Deep Learning-Based Intelligent Intrusion Detection With Optimized Feature Selection for IoT Network Security

Surendra Kumar, Avinash Kumar, Mridula Dwivedi, Mohit Kumar, Huaming Wu, *Senior Member, IEEE*, and Sukhpal Singh Gill

***Abstract*—Intrusion detection is crucial to network security, especially in Internet of Things (IoT) contexts with rising cyberthreats. Traditional Intrusion Detection Systems (IDSs) can cause false alerts and are difficult to detect new attack types. Metaheuristic optimization can enhance detection rate, model complexity, and feature selection, but it lacks scalability and flexibility to changing situations. Nature-inspired methods enable deep learning models to focus on key features. Current existing methods face challenges in securing the network from several attack types and adapting to evolving attack patterns effectively, identifying changing attack patterns in IoT networks with great accuracy. This paper presents a novel IoT-aware intrusion detection framework that integrates the Whale Optimization Algorithm (WOA), Dipper Throated Optimization (DTO), and Ant Lion Optimization (ALO) for optimized feature selection, alongside Long Short-Term Memory and eXtreme Gradient Boosting for accurate anomaly detection. The proposed method effectively balances exploration and exploitation to select informative features, while the learning models capture temporal and nonlinear patterns in network traffic. Experimental results evaluated on benchmark datasets, including NSL-KDD, KDDCup99, CICDDoS2019, and NF-UNSW-NB15-v2, demonstrate that the proposed framework achieves high detection performance with reduced false alarms. The proposed frameworkimproves computational efficiency and maintains strong detection capability across multiple attack categories,making it scalable and secure for real-time IoT intrusion detection.**

***Index Terms*—IoT security, deep learning, metaheuristic optimization, intrusion detection, cyber threats.**

I. Introduction

ARTIFICIAL Intelligence (AI) is frequently used to secure IoT networks against cyberattacks. Traditional Intrusion Detection Systems (IDSs) can have difficulty handling complex and developing threats, but machine and deep learning models can detect unusual behaviors and novel forms of attacks [1]. As Internet usage and connected devices proliferate, data privacy and security concerns increase [2]. Virtualized cloud services raise security and trust issues, increasing the issue. Intrusions are becoming more frequent, putting individuals, companies, and society at risk [3]. Firewalls, access controls, and other fundamental security measures will no longer detect advanced attacks quickly. Thus, robust and intelligent defense measures are necessary [4]. Modern IDSs detect suspicious activities quickly, improving computer and network security [5]. These systems enhance device availability by detecting and recognizing various harmful activities [6]. In general, IDSs are host-based, network-based, or hybrid. Host-based IDSs evaluate system logs and audit data related to a machine, while network-based IDSs analyze network traffic for suspicious behavior [7]. Hybrid IDSs, combining both views, are prevalent and offer enhanced detection capabilities and system performance [8].

Various optimization methods represent a single optimiser, which can constrain exploration versus exploitation when selecting informative characteristics from complex Internet of Things (IoT) traffic data. Anomaly-based IDS can identify new attacks, but false positive rates and high-dimensional network features are weak. Various optimisation techniques are used for feature selection, e.g., Ant Colony Optimisation (ACO), Grey Wolf Optimisation (GWO), Whale Optimisation Algorithm (WOA), Optimised Sine Swarm (OSS), Gradient Descent Spider Monkey Optimisation (GDSMO), and Teaching-Learning-Based Optimisation (TLBO) [9]. This paper presents a Hybrid WOA-DTO-ALO Feature Selection Framework to enhance Intrusion Detection. Pre-Processing, Optimised Feature Selection, Classifier Training, and Attack Decision comprise the IoT-aware

Received 2 December 2025; revised 2 April 2026 and 5 June 2026; accepted 23 June 2026. Date of publication 1 July 2026; date of current version 16 July 2026. This work was supported in part by the Emerging Frontiers Cultivation Program of Tianjin University Interdisciplinary Center, Tianjin Natural Science Foundation Project under Grant 25JCYBJC01540, in part by Tianjin Municipal Science and Technology Major Program under Grant 25ZXZSSS00390, and in part by Tianjin Future Industry Science and Technology Major Project under Grant 25ZXWCSY00290. *(Corresponding author: Huaming Wu.)*

Surendra Kumar and Avinash Kumar are with the Department of Computer Engineering and Applications, GLA University, Mathura 281406, India (e-mail: surendra.kumar@gla.ac.in; avinash.kumar_mt.cs23@gla.ac.in).

Mridula Dwivedi is with the Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow 226025, India (e-mail: mridula.rs.cs@bbau.ac.in).

Mohit Kumar is with the Department of IT, Dr. B.R. Ambedkar National Institute of Technology, Jalandhar 144008, India (e-mail: kumarmohit@nitj.ac.in).

Huaming Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

Sukhpal Singh Gill is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: s.s.gill@qmul.ac.uk).

Digital Object Identifier 10.1109/TICPS.2026.3708811

Intrusion Detection Framework. The main contributions are summarized as follows:

- We propose a hybrid Whale Optimization Algorithm (WOA), Dipper Throated Optimization (DTO), and Ant Lion Optimization (ALO) feature selection model that effectively reduces feature dimensionality while preserving discriminative information, thereby improving intrusion detection performance.
- We develop an intrusion detection framework by combining Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost), where LSTM captures sequential dependencies in traffic data, and XGBoost provides efficient and accurate classification. The resulting framework achieves an inference time of 1.2 ms per sample, making it suitable for real-time IoT intrusion detection.
- We integrate optimizing feature selection and classification, the proposed framework significantly reduces false alarms and achieves a false positive rate (FPR) of 0.015, which is 43% lower than that of competing methods.
- We conduct extensive experiments on four benchmark datasets, namely NSL-KDD, KDDCup99, CICDDoS2019, and NF-UNSW-NB15-v2, demonstrating the effectiveness and robustness of the proposed framework. The model achieves average detection accuracies of 99.41%, 99.93%, 99.99%, and 99.15%, respectively, while maintaining strong performance across various attack types.

