A Deep Learning–Enhanced Blockchain Architecture for Intrusion Detection and Classification

C. Ananth
Department of Computer and Information Science, Annamalai University, Annamalainagar, Tamil Nadu, India
ananth.prog@gmail.com

S. Sathiyarani
Department of Computer and Information Science, Annamalai University, Annamalainagar, Tamil Nadu, India
sathiyaranis86@gmail.com (corresponding author)

N. Mohananthini
Department of Electrical and Electronics Engineering, Muthayammal Engineering College, Rasipuram, Tamil Nadu, India
mohananthini@yahoo.co.in

Received: 13 October 2025 | Revised: 28 October 2025, 10 November 2025, and 27 November 2025 | Accepted: 29 November 2025

 DOI: https://doi.org/10.48084/etasr.15481

ABSTRACT

An Intrusion Detection System (IDS) is a significant cybersecurity process that comprises network traffic monitoring for malicious activity and taking appropriate protective actions. However, inadequate training data or inappropriately selected thresholds often restrict the performance of these systems, leading to low detection rates. Blockchain (BC) technology can offer a secure, decentralized, and immutable ledger to monitor suspicious activities over time and classify intrusions globally. Integrating advanced Deep Learning (DL) and BC improves detection and overall security. The decentralized nature of BC eliminates single points of failure. DL can efficiently detect patterns and anomalies, mitigating false alerts, and when integrated with public BC, it ensures secure, transparent, and tamper-proof storage of intrusion data. This study presents a BC-assisted Coati Optimization Algorithm (COA) with DL for Intrusion Detection and Classification (BCOADL-IDC) method, using the BC architecture to ensure data integrity and immutability. Initially, min-max normalization is applied to scale the input data. Then, an Attention-based Bidirectional Recurrent Neural Network (ABiRNN) model is utilized for intrusion detection. COA is used to fine-tune the crucial parameters of the ABiRNN model to improve detection performance. Finally, the integration of BC helps to ensure the integrity of the detection results, prevent tampering, and provide a transparent and secure record of network actions. The comparative study of the BCOADL-IDC approach showed a higher accuracy of 98.79% over existing models on the ToN_IoT dataset.

Keywords-blockchain; deep learning; security; coati optimization algorithm; intrusion detection; cybersecurity

I. INTRODUCTION

Cyberspace has great potential to transform lives, but its access and use must be carefully managed [1]. Artificial Intelligence (AI), BC, Machine Learning (ML), and cybersecurity are significant modules of a wide-ranging digital conversion approach [2]. BC ensures secure data storage, and integrating it with AI, ML, and cybersecurity improves security and efficiency, and reduces costs [3]. Cyberattacks often target web data, brand reputation, and e-commerce [4]. Combining BC and AI can produce more effective, safer, smarter, and protective systems [5]. However, Quantum technology has made most BC models susceptible to quantum attacks [6]. Combining BC and DL offers robust models to increase security in diverse fields. IDSs are effective for monitoring network activities, detecting and protecting against internal and external intrusions [7], suitable for IoT and smart city applications. However, they often produce high false alarms, which reduce effectiveness, increase computational costs, and burden security analysts [9]. Integrating BC and DL can enable robust, transparent, and reliable IDS for large-scale use [10].

This study presents a BC-Assisted Coati Optimization Algorithm with DL for Intrusion Detection and Classification (BCOADL-IDC) method, with the following key contributions:

- Uses min-max normalization to scale the input data into a uniform range, improving data consistency. This step ensures faster convergence during training and contributes to improved overall model performance and detection accuracy.
- The ABiRNN technique effectively captures temporal dependencies, and the attention mechanism highlights critical features. Thus, accuracy is improved, and the model is strengthened to detect intrinsic attack behaviors.
- Employs the COA method to fine-tune the key hyperparameters of the ABiRNN, improving detection accuracy and reducing false positives. A BC layer is incorporated to securely store the detection results, ensuring tamper-proof logging and transparency, thus strengthening trust, data integrity, and reliability.
- The novelty of the BCOADL-IDC model is in the unified application of ABiRNN, COA optimization, and BC technology. This hybrid technique improves detection accuracy, ensures secure and immutable result storage, and prevents data tampering.

