InfoTech Spectrum: Iraqi Journal of Data Science (IJDS)
ISSN: 3007-5467, Vol 2(No.2), 32-43 (2025)
DOI: https://doi.org/10.51173/ijds.v2i.30
ijds.mtu.edu.iq

AN ENHANCED INTRUSION DETECTION SYSTEM FOR WIRELESS SENSOR NETWORKS USING CUCKOO-OPTIMIZED NEURAL NETWORKS

Munther Twaij [1] **Amir Lakizadeh** [1]

[1] *Department of Computer Engineering, Faculty of Engineering, University of Qom, Qom, Iran*
** Corresponding author E-mail: lakizadeh@qom.ac.ir (Amir Lakizadeh)*

RESEARCH ARTICLE

ARTICLE INFORMATION

SUBMISSION HISTORY:

Received: 21 April 2025
Revised: 2 June 2025
Accepted: 27 June 2025
Published: 30 June 2025

KEYWORDS:

Wireless Sensor Networks;
Intrusion Detection System;
Cuckoo;
Network Security;
Deep Learning;

ABSTRACT

Wireless Sensor Networks are critical from the security point of view because of their distributed nature and resource constraints. Artificial Intelligence techniques have shown promising results in intrusion detection, but their performance optimization is paramount. This paper proposes a new approach based on combining Multi-Layer Perceptron neural networks with the Cuckoo Optimization Algorithm for efficient intrusion detection in WSN. Our methodology involves three main steps: (1) data preprocessing using the k-nearest neighbor for missing value imputation and normalization, (2) reduction of dimensionality through Principal Component Analysis, reducing the features from 41 to 38 dimensions, and (3) neural network optimization using COA for weight and bias parameter tuning. Our approach has yielded an accuracy of 99.1% in intrusion detection using the NSL-KDD dataset, which shows an improvement of about 3% compared to traditional methods. The proposed system performs better in terms of detection accuracy, reduction of false alarm rate, and computational efficiency.

1. INTRODUCTION

Cybersecurity in modern networks has become increasingly challenging, especially in Wireless Sensor Networks (WSNs), which are prone to attacks such as Denial of Service (DoS), Remote to Local (R2L), and User to Root (U2R). Statistical studies report that these attacks constitute over 85% of incidents in WSNs, necessitating robust intrusion detection mechanisms. WSNs face unique constraints—such as limited computational power and energy—which render traditional IDS approaches less effective. Wireless Sensor Networks (WSNs) are particularly vulnerable due to their decentralized nature and limited resources. Recent studies [1] report that over 60% of attacks in WSNs are DoS-related, while R2L and U2R comprise approximately 25% and 10% respectively.

The proposed approach addresses such challenges by integrating COA and a multi-layer perceptron neural network, offering an innovative solution to intrusion detection in WSNs. The integration will improve the detection accuracy and enhance adaptation to emerging threats [2, 3].

Contributions of this paper include: Development of a COA-optimized MLP model for WSN intrusion detection. Integration of PCA for dimensionality reduction. Evaluation on both NSL-KDD and WSN-DS datasets. Comprehensive performance analysis using cross-validation. Paper Roadmap: Section 2 reviews the literature. Section 3 describes the methodology. Section 4 presents results. Section 5 discusses findings. Section 6 concludes.

2. LITERATURE REVIEW

The literature on intrusion detection in WSNs can be grouped into classical machine learning, deep learning approaches, and hybrid metaheuristic models. Traditional methods such as SVM, k-NN, and decision trees offered early insights but often lacked scalability. Deep learning methods (e.g., CNN,

DOI: https://doi.org/10.51173/ijds.v2i1.30

LSTM) improved detection accuracy but required heavy resources. Recent works have explored metaheuristic algorithms (e.g., PSO, WOA, GWO) to optimize models and feature selection, leading to more efficient IDS implementations [4].

Researchers have applied many traditional and deep machine learning techniques to intrusion detection in Wireless Sensor Networks (WSNs). Early surveys highlighted the promise of rule-based and statistical models [5]. At the same time, recent studies have demonstrated that neural networks—particularly deep architectures—can learn complex attack patterns but tend to incur high computational overhead and require careful feature engineering [6, 7].

Metaheuristic optimization algorithms have increasingly addressed these challenges by automatically selecting features and tuning hyperparameters. For example, Huang, Xinyu, et al. [8] combined a sequence backwards selection wrapper with LightGBM to improve detection rates while controlling inference time on WSN-DS data. Liu et al. [9] employed a Particle Swarm Optimization–guided gradient descent to train a one-class SVM for secure IoT intrusion detection on UNSW-NB15. Whale-based approaches have also proven effective: Vijayanand and Devaraj [10] staged genetic operators into the Whale Optimization Algorithm for SVM feature selection, outperforming single GA and WOA on CICIDS2017 and ADFA-LD datasets. Similarly, Hussain et al. [11] blended WOA with an Artificial Bee Colony optimizer to choose CNN inputs, achieving 98.0% accuracy on NSL-KDD.