The rest of the paper is structured as follows: Section II gives a review of related work. Section III details the proposed methodology and describes the data sets used. Section IV contains results from experiments conducted to evaluate our method compared to currently existing methods, and Section V concludes our paper.

II. Related Work

The application of deep learning as a promising approach for implementing Network IDSs is shown through research to be practical at detecting sophisticated and changing attacks. The LSTM Network Intrusion Detection System (NIDS) is augmented through the optimization of hyper-parameters, using Salp Swarm Algorithm (SSA), JAYA, and Particle Swarm Optimization (PSO), and exhibited its capability to detect multiple classes of attacks: Probe, DoS (Denial of Service), DDoS (Distributed Denial of Service), and R2L (Remote to Local) when evaluated on the dataset Canadian Institute for Cybersecurity Intrusion Detection System (CICIDS), Botnet Internet of Things (Bot-IoT), and Network Security Laboratory - Knowledge Discovery in Databases (NSL-KDD) [10]. Similarly, a method using a deep auto-encoder (NIDS) to identify malicious input provided greater accuracy of detection and fewer false positives for UNSW-NB15, NSL-KDD, and CICIDS2017 [11].

Metaheuristic optimization was used to choose features to increase IDS detection. GWO and Particle Swarm Optimization (PSO) were combined to reduce false alarms and improve feature selection accuracy on UNSW-NB15 and NSL-KDD99 datasets [12]. Additionally, integrating WOA with DTO and a voting classifier enhanced IDS performance by balancing datasets using the Synthetic Minority Oversampling Technique (SMOTE) and locality sensitive hash (LSH) techniques [13]. A Gradient Descent Spider Monkey Optimization (GDSMO) approach can enhance Supervisory Control and Data Acquisition (SCADA) security application classification performance by optimising parameters from clustered datasets [14].

IDS and optimization have been tested on hybrid and intelligent architectures. In data pre-processing, feature selection, and time-based analysis, dynamic fuzzy logic sets improve IDS. [15]. Federated Learning Recommender Hybrid Intrusion Detection System (FRHIDS) can improve the security of SDN-based IoT by identifying threats and securing cloud communications between connected devices [16]. Optimal Deep Learning-based Intrusion Detection for Next-Generation Wireless Networks (ODLID-NGWN) uses LSTM-Gated Recurrent Units (GRU) to detect unauthorized access and malicious actions [17]. Anomaly-based intrusion detection examined residual autoencoder algorithms for better characterisation and reconstruction of abnormalities in various network contexts [18]. In addition, GWO has been used to classify incoming traffic and identify DDoS attacks within a wireless body sensor network [19]. Multiple-layer defence strategies employing deep reinforcement learning have also been examined to help improve the resilience of IoT against DDoS attacks [20].

Recent methods have several limitations despite their effectiveness. Many use only one optimization approach. Many are meant to improve classifying accuracy without considering feature selection quality, optimization/feature selection efficiency, or false positive detection reduction. These restrictions inspired the new framework, which employs several metaheuristic optimization methods and deep learning and machine learning classification algorithms to improve intrusion detection. Table I presents a comparative analysis between the proposed framework and existing state-of-the-art methods.

TABLE I
Comparative Analysis With Existing Works

Ref	Coop. Opt.	FPR Red.	Sequential Attack Det.	Multi-Dataset	Multi-Attack
[10]	✗	✗	✓	✓	✓
[11]	✗	✗	✗	✗	✓
[12]	✗	✓	✗	✗	✓
[13]	✗	✗	✗	✗	✓
[16]	✗	✓	✗	✓	✓
[17]	✗	✓	✓	✓	✓
This Work	✓	✓	✓	✓	✓

(Cooperative Optimization (Coop. Opt.), False Positive Rate Reduction (FPR Red.) and Sequential Attack (Seq. Attack))

III. Proposed Methodology

The proposed model is developed to cope with the complexity of modern IoT networks, in which large-scale, irregular, and heterogeneous traffic flows considerably increase the risk of cyber-attacks. Fig. 1 shows the proposed IoT-aware IDS framework's architectural design.

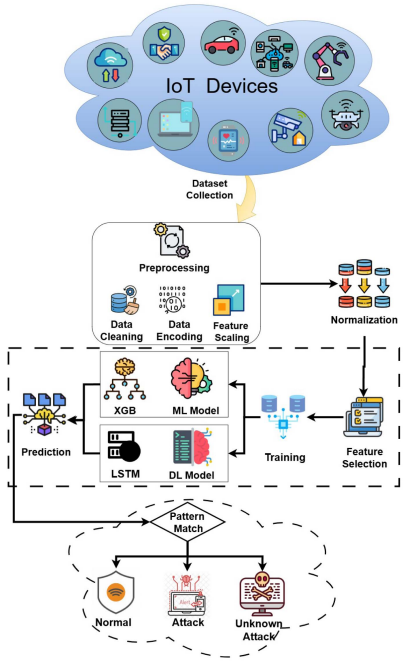


Fig. 1. The proposed IoT-aware Network Intrusion Detection System.

The hybrid integration of WOA, DTO, and ALO uses the effects to determine features. WOA's spiral search algorithm improves global exploration by finding candidate feature subsets in higher-dimensional domains. ALO increases local exploitation by using stochastic random walks, improving malicious traffic pattern discrimination. DTO models the search process as a cooperative dynamic, improving each search by updating adaptive solutions and reducing premature convergence. The hybrid combination of these three approaches gives the suggested IoT-aware IDS framework a better balance between exploration and exploitation for intrusion-related features. The proposed framework integrates data preparation, feature normalisation, optimal feature selection, and classifier training in a pipeline. Missing values are deleted and categorical attributes are encoded during data preprocessing. Before feature filtering, processed features are standardized to a uniform numerical scale. After the hybrid WOA-DTO-ALO algorithm filters out feature subsets without discriminatory power, the resultant feature subsets are transferred to either the LSTM network or the XGBoost classifier to analyze and capture their time dependencies and non-linear characteristics for real-time intrusion detection. IoT traffic attributes, such as protocol type, service type, and packet size, are processed through these stages to distinguish malicious behaviors from normal network activities. The overall performance of the framework is evaluated using accuracy, precision, recall, F1-score, MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and ROC-AUC (Receiver Operating Characteristic - Area Under Curve) based analysis.

