperparameter optimization to reduce time and overhead. Using essential features extracted through GA for feature selection, color images were created as inputs for various CNN frameworks, such as VGG16, Xception, and VGG19. However, the ensembled model developed by integrating and combining VGG16, Xception, and VGG19 classifiers, along with a GA-based best feature selector,

reached an impressive 98.7% accuracy, standing 5% higher than independent classifiers. Thus, it largely reduced false positives and the time consumed for calculations.

The LSTM-based model was proposed in [15] for the detection of cyber-attacks and threats in IoT, giving rise to the CIC-IoT2023 model. LSTM application in IoT security has been widely considered. Results suggested that the LSTM model exhibits superior performance, given an appropriate architecture. The F1 score, recall, and precision of LSTM were calculated to be 0.9859, 0.9875, and 0.9866, respectively, resulting in an outstanding accuracy of 98.75%.

This proved the importance and effectiveness of LSTM, and hence, it became essential for classification in cases where IoT security was critical. Furthermore, it presented a complete analysis of CIC-IoT2023 datasets, which were used for conducting the research and also proved to be beneficial in investigating other problems in the domain.

In [16], a DL-based classification system for intrusion detection was designed for IoT settings with three separate architectures: Transformer, CNNs, and DNNs-based models. 12-class, 3-class, and binary classification performances were evaluated using the CICIoT2023 data set. The preprocessing of data included log normalizations to stabilize feature distribution and oversampling according to SMOTE to treat class imbalances.

The evaluations carried out on the dataset showed that in the binary classification results, the DNN achieved the highest accuracy of 99.2%, the CNN next with 99.0%, while the Transformer came at 98.8%. Although in the 3-class classification, all models performed better, ranging from about 99.9% to 100%. In the 12-class scenario, the Transformer, CNN, and DNN attained accuracies of 92.7%, 93.0%, and 92.5%, respectively.

A method for data pre-processing in combination with DL models to develop a lightweight and portable IDS was developed in [17]. The work introduced an innovative data preprocessing approach as a fundamental framework for creating a lightweight sensory IoT-IDS. A unique data pre-processing method employing tree-based feature importance to identify optimal features through a feature importance assessment. Consequently, to eliminate noise from the data, the features of minimal importance were taken out. Experimental suggestions were made to use hybrid models in determining the efficacy of various DL classifiers.

For multiclass classification, an Extra Tree Classifier with LSTM (ETLSTM) and an Extra Tree Classifier with 1D-CNN (ETCNN) have been used. The work evaluated the effectiveness of 1D-CNN and LSTM on the CIC IoT 23 dataset and found that the hybrid technique yielded better success.

A framework in [18] proposed integrating incremental learning, blockchain, and hybrid encryption techniques in three phases to improve the safety and efficiency of IoT-based IDS. The first phase develops a model to detect cyber-attacks with Incremental Learning (SGD Classifier). The second phase establishes a blockchain framework to secure the encrypted data. In the third phase, the incremental model is retrained with new data obtained from the blockchain through multiple iterations to achieve the best accuracy. The incremental learning model was first trained on the CICIoT2023 data set. Upon retraining the model, it was evaluated with previously unseen data in the two scenarios of CICIDS2019 and TON IoT2020. The framework proved capable of protecting the IDS data in IoT environments and was further improved by retraining on the newer traffic data.

A hybrid methodology combining DL and ML was developed in [19] to enhance the detection of DDoS and spoofing attacks, minimize false alarms, and facilitate the implementation of requisite defensive measures. The initial phase involved a hybrid approach for feature selection that incorporated correlation coefficient and sequential feature selector techniques; the subsequent phase entailed the development of a hybrid model that merges DL neural networks with an ML classifier, specifically a cascaded LSTM network and a Naive Bayes classifier; finally, the third phase focused on enhancing network defense mechanisms by blocking ports post-threat detection and preserving network integrity. Three datasets (CIC-IoV2024, CIC-IoT2023, and CIC-DDoS2019) were utilized in the training and evaluation of the methodology, with the data also being balanced to provide optimal results.

An enhanced intrusion detection for security in IoT that utilized multimodal representation of big data and Transfer Learning (TL) was developed in [20]. The files of packet capture were analyzed to extract the requisite data bytes and attacks. Secondly, big data based on Spark optimization approaches manages substantial data volumes. A TL methodology like Word2Vec extracted semantically grounded observable features. Then, a technique was devised to transform network data bytes into images, and features of texture were recovered by employing the attention-based Residual Networks. The texture and text features trained were integrated and utilized as multimodal features for the classification of diverse attacks. The methodology was assessed using Edge-IIoT, CIC-IoT 2023, and CIC-IoT 2022 datasets. The methodology attained exceptional classification performances, with a 98.2% accuracy.

The research in [21] developed two improved Intrusion Detection Frameworks (IDFs). SelectKBest-MI integrated two filter-based feature selection methodologies, specifically SelectKBest and Mutual Information (MI), to identify optimal features, while employing RF as the classifier. The second IDF was designated as CNN-SVM-GWO. CNN was

employed for feature extraction, SVM and Grey Wolf Optimizers (GWO) were utilized for optimal feature selection, while RF and XGBoost classifiers were applied for intrusion detection. Two datasets, CIC-IoT-2023 and CICIDS2017, were utilized. There was an improvement in accuracy when using the SelectKBest-MI method with the CICIDS2017 dataset.

An integration of AE for feature extraction, LSTM for temporal dynamics, and CNN was implemented as a hybrid model in [22]. The Synthetic Minority Over-Sampling Technique (SMOTE) was used to solve the problem of data imbalance. This structured approach takes advantage of all the models in order to process and categorize security data from IoT. This study emphasized the potential benefits of applying DL techniques to boost the performance of IDS in IoT security. The results suggested a possible enhancement of security measures and the mitigation of upcoming threats.

A model for IDS combining ensemble learning techniques with rule induction to enhance model explainability is proposed in [23]. The study analyzed the efficacy of five algorithms (RF, XGBoost, AdaBoost, CatBoost, and LightGBM) in building efficient IDSs for the IoT. Results indicated that XGBoost was superior to other ensemble methods on two public datasets in intrusion detection. The model was made explainable through rule inductions via a DT used as the basic weak learner in the ensemble algorithm compared in the study. Experimental results showed the approach to be effective in delivering a transparent, lightweight, and reliable IDS system.

The research in [24] proposed an IDS model for big data environments by hybridizing DL techniques. The method was trained and tested on the TON_IoT and CICIoT2023 datasets. Feature reduction using a correlation technique was utilized to ensure the selection of all relevant features for analysis. A hybrid DL technique was designed utilizing LSTM and 1D-CNN. DL methods like LSTM and CNN were evaluated independently. The hybrid method had a better performance against their individual implementations. It was, thus, established that the hybrid technique was more accurate.

The investigation carried out in [25] used the CICIoT2023 data set for a thorough analysis and evaluation of intrusion detection on IoT networks. DL techniques were employed to improve the accuracy and reliability of the detection of anomalous activities and intrusions. The study then implemented Transformer for classification in a multi-class setting, further comparing against existing models. The experiment findings indicated that in the classification of binary class, CNN + LSTM and DNN attained the maximum results; however, in the classification of multi-class, the Transformer method attained the best accuracy. The results demonstrated the prospective utility of DL techniques in IoT-IDS.

In [26], the application of ML predictive models for the detection and mitigation of cyberattacks on IoT networks was studied. The principal focus was on the analysis of the CIC-IoT2023 dataset, an extensive collection that includes several Distributed Denial of Service (DDoS) attacks on multiple IoT devices, to ensure the employment of a competent and thorough benchmarking standard. The work built, trained, and tested four standalone ML models, K-Nearest Neighbors (KNN), Logistic Regression (LR), RF, and DT, to check their effectiveness in detecting and mitigating cyber threats to the IoT. The results suggest the great performance of DT and RF with a successful accuracy of detection.

In [27], an IDS model based on DL methods, Multilayer Perceptron (MLP), and AE on the dataset CICIoT2023. The model implements mechanisms of size reduction and efficiency enhancement to tackle the problem of high dimensionality present in intrusion detection datasets. A novel clustering strategy for intrusion detection datasets, derived from a combination of a Gaining-Sharing Knowledge (GSK) optimization algorithm and static tools, has been developed. The model was assessed for its efficiency through various evaluation metrics. The findings indicated that, relative to prior studies utilizing the same datasets, the DL model exhibited superior performance in attack detection.

The study in [28] developed an approach to IDS through the hybridization of unsupervised learning-based intrusion detection, a non-parametric Siamese cross-dataset dissimilarity filtering technique, and Proximal Policy Optimizer (PPO)-based adaptive defense. Unsupervised models detect anomalies, Siamese correlation helps in detecting rare zero-days, and the PPO agent learns effective defensive strategies using environmental feedback. The model provided improved results in the CIC-IoT-2023 data set with 99.07% accuracy. This methodology offered an adaptable, proactive, and self-learning protection framework for automated IoT cybersecurity.

A lightweight and high-accuracy IDS framework based on feature selection and DNN architecture tuning was proposed in [29]. The Robust Scaler was employed for statistical preprocessing, which helped to avoid using the Z-score normalization method, and thus minimized the negative effects. A self-normalizing residual block with channel attention was applied. By combining compressed hidden layers and the Scaled Exponential Linear Unit (SELU) activation function, the network design addressed the issue of vanishing gradients and improved the visibility of important traffic characteristics. Moreover, significantly, the number of parameters and computation time were reduced for the inference process. A learning rate adjustment technique was applied using the cosine annealing approach. The findings on the CIC-IoT-2023 dataset showed that the model was highly accurate and computationally efficient.

Edge AI Bridge, a micro-computing security layer that operated between the IoT device and the gateway in order to provide early detection of any threats, was developed in [30]. This concept involved embedding AI hardware into a hybrid pipeline that utilized anomaly detection and signature matching. Operations in the system included traffic inspection, protocol analysis, and feature extraction at the edge before data aggregation, ensuring privacy of devices and reducing the computational load on the IoT gateway.