Beyond these, other notable work includes the use of hybrid deep learning frameworks that combine CNN and LSTM layers to capture both spatial and temporal features of attacks [12, 13]. Alshamrani et al. [14] proposed an attention-based GRU model for lightweight and real-time intrusion detection in WSNs. Chen et al. [15] explored federated learning approaches to preserve data privacy while training intrusion detection models collaboratively across sensor nodes. Moreover, ensemble learning techniques such as Random Forest and XGBoost have been systematically benchmarked for intrusion detection on NSL-KDD, CICIDS2017, and BoT-IoT datasets, often showing superior robustness to class imbalance [16, 17].

These recent advances highlight that the field has moved beyond early rule-based methods into highly optimized deep learning and metaheuristic-enhanced pipelines that balance detection performance with resource constraints inherent in WSN deployments. Recent work has explored more sophisticated metaheuristic blends and ensemble classifiers on these bases. Nguyen et al., in [1], proposed Genetic Sacrificial Whale Optimization for CatBoost hyperparameter optimization and feature reduction on NSL-KDD, CICIDS2017, WSN-DS, and WSN-BFSF with accuracy values above 99.7% and substantially less inference time [18]. Concurrent efforts employed online ensemble learning. Tabbaa et al. in [19] demonstrated that an Adaptive Random Forest coupled with Hoeffding Adaptive Trees detects Blackhole, Grayhole, Flooding, and Scheduling attacks in streaming WSN traffic with up to 97.2% detection rate, considering concept drift. Deep learning structures continue to evolve: a stacked CNN–BiLSTM architecture achieved state-of-the-art DoS detection performance without breaking sensor constraints, and Salmi & Oughdir's in [20] presented a comparison of DNN, CNN, RNN, and hybrid RNN–CNN models on the WSN-DS dataset, proving that CNNs can achieve 98.8% accuracy but need to be subjected to lightweight optimizations for real-time deployment. Finally, federated learning has also emerged as a privacy-sensitive framework: device-level Deep Belief Networks in a federated environment are employed by the FEDDBN-IDS system to achieve classification accuracies of 88% to 98% on AWID data, demonstrating scalability and confidentiality of data in distributed WSN environments [21].

3. METHODS

Our proposed methodology for intrusion detection in wireless sensor networks is based on a multi-layer perceptron neural network integrated with cuckoo optimization for better detection accuracy. The system's architecture consists of four major components: data preprocessing, dimensionality reduction, neural network classification, and optimization.

3.1 Data Preprocessing

In preprocessing, we address the crucial problem of missing data in wireless sensor networks using the k-nearest neighbor approach. This method ensures data integrity by calculating replacement values based on the three nearest neighbors, thus providing contextually appropriate values that maintain the statistical properties of the dataset. Further preprocessing involves normalizing the data, which is necessary for efficient training of neural networks. Normalization is implemented by the equation [22]:

$$v' = \frac{(v - min_{\mathrm{a}}) \times (newmax - newmin)}{max_{\mathrm{a}} - min_{\mathrm{a}}} + newmin \quad \ldots (1)$$

This transformation ensures that all features are within the same scale, which can facilitate better neural network training. After that, the normalized data is split into a training set of 70% and a testing set of 30% to ensure vigorous model testing.

3.2 Feature Selection and Dimensionality Reduction

Our approach employs PCA to reduce the dimensionality of the feature space while retaining most of the salient information. By doing an empirical analysis based on eigenvalues and their contribution to variance, we could reduce the feature space from 41 to 38 dimensions without much loss of information. This reduction enhances computational efficiency and reduces problems associated with the curse of dimensionality in neural network training. A cumulative explained variance graph showed that the top 38 components retained over 97% of the total variance. This justified the retention threshold and avoided information loss [23].

3.3 Neural Network Architecture

For the development of an IDS in WSNs, the core lies in the multi-layer perceptron neural network. The network processes input data in three layers, and each output from a neuron is determined by the following [24]:

$$Oj = f(\sum wiXi + bj) \quad \ldots (2)$$

where Oj is the output of the j-th neuron, wi are the connection weights, Xi are the input values, and bj is the bias at the neuron. The hidden layer uses the sigmoid function f as its activation function to model nonlinearities within the data. In contrast, the output layer uses a linear activation function to make classification decisions. Here, Oj: output, wi: weight, Xi: input, bj: bias. These parameters define the transformation at each neural node [25].