To further emphasize discriminative attributes for intrusion detection, the feature selection objective is defined as

$$\Phi(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|\theta \odot \tilde{\mathbf{x}}_i - \boldsymbol{\mu}_\theta\|_2^2, \tag{1}$$

where $\tilde{\mathbf{x}}_i$ denotes the normalized feature vector of the i-th sample, θ is the feature selection or weighting vector, $\odot$ represents element-wise multiplication, and $\boldsymbol{\mu}_\theta$ denotes the mean vector of the weighted features. The objective of this study is to maintain informative variations of the features in the feature space being evaluated so as to improve the overall sensitivity of an IDS to anomalous traffic patterns.

The fitness function is intended to maximize optimization stability by reducing the number of redundant mutations in candidate feature representations during the search process. The overall intrusion detection accuracy of the ensemble learning stage in the next two steps (LSTM and XGBoost) reflects the complex inter-class decision boundary between normal and intrusion traffic; no decrease in class discriminability is guaranteed by the minimization goal.

To evaluate the characteristics of selected traffic features, we created a fitness function. A smaller standard deviation means that the features can better consistently distinguish between attack traffic and normal traffic, resulting in more reliable detection of anomalous traffic and lower false alerts. To identify which subset of features most accurately characterizes intrusion patterns, WOA was used. During the optimization process, the control parameter was updated so that there was both global exploration and local exploitation.

$$\alpha_t = 2 - 2\frac{t}{T}, \tag{2}$$

where t denotes the current iteration and T is the maximum number of iterations.

Early WOA experiments examine DoS, Probe, and scanning attack behaviors. The most discriminative attack indications are refined throughout optimization. Whale to the best global solution distance:

$$\mathbf{D}_w = \left|\mathbf{C}_t \odot \mathbf{X}_t^* - \mathbf{X}_{w,t}\right|. \tag{3}$$

where C_t denotes the coefficient vector controlling the search behavior, $X_{w,t}$ represents the position of the w-th whale at iteration t, and X_t^* is the current best solution. The degree to which a candidate's feature subset matches known adverse behaviors. Large distances indicate underperforming subsets, requiring correction. Update whale position:

$$\mathbf{X}_{w,t+1} = \mathbf{X}_t^* - \alpha_t \mathbf{D}_w. \tag{4}$$

Each whale identifies variables significantly connected with adverse traffic to determine the most intrusion-representative solution. ALO focuses on the exploitation of attack-specific patterns, such as packet intervals or abnormal traffic patterns.

$$\mathbf{R}(t) = \int_0^t \eta \, \mathrm{d}t, \qquad \eta \sim \mathcal{N}(0, \sigma^2). \tag{5}$$

The optimizer can examine multiple threat signatures by simulating unpredictable attacker behavior like DDoS.

$$\mathbf{X}_{A,t+1} = \frac{\mathbf{R}(t) + \mathbf{X}_t^*}{2}. \tag{6}$$

Exploration and advanced instruction help ALO distinguish between typical traffic and malicious attacks like R2L or U2R

(User to Root). DTO models coordinated attacks like distributed probing and multi-stage exploitation. Typical position change:

$$\beta_{nd}(t+1) = \beta_{best}(t) - \gamma_1 \left|\gamma_2 \beta_{best}(t) - \beta_{nd}(t)\right|. \quad (7)$$

This resembles attackers consolidating on weak nodes. DTO learns convergence patterns to improve attack detection feature relevance. Performance update:

$$\beta_s(t+1) = \gamma_3 \beta_s(t) + \gamma_4 \rho_1 (\beta_{best}(t) - \beta_{nd}(t)) + \gamma_5 \rho_1 (\beta_{Gbest} - \beta_{nd}(t)) \quad (8)$$

The parameters $\gamma_1, \gamma_2, \gamma_3, \gamma_4$, and γ_5 are control coefficients governing the exploration and exploitation balance in the DTO algorithm, while ρ_1 is a uniformly distributed random variable in [0,1]. β_{nd} and β_{best} denote the current and best candidate solutions, respectively, and β_{Gbest} represents the global best solution. Υ_t indicates the DTO population at iteration t. The equation shows dynamic attacker behavior using local exploits (e.g., repetitive probing) and global methods (e.g., DDoS locations). Updated position:

$$\beta_{nd}(t+1) = \beta_{nd}(t) + \beta_s(t+1). \quad (9)$$

This guarantees that DTO adapts to capture micro- and macro-attack behaviors. Population merge as:

$$\Psi_t = \Omega_t \cup \Lambda_t \cup \Upsilon_t. \quad (10)$$

A dynamic search combines exploration (WOA), stochastic attacker-like behavior (ALO), and coordinated attack modeling. Ideal feature masking agents:

$$\theta^* = \arg\min_{\psi \in \Psi_t} \Phi(\psi). \quad (11)$$

To reduce FPR and improve minority attack detection, the optimizer chooses features that reliably differentiate malicious patterns from benign traffic. Hidden LSTM dynamics:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t). \quad (12)$$

Recurrent Neural Networks (RNNs) determine sequence modelling; however, their potential to detect long-term dependencies is constrained by the vanishing gradient issue [21]. For intrusion detection, this study uses LSTM networks with long-range temporal dependencies. Keras builds an LSTM model with TensorFlow as the backend and three layers: input, hidden, and output. The proposed LSTM model incorporates a dropout approach to avoid overfitting and enhance generalisation [22]. A categorical cross-entropy loss using the Adam optimiser is used during training, and "fit()" sets the epochs and batch size. Sequential LSTM models capture temporal attack patterns like progressive scanning and low-rate DDoS better than static models. The LSTM-based output model is represented as follows:

$$\hat{\mathbf{y}}_{\text{LSTM}} = \text{softmax}(W_h \mathbf{h}_T + b_h), \quad (13)$$

where $\mathbf{h}_T$ denotes the hidden state at the final time step, and W_h and b_h are the trainable weight matrix and bias vector, respectively.