The CIC-IoT-2023 dataset was utilized for evaluation purposes. The Edge AI Bridge presented an approach that could provide a scalable, modular, and privacy-preserving solution to increase cyber resiliency in the IoT. The study by [31] focused on evaluating a federated IDS model in IoT networks using datasets and comparing the efficiency of FL algorithms like FedNova, FedProx, and FedAvg in combination with LSTM and Transformer neural network architectures. It was established that FL was capable of achieving excellent performance results comparable to those of central learning schemes and provided extra benefits in cases where the data was heterogeneous or not identically distributed. Federated training across multiple domains enhanced the generalizability of the model to different scenarios, while FedNova decreased the number of communication rounds. The research work by [32] proposed and analyzed a Hybrid Improved Binary Grey Wolf Optimizer (IBGWO)-Particle Swarm Optimizer (PSO) with an RF model for identifying intrusion in a larger IoT network. In addition, the designed IDS model was analyzed using the latest CIC-ToT-2023 dataset. The major contribution of the model relied on the use of IBGWO and PSO for selecting an optimal subset of features from the dataset. The two algorithms were combined in order to integrate the strength of exploitation and exploration in order to ensure convergence. The feature selection approach ensured optimized convergence with a lower fitness value.

Table 1. Comparative review of the analyzed research models

Refs	Methodologies	Applications	Advantages	Limitations
[13]	Comparative study of ML (RF, IF, AE, CNN/RNN/LSTM, GB/XGBoost, KNN, SVM, DT)	Anomaly detection on CIC-IoT-2023 and IoT-23	RF and AE are highly effective; broad algorithmic coverage; highlights the critical role of preprocessing for accuracy.	Lacks a single unified model; results are dataset-specific; limited discussion of deployment cost/latency.
[14]	GA-based feature selection; features converted to images; CNN ensemble (VGG16, Xception, VGG19) with GA for fusion	Multi-attack IDS for intelligent/IoT apps	98.7% accuracy; ~5% over single CNNs; lower false positives; reduced processing time via feature reduction.	Image conversion pipeline adds complexity; heavy CNNs (VGG/ Xception) can be computationally expensive; explainability is limited.
[15]	LSTM-based IDS; careful architecture tuning; dataset overview (CIC-IoT2023)	IoT attack detection with sequence modelling	Strong metrics (Acc 98.75%, F1 0.9859, Rec 0.9875, Prec 0.9866); robust to temporal patterns.	Performance sensitive to architecture; potential training cost; limited interpretability.
[16]	DNN, CNN, Transformer; log normalization; SMOTE; binary / 3-class / 12-class tasks on CIC-IoT-2023	General IDS with varying label granularity	High accuracy (binary: 98.8–99.2%; 3-class: ~99.9–100%); solid 12-class performance (92.5–93.0%).	Multi-class still lower; Transformers are heavier; oversampling risks overfitting minority patterns.
[17]	Tree-based feature importance for pruning; hybrid ET+LSTM (ETLSTM) and ET+1D-CNN (ETCNN)	Lightweight sensory IoT-IDS; evaluated on CIC IoT 23	Reduces noisy features; hybrid models outperform single DL models; targets portability.	"Extra Trees + DL" pipeline adds steps; feature importances miss interactions; potential concept drift issues.
[18]	Three-stage framework: Incremental Learning (SGD), Blockchain for secure storage, Hybrid encryption, retraining on new data	Cross-dataset learning: train on CIC IoT 2023; test on CICIDS2019 & TON IoT 2020	Security hardening via blockchain; handles evolving traffic (incremental learning); improved performance after retraining.	Blockchain introduces latency/storage overhead; the SGD baseline is weaker than state-of-the-art; integration complexity.

[19]	Hybrid FS (correlation + sequential selector); Cascaded LSTM + Naive Bayes; post-detection defense (port blocking)	DDoS/spoofing detection on CIC-IoV2024, CIC-IoT2023, CIC-DDoS2019 (balanced data)	Reduces false alarms; actionable defense step; leverages temporal (LSTM) + probabilistic (NB) strengths.	Port-blocking can disrupt benign traffic if misclassified; pipeline complexity, generalization across devices must be verified.
[20]	Multimodal big-data IDS: Spark optimization; word2vec-like TL for text features; bytes→image transform; attention-ResNet for textures; feature fusion	Large-scale IoT/IIoT (Edge-IIoT, CIC-IoT 2023/2022); PCAP-level analysis	High accuracy (98.2%); scalable via Spark; exploits both textual and texture cues.	Requires heavy compute (Spark + ResNet + fusion); bytes→image step adds preprocessing burden; interpretability challenges.
[21]	Two IDFs: (a) SelectKBest+MI + RF; (b) CNN feature extraction + SVM + GWO for selection; RF/XGBoost for detection	CIC-IoT-2023 and CICIDS2017; feature selection + ensemble classification	Demonstrates FS benefits; RF/XGBoost strong baselines; improved accuracy (noted on CICIDS2017).	Mixed datasets complicate comparability; GWO adds tuning overhead; model overfits if FS is not cross-validated carefully.
[22]	Hybrid DL: AE + LSTM + CNN; SMOTE for imbalance	General IoT IDS classification with integrated representations	Leverages complementary strengths; improved learning on imbalanced data; structured architecture.	Larger model footprint; SMOTE risks synthetic noise; harder to explain and tune.
[23]	Ensemble learners (RF, XGBoost, AdaBoost, CatBoost, LightGBM); rule induction via decision trees for explainability	Lightweight, transparent IDS on public datasets	XGBoost is best among ensembles; adds explainable rules; efficient and reliable.	Rule induction may oversimplify complex patterns; potential drop vs deep models on subtle temporal attacks.
[24]	Correlation-based feature reduction; Hybrid LSTM + 1D-CNN; individual vs hybrid comparison	Big-data IDS on TON_IoT and CICIoT2023	Hybrid outperforms individual LSTM/CNN; retains salient features; strong accuracy.	Correlation FS discards interacting features; training hybrid networks increases complexity and computation.
[25]	DL comparison incl. Transformer; binary vs multi-class; CNN+LSTM/DNN best for binary, Transformer best for multi-class	CIC-IoT-2023 comprehensive study	Identifies task-dependent winners; validates the Transformer for multi-class; provides practical guidance for model choice.	Transformers are heavier; results are specific to label design; requires careful HP tuning.
[26]	Classical ML comparison: KNN, LR, RF, DT on CIC-IoT2023	Baseline ML for IoT cyberattack detection/mitigation	RF/DT high accuracy; simpler, faster, easier to deploy; good baselines.	Limited temporal modeling; underperforms on complex/novel attacks; feature engineering needed.
[27]	DL IDS with MLP and AE; dimensionality reduction; novel clustering using GSK optimization + static tools	CICIoT2023 handles high dimensionality.	Better attack detection than prior work on the same data; efficiency via reduction/clustering.	Adding clustering/optimization stages raises complexity, potential sensitivity to clustering settings; interpretability is modest.
[28]	Unsupervised IDS + Siamese filtering + PPO	IoT intrusion detection	Detects zero-day attacks; adaptive and self-learning.	High computational complexity.
[29]	Lightweight DNN with feature selection and SELU	Lightweight IoT IDS	High accuracy; reduced computation time.	Complex hyperparameter tuning.
[30]	Edge AI Bridge with anomaly and signature detection	Edge-based IoT security	Privacy-preserving and scalable.	Requires specialized edge hardware.

[31]	Federated Learning with LSTM and Transformer	Distributed IoT IDS	Preserves privacy; improves generalization.	Communication and synchronization overhead.
[32]	IBGWO-PSO with RF	Large-scale IoT intrusion detection	Optimal feature selection and convergence.	High optimization computational cost.

2.1. Research Gap Analysis

Existing IoT-IDS models have achieved considerable advancements by using ML, DL, ensemble learning, and optimization techniques. Despite that, there are still some limitations left to be addressed. The majority of the existing ML algorithms, like RF, DT, and XGBoost, highly rely on handcrafted features and are unable to detect novel and zero-day attacks.

DL approaches such as CNN, RNN, Transformer, and LSTM-based methods can enhance detection performance; however, they usually encounter challenges such as computational complexity, overfitting, and inadequate feature optimization. Although hybrid techniques have tried to integrate different approaches, they tend to add further architectural complexity without sufficiently tackling the problem of redundancy in the high-dimensional IoT traffic dataset.

Moreover, some research studies emphasize accuracy but ignore other important aspects, such as reducing the number of false positives, scalability, and optimization of hyperparameters. In addition, current research lacks a unified approach that could handle feature selection, temporal dependency learning, data imbalance, and optimization at the same time. As a result, an effective and intelligent IDS system with efficient learning capacity is required.

3. Materials and Methods

This research proposes a novel DL-based intrusion detection model for classifying attacks in an IoT environment. An outline for IoT intrusion detection based on the general workflow is shown in Figure 2, a model to maintain the pipeline with guaranteed accurate and efficient detection of malicious activities within IoT networks.

The entire process starts with data preprocessing, in which raw IoT traffic is balanced using SMOTE against class imbalance and normalized through Min-Max scaling to set all features in a uniform range suitable for DL models. The dataset after processing through the preprocessing phase is divided into two parts, namely the training dataset and the testing dataset, which are used for evaluating the model's performance.

The discriminative features are chosen using BBFS, a technique that simulates the selectivity and optimality of the Bowerbird to eliminate any redundancies, thereby reducing the computational burden and increasing the classification accuracy. These features are then passed on to the LSTM-AE, where the encoder compresses the input data, and the decoder decodes the outputs, thereby detecting the anomalies in traffic flow. SGO is used for enhancing learning performance via tuning of the models' hyperparameters in order to achieve faster convergence, avoid overfitting, and obtain high generalization capability. After that, the classified data will be evaluated using some performance metrics like accuracy, precision, recall, F1-score, and FAR, which will prove the stability of the LSTM-AE model in detecting the attacks in the IoT network.