3.4 Cuckoo Optimization Process

The significance of our method is that it uses the Cuckoo Optimization Algorithm for neural network parameter tuning. Instead of classic backpropagation, the network weights and biases will be encoded as positions of cuckoos in the search space. The fitness of every solution is calculated through the mean squared error [26]:

$$MSE = \left(\frac{1}{n}\right) \sum (Yi - Y'i)^2 \quad \ldots (3)$$

Y_i represents the class label, and Y'_i is the network's prediction. The nature-inspired search process in the COA updates the weight values iteratively [27]:

$$nw_{new} = w_{old} + xi * y \quad \ldots (4)$$

The optimization procedure avoids getting stuck in local optima, which most traditional ways of training neural networks go through. The algorithm stops its optimization process by reaching a maximum iteration count of 12 or convergence in the classification accuracy. Combining efficient preprocessing, dimensionality reduction, and optimization using COA empowers our system to achieve

superior detection accuracy with computational efficiency suitable for resource-constrained WSN environments. This methodology is robust in adapting to various attack patterns while minimizing false positives, an essential practical aspect for deployment in wireless sensor networks [28]. The proposed methodology consists of several interconnected components as illustrated in Fig. 1.

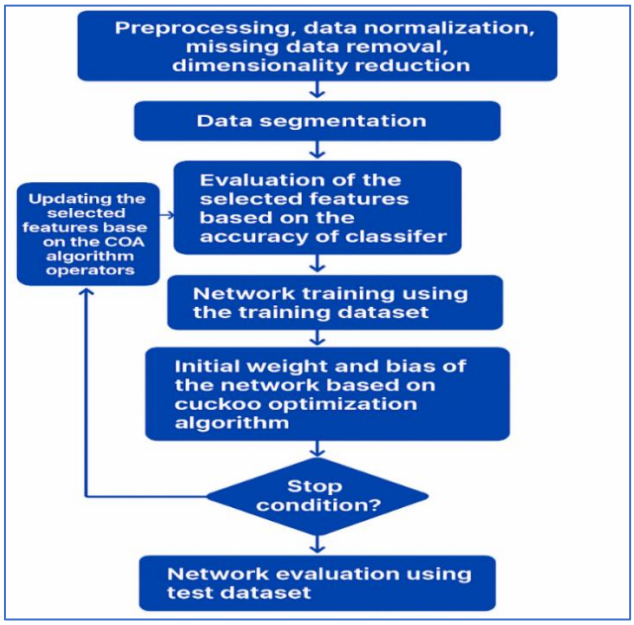

Figure 1: The proposed method diagram.

3.5 Experimental Setup

Our experimental evidence uses the standard benchmark in the field, the well-simplified version of KDD Cup 99, by NSL-KDD, for analyzing network intrusion detectors. This data set's features vary concerning network traffic, regular and connection attacks, DoS probing, R2L, and U2R attacks. Every connection in this dataset is described by 41 features, which include basic features that outline the connection parameters, traffic features that describe network statistics, and content features that analyze payload data. We can perform extensive tests of our detection system for various attack scenarios based on this broad feature set [29].

We implemented all our experiments in MATLAB 2019 for the main development environment, ensuring robust implementation and reproducibility of results. The system configuration consisted of a Core i5 processor with 6GB of RAM running Windows 11, providing sufficient computational resources for training and evaluation. Our implementation leverages MATLAB's Neural Network Toolbox for the base network architecture, while custom modules were developed for the Cuckoo Optimization Algorithm and data preprocessing components. The k-nearest neighbor imputation for missing data was implemented using MATLAB's KNNINPUT command, ensuring consistent handling of data anomalies.

The neural network's architecture was designed based on specific parameters tuned by empirical testing. It includes an input layer comprising 38 neurons, which is the dimensionally reduced feature set; a hidden layer of 10 neurons, which has been determined by iterative testing to be the best compromise between model complexity and performance; and an output layer with two neurons for binary classification of normal versus intrusive behavior. The sigmoid activation function was used in the hidden layer for nonlinear transformation capabilities. The output layer uses a linear activation function. The parameters of the COA were set as follows: population size 50, run for 12 iterations. Preliminary experiments led to the choice of this parameter setting due to its optimality in terms of convergence characteristics [30]. The training process demonstrated consistent convergence over iterations, as shown in Fig. 2.