Updating instance labels and weights during analysis improves prediction accuracy. XGBoost, a gradient-boosting technique [23], integrates weak prior models into a strong model by training the next weak model with the errors of the preceding weak model and identifying wrong features until a specific accuracy is achieved. Students can learn temporal patterns for each class type to calculate class probabilities and detect attacks.

$$\hat{\Psi} = \sum_{\ell=1}^{L} f_\ell(\mathbf{X}^*). \quad (14)$$

In selected features like flag bit combinations or incursion packet counts, XGBoost captures complex nonlinear relations.

$$\hat{\mathbf{y}}_{\text{XGB}} = \text{softmax}(W_x \hat{\Psi} + b_x). \quad (15)$$

Real-time IDS environments can use XGBoost's fast, accurate classification. Classifier output combined:

$$\hat{\mathbf{y}} = \lambda \hat{\mathbf{y}}_{\text{LSTM}} + (1-\lambda)\hat{\mathbf{y}}_{\text{XGB}}. \quad (16)$$

Equal weighting was adopted for classifier fusion, where $\lambda = 0.5$ was assigned to both the LSTM and XGBoost components during the final prediction stage.

LSTM and XGBoost's output C, where C is the set of possible output classes, is combined using the fusion weight parameter λ, which ranges from [0,1]. Fusion can identify periodic events like DDoS attacks and non-periodic events like U2R attacks using the LSTM's temporal awareness and XGBoost's tabular processing. According to the last attacking incident's input class, these attacks were identified:

$$\hat{\mathbf{y}} = \arg\max_{c \in \mathcal{C}} \hat{y}_c. \quad (17)$$

The system receives typical, known, and unknown attacks to manage zero-day threats broadly. Improving discriminative features, capturing damaging temporal patterns, and using high-performance classifiers for IoT-scale systems would improve IDSs in the hybrid model.

Since the dataset had many attributes, the highest-value ones will be chosen to improve detection accuracy and reduce false alerts. The model should be validated using data not included in the development process, so two sets of data were created: one for training and one for testing. The $sklearn.model\_selection$ function $train\_test\_split$ created both sets. 20 percent of the original dataset was in the testing set and 80 percent in the training set. Selecting a random seed for the divide is optional. The generalizability of this model is assessed by this function. The feature set with the lowest fitness function value is ideal. Optimal feature selection set:

$$\theta^* = \arg\min_{\psi \in \Psi_t} \Phi(\psi). \quad (18)$$

During the preprocessing stage, feature-wise standardization is applied to all attributes that represent heterogeneous types of traffic as well as to avoid the situation where higher magnitude features dominate the lower magnitude features. For a given dataset consisting of N samples with d features, the normalized value of the j^{th} feature of the i^{th} sample will be computed as:

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \quad (19)$$

Algorithm 1: Hybrid WOA–DTO–ALO.

function Preprocess_DataΔ
  Remove missing values from Δ
  Encode categorical attributes using label encoding
  Extract features $\mathbf{X}$ and labels γ
  $\mathbf{X} \leftarrow \frac{\mathbf{X}-\mu(\mathbf{X})}{\sigma(\mathbf{X})}$
  return $\mathbf{X}, \gamma$
end function
function Fitness_Function$\theta, \mathbf{X}$
  $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \odot \theta$
  $\Phi(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\theta_i - \mu_i)^2$
  return $\Phi(\theta)$
end function
function Initialize_Populationν, δ
  $\Omega, \Theta, \Lambda_A \sim \mathcal{U}(\varsigma_{\min}, \varsigma_{\max})^{\nu \times \delta}$
  return $\Omega, \Theta, \Lambda_A$
end function
function Hybrid_Optimization$\Delta, \tau, \nu, \delta$
  $(\mathbf{X}, \gamma) \leftarrow$ Preprocess_Data(Δ)
  $(\Omega, \Theta, \Lambda_A) \leftarrow$ Initialize_Population(ν, δ)
  Evaluate fitness and determine ω^*, λ^*
  for$t = 1$ to τ **do**
    $\alpha_t = 2 - \frac{2t}{\tau}$
    for each whale $\omega_j \in \Omega$ **do**
    $\omega_j \leftarrow \omega^* - \alpha_t|\mathbf{C}_j\omega^* - \omega_j|$
    end for
    for each DTO candidate $\theta_j \in \Theta$ **do**
      $\theta_j(t+1) \leftarrow \theta_j(t) + \beta(\omega^* - \theta_j(t))$
    end for
    for each ant $\alpha_j \in \Lambda_A$ **do**
      $\alpha_j(t+1) \leftarrow \frac{\alpha_j(t)+\eta\lambda^*}{2}$
    end for
    Clip all candidate solutions within $[\varsigma_{\min}, \varsigma_{\max}]$
    Update fitness values and best solutions
    $\Psi \leftarrow \Omega \cup \Theta \cup \Lambda_A$
  end for
  return ω^*, λ^*
end function

where μ_j and σ_j represent the mean and standard deviation of the j^{th} feature respectively and are defined as:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j}, \quad \sigma_j = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(x_{i,j} - \mu_j)^2}. \tag{20}$$

This normalization provides scale invariance for all features, enhances numerical stability and increases the ability of the model to identify small differences in normal traffic as opposed to malicious traffic.