II. LITERATURE SURVEY

Diverse methods have been proposed for the detection of cyberattacks. In [11], African Buffalo Optimization (ABO) was used in the Recurrent Neural Network (RNN) prediction stage. In [12], a multi-head attention Bidirectional Gated Recurrent Unit (BiGRU) model was optimized using Wolf Pack Predation (WPP) and Improved Secretary Bird Optimizer Algorithm (ISBOA). In [13], a self-adaptive, BC-assisted LSTM-based IDS was introduced. In [14], a framework combined BC with a multi-attention Deep Convolutional RNN (DeepCRNN). In [15], a Self-Attention-based Deep Convolutional Neural Network (SA-DCNN) was presented. In [16], the Automated Cyberattack Detection using Binary Metaheuristics with DL (ACAD-BMDL) method utilized the Binary Gray Wolf Optimizer (BGWO) for feature selection, the Enhanced Elman spike NN (EESNN) for detection, and the Archimedes Optimization Algorithm (AOA) for tuning.

In [17], the Optimized Graph Transformer with Molecule Attention Network (OGTMAN) method integrated a Secure Multi-party Computation (SMC) model and differential privacy. In [18], a DL model used the Pigeon-Inspired Optimizer for feature selection (PIOFS) and COA to tune the model. In [19], a 1D-CNN model was proposed. In [20], a distributed Federated IDS system was developed. In [21], an IDS system incorporated Conditional Generative Adversarial Networks (CGANs) and foundation models within an FL framework. In [22], a real-time IDS used a hybrid DL model, with CNN and Long Short-Term Memory (LSTM) to accurately detect attacks.

III. THE PROPOSED METHOD

This study presents the BCOADL-IDC technique, which involves preprocessing, classification, and tuning processes. Figure 1 shows the flow of the BCOADL-IDC approach.

A. Data Preprocessing: Min-Max Normalization

Initially, min-max normalization is applied to scale the input data to a useful format [23]. This approach was chosen for its ability to scale the data to [0, 1], ensuring equal feature contribution during learning. This technique is appropriate for scale-sensitive algorithms, such as NNs, as it prevents dominant features and ensures faster convergence. Compared to Z-score normalization, it is more effective for models requiring bounded inputs. Regarding intrusion detection, where varying features with diverse scales are frequently used to describe network traffic patterns, min-max normalization ensures that all features are scaled to a uniform range. Due to their larger magnitudes, this normalization step is crucial to prevent specifics from dominating the learning process.

B. Classification Model: ABiRNN Model

For intrusion detection, the BCOADL-IDC approach implements the ABiRNN method, which identifies the complex patterns in the network traffic [24]. This method was selected for its ability to capture both forward and backward dependencies in sequential data, making it highly effective for time-series and pattern recognition tasks. The attention mechanism improves its focus on crucial features, improving classification accuracy by prioritizing relevant data. The ABiRNN model overcomes vanishing gradient issues and better handles long-term dependencies, making it more robust for complex classification tasks in dynamic environments.

RNN models process sequential data but have difficulty with long sequences. BiRNN uses bidirectional input for better prediction and consists of forward and backward RNNs.

The forward RNN $\vec{f}$ reads an input series from x_1 to x_T and computes a series of forward Hidden Layers (HLs) $(\vec{h}_1, \cdots, \vec{h}_T)$ $(\vec{h}_i \in \mathbb{R}^p$, p refers to the dimension of the HLs). The backward RNN $\overleftarrow{f}$ reads the series in reverse order, from x_T to x_1, resulting in a series of backward HLs $(\overleftarrow{h_1}, \cdots, \overleftarrow{h_T})$, $(\overleftarrow{h_i} \in \mathbb{R}^p)$. By connecting the backward $\overleftarrow{h_i}$ and forward $\vec{h}_i$, the last latent vector is obtained as $h_i = [\vec{h}_i; \overleftarrow{h_i}]^T (h_i \in \mathbb{R}^{2p})$.

Only past data is used to predict future cases during the testing stage. In the prediction function, the last objective is to forecast the class-level codes of the $(t+1)$-th series, i.e., y_t. The t-th $x_t(h_t)$ output is the assessed vector symbol of the $(t+1)$ -th. Therefore, the main problem is originating a context vector c_t that takes related data to aid in forecasting the y_t. Three models are employed to evaluate the context vector c_t.