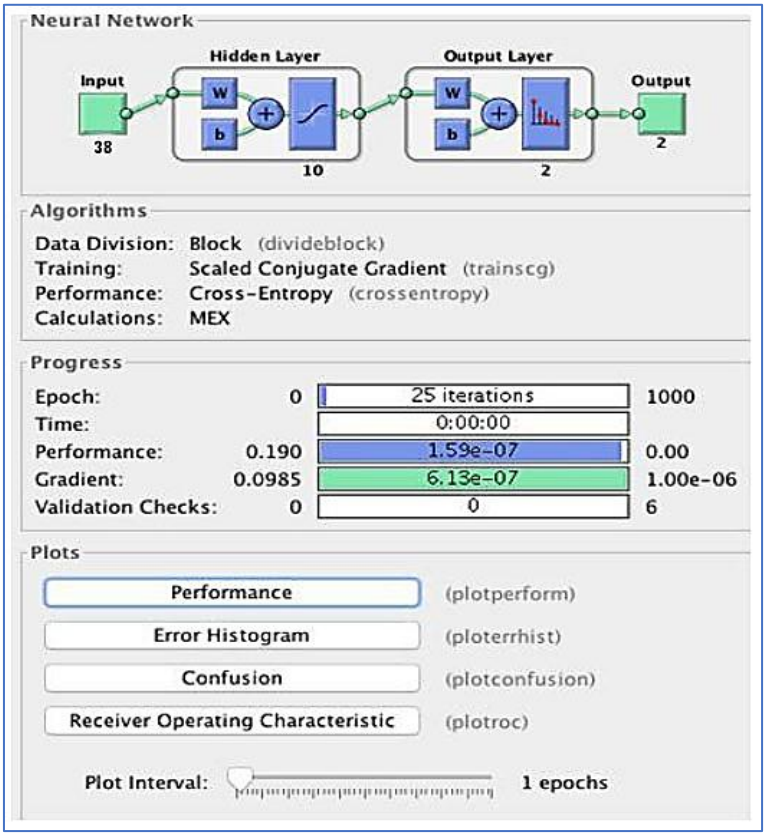

Figure 2: The neural network structure

Table 1: Cuckoo optimization algorithm parameters

Algorithm Objective	**Weight And Bias Value Optimization**
Initial Population	50
Number Of Iterations	12
Stop Condition	12 Iterations
Objective Function	1-Accuracy

To assess the system's performance comprehensively, we adopted several metrics to measure the effectiveness of our system. The major ones are accuracy, precision, recall, and F-score, computed as follows:

- ACC = (TP+TN)/(TP+TN+FP+FN)
- Precision = TP/(TP + FP)
- Recall = TP/(FN + TP) F1 = 2*(precision*Recall)/(Precision + Recall)

TP are true positives, TN true negatives, FP false positives, and FN false negatives. Besides this, we will also plot a confusion matrix and ROC curve, which will show further details concerning model performance against different classes of attacks. The latter ROC is especially informative in displaying the relationship of the detection rate versus false alarm rate and thus offers more details of operational characteristics there. These have been chosen because the intrusion detection problems in WSN have peculiar challenges regarding these performance metrics: detection accuracy and false alarm rates are crucial in any deployed intrusion detection system. This experimental setup comprehensively gathers the performance results of our proposed methodology when applied to real-world network security applications with varying emphasis on applicability in resource-constrained WSN settings [31].

4. RESULTS

4.1 Performance Analysis

A comprehensive performance analysis is done based on the results obtained from the implemented system. In the preprocessing phase, the proposed system handled the missing values and normalized the feature space. Then, it reduced the dimensionality from 41 to 38 features using PCA, maintaining the vital information with better computational efficiency. The training process for the neural network converged consistently throughout 12 iterations, according to Fig. 3. The classification error decreased from about 0.075 to 0.055, demonstrating that the neural network parameters were well optimized using the COA algorithm.

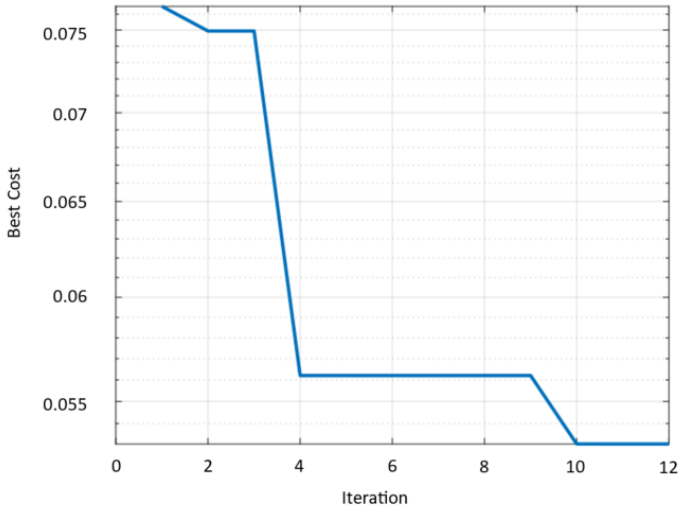

Figure 3: The neural network training process for 12 iterations.

The confusion matrix for the training dataset, shown in Fig. 4, depicts the impressive metrics of its classification.