3.1. Dataset Details

For the research study, we use the CIC-IoT-2023 dataset, which is open access and created specifically to perform security analysis on IoT. The dataset contains network traffic collected from the IoT devices and contains attacks like DoS, DDoS, Brute Force, Mirai Reconnaissance, Spoofing, and Web attacks. Table 2 indicates that this dataset contains 46,686,579 attack samples, and they can be classified into 33 different types of attack. This data set stands out from others for having a great level of comprehensiveness since there are various types of attacks divided into seven categories. Each of these attacks takes place among IoT nodes [28].

Table 2. Data distribution of the CIC-IoT-23 dataset

No. of Records	Attack Types	Distributions %
33984560	DDoS	72.79
486504	Spoofing	1.04
1098195	Benign/Normal	2.35
8090738	DoS	17.33
24829	WebBased	0.05
354565	Reconnaissance	0.76
2634124	Mirai	5.64
13064	Bruteforce	0.03

CICIoT-2023 provides a comprehensive dataset, comprising raw network data in PCAP formats and a collection of 47 features extracted in CSV formats. The features are derived from the examination of the network packets and encompass data on headers, flow durations, packet lengths, protocols, TCP flags, and other timing measures.

This comprehensive data facilitates a thorough analysis of network activity and the detection of trends that might signify potential threats. The CICIoT-2023 dataset is applicable for training and assessing diverse AI models, particularly those focused on anomaly detection. These algorithms can detect anomalous trends in network activity that could signify the existence of unidentified attacks previously.

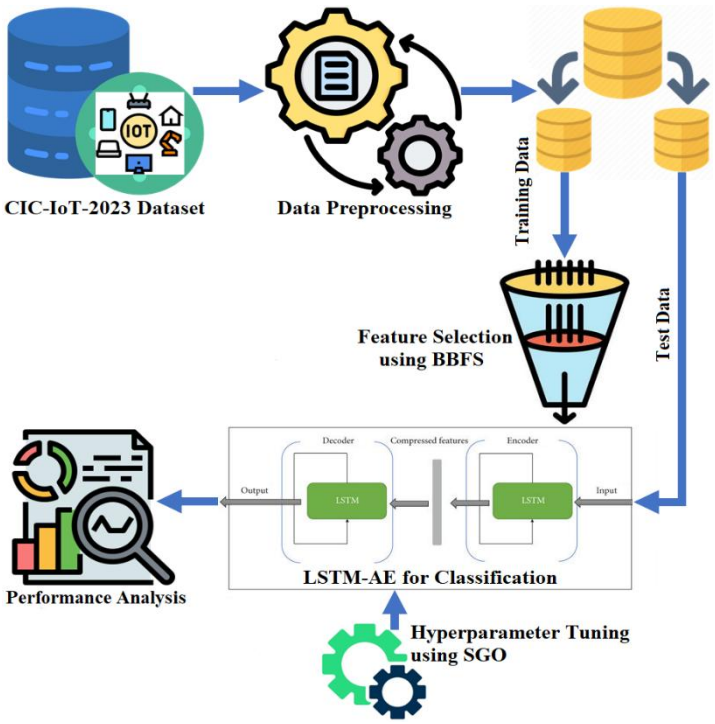

Fig. 2 Workflow of the proposed BBFS-LSTM-AE model

3.2. Data Preprocessing

Utilizing a dataset in DL algorithms without preprocessing is inappropriate. Data gathered from real-world settings frequently exhibits several inaccuracies and inconsistencies, necessitating data cleaning. Preprocessing seeks to provide the DL algorithm with more refined data, hence improving the model's efficiency. The preprocessing phase encompasses several methods to clean, balance, and normalize the data set, hence assuring optimal performance for DL algorithms. SMOTE is applied to the data set to mitigate the class imbalance issue. SMOTE functions by generating synthetic samples within minority attack classes, so rectifying imbalance in the distribution of attack classes. Despite being an efficient technique that deals with class imbalance by creating extra samples for the minority classes, there are certain weaknesses associated with SMOTE. A possible disadvantage is the insertion of unrealistic synthetic samples or noise that might fail to adequately reflect distributions of real-time data. This issue was especially pertinent in IoT systems, where patterns of attack frequently demonstrate intricate spatial and temporal correlations that SMOTE might not adequately address. In order to overcome major class imbalance, the use of SMOTE was employed to develop synthetic samples for the minority classes, resulting in a balanced dataset using Equation (1).

$$x_{new} = x_i + \alpha(x_j - x_i) \tag{1}$$

Here, the minority class samples are denoted by x_i, its nearest neighbor is denoted by x_j, and a random value within the interval [0, 1] is denoted by α. This approach was used only in the training dataset. Each time, the test set stayed untouched, ensuring that there was no effect from synthetic data on testing and that data leak was effectively prevented [29].

The next step after oversampling with SMOTE is normalization of features using Min-Max normalization. It is important because some features have significantly larger values than others. Equation (2) shows how to compute values after normalization using the min-max approach. Min-max normalization generates new values for all variables within the closed interval of [0, 1]. This is important because it ensures that variables with smaller values do not overpower other variables that have larger values in their particular variables.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

Here, x denotes a particular feature value before scaling, $max(x)$ and $min(x)$ represent the maximum and minimum attained values for the feature set, and x' signifies the newly standardized values for the features. For fast learning, a random sample of 10% from the data set was taken, ensuring that balanced classes were kept intact during this process. Any duplicity in the data is identified and removed in order to prevent the possibility of overfitting the model. Logarithm transformations were used on features with a diverse range of values, as shown in Equation (3).

$$x' = \log(1 + x) \tag{3}$$

Preprocessing made sure that the dataset was ready to be used for training deep learning models. The representation of the features was improved, and class distributions were balanced [30].

In this study, a 10% data subset is used for testing purposes. It is divided into an 80:20 ratio to train and evaluate the developed model.

3.3. BBFS-based Feature Selection

BBFS was chosen due to the presence of high-dimensional and redundant traffic data within the IoT intrusion dataset, which impacts the accuracy and efficiency of classification. In contrast to traditional filter-based methods that consider each feature individually, BBFS utilizes wrapper-based optimization techniques by evaluating feature subsets based on classifier performance. The biological-inspired searching procedure used in BBFS allows for an optimal balance between exploration and exploitation, thus making it possible to identify discriminant features without falling into local minima. Moreover, BBFS performs better than other conventional methods, such as PSO and GA, in terms of convergence and feature redundancy. The features are very useful in increasing learning efficiency, preventing overfitting, and improving the classification performance of the LSTM-AE model.

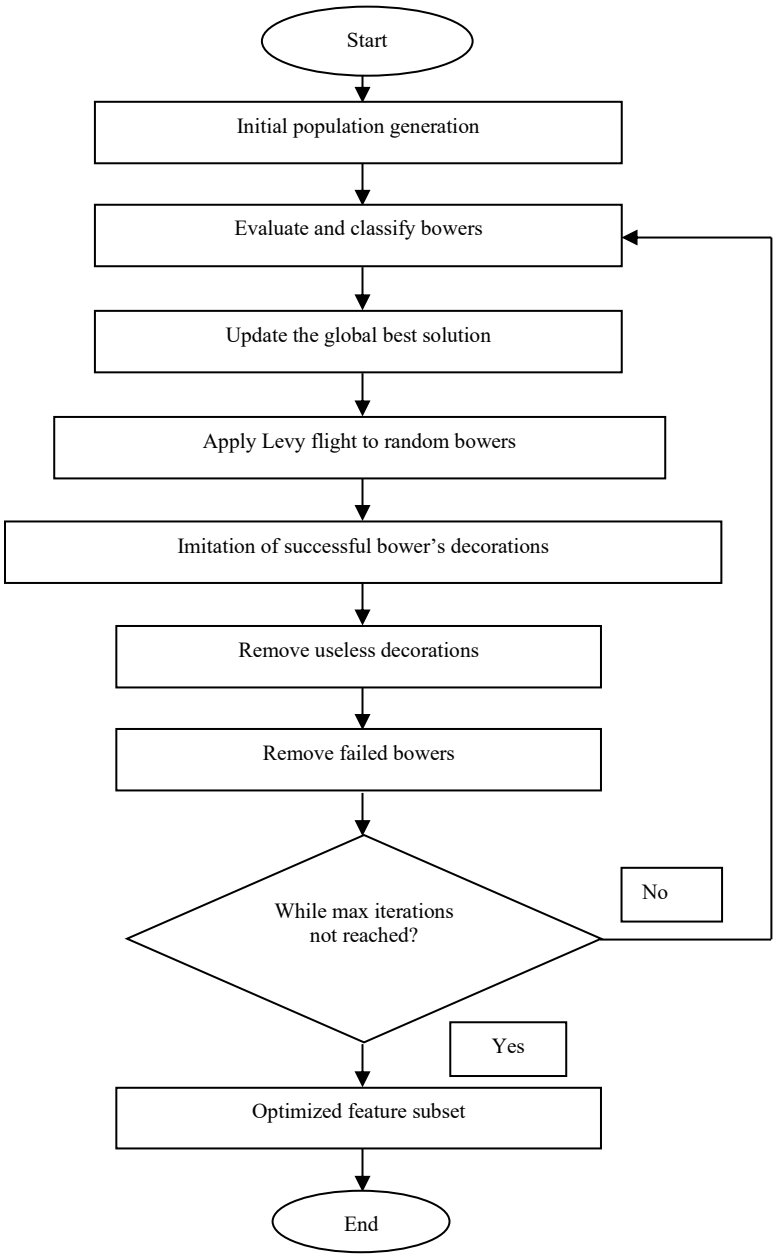

Fig. 3 Flowchart of BBFS-based feature selection

The concept behind the BBFS approach is to improve the process of feature selection based on the mating and decoration behavior observed in bower birds. This includes forming the first sets of features and evaluating their performance while improving them through biological processes. BBFS uses a bio-inspired method in which feature selection is done using the behavior of bowerbirds in their decoration process. The method uses a wrapper strategy and starts with an initial population of potential solutions, "bowers," each containing a set of features.

The algorithm first defines various parameters: the count of bowers (N), the failed bower's fraction (ρ), the count of iterations (T), and the frequency of bower failure evaluations (τ). An essential component of the technique was the implementation of the Levy flight strategy, characterized by λ, which facilitates the random walks to choose features from a bower for possible inclusion in another, emulating the bower bird's method of incorporating new decorations selectively from its environment.