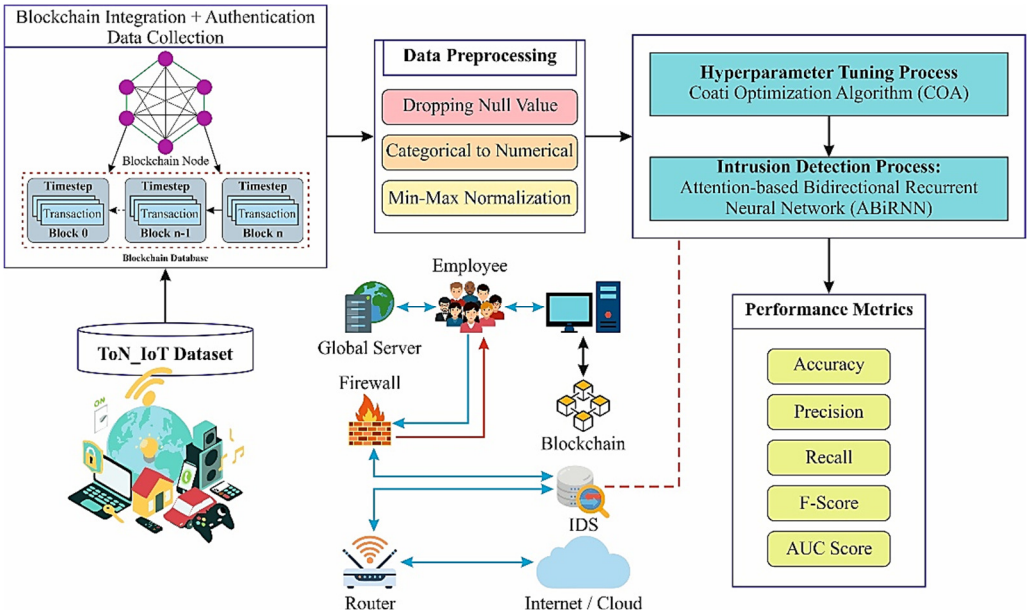

Fig. 1. Workflow of BCOADL-IDC technique.

1) Location-Based Attention

This task computes the weights only from the present HL h_i as:

$$\alpha_{ti} = W_\alpha^T h_i + b_\alpha \quad (1)$$

where $W_\alpha \in \mathbb{R}^{2p}$ and $b_\alpha \in \mathbb{R}$ denote the parameters to be acquired. Based on (1), an attention weight vector α_t is obtained utilizing the softmax function as follows:

$$\alpha_t Softmax\left([\alpha_{t1}, \alpha_{t2}, \cdots, \alpha_{t(t-1)}]\right) \quad (2)$$

Then, $c_t \in \mathbb{R}^{2p}$ depend on the weights attained from (2) and the HLs from h_1 to h_{t-1} as:

$$c_t = \sum_{i=1}^{t-1} \alpha_{ti}\, h_i \quad (3)$$

The location-based attention only reflects every individual HL's info and does not capture the relations among the present and prior HLs.

2) General Attention

A simple method to take the relation among h_t and $h_i (1 \leq i \leq t-1)$ is utilizing a matrix $W_\alpha \in \mathbb{R}^{2p \times 2p}$ and computing the weight as:

$$\alpha_{ti} = h_t^T W_\alpha h_i \quad (4)$$

The context vector c_t is acquired using (2) and (3).

3) Concatenation-Based Attention

A Multi-Layer Perceptron (MLP) is another method to compute the context vector c_t. The present HL h_s is initially connected with the prior h_i, and then a hidden vector is attained by increasing the weight matrix $W_\alpha \in \mathbb{R}^{q \times 4p}$, whereas q denotes the latent dimension. The $tanh$ was chosen as the activation function, as:

$$\alpha_{ti} = v_\alpha^T \tanh(W_\alpha [h_t; h_i]) \quad (5)$$

where $v_\alpha \in \mathbb{R}^q$ represents the parameter to be acquired. This is used to capture relations between the current and all prior hidden states, enhancing prediction accuracy.

C. Hyperparameter Tuning: COA Model

In this stage, the COA is utilized to fine-tune the parameters of the ABiRNN model to ensure effective detection performance [25]. This approach was chosen for its efficiency in solving complex optimization problems with a balance of exploration and exploitation. This approach shows superiority over conventional optimization techniques by offering faster convergence and improved handling of high-dimensional search spaces. COA's ability to avoid local minima and efficiently tune hyperparameters makes it appropriate for enhancing the performance of DL models in dynamic environments. COA is a metaheuristic optimization model derived from the raccoon population, where a coati is a member of the population. The value of the decision parameter is the location of the coati in space, and the coati's position defines a candidate solution. Initially, the coatis are initialized in the space as:

$$X_a : x_{ab} = lb_a + \cdot (ub_b - lb_b) \quad (6)$$

where X_a is the position of the a-th individual, and lb_b and ub_b are the lower and upper limits of the b^{th} variable. Each position encodes key ABiRNN hyperparameters such as learning rate, number of recurrent units, dropout rate, and attention weight parameters.