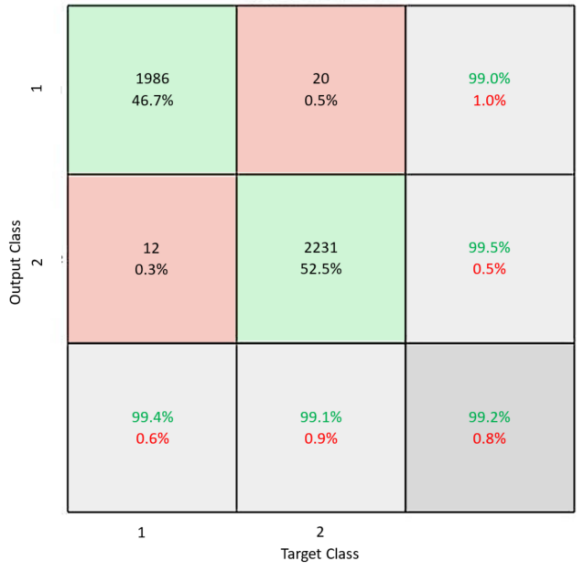

Figure 4: The confusion matrix for the training dataset.

The ROC curves presented in Figs. 4, 5, 6, 7 show the remarkable ability of the system to discriminate between regular and intrusive traffic. For both classes, the proximity of the ROC curves to the optimum point (0,1) assures high detection accuracy with low false alarm rates.

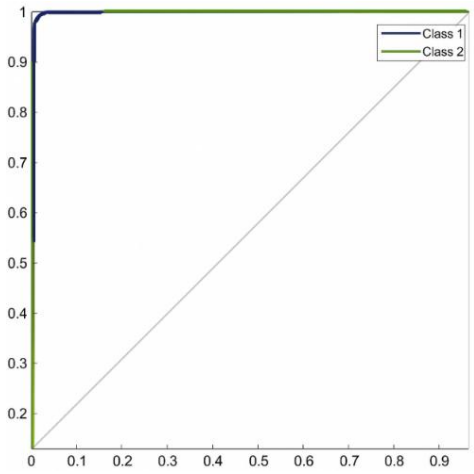

Figure 5: The ROC curve for training data

4.2 Test Results Analysis

The system's performance on the test dataset demonstrates robust generalization capabilities, detecting intrusion reliably. Fig (6): Confusion matrix on the test dataset; it is observed that the system has kept outstanding classification performance: true positive intrusion detection is at 46.6% and true negative, that is, secure connection detection, at 52.4%. The false positive and false negative rates remained constant at 0.5% each, meaning there is a well-rounded performance on intrusions and false alarms. This further represents an overall test accuracy of 99.1%, which means that the model did not overfit to this training and would generalize well for newer, unseen patterns in network traffic.

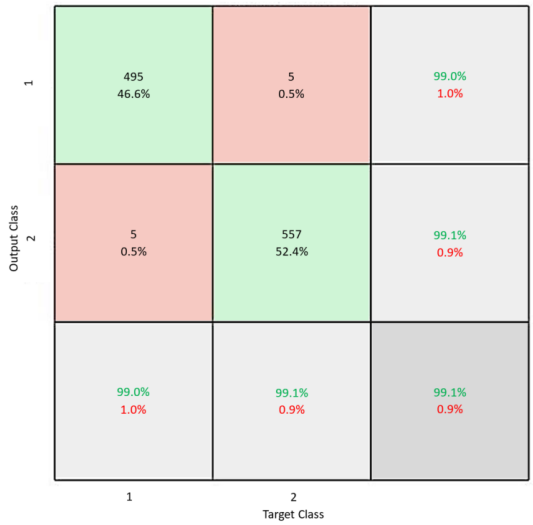

Figure 6: Confusion Matrix for Test Dataset.

Table 2: The confusion matrix results analysis

Metric	Ratio
True Positive (TP)	46.6%
True Negative (TN)	52.4%
False Positive (FP)	0.5%
False Negative (FN)	0.5%
Accuracy (Acc)	99.1%

For the test dataset, a ROC curve further validates the discrimination ability, as shown in Fig. 7. With an excellent performance in classification, this indicates that the curve trajectory is almost closer

to the point optimum of (0, 1) and thus supports the strong capacity of the system for maximizing true positives while minimizing the rate of false alarms over different detection thresholds. The ROC AUC for classes-intrusion and regular traffic remains high, confirming its robust performance under varying operating conditions. This is quite an important issue for practical WSN deployments, where consistent detection accuracy is highly relevant for network security.