The method initiates by creating a bower's initial population randomly. Every bower is evaluated using a fitness function that assesses the efficacy of the feature subsets alongside their simplicity of computation, so reconciling efficiency with the difficulty of the model. The fitness parameter was essential since it affects the selection directly and optimization of the feature subset. Figure 3 depicts the flowchart of BBFS.

In every iteration of the algorithm, the bowers are ranked according to their fitness, and the minimal effective ones are selected for possible reinitialization. For every bower B_i, the method does a Levy flight to identify the bower randomly B_j. The feature selection for the potential transfer from B_j to B_i was determined by the assessed potential enhancement each feature may provide, as described in Equation (4):

$$\Delta F_k = F\big(B_i \cup \{f_k\} - F(B_i)\big) \tag{4}$$

Here, F denotes the Fitness Function (FF). The aggregate prospective enhancement from incorporating features from B_j is given in Equation (5):

$$\sum \Delta F = \sum_{f_k \in B_j - B_i} \Delta F_k \tag{5}$$

The probability of incorporating a given feature f_k into B_i is ascertained by Equation (6):

$$P_k = \frac{\Delta F_k}{\sum \Delta F} \tag{6}$$

Features are incorporated or eliminated according to the probabilities, mirroring the Bowerbird's activity of modifying its nest. This active optimization persists through designated iterations till convergence is attained, with the technique consistently enhancing the feature subsets towards ideal configurations.

The BBFS technique functions as a feature selection of a wrapper-based technique that assesses and optimizes the feature subset selection utilizing a designated DL model. This strategy guarantees that the chosen features significantly enhance the efficacy of the DL model employed for the classification of attacks.

The efficacy of every feature subset is evaluated according to the efficiency of the developed DL model and the quantity of selected features. This metric directly assesses the efficacy of the chosen features in predicting the output with low errors and maximum explanatory capacity. The wrapper method was advantageous as it assesses the predictive efficacy of a feature subset immediately via the performance of the model, thus incorporating the selection of features dynamically into the training procedure of the model.

The efficacy of the feature subsets in Equation (7), denoted as B, is assessed using the preceding equation, whereby ω serves as a parameter that equilibrates the significance of the model's accuracy and the feature's complexity.

$$F_{Classification}(B) = \omega \cdot Accuracy + (1-\omega) \cdot \left(1 - \frac{|B|}{|F|}\right) \tag{7}$$

The BBFS technique is predominantly utilized as the wrapper approach in this research, while it could be modified for filter-based feature selection. This facilitates a more expedient first feature selection approach that could diminish the feature spaces before the application of more computationally demanding wrapper techniques [31]. After applying this BBFS technique, fifteen optimal features were selected from the dataset. The chosen features were Covariance, flow_durations, Duration, Protocol_Types, Srate, Headers_Length, fin_counts, Min, syn_counts, urg_counts, rst_counts, HTTPS, IAT, Weight, and Tot size.

3.4. LSTM-AE Classification

LSTMs were particularly suited for managing dependencies of long-range, rendering them effective for data sets with inherent temporal dynamics. The AE enhances the model's ability for learning adaptable and significant representations from the data of input while filtering noises and accentuating feature extraction.

Thus, in highly complicated scenarios of nonlinear relationships and high-dimensional information, LSTM–AE stands out. Further, it might extrapolate on unknown data, which was vital for classifying the ever-changing environment.

Compared to classic ML algorithms, LSTM–AE presents precision in time-series forecasting, thus being suitable for real-life applications. Besides, the ability to recognize the temporal association existing in sequential data will undoubtedly enhance its performance in a time-series process. The training approach of AE is undertaken with inputs wherein the model tries to output and tries to obtain its encoded representation during decoding, from which it tries to generate reconstructed outputs.

This research employs the AE sequence utilizing LSTM for predicting the result. Two different layers of the LSTM were utilized for both encoding and decoding, while the recurrent vector layers function as the intermediary within the encoding and decoding layers.

The LSTM was the foundational structure of the Recurrent Neural Network (RNN), adept in mitigating the vanishing gradient issues commonly encountered in RNNs with extended sequential information. The architecture of LSTM comprises RNN elements, each featuring three control gates: input, output, and forget gates, constructed using sigmoid neural network and pointwise multiplication operations. The output sigmoid layer (σ) scales from zero to one for input data.

The time series analysis with LSTM techniques utilizes the input vectors sequence $U = \{U1, U2, \dots, Ut, \dots\}$, where $Ut \in R^m$ represents an m-dimensional vector containing m the variable at the t-th time point. The input vector U_t is utilized as illustrated by the LSTM module. Initially, it identifies the prior data to be discarded via the function. F_t, delineated as follows in Equation (8):

$$F_t = \sigma_1\big(W_f \cdot [H_{t-1}, U_t] + B_f\big) \tag{8}$$

Meanwhile, H_{t-1} signifies the outcome for the $t-1$ time-stamp, whereas B_f and W_f imply bias features and weighted matrices, respectively. Subsequently, identify the data to be discarded, and then manage. U_t. The function of the input gate I_t and the layer of $tanh$ generated candidate values of the vector $\tilde{C}_t$ to enhance the new cell states C_t. These factors are derived using Equations (9)-(11).

$$I_t = \sigma_2(W_i \cdot [H_{t-1}, U_t] + B_i) \tag{9}$$

$$\tilde{C}_t = tanh_1(W_c \cdot [H_{t-1}, U_t] + B_c) \tag{10}$$

$$C_t = F_t \times C_{t-1} + I_t \times \tilde{C}_t \tag{11}$$

The variables (W_c, B_c) and (W_i, B_i) denote the biased features and weighted matrices of the inputs and memory state, respectively. The output gate (O_t) has been identified using Equations (12) and (13).

$$O_t = \sigma_3(W_O \cdot [H_{t-1}, U_t] + B_O) \tag{12}$$

$$H_t = O_t * \tanh(C_t) \tag{13}$$

The variables B_O and W_O represent biased features and weighted matrices of the output gates, accordingly, which regulate the generation of the cell states component. The LSTM encompasses multiple models, a few of which were more efficient than others, depending on the specific problems.

Encoders and decoders signify complementary AE modules typically utilized for dimensionality reduction or feature extraction. Autoencoders are executed by the encoder's (input data) preliminary learning models within a low-dimensional space after employing this method to reconstruct the new decoder (input). The primary objective of the AE was not simply to replicate input as output data. The AE model's defining feature is its capability to compress data sizes while preserving most of the features that are inherent in the data structure. The study presents many Autoencoders, including LSTM–AE, regularized AE, convolution AE, and vanilla AE. The category of LSTM encompasses the encoder or decoder, the LSTM methodology.

LSTM was employed to capture long-range relationships in multivariable time-series information for the purpose of predicting anomaly detection problems. Post-training, the encoding and decoding methodology, which employs conventional data samples for recovery, may fail to recreate well when confronted with aberrant information, leading to increased reconstruction errors. The implemented AE integrates a recurrent vector layer to create a connection and latent spaces within the encoding and decoding layers [32].

3.5. SGO-based Fine-Tuning

The SGO method effectively fine-tunes the LSTM–AE method's hyperparameters, resulting in improved classification performances. This methodology is selected for its exceptional capacity to adjust hyperparameters effectively, hence enhancing the performance and convergence of deep learning algorithms.

The SGO method is a contemporary metaheuristic intelligence technique derived from the attacking and migratory behaviors of seagulls. The fundamental premise is to identify the global optimal solution by correlating both the global and local search advancements of various individuals within the population.

Simultaneously, attack represents a local optimization, whereas migration constitutes a global optimization. Compared to alternative optimization methods, the SGO offers the following two advantages:

- The algorithm's structure is open, with few parameters to manage, making it simple to address various types of problems.
- A global optimization technique known as SGO has enhanced efficacy for global search and local exploitation, addressing the challenge of high dimensionality.

During the flight, it influences the local developmental capacity of the SGO; yet, predatory behavior is exhibited by seagulls when they attack for sustenance in aquatic and terrestrial environments. A migration technique is akin to a seagull's flight in various directions, optimized for survival in its current environment but not necessarily in a different location.

Migration: The SGO conducts a comprehensive search by emulating the random flying patterns of seagulls in various directions. Each random seagull must progressively fulfill the criteria during this process.

Collision Avoiding: The SGO assesses the seagull's post-migration position $C_s(t)$ by considering the current position $P_s(t)$ and an extra parameter A to prevent collisions with neighboring seagulls.

$$C_s(t) = A \times P_s(t) \tag{14}$$

Here, in this Equation (14), t indicates the count of iterations present; A was the propulsion of seagulls in space.

$$A = f_c - \left(t \times \frac{f_c}{Max_{iter}}\right) \tag{15}$$

Here in Equation (15), f_c is the linear function that decreases the value of A linear from f_c to zero; Max_{iter} is the maximum iteration count.

Best Location Orientation: Considering that the seagulls do not collide with one another, they must navigate towards the optimum location $M_s(t)$.

$$M_s(t) = B \times \left(P_{best}(t) - P_s(t)\right) \tag{16}$$

Here in Equation (16), $P_{best}(t)$ denotes the optimum location for seagulls; B represents the random value that equilibrates both global and local optima. In the following Equation (17), rd is the generated integer randomly within the range [0,1].

$$B = 2 \times A^2 \times rd \tag{17}$$

Movement to Optimal Location: Upon fulfilment of the dual conditions, the seagull must move toward the optimal location until it reaches its newest position $D_s(t)$, as defined by the following Equation (18):

$$D_s(t) = |C_s(t) + M_s(t)| \tag{18}$$

Attacking behaviour: Seagulls utilize their mass and wings while migrating to maintain a particular altitude. They execute a spiral dive towards the target upon identifying an objective for attack. The flying behaviors of seagulls are depicted as x, y, and z three-dimensional planes:

$$x = r \times \cos(\theta) \tag{19}$$

$$y = r \times \sin(\theta) \tag{20}$$

$$z = r \times \theta \tag{21}$$

$$r = u \times e^{\theta v} \tag{22}$$

Here, in these Equations (19)-(22), r denotes the radius of the helix; θ represents the angle within the interval [0, 2π]; u signifies the correlation constants for the helix. Calculation formula for seagull attack location is given in Equation (23):

$$P_s(t) = D_s(t) \times x \times y \times z + P_{best}(t) \tag{23}$$

The FF is an essential component of the SGO approach. The encrypted approach is structured to compute the optimal outcome for candidate results. The values of accuracy are the primary metrics established for estimating an FF using Equation (24).

$$FF = \max(P) \tag{24}$$

$$P = \frac{TP}{TP+FP} \tag{25}$$

Here, in Equation (25), FP and TP denote the positive values of false positives and true positives, respectively [33]. Table 3 presents the developed LSTM-AE model's hyperparameters that were tuned using the SGO technique. The pseudocode for this developed BBFS-LSTM-AE-SGO model is presented in the following.