IV. Performance Evaluation

A. Experimental Setup

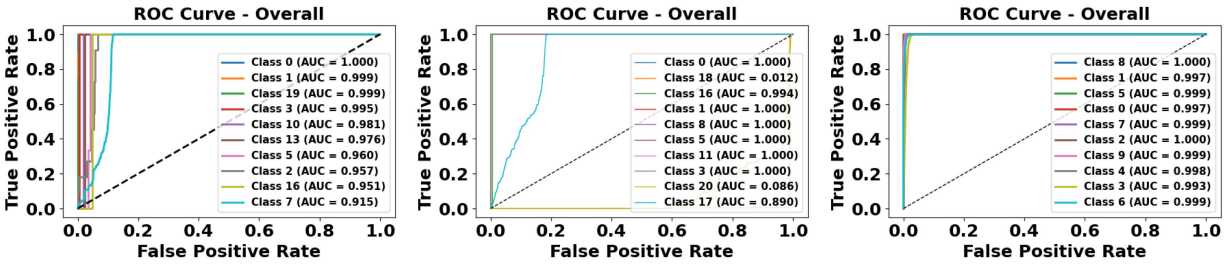
The experiments were implemented in Python and conducted in the Google Colab environment with GPU acceleration (Tesla T4/P100). The proposed models were developed using Keras 2.13.1. Network traffic data were processed using 5-s sliding windows, and 50 iterations were performed for each window to support feature selection and model training. The proposed framework was evaluated on four benchmark intrusion detection datasets under different attack categories and traffic conditions: NSL-KDD Train[1] (125,974 flows), KDDCup99[2] (494,021 flows), CICDDoS2019[3] (431,372 flows), and NF-UNSW-NB15-v2[4] (1,048,576 flows). NSL-KDD is an improved version of KDDCup99 and contains 42 features for labeling network connections as normal or as attack classes such as U2R, DoS, Probe, and R2L. KDDCup99, derived from the DARPA 1998 evaluation, is one of the earliest benchmark datasets for intrusion detection. CICDDoS2019 has real traffic from modern DDoS attack types and over 80 flow features to evaluate detection performance against volumetric, protocol-based, and application-layer attacks. NF-UNSW-NB15-v2 is a NetFlow-based version of UNSW-NB15 with 45 features covering both normal traffic and multiple attack categories, providing a realistic benchmark for large-scale network intrusion detection. Using the `train_test_split` function in the `sklearn.model_selection` library, the feature subsets were split into training and testing sets, with 80% of samples utilised for training and 20% for testing. Each experiment used the same preprocessing pipeline and data split strategy to increase consistency and fairness. A fixed random seed was used for all training and evaluation. To reduce implementation bias in results, all hyperparameters were kept constant across datasets unless otherwise stated.

Before model training and evaluation, each benchmark dataset was first split into training and test sets at an 80%-20% ratio using stratified sampling to preserve the class distribution. To prevent data leakage, the sliding-window-based preprocessing was then performed independently on the training and test sets, ensuring that no traffic segments overlapped across the two partitions. The LSTM model consists of two stacked LSTM layers, each with 50 hidden units, followed by dropout layers with a dropout rate of 0.2 to mitigate overfitting. The model was trained for 10 epochs using the Adam optimizer with a batch size of 32. For the XGBoost classifier, multiclass log loss (mlogloss) was used as the evaluation metric, while the remaining hyperparameters were kept at their default settings.

B. Performance Evaluation

The proposed IDS will be compared with existing approaches using classification measures like Precision, Recall, F1-score,

[1]https://www.kaggle.com/datasets/kiranmahesh/nslkdd
[2]https://www.kaggle.com/datasets/venkatakanumuru/kddcup99csv
[3]https://www.unb.ca/cic/datasets/ddos-2019.html
[4]https://www.kaggle.com/datasets/sankurisrinath/nf-unsw-nb15-v2csv



(a) NSL-KDD dataset. (b) KDDCup99 dataset (c) NF-UNSW-NB-15-v2 dataset

Fig. 2. ROC Comparison on various datasets.

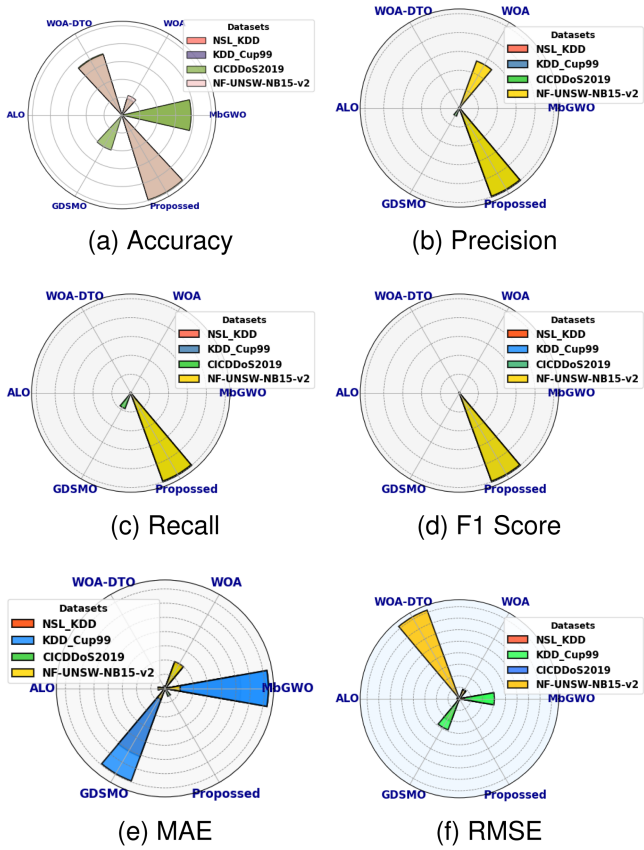
TABLE II
TRAINING, TESTING LOSS AND COMPUTATIONAL ANALYSIS

Dataset	Train Loss	Test Loss	Selection Time(s)	Train Time(s)
NSL-KDD	0.0048	0.0061	12.4	85.2
KDDCup99	0.0021	0.0038	10.7	79.5
CICDDoS2019	0.0015	0.0027	14.1	91.8
NF-UNSW-NB15-v2	0.0052	0.0074	16.3	97.2

MSE, MAE, RMSE, and Accuracy. These measures evaluate detection efficacy from several perspectives, with accuracy reducing false alarms and recall identifying real intrusion occurrences. The F1 Score balances precision and recall, making it optimal for 'imbalanced' attack data. Accuracy evaluates the overall correctness of classification, whereas MAE and RMSE quantify prediction errors and provide additional insight into model robustness. Together, these metrics enable a systematic evaluation of the proposed framework in terms of accuracy, reliability, and robustness for IoT intrusion detection.