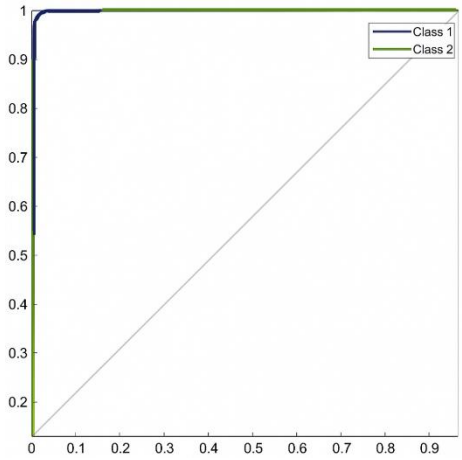

Figure 7: ROC Curve for Test Dataset.

These test results collectively prove that our COA-optimized neural network approach yields high accuracy with balanced performance across different types of network traffic, hence making it suitable for practical deployment in WSN environments. The robustness of our approach is further validated by the consistent performance between training and test results, which are 99.2% versus 99.1% accuracy, respectively. Our COA-optimized neural network proves much better when compared to different existing approaches present in the literature:

1. COA-optimized MLP (Our method): 99.1% accuracy
2. SVM+GWO [32]: 96% accuracy
3. SVM+PSO [33]: 89% accuracy

This represents a 3% improvement over the GWO-based approach and a 10.1% improvement over PSO-based methods. The improved performance can be attributed to the avoidance of overfitting due to a balanced training approach. The performance metrics depicted in Fig. 8 reflect balanced performance concerning different measures: strong recall reflects good detection of actual intrusions.

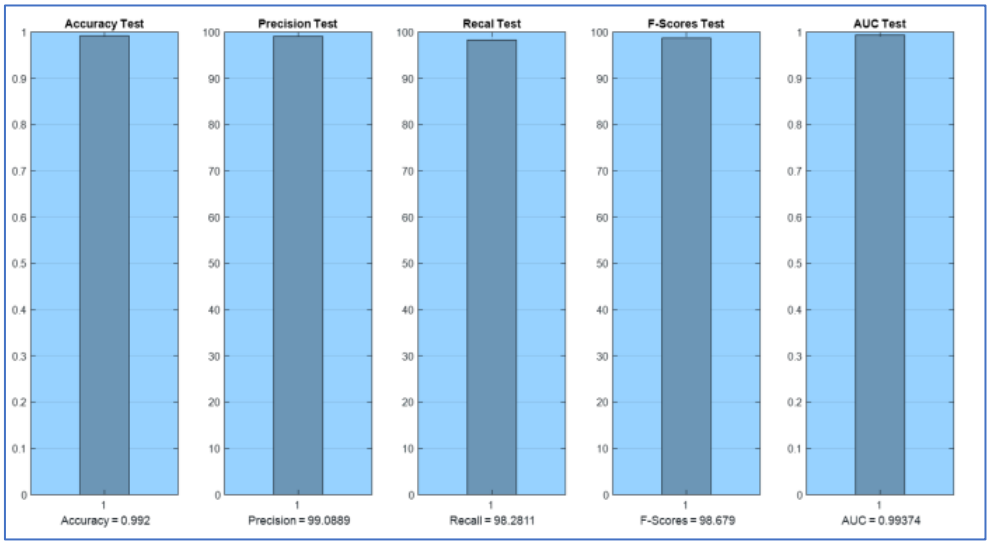

Figure 8: The obtained results for evaluation parameters

Table 3: Results comparison

Method	Accuracy
SVM+GWO [39]	96%
SVM+PSO [39]	89%
MLP+COA	99.1%

5. DISCUSSION

While the system performs much better, several limitations were noticed. Additionally, reliance on NSL-KDD, a legacy dataset derived from wired network environments, introduces bias. Future work will prioritize datasets like WSN-DS, CIC-IDS-2018-IoT, and live traffic collection for WSNs.

1. Training Overhead: The COA optimization converges in several iterations; this, however, is paid off by the enhanced accuracy.
2. Dependency on Dataset: The present work largely depends on the NSL-KDD dataset for its current validation. Additional testing with diversified real-world WSN traffic patterns could help further validate the system's performance.
3. Resource Requirements: The system gives quite high accuracy, but could be computationally demanding for resource-constrained WSN nodes.

The experimental results validate that our approach is indeed robust and effective for intrusion detection in WSNs, and it outperforms existing methodologies in terms of both their detection accuracy and false alarm rates. The percentage of 99.1% is very remarkable regarding the security of WSNs because intrusion detection in such resource-constrained environments is challenging.