The BBFS-LSTM-AE-SGO algorithm is composed of four main steps, including preprocessing, feature selection, DL classifier, and hyperparameter tuning. In the first step, the CIC-IoT-2023 dataset was preprocessed using the SMOTE technique for balancing the minority class and Min-Max normalization for normalizing the features. In the next step, the BBFS method was used to choose the most relevant traffic features based on a fitness function. The selected feature set was then fed into the LSTM-AE framework, where the encoder learns to represent temporal traffic features in latent vectors, whereas the decoder decodes the input sequence back to find any anomalies using reconstruction error. The LSTM neural network model was useful for capturing temporal dependencies in IoT traffic flows. In order to improve the performance further, the SGO technique was utilized for the

optimization of important hyperparameters like learning rate, batch size, hidden units, dropout rate, and the number of epochs. The movements of the seagulls in terms of migration and attacks are mathematically modeled to ensure a good compromise between global exploration and local exploitation.

Table 3. Hyperparameters of LSTM-AE tuned by SGO

Hyperparameter	Optimized Values	Description
Learning Rate	0.0012	Step size for weight updates during training
Batch Size	64	Number of samples processed before weight update
Number of LSTM Layers	2	Depth of LSTM encoder-decoder structure
Hidden Units per Layer	128	Number of neurons in each LSTM layer
Dropout Rate	0.3	Regularization factor to prevent overfitting
Activation Function	Tanh	Non-linearity in hidden layers
Optimizer	Adam	Optimization algorithm for training
Epochs	100	Maximum training iterations

Algorithm: BBFS-LSTM-AE with SGO

Input: CIC-IoT-2023 dataset D
Output: Classification results (Normal / Attack categories)
Load CIC-IoT-2023 dataset
  Apply SMOTE on D to handle class imbalance
  Normalize features of D using Min-Max scaling
  Split D into training set (D_train) and testing set (D_test)
Feature Selection using BBFS
  Initialize population of candidate feature subsets
  For each iteration:
    Evaluate the fitness of each candidate using classification accuracy
    Update the positions of candidates based on the BBFS strategy
  Select optimal feature subset F_opt
  Transform D_train and D_test using F_opt
Classification with LSTM Autoencoder (LSTM-AE)
  Initialize LSTM-AE with H_opt
  Train LSTM-AE on D_train (with selected features F_opt)
  Test model on D_test to obtain predictions P
Hyperparameter Optimization using SGO
  Define search space for LSTM-AE hyperparameters:
    - learning rate, batch size, hidden units, dropout, epochs
  Initialize population of candidate hyperparameter sets
  For each iteration:
    Train LSTM-AE on D_train with candidate hyperparameters
    Evaluate performance
    Update candidate hyperparameters using the SGO strategy
  Select the best hyperparameter set H_opt
Performance Analysis
  Compute evaluation metrics
  Return classification results and performance metrics
End

The pseudocode highlights the entire workflow of the IoT-IDS, starting with loading the CICIoT2023 data set and preprocessing it, where SMOTE balances class distributions and Min-Max normalization scales features, after which the dataset is split into training and testing sets. Next, it applies BBFS to select the most relevant features by iteratively evaluating candidate subsets based on classification accuracy, which eliminates redundancy and boosts efficiency. These selected features, then, enter an SGO-based hyperparameter optimization phase for the LSTM-AE to optimize learning rate, batch size, hidden units, dropout rate, and epochs by assessing candidate configurations and updating them according to the SGO search strategy. The LSTM-AE was trained using the optimal hyperparameters with the refined feature set to learn temporal patterns and reconstruct data, thereby being sorted into normal and attack traffic. Finally, the

model is applied to unseen data, and standard performance metrics such as accuracy, precision, recall, F1-score, and FAR are used to test the intrusion detection framework adequately.

4. Results and Discussion

4.1. Experimental Setup

This section highlights the experiments conducted on the extensive CIC-IoT-2023 dataset. The developed BBFS-LSTM-AE-SGO model was experimented with in an evaluation configuration with a system with Windows 11, a 64-bit operating system, an Intel i7 CPU operating at 4.6 GHz, and 16GB RAM. The BBFS-LSTM-AE-SGO model was developed utilizing the Python 3.11.4 64-bit programming and relied on the Scikit-learn and Numpy library modules. The attained results for the developed BBFS-LSTM-AE-SGO model are compared and validated with the current existing IoT-IDS frameworks. To ensure that all comparative models had the same conditions of analysis, identical preprocessing parameters, dataset splits, and measures of performance were used for each of the algorithms. To avoid experimental variance, random initialization seeds were preserved for all learning algorithms. The developed framework is simple to implement by using Python libraries, which makes it possible to replicate the research in further studies.

4.2. Evaluation Metrics

The study conducted an evaluation on the performance of the developed BBFS-LSTM-AE-SGO algorithm using binary classification and multiclass classification.

These parameters include Accuracy, Precision, Detection Rate, and F1-score. The mathematical representation of the above-listed parameters is provided in the following Equations (26)-(30).

TP-True Positive: Number of samples which are actually positive but have been correctly classified.

TN-True Negative: Number of samples which are actually negative but have been correctly classified.

FP-False Positive: Number of samples which are actually negative but have been falsely classified as positive.

FN-False Negative: Number of samples which are actually positive but have been wrongly classified as negative.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (26)$$

Accuracy is an assessment of how accurately the model is at correctly categorizing the results, hence giving an overall analysis of the model.

$$Precision = \frac{TP}{TP+FP} \quad (27)$$

Precision focuses on the reliability of the positive prediction by ensuring that there are no FP. The accuracy and precision metrics are very important in the IDS.

$$Detection\ Rate = \frac{TP}{TP+FN} \quad (28)$$

The Detection Rate (DR), also known as recall, determines how well the BBFS-LSTM-AE-SGO detects intrusions. A DR that produces a high recall must be implemented in order to avoid missing any detections in providing a robust system that is efficient in identifying intrusions.

$$F1score = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (29)$$

The F1-score is the harmonic mean of recall and precision. It signifies a balanced assessment of the classifier's overall performance. A high value of F1 indicates the performance of the model in achieving an optimal trade-off between recall and precision. False Positive Rate (FPR) in IDS refers to the ratio of the total number of alerts generated that happen to be incorrect or non-malicious.

$$FPR = \frac{FP}{FP+TN} \quad (30)$$

False positives can be caused by many reasons, such as incorrect configurations, out-of-date rules or signatures, and normal traffic that triggers the IDS. Therefore, lowering false positives becomes an important consideration when deploying and designing the IDS [13-27].

4.3. Discussion of Results

In this section, a detailed analysis of the experimental results obtained from the BBFS-LSTM-AE-SGO model will be presented. The effectiveness of the BBFS-LSTM-AE-SGO model in detecting IoT attacks was evaluated based on performance measures.

Table 4. Binary classification results of the BBFS-LSTM-AE-SGO model

Metrics	Training Result	Testing Result
Accuracy	99.88	99.63
DR	99.82	99.55
F1-score	99.75	99.59
Precision	99.89	99.71
FPR	0.12	0.37

The performances of the BBFS-LSTM-AE-SGO model in binary classifications for the CIC-IoT-2023 dataset are presented in Table 4. In the process, 99.88% training accuracy and 99.63% testing accuracy are realized, showing that the model is able to learn discriminatory patterns with minimal overfitting.

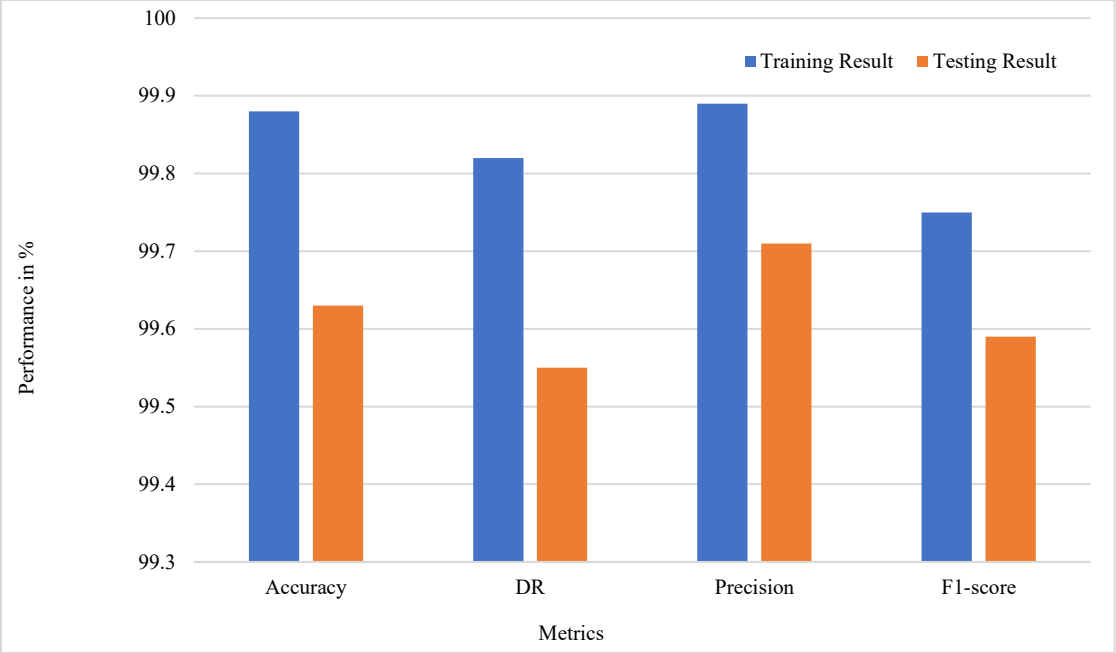

Fig. 4 Graphical illustration of binary classification results

With 99.82% detection accuracy in training and 99.55% in testing, it is shown that the model performs well in classifying malware traffic cases, an important factor in intrusion detection. Objective measures such as 99.89% training accuracy and 99.71% testing precision show that the system is very precise in minimizing false positives. In addition, the F1-score of both precision and detection rates (99.75% during training, 99.59% during testing) indicates that the proposed model is also strong when handling imbalanced intrusion datasets.