C. Results & Discussion

The presented framework was evaluated on benchmark real-world datasets like NSL-KDD, CICDDoS2019, NF-UNSW-NB15-v2, and KDDCup99. The integration of LSTM and XGBoost enhances feature classification performance for intrusion detection in IoT networks. The proposed framework outperforms in evaluation and minimizes the training and testing loss metrics in all four real-world datasets shows in Table II. The proposed framework shows the optimal training loss of 0.0048 and a testing loss of 0.0061 on the NSL-KDD dataset. The loss values evaluated on the CICDDoS2019 dataset are 0.0015 during training and 0.0027 during testing, showing minimum overfitting. The computational analysis of the proposed model shows the optimal integration to maintain the balance between feature selection and model training in Table II. The proposed method effectively detects relevant features before training on the NSL-KDD dataset. Feature selection requires 12.4 seconds, whereas model training necessitates 85.2 seconds to detect the intrusions in the IoT network. The evaluation done on the NF-UNSW-NB15-v2 dataset takes 97.2 s for training and 16.3 s for feature selection, showing the complexity of the dataset. The proposed framework demonstrates effective detection accuracy across various attack patterns in network traffic. The proposed model shows intrusion detection rates of 99.85% on NSL-KDD,



(a) Accuracy (b) Precision (c) Recall (d) F1 Score (e) MAE (f) RMSE

Fig. 3. Performance analysis on various datasets.

99.92% on KDDCup99, and 99.99% on CICDDoS2019 for DDoS attacks. The proposed framework provides an optimal accuracy of 98.97% on NSL-KDD and 99.21% on KDDCup99, for the U2R attacks. As shown in Fig. 2, the proposed framework exhibits superior ROC characteristics, achieving high true positive rates while maintaining low false positive rates across all benchmark datasets. The performance analysis is shown in Fig 3. Furthermore, Fig. 4 provides a heatmap-based comparative analysis, demonstrating consistent performance gains of the proposed method over existing approaches. The experimental results show that the proposed framework effectively detects sequential attack patterns and improves feature selection, making it highly useful for IoT-based network intrusion detection shows in Table IV.

TABLE III
PERFORMANCE COMPARISON OF PROPOSED WORK WITH BASELINES

Dataset	Methods	Precision	Recall	F1-Score	MAE	MSE	RMSE	Accuracy
NSL-KDD	ANN [24]	96.50%	95.18%	92.14%	0.0416	0.0017	0.0421	79.45%
	RNN [25]	95.65%	87.62%	93.48%	0.0854	0.0073	0.0857	74.28%
	DNN [26]	96.52%	97.25%	96.87%	0.0312	0.0090	0.0312	82.16%
	SVM-CSA [27]	98.01%	98.56%	97.68%	0.0170	0.0029	0.0175	99.11%
	GWO-LOA-RF [28]	98.49%	98.61%	98.16%	0.0100	0.0017	0.0101	99.37%
	Proposed	**99.39%**	**99.41%**	**99.39%**	**0.00038**	**0.00319**	**0.00516**	**99.41%**
KDDCup99	ANN [24]	95.75%	94.82%	88.75%	0.0475	0.0022	0.0547	90.96%
	RNN [25]	93.37%	89.16%	87.45%	0.0878	0.0079	0.0869	91.37%
	DNN [26]	96.79%	90.42%	91.59%	0.0650	0.0047	0.0654	89.94%
	SVM-CSA [27]	97.70%	95.78%	93.48%	0.0332	0.00107	0.0333	94.67%
	GWO-LOA-RF [28]	98.07%	97.87%	95.23%	0.0210	0.0019	0.0216	97.63%
	Proposed	**99.93%**	**99.93%**	**99.93%**	**0.000052**	**0.00053**	**0.0021**	**99.93%**
CICDDoS2019	ANN [24]	96.40%	95.15%	92.09%	0.0425	0.00183	0.0425	79.32%
	RNN [25]	95.60%	87.59%	93.27%	0.0864	0.00716	0.0616	73.58%
	DNN [26]	96.50%	91.17%	95.87%	0.6370	0.00389	0.0625	81.33%
	SVM-CSA [27]	95.00%	99.00%	97.00%	0.0307	0.0016	0.0307	88.58%
	GWO-LOA-RF [28]	99.13%	99.14%	99.12%	0.0089	0.0008	0.0093	99.13%
	Proposed	**99.99%**	**99.99%**	**99.99%**	**0.0115**	**0.01159**	**0.000024**	**99.99%**
NF-UNSW-NB15-v2	ANN [24]	95.72%	96.42%	97.26%	0.0412	0.00165	0.0393	87.52%
	RNN [25]	93.35%	95.51%	96.74%	0.0559	0.00376	0.0567	75.27%
	DNN [26]	96.71%	96.71%	96.23%	0.0343	0.00109	0.0341	77.16%
	SVM-CSA [27]	97.65%	97.12%	97.92%	0.0265	0.000071	0.0291	87.78%
	GWO-LOA-RF [28]	99.23%	98.57%	99.07%	0.0075	0.00089	0.0077	98.45%
	Proposed	**99.26%**	**99.15%**	**99.17%**	**0.00022**	**0.00081**	**0.00282**	**99.15%**