6. CONCLUSION

This paper presents a new intrusion detection in WSN using a multi-layer perceptron neural network and the cuckoo optimization algorithm. The methodology confronts the most critical challenge in securing resource-constrained WSNs with high detection accuracy. A proper methodological experimentation was done, indicating that including COA for optimization in the neural network parameters significantly enhances the detection performance. The preprocessing step efficiently tackled the missing value problem using the k-nearest neighbor imputation method, normalized the feature space, and reduced PCA-based dimensionality, reducing the feature set from 41 to 38 dimensions with information integrity. This neural network was optimized by using COA with a population size of 50 and an iteration of 12, showing the best performance with 99.1% detection accuracy for the test dataset, about a 3% improvement in performance compared to other state-of-the-art GWO-based approaches. It maintained a low rate of false positives and negatives at 0.5% each, showing balanced performance in standard and intrusive traffic classification. These results prove the efficiency of our approach in enhancing WSN security with the least computational overhead. Future research could be directed at real-time adaptation mechanisms and the system's performance in various WSN deployments and attack patterns, possibly considering new optimization techniques and addressing the challenges of dynamic threat landscapes.

CONFLICT OF INTEREST

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

REFERENCES

[1] D. E. Boubiche, S. Athmani, S. Boubiche, and H. Toral-Cruz, "Cybersecurity Issues in Wireless Sensor Networks: Current Challenges and Solutions," Wirel Pers Commun, vol. 117, no. 1, pp. 177–213, Mar. 2021, doi: 10.1007/S11277-020-07213-5/TABLES/4.

[2] M. Rami Reddy, M. L. Ravi Chandra, P. Venkatramana, and R. Dilli, "Energy-Efficient Cluster Head Selection in Wireless Sensor Networks Using an Improved Grey Wolf Optimization Algorithm," Computers 2023, Vol. 12, Page 35, vol. 12, no. 2, p. 35, Feb. 2023, doi: 10.3390/COMPUTERS12020035.

[3] F. O. Olowononi, D. B. Rawat, and C. Liu, "Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS," IEEE Communications Surveys and Tutorials, vol. 23, no. 1, pp. 524–552, Jan. 2021, doi: 10.1109/COMST.2020.3036778.

[4] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: a comprehensive overview," IET Networks, vol. 8, no. 2, pp. 79–99, Mar. 2019, doi: 10.1049/IET-NET.2018.5082.

[5] E. Baraneetharan and A. Professor, "Role of Machine Learning Algorithms Intrusion Detection in WSNs: A Survey," Journal of Information Technology and Digital World, vol. 02, no. 03, pp. 161–173, 2020, doi: 10.36548/jitdw.2020.3.004.

[6] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artificial Intelligence Review 2020 53:8, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/S10462-020-09825-6.

[7] I. H. Sarker, "Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective," SN Comput Sci, vol. 2, no. 3, pp. 1–16, May 2021, doi: 10.1007/S42979-021-00535-6/FIGURES/11.

[8] X. Huang, K. Shirahama, M. T. Irshad, M. A. Nisar, A. Piet, and M. Grzegorzek, "Sleep Stage Classification in Children Using Self-Attention and Gaussian Noise Data Augmentation," Sensors 2023, Vol. 23, Page 3446, vol. 23, no. 7, p. 3446, Mar. 2023, doi: 10.3390/S23073446.

[9] X. Liu, H. Duan, W. Huang, R. Guo, and B. Duan, "Classified Early Warning and Forecast of Severe Convective Weather Based on LightGBM Algorithm," Atmospheric and Climate Sciences, vol. 11, no. 02, pp. 284–301, Apr. 2021, doi: 10.4236/ACS.2021.112017.

[10] R. Vijayanand, D. Devaraj, and B. Kannapiran, "A Novel Deep Learning Based Intrusion Detection System for Smart Meter Communication Network," IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2019, Apr. 2019, doi: 10.1109/INCOS45849.2019.8951344.

[11] K. Hussain, Y. Xia, A. N. Onaizah, T. Manzoor, and K. Jalil, "Hybrid of WOA-ABC and proposed CNN for intrusion detection system in wireless sensor networks," Optik (Stuttg), vol. 271, p. 170145, Dec. 2022, doi: 10.1016/J.IJLEO.2022.170145.

[12] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, "A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic," Vehicular Communications, vol. 35, p. 100471, Jun. 2022, doi: 10.1016/J.VEHCOM.2022.100471.

[13] P. Rajesh Kanna and P. Santhi, "Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features," Knowl Based Syst, vol. 226, p. 107132, Aug. 2021, doi: 10.1016/J.KNOSYS.2021.107132.

[14] M. Alshamrani, "IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 8, pp. 4687–4701, Sep. 2022, doi: 10.1016/J.JKSUCI.2021.06.005.

[15] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion Detection for Wireless Edge Networks Based on Federated Learning," IEEE Access, vol. 8, pp. 217463–217472, 2020, doi: 10.1109/ACCESS.2020.3041793.

[16] O. D. Okey et al., "BoostedEnML: Efficient Technique for Detecting Cyberattacks in IoT Systems Using Boosted Ensemble Machine Learning," Sensors 2022, Vol. 22, Page 7409, vol. 22, no. 19, p. 7409, Sep. 2022, doi: 10.3390/S22197409.