The False Positive Rate (FPR) for this system was still extremely low, at only 0.12% during training and 0.37% during testing. Based on these findings, it is evident that BBFS-driven feature selection, LSTM-AE for time-series representation, and SGO-driven hyperparameter optimization will make it possible for the model to detect intrusions accurately with few or no misclassifications, hence making an important contribution to IoT security applications. Figure 4 shows the graphical representation of the binary classification of the proposed BBFS-LSTM-AE-SGO model.

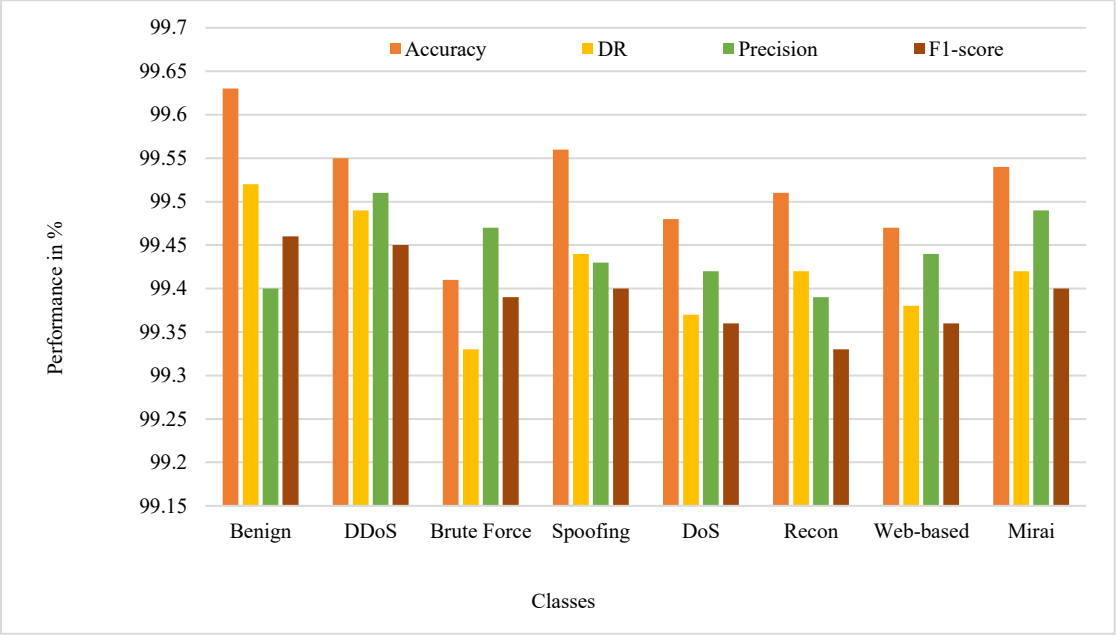

Fig. 5 Graphical illustration of multiclass classification results

Table 5. Multiclass classification results of the BBFS-LSTM-AE-SGO model

Classes	Acc	DR	Prec	F1
DDoS	99.55	99.49	99.51	99.45
Benign	99.63	99.52	99.40	99.46
Spoofing	99.56	99.44	99.43	99.40
DoS	99.48	99.37	99.42	99.36
Recon	99.51	99.42	99.39	99.33
Brute Force	99.41	99.33	99.47	99.39
Web-based	99.47	99.38	99.44	99.36
Mirai	99.54	99.42	99.49	99.40
Average	99.52	99.41	99.45	99.38

The results of the multiclass classification for the BBFS-LSTM-AE-SGO algorithm for the CICIoT2023 dataset have been presented in Table 5. It clearly shows that the model is able to classify accurately several classes of traffic belonging to an IoT network, such as benign and attacks, which shows the capability of the model. All classes of the model achieve accuracy in the range of 99.41% (Brute Force) to 99.63% (Benign). The detection rates are also found to be on the high side, starting from 99.33% for Brute Force to the maximum value of 99.52% for Benign. This clearly indicates that the model is capable of detecting different kinds of attack vectors and keeping the rate of false negatives to a minimum. Furthermore, the precision rate remains consistent for all the classes, ranging between 99.39%-99.51%, thus proving that the model has validity to ensure that there will be no false alarms concerning both common and rare attacks. Based on the average values of the metrics, such as accuracy at 99.52%, detection rate at 99.41%, precision at 99.45%, and F1-score at 99.38%, it is quite clear that the model provides consistently accurate performance despite the problem of class bias, which normally exists when multiple classes of attacks need to be detected. Amongst the attacks that are difficult to identify, detection of Mirai and spoofing is done with comparable accuracy and precision, indicating that the feature selection method implemented in this study via Bowerbird has the ability to filter out those features that are crucial for identifying attacks, and LSTM-AE models sequential attack signatures while SGO optimizes model parameters. These findings clearly indicate the applicability of the proposed methodology in the context of multiclass intrusion detection, hence providing a safe, scalable, and robust solution for the IoT ecosystem. Figure 5 shows the graphical representation of the multiclass classification results achieved by the developed BBFS-LSTM-AE-SGO model.

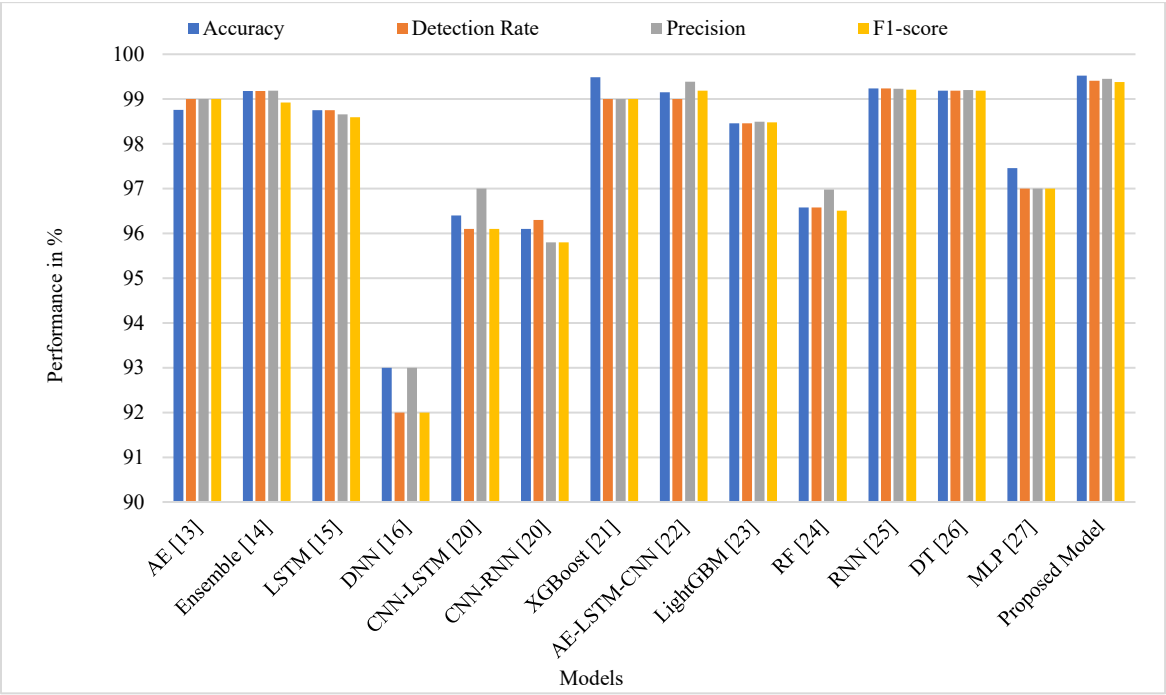

Fig. 6 Graphical view of comparison of results

Table 6. Comparison of performance with existing models

Model	Acc	DR	Prec	F1
AE [13]	98.76	99.00	99.00	99.00
Ensemble [14]	99.18	99.18	99.19	98.92
LSTM [15]	98.75	98.75	98.66	98.59
DNN [16]	93.00	92.00	93.00	92.00
CNN-LSTM [20]	96.40	96.10	97.00	96.10
CNN-RNN [20]	96.10	96.30	95.80	95.80
XGBoost [21]	99.49	99.00	99.00	99.00
AE-LSTM-CNN [22]	99.15	99.00	99.39	99.19
LightGBM [23]	98.46	98.46	98.49	98.48
RF [24]	96.58	96.58	96.98	96.51
RNN [25]	99.24	99.24	99.23	99.21
DT [26]	99.19	99.19	99.20	99.19
MLP [27]	97.46	97.00	97.00	97.00
Proposed Model	99.52	99.41	99.45	99.38

Table 6 presents a comprehensive performance comparison between the proposed BBFS-LSTM-AE-SGO framework and other prominent ML and DL models from prior works, thus visually emphasizing the superiority of the proposed framework.

However, conventional models like DNN with accuracies of 93.00% and precisions of 93.00% may not be able to deal with such complex network traffic patterns by shallow architecture design or by a lack of optimization strategy. Other models with contemporary architectures, for instance, AE (98.76% accuracy), LSTM (98.75% accuracy), LightGBM (98.46% accuracy), and MLP (97.46% accuracy), performed better under some circumstances but still do not guarantee either robustness or generalizability in changing environments in which attacks evolve.