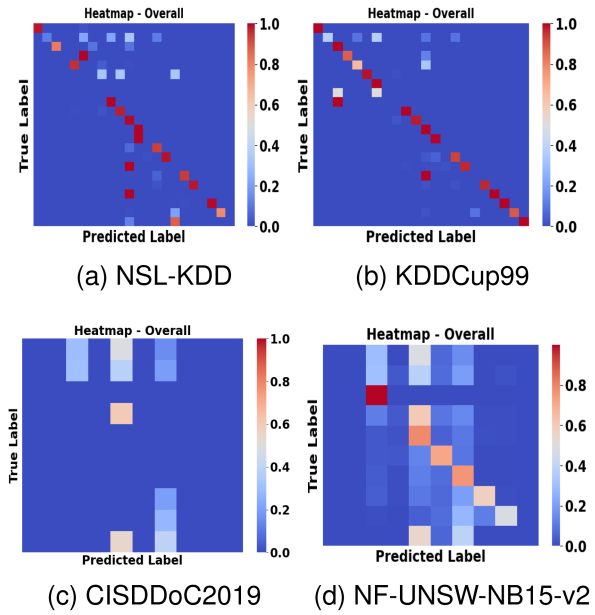


Fig. 4. Heatmap comparison on various datasets.

TABLE IV
DETECTION PERFORMANCE ACROSS ATTACK TYPES (MEAN ± STD. DEV.)

Attack	NSL-KDD	KDDCup99	CICDDoS2019	NF-UNSW
DoS	99.91 ± 0.04	99.96 ± 0.02	99.99 ± 0.01	99.12 ± 0.08
DDoS	99.85 ± 0.05	99.92 ± 0.03	99.99 ± 0.01	99.32 ± 0.07
Probe	99.63 ± 0.07	99.81 ± 0.05	99.95 ± 0.02	98.91 ± 0.11
R2L	99.32 ± 0.09	99.47 ± 0.06	99.90 ± 0.03	97.85 ± 0.14
U2R	98.97 ± 0.11	99.21 ± 0.08	99.89 ± 0.03	97.43 ± 0.16

Table III presents a comparative analysis between the proposed method and several state-of-the-art approaches, including Artificial Neural Network (ANN) [24], Recurrent Neural Network (RNN) [25], Deep Neural Network (DNN) [26], Crow Search Algorithm-based Support Vector Machine (SVM-CSA) [27], and a hybrid method that integrates Grey Wolf Optimization (GWO), Lion Optimization Algorithm (LOA), and Random Forest for feature selection (GWO-LOA-RF) [28]. The performance of these methods is evaluated using Precision, Recall, F1-score, MSE, MAE, RMSE, and Accuracy. ANN models have predefined feature representations and may struggle with high-dimensional IoT traffic data, reducing detection accuracy. RNN model sequential dependencies, however, vanishing gradients make it challenging to simulate complicated feature interactions in huge data sets. The scalability and performance of Support Vector Machine (SVM)-based algorithms in heterogeneous IoT environments depend on feature dimensionality and kernel function selection. The datasets used for testing the proposed approach are the NSL-KDD, KDDCup99, CICDDoS2019, and NF-UNSW-NB15-v2 benchmark datasets for evaluating the proposed method. The datasets considered in this study mainly represent benchmark network intrusion detection scenarios. Evaluation of the proposed framework under Industrial IoT communication environments and industrial protocol-specific traffic conditions remains an important direction for future investigation.

V. CONCLUSION

This study provides a hybrid IoT-Based intrusion detection framework that integrates WOA, DTO, and ALO for feature selection, and uses LSTM and XGBoost for anomaly detection. The proposed framework improves intrusion detection capabilities in IoT systems by utilizing multi-strategy feature optimization, temporal modeling, and effective classification. Experimental results on NSL-KDD, KDDCup99, CICDDoS2019, and NF-UNSW-NB15-v2 verified the effectiveness of the proposed

method. The framework achieved high detection accuracies of 99.41% on NSL-KDD, 99.93% on KDDCup99, 99.96% on UNSW-NB15, and 99.99% on CICDDoS2019 datasets and low false-positive rates across multiple attack categories while maintaining computational efficiency through feature dimensionality reduction. These results demonstrate that the proposed framework provides a reliable and practical solution for real-time IoT intrusion detection.

Future work will focus on improving model interpretability, reducing computational overhead, extending the framework to adaptive zero-day attack detection and explainable intrusion analysis, evaluating the framework under real-time IoT and edge deployment environments, and improving the computational efficiency for massive network traffic analysis.

REFERENCES

[1] M. D. Mauro, G. Galatro, and A. Liotta, "Experimental review of neural-based approaches for network intrusion management," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2480–2495, Dec. 2020.

[2] Z. U. I. Nasir, A. Iqbal, and H. K. Qureshi, "Securing cyber-physical systems: A decentralized framework for collaborative intrusion detection with privacy preservation," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 2, pp. 303–311, 2024.

[3] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly detection in smart home operation from user behaviors and home conditions," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 183–192, May 2020.

[4] D. Javeed, M. S. Saeed, I. Ahmad, P. Kumar, A. Jolfaei, and M. Tahir, "An intelligent intrusion detection system for smart consumer electronics network," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 906–913, Nov. 2023.

[5] V. Jain, A. Gupta, P. Verma, and S. S. Gill, "TinyFed6G: Federated learning with TinyML for resource-constrained intelligence in 6G edge networks," *IEEE Wireless Commun.*, vol. 33, no. 2, pp. 25–31, Apr. 2026.

[6] Y. Yao, Z. Zhang, K. Zhao, P. Wang, and R. Wu, "NGMO: A novel geometric mean optimizer for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 3, pp. 296–308, 2025.

[7] J. Ahmad et al., "An interpretable deep learning framework for intrusion detection in industrial Internet of Things," *Internet Things*, vol. 33, 2025, Art. no. 101681.