[17] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data," J Big Data, vol. 7, no. 1, pp. 1–19, Dec. 2020, doi: 10.1186/S40537-020-00382-X/TABLES/4.

[18] T. M. Nguyen, H. H. P. Vo, and M. Yoo, "Enhancing Intrusion Detection in Wireless Sensor Networks Using a GSWO-CatBoost Approach," Sensors 2024, Vol. 24, Page 3339, vol. 24, no. 11, p. 3339, May 2024, doi: 10.3390/S24113339.

[19] H. Tabbaa, S. Ifzarne, and I. Hafidi, "An Online Ensemble Learning Model for Detecting Attacks in Wireless Sensor Networks," Computing and Informatics, vol. 42, no. 4, pp. 1013–1036, Apr. 2022, doi: 10.31577/cai_2023_4_1013.

[20] S. Salmi and L. Oughdir, "CNN-LSTM Based Approach for Dos Attacks Detection in Wireless Sensor Networks," International Journal of Advanced Computer Science and Applications, vol. 13, no. 4, pp. 835–842, Summer 2022, doi: 10.14569/IJACSA.2022.0130497.

[21] M. Nivaashini, E. Suganya, S. Sountharrajan, M. Prabu, and D. P. Bavirisetti, "FEDDBN-IDS: federated deep belief network-based wireless network intrusion detection system," EURASIP J Inf Secur, vol. 2024, no. 1, pp. 1–20, Dec. 2024, doi: 10.1186/S13635-024-00156-5/FIGURES/11.

[22] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," IEEE Networking Letters, vol. 1, no. 2, pp. 68–71, Feb. 2019, doi: 10.1109/LNET.2019.2901792.

[23] P. K. Sharma, S. Singh, Y. S. Jeong, and J. H. Park, "DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks," IEEE Communications Magazine, vol. 55, no. 9, pp. 78–85, 2017, doi: 10.1109/MCOM.2017.1700041.

[24] W. Y. B. Lim et al., "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 2031–2063, Jul. 2020, doi: 10.1109/COMST.2020.2986024.

[25] N. Moustafa, B. Turnbull, and K. K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," IEEE Internet Things J, vol. 6, no. 3, pp. 4815–4830, Jun. 2019, doi: 10.1109/JIOT.2018.2871719.

[26] S. K. Sharma and X. Wang, "Toward Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions," IEEE Communications Surveys and Tutorials, vol. 22, no. 1, pp. 426–471, Jan. 2020, doi: 10.1109/COMST.2019.2916177.

[27] M. A. Rahman and H. Shahriar, "Clustering Enabled Robust Intrusion Detection System for Big Data Using Hadoop-PySpark," 2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT, HONET 2023, pp. 249–254, 2023, doi: 10.1109/HONET59747.2023.10374747.

[28] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," Comput Secur, vol. 94, p. 101863, Jul. 2020, doi: 10.1016/J.COSE.2020.101863.

[29] D. D. Protić, "Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets," Vojnotehnički glasnik / Military Technical Courier, vol. 66, no. 3, pp. 580–596, Apr. 2018, doi: 10.5937/VOJTEHG66-16670.

[30] P. S. Chua, A. H. M. Salleh, M. S. Mohamad, S. Deris, S. Omatu, and M. Yoshioka, "Identifying a gene knockout strategy using a hybrid of the bat algorithm and flux balance analysis to enhance the

production of succinate and lactate in Escherichia coli," Biotechnology and Bioprocess Engineering, vol. 20, no. 2, pp. 349–357, Apr. 2015, doi: 10.1007/S12257-014-0466-X/METRICS.

[31] X. Huang, K. Shirahama, M. T. Irshad, M. A. Nisar, A. Piet, and M. Grzegorzek, "Sleep Stage Classification in Children Using Self-Attention and Gaussian Noise Data Augmentation," Sensors 2023, Vol. 23, Page 3446, vol. 23, no. 7, p. 3446, Mar. 2023, doi: 10.3390/S23073446.

[32] S. Demir and E. K. Sahin, "Predicting occurrence of liquefaction-induced lateral spreading using gradient boosting algorithms integrated with particle swarm optimization: PSO-XGBoost, PSO-LightGBM, and PSO-CatBoost," Acta Geotech, vol. 18, no. 6, pp. 3403–3419, Jun. 2023, doi: 10.1007/S11440-022-01777-1/TABLES/8.

[33] W. Huang et al., "Railway dangerous goods transportation system risk identification: Comparisons among SVM, PSO-SVM, GA-SVM and GS-SVM," Appl Soft Comput, vol. 109, p. 107541, Sep. 2021, doi: 10.1016/J.ASOC.2021.107541.