Further competitive baselines such as Ensemble (99.18%), AE-LSTM-CNN (99.15% accuracy, 99.39% precision, 99.19% F1-score), DT (99.19% accuracy), RNN (99.24% accuracy, 99.21% F1-score), and XGBoost (99.49% accuracy) can be considered as almost the best alongside the proposed model, yet showing slight trade-offs in terms of detection rate, precision, and F1-score. Completing the performance balance, the BBFS-LSTM-AE-SGO model attains the highest accuracy (99.52%) and highest detection rate (99.41%), precision (99.45%), and F1-score (99.38%). Figure 6 depicts the graphical illustration of the developed BBFS-LSTM-AE-SGO model's multiclass classification results compared with current models. These results, in comparison, indicate that the BBFS-LSTM-AE-SGO design is equally excellent in all the evaluation metrics of importance. This improvement was mainly due to BBFS, which eliminates redundancy and extracts highly discriminative features; the LSTM-AE architecture, which modeled both temporal and nonlinear dependencies in attack sequences; and SGO-enabled hyperparameter optimization, allowing for the optimal tuning of model parameters toward rapid learning.

Therefore, this multi-way technique allows the proposed scheme to not only outperform shallow ML classifiers and traditionally constructed DL models but also outperform Ensemble and hybrid implementations considered to be strong baselines previously. Hence, the comparative analysis places the proposed model as the best high-performance and scalable solution for intrusion detection in IoT networks, capable of strong detectability and resisting a variety of attack categories.

The performance of the proposed BBFS-LSTM-AE-SGO architecture was better than the existing state-of-the-art IoT IDS models. This improvement is due to the efficient utilization of optimized feature selection, temporal learning, and hyperparameter optimization in one system. The BBFS technique selected only the most distinctive features while ignoring other redundant traffic features, unlike traditional ML and DL models, which utilized redundant features and had fixed values for their parameters. The LSTM-AE design successfully modeled the sequential and nonlinear nature of attacks occurring in IoT traffic, resulting in more efficient detection capabilities for various types of attacks. In addition to this, the use of SGO optimization techniques increased convergence speed as well as improved the generalization ability of the proposed method by determining the best values for hyperparameters. Both contributions greatly contributed to minimizing false positives and improving classification performance.

The ablation experiment in Table 7 clearly shows how each of the implemented models contributes to the performance of the proposed intrusion detection framework individually and combined. The baseline LSTM-AE showed good results in terms of performance; nevertheless, its relatively high false-positive rate and low detection accuracy suggest that the model struggles with dealing with redundant and high-dimensional IoT traffic features. With the inclusion of BBFS, the performance of the classifier was greatly enhanced owing to the removal of unnecessary attributes, thus

increasing feature discriminability. In addition, the integration of SGO into the LSTM-AE model helped to improve its convergence ability and hyperparameter tuning, thus improving its generalization performance. Overall, the full BBFS-LSTM-AE-SGO architecture had the best performance, which proved that the combination of intelligent feature selection and metaheuristic hyperparameter optimization greatly enhances intrusion detection capability while reducing false positives in IoT systems. Figure 7 depicts the graph of the ablation study results comparison.

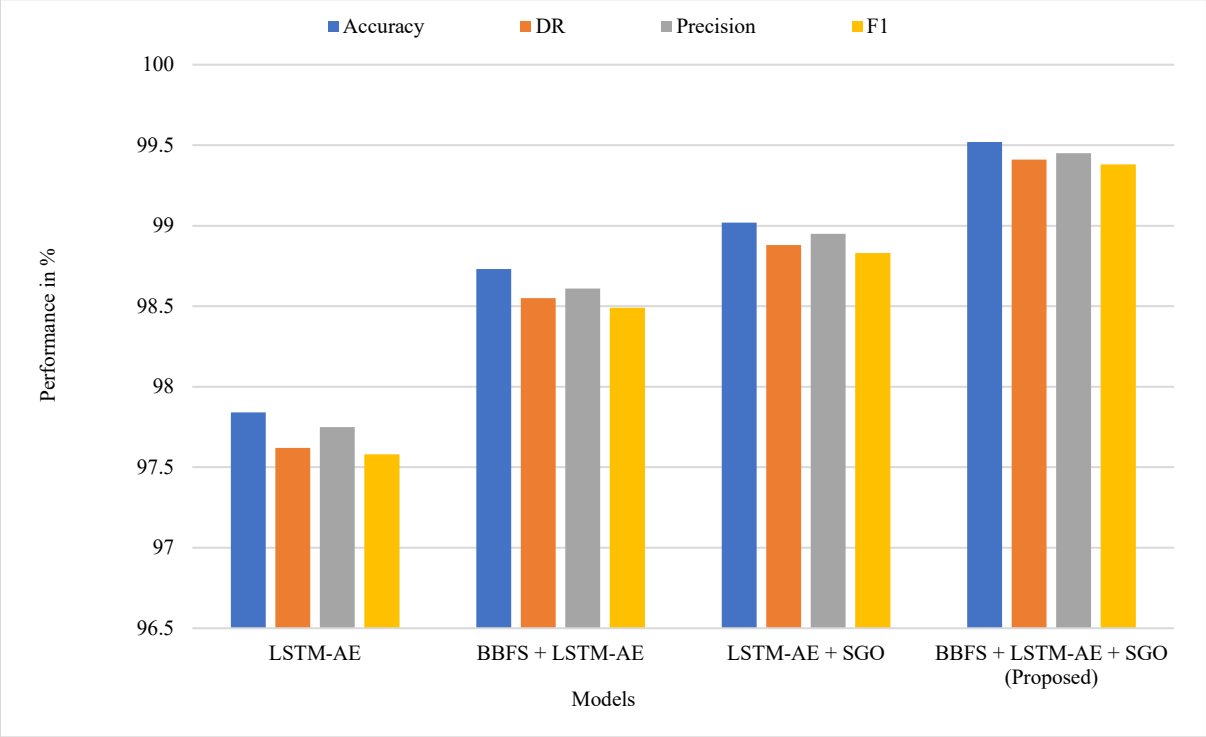

Fig. 7 Graphical illustration of ablation study results comparison

Table 7. Ablation study results comparison

Model Configuration	Accuracy	DR	Precision	F1
LSTM-AE	97.84	97.62	97.75	97.58
BBFS + LSTM-AE	98.73	98.55	98.61	98.49
LSTM-AE + SGO	99.02	98.88	98.95	98.83
BBFS + LSTM-AE + SGO (Proposed)	99.52	99.41	99.45	99.38

While existing IDS models that incorporate DL and optimization have been proposed recently, most of the existing IDS models face problems like high computational cost, inadequate optimization of features, inability to address temporal dependencies, or lack of adaptive tuning of hyperparameters.

For instance, CNN-LSTM and AE-LSTM-CNN models offer enhanced intrusion detection capabilities, but they include an added cost of increased complexity and high computational costs. Models based on ensembles and Transformers provide better performance, but they usually fail to have efficient feature reduction and optimization approaches. Conversely, the BBFS-LSTM-AE-SGO model presents a complete and optimized IDS structure that utilizes intelligent feature selection, temporal anomaly detection, and bio-inspired hyperparameter tuning. The application of BBFS helps minimize unnecessary features, the utilization of LSTM-AE facilitates the detection of attack behavior sequences, while SGO increases learning efficiency and stability. This approach results in enhanced detection accuracy, reduced false positives, and increased scalability as opposed to other existing hybrid IDS models.

4.4. Advantages & Limitations

The proposed BBFS-LSTM-AE-SGO model offers many benefits that make it the best solution for IoT intrusion detection. Use of BBFS reduces the redundant attributes and increases discriminability. The LSTM-AE architecture captures complex temporal dependencies and nonlinear relationships present in IoT traffic data, making detection highly precise. Also, hyperparameter tuning using bio-inspired SGO algorithms finds the best configuration of the model and achieves balancing precision and generalization

over various kinds of attacks, as evidenced by its superior performance to state-of-the-art models in the literature. Though the model has a lot of advantages, however, there are certain limitations that still prevail in this model.

One of these limitations is that due to complexity in feature selection, autoencoder training, and meta-heuristic training, training time will increase. In addition to this, the model performs exceptionally well in benchmark data sets, but in real-world situations, it might face difficulties owing to changes in attack patterns over time.

In addition, just as is common in the case of most techniques based on AI systems, there is a possibility that the model will suffer from the loss of interpretability, which could mean that security analysts cannot understand the reasoning behind the decisions taken. Therefore, with regard to real-time scalability and interpretability, it appears to be an area where improvements can be made.

As far as its practical deployment is concerned, the proposed BBFS-LSTM-AE-SGO model can be implemented in the edge-cloud IoT security system architecture, where the processing and feature extraction are done at the edge side, whereas the training and optimization phases take place at the centralized cloud servers due to their high computational capacity. Such an implementation approach would help achieve low latency in intrusion detection in large IoT environments. Moreover, the optimal feature subset identified through BBFS contributes to memory and communication efficiency.

However, the actual IoT environment might produce streaming data continuously in high volumes and varying attack behavior that can impact the sustainability of detection. Also, constant retraining and optimization might lead to energy and computation overhead in actual implementations. Heterogeneous network characteristics, diverse devices, and encrypted traffic flows can have an impact on the effectiveness of the approach in the real world. Hence, any future improvements in this regard should be concentrated on lightweight model compression, online learning, distributed IDS implementation, and explainable AI technology.

5. Conclusion

In this study, a novel framework of IDS referred to as the BBFS-LSTM-AE-SGO model has been developed using the state-of-the-art technology of feature selection, DL, and meta-heuristics optimization to ensure IoT cybersecurity. The BBFS technique was applied for feature selection as well as for eliminating irrelevant features. The LSTM-AE framework is capable of detecting complex temporal dependencies and nonlinear relationships in IoT traffic, which makes the system scalable enough to detect any anomalies.