[8] M. Bakro et al., "An improved design for a cloud intrusion detection system using hybrid features selection approach with ML classifier," *IEEE Access*, vol. 11, pp. 64228–64247, 2023.

[9] B. Naik, M. S. Obaidat, J. Nayak, D. Pelusi, P. Vijayakumar, and S. H. Islam, "Intelligent secure ecosystem based on metaheuristic and functional link neural network for edge of things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1947–1956, Mar. 2020.

[10] N. Dash, S. Chakravarty, A. K. Rath, N. C. Giri, K. M. AboRas, and N. Gowtham, "An optimized LSTM-based deep learning model for anomaly network intrusion detection," *Sci. Rep.*, vol. 15, no. 1, 2025, Art. no. 1554.

[11] V. Kumar, K. Kumar, M. Singh, and N. Kumar, "NIDS-DA: Detecting functionally preserved adversarial examples for network intrusion detection system using deep autoencoders," *Expert Syst. Appl.*, vol. 270, 2025, Art. no. 126513.

[12] Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization," *Expert Syst. Appl.*, vol. 204, 2022, Art. no. 117597.

[13] D. S. Khafaga et al., "Voting classifier and metaheuristic optimization for network intrusion detection," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 3183–3198, 2023.

[14] A. O. Khadidos, H. Manoharan, S. Selvarajan, A. O. Khadidos, K. H. Alyoubi, and A. Yafoz, "A classy multifacet clustering and fused optimization based classification methodologies for scada security," *Energies*, vol. 15, no. 10, 2022, Art. no. 3624.

[15] J. Xie, H. Wang, J. M. Garibaldi, and D. Wu, "Network intrusion detection based on dynamic intuitionistic fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 9, pp. 3460–3472, Sep. 2022.

[16] H. Babbar and S. Rani, "FRHIDS: Federated learning recommender hybrid intrusion detection system model in software-defined networking for consumer devices," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2492–2499, Feb. 2024.

[17] E. B. Ashary et al., "Enhancing resilience in next-generation wireless networks through deep learning for security enhancement," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 6284–6292, Aug. 2024.

[18] N. Ijaz, S. U. Jan, and I. Koo, "A robust residual autoencoder framework for anomaly-based network intrusion detection," in *Proc. 7th Int. Conf. Advancements Comput. Sci.*, 2026, pp. 1–6.

[19] A. P. Muniyandi et al., "Intelligent security system for preventing DDoS attacks for 6G enabled WBSN using improve grey wolf optimization," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 5775–5782, Aug. 2024.

[20] Y. Liu, K.-F. Tsang, C. K. Wu, Y. Wei, H. Wang, and H. Zhu, "IEEE P2668-compliant multi-layer IoT-DDoS defense system using deep reinforcement learning," *IEEE Trans. Consum. Electron.*, vol. 69, no. 1, pp. 49–64, Feb. 2023.

[21] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization," in *Proc. Int. Conf. Platform Technol. Serv.*, 2017, pp. 1–6.

[22] S. A. Althubiti, E. M. Jones, and K. Roy, "LSTM for anomaly-based network intrusion detection," in *Proc. 28th Int. Telecommunication Netw. Appl. Conf.*, 2018, pp. 1–3.

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[24] R. Gopi et al., "Enhanced method of ANN based model for detection of DDoS attacks on multimedia Internet of Things," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 26739–26757, 2022.

[25] R. SaiSindhuTheja and G. K. Shyam, "An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," *Appl. Soft Comput.*, vol. 100, 2021, Art. no. 106997.

[26] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12499–12514, 2020.

[27] J. Kumar Samriya, S. Kumar, M. Kumar, H. Wu, and S. Singh Gill, "Machine learning-based network intrusion detection optimization for cloud computing environments," *IEEE Trans. Consum. Electron.*, vol. 70, no. 4, pp. 7449–7460, Nov. 2024.

[28] A. Kumar and S. Kumar, "Metaheuristic-enhanced feature selection for high-accuracy intrusion detection in cloud computing," *Procedia Comput. Sci.*, vol. 259, pp. 640–649, 2025.

Surendra Kumar received the Ph.D. degree from Babasaheb Bhimrao Ambedkar University (A Central University), India. He is currently an Assistant Professor with GLA University, Mathura, India. He has coauthored various peer-reviewed papers published in leading journals and conferences. His research focuses on resource management, cloud security, cryptography, and distributed computing.

Avinash Kumar received the Master of Technology degree from GLA University Mathura, India. He has authored or coauthored in prominent international journals and conferences. His research interests include explainable AI, federated learning, social network and communication and AI.

Mridula Dwivedi is currently working toward the Ph.D. degree with the Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India. She has authored or coauthored in prominent international journals and conferences. Her research interests include explainable AI, federated learning, social network and communication and AI.

Mohit Kumar received the Ph.D. degree in cloud computing from the Indian Institute of Technology Roorkee in 2018. He is currently an Assistant Professor with the Department of Information Technology, Dr. B R Ambedkar National Institute of Technology, Jalandhar, India. His research focuses on the areas of cloud computing, fog/ edge computing, Internet of Things, federated learning, blockchain, and AI.

Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, China in 2009 and 2011, respectively, and the Ph.D. degree with highest honor in computer science from Freie Universität Berlin, Germany, in 2015. He is currently a Professor with the Center for Applied Mathematics, Tianjin University, Tianjin, China. His research interests include mobile cloud computing, edge computing, Internet of Things, deep learning, complex networks, and DNA storage.

Sukhpal Singh Gill is currently an Assistant Professor with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. His research interests include cloud computing, edge computing, and energy efficiency. He is an Editor-in-Chief of IGI Global IJAEC, Executive Editor of Elsevier IoT Journal and Area Editor of Springer Cluster Computing Journal, also an Associate Editor for IEEE INTERNET OF THINGS JOURNAL, *Scientific Reports* (Nature), Wiley SPE Journal, Wiley ETT Journal, Elsevier Telematics and Informatics Reports and IET Networks Journal.