Combining SGO-based parameter optimization also helped improve the system by tuning it properly, thereby enhancing the detection process. Experimental findings from the benchmarking on the CIC IoT 2023 data set showed that the proposed model exhibited a binary classification accuracy score of 99.63% and a multiclass average accuracy of 99.52%. The proposed model outperformed the state-of-the-art models, which include CNN-LSTM, XGBoost, AE-LSTM-CNN, and RNN. It also demonstrated an extremely low rate of false positives, at only 0.37% during testing. This is an excellent result, especially for real-time IT cybersecurity operations where false positives should not be allowed. Result comparison also proved that this was the appropriate model, highlighting its balance in all enhancements regarding accuracy, detection, precision, and F1-Score. On theoretical grounds, being robust on results would make implementation complex, especially when deployed on a large scale and on a real-time IoT basis, in addition to interpretability because of DL. In the future, it will be necessary to overcome computational bottlenecks, introduce models for explainable AI, and verify flexibility on continuously evolving attack vectors in real IoT environments. In general, BBFS-LSTM-AE-SGO can be considered a promising and resilient solution for intrusion detection, significantly contributing to developing intelligent and secure IoT solutions.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

Nil.

References

[1] Arjun Kumar Bose Arnob et al., "A Comprehensive Systematic Review of Intrusion Detection Systems: Emerging Techniques, Challenges, and Future Research Directions," *Journal of Edge Computing*, vol. 4, no. 1, pp. 73-104, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ya Zhang, Ravie Chandren Muniyandi, and Faizan Qamar, "A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance," *Applied Science*, vol. 15, no. 3, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Md Mahbubur Rahman, Shaharia Al Shakil, and Mizanur Rahman Mustakim, "A Survey on Intrusion Detection System in IoT Network," *Cyber Security and Application*, vol. 3, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[4] Nuha A. Hamad et al., "Systematics Analysis of Federated Learning Approaches for Intrusion Detection in the Internet of Things Environment," *IEEE Access*, vol. 13, pp. 95410-95444, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] S. Kumar Reddy Mallidi, and Rajeswara Rao Ramisetty, "Optimizing Intrusion Detection for IoT: A Systematic Review of Machine Learning and Deep Learning Approaches with Feature Selection and Data Balancing," *Wiley Interdisciplinary Review: Data Mining and Knowledges Discovery*, vol. 15, no. 2, pp. 1-42, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Saeid Jamshidi et al., "Application of Deep Reinforcement Learning for Intrusion Detection in Internet of Things: A Systematic Review," *Internet of Things*, vol. 31, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Márton Bendegúz Bankó et al., "Advancements in Machine Learning-based Intrusion Detection in IoT: Research Trends and Challenges," *Algorithms*, vol. 18, no. 4, pp. 1-34, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Imane Rakine et al., "Comprehensive Review of Intrusion Detection Techniques: ML and DL in Different Networks," *IEEE Access*, vol. 13, pp. 104345-104367, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Tamara Al-Shurbaji et al., "Deep Learnings-based Intrusions Detections Systems for Detecting IoT Botnet Attack: A Review," *IEEE Access*, vol. 13, pp. 11792-11822, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] S. Kumar Reddy Mallidi, and Rajeswara Rao Ramisetty, "Advancements in Training and Deployment Strategies for AI-based Intrusion Detection Systems in IoT: A Systematic Literature Review," *Discover Internet of Things*, vol. 5, no. 1, pp. 1-33, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mohammad Shamim Ahsan, Salekul Islam, and Swakkhar Shatabda, "A Systematic Review of Metaheuristics-based and Machine Learning-Driven Intrusion Detection Systems in IoT," *Swarm and Evolutionary Computations*, vol. 96, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Muhannad Almohaimeed et al., "Use of Machine Learning and Deep Learning in Intrusion Detection for IoT," *Advances in Internet of Things*, vol. 15, no. 2, pp. 17-32, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Emanuel Krzysztoń, Izabela Rojek, and Dariusz Mikołajewski, "A Comparative Analysis of Anomaly Detection Methods in IoT Networks: An Experimental Study," *Applied Sciences*, vol. 14, no. 24, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Hidangmayum Satyajeet Sharma, and Khundrakpam Johnson Singh, "A Genetic Algorithm based Feature Selection and CNN based Ensemble Model for Intrusion Detection in IoT Smart Environments," *Journal of Scientific and Industrial Research (JSIR)*, vol. 84, no. 1, pp. 48-59, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Akinul Islam Jony, and Arjun Kumar Bose Arnob, "A Long Short-Term Memory based Approach for Detecting Cyber Attacks in IoT using CIC-IoT2023 Dataset," *Journal of Edge Computing*, vol. 3, no. 1, pp. 28-42, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sheikh Abdul Wahab et al., "A Multi-Class Intrusion Detection System for DDoS Attacks in IoT Networks using Deep Learning and Transformers," *Sensors*, vol. 25, no. 15, pp. 1-35, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shahbaz Ahmad Khanday, Hoor Fatima, and Nitin Rakesh, "A Novel Data Preprocessing Model for Lightweight Sensory IoT Intrusions Detections," *International Journal of Mathematical*, *Engineering and Managements Science*, vol. 9, no. 1, pp. 188-204, 2024. [CrossRef] [Google Scholar]

[18] Zaed S. Mahdi, Rana M. Zaki, and Laith Alzubaidi, "A Secure and Adaptive Framework for Enhancing Intrusion Detection in IoT Networks using Incremental Learning and Blockchain," *Security and Privacy*, vol. 8, no. 4, pp. 1-17, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zaed S. Mahdi, Rana M. Zaki, and Laith Alzubaidi, "Advanced Hybrid Techniques for Cyberattack Detection and Defense in IoT Networks," *Security and Privacy*, vol. 8, no. 2, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Farhan Ullah et al., "Enhanced Network Intrusion Detection System for Internet of Things Security using Multimodal Big Data Representation with Transfer Learning and Game Theory," *Sensors*, vol. 24, no. 13, pp. 1-31, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ram Ji, Neerendra Kumar, and Devanand Padha, "Hybrid Enhanced Intrusion Detection Frameworks for Cyber-Physical Systems via Optimal Features Selection," *Indian Journal of Sciences and Technology*, vol. 17, no. 30, pp. 3069-3079, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Bambang Susilo, Abdul Muis, and Riri Fitri Sari, "Intelligent Intrusion Detection System Against Various Attacks based on a Hybrid Deep Learning Algorithm," *Sensors*, vol. 25, no. 2, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Kayode S. Adewole, Andreas Jacobsson, and Paul Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," *Sensors*, vol. 25, no. 6, pp. 1-27, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Sami Yaras, and Murat Dener, "IoT-based Intrusion Detection System using New Hybrid Deep Learning Algorithm," *Electronics*, vol. 13, no. 6, pp. 1-28, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shu-Ming Tseng, Yan-Qi Wang, and, Yung-Chung Wang "Multi-Class Intrusions Detections based on Transformers for IoT Network using CIC-IoT 2023 DataSet," *Future Internet*, vol. 16, no. 8, pp. 1-25, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Akinul Islam Jony, and Arjun Kumar Bose Arnob, "Securing the Internet of Things: Evaluating Machine Learnings Algorithm for Detecting IoT Cyberattack using CIC IoT2023 DataSet," *International Journal of Information Technology and Computer Science*, vol. 16, no. 4, pp. 56-65, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Hadeel Q. Gheni, and Wathiq L. Al-Yaseen, "Two-Steps Data Clustering for Improved Intrusions Detections Systems using CIC IOT2023 DataSet," *e-Prime-Advance in Electrical Engineering, Electronics and Energy*, vol. 9, pp. 1-8, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Md. Meheraj Hossain et al., "A Cross-Dataset based Zero-Day Intrusion Detection System by Integrating Siamese Network and Reinforcement Learning," *ICT Express*, vol. 12, no. 3, pp. 752-757, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[29] Jiantao Cui, and Yixiang Zhao, "Design of Network Traffic Analysis Models based on Deep Neural Networks," *Future Internet*, vol. 18, no. 3, pp. 1-16, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[30] Sethu Subramanian N et al., "Edge AI Bridge: A Micro-Layer Intrusion Detection Architecture for Smart-City IoT Networks," *IoT*, vol. 7, no. 2, pp. 1-40, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[31] Muhammad Ahmad Bilal et al., "Dataset-Centric Evaluation of Federated Intrusion Detection Models in IoT Networks," *Scientific Reports*, vol. 16, no. 1, pp. 1-19, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[32] Abdulwahid Al Abdulwahid et al., "A Hybrid Improved Binary GWO-PSO with Random Forests (IBGWO-PSO-RF) based Intrusions Detections Models for Large-Scaled Attack in IoT Environments," *Scientific Reports*, vol. 16, no. 1, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[33] Euclides Carlos Pinto Neto et al., "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, pp. 1-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Arati Behera et al., "Enhancing DDoS Detections in SDIoT Through Effective Features Selections with SMOTE-ENN," *PloS one*, vol. 19, no. 10, pp. 1-29, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[35] Ramya Chinnasamy et al., "Deep Learning-Driven Methods for Network-based Intrusion Detection Systems: A Systematic Review," *ICT Express*, vol. 11, no. 1, pp. 181-215, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[36] S. Kumar Reddy Mallidi, and Rajeswara Rao Ramisetty, "Bowerbird Courtship-Inspired Feature Selection for Efficient High-Dimensional Data Analysis using a Novel Meta-Heuristic," *Discover Computing*, vol. 28, no. 6, pp. 1-28, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[37] Zhan Wang et al., "Explanatory LSTM-AE-based Anomaly Detections for Time Series Data in Marines Transportations," *IEEE Access*, vol. 13, pp. 23195-23208, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[38] Vimal Kumar Pathak, Swati Gangwar, and Mithilesh K. Dikshit "A Comprehensive Survey on Seagulls Optimizations Algorithms and its Variant," *Archive of Computational Method in Engineering*, vol. 32, no. 1, pp. 3651-3685, 2025. [CrossRef] [Google Scholar] [Publisher Link]