The best individual (igu) guides the search. Half of the population climbs toward igu as demonstrated in:

$$X_a^{L1} : x_{a,b}^{L1} = x_{a,b} + m \cdot \left(igu_b - I \cdot x_{a,b}\right)$$

$$\text{for } a = 1,2,\ldots,\left[\frac{N}{2}\right], b = 1,2,\ldots,m \quad (7)$$

After igu falls to a random point in the space, its new random location is determined using:

$$igu^G = lb_a + m \cdot (ub_b - lb_b) \quad (8)$$

$$X_a^{L1}: x_{ab}^{L1} = \begin{cases} x_{a,b} + m \cdot \left(igu_b^G - I \cdot x_{a,b}\right) & F_{iguG} < F_i \\ x_{a,b} + m \cdot \left(x_{a,b} - igu_b^G\right), \end{cases} \quad (9)$$

$$i = \left[\frac{N}{2}\right] + 1, \left[\frac{N}{2}\right] + 2, \ldots, N, \;\; j = 1,2, \ldots, M$$

A new position is accepted only if it improves the objective value:

$$X_a = \begin{cases} X_a^{L1}, & F_a^{L1} < F_i \\ X_a, & \text{else} \end{cases} \quad (10)$$

where the new location calculated for the a^{th} coati is X_a^{L1} and $x_{a,b}^{L1}$ is the b^{th} decision variable. The objective function value corresponding to the coati is F_a^{L1}. igu denotes the igu's location in the space, which represents the optimal positioning member of the space, igu_b is its corresponding b-th decision variable, I randomly chooses an integer in the set, igu^G denotes the igu's randomly produced location on the ground, and igu_b^G is the b-th decision variable, and F_{guG} is the value of the objective function. A random location near the co-location is generated to simulate the coati characteristics while avoiding predators as:

$$lb_b^{loc} = \frac{lb_b}{t}, ub_b^{loc} = \frac{ub_b}{t}, \text{ where } t = 1,2, \ldots, T \quad (11)$$

$$X^{L2}: x_{a,b}^{L2} = x_{a,b} + (1 - 2r) \cdot \left(lb_b^{loc} + r \cdot \left(ub_b^{loc} - lb_b^{loc}\right)\right), a = 1,2, \ldots, N, b = 1,2, \ldots, m \quad (12)$$

The new location is accepted if the newly calculated location augments the value, and the updating condition is formulated by:

$$X_a = \begin{cases} X_a^{L2}, & F_a^{L2} < F_a \\ X_a, & \text{else} \end{cases} \quad (13)$$

Here, the new position of the a-th coati is X_a^{L2}, $x_{a,b}^{L2}$ is its both dimensions due to the updating of the second stage, F_a^{L2} shows the new position value of the coati, t indicates the iteration count, and lb_b^{loc} and ub_b^{loc} are the lower and upper boundaries of the b-th dimension, correspondingly.

The hyperparameter selection process uses encoded results to evaluate candidate solutions. Here, the COA approach considers precision as the main criterion for designing the Fitness Function (FF):

$$Fitness = \max(P) \quad (14)$$

$$P = \frac{TP}{TP + FP} \quad (15)$$

where FP and TP depict the false and true positive values.

D. BC Integration

Integrating BC helps ensure the integrity of the detection results, prevents tampering, and delivers a transparent and secure record of network actions [26]. BC is integrated into the IDS to improve data integrity and security by recording validated intrusion events in an immutable decentralized ledger.

BC stores data in blocks linked by cryptographic hash functions and Merkle trees. It eliminates intermediaries, allowing only authorized nodes to add or read blocks, with deletion not permitted. BC types include Public (e.g., Ethereum), Private, and Consortium, differing in access and participation. The nodes follow consensus protocols such as proof-of-capacity (PoC), proof-of-work (PoW), proof-of-stake (PoS), and proof-of-burn (PoB) to validate transactions. Cryptographic hashing further ensures security. In summary, BC provides a distributed, flexible, immutable, and secure framework for information sharing.

IV. EXPERIMENTAL VALIDATION

The proposed BCOADL-IDC technique was run on Python 3.6.5 with an i5-8600k CPU, 4 GB GPU, 16 GB RAM, 250 GB SSD, and 1 TB HDD, using a 0.01 learning rate, ReLU, 50 epochs, 0.5 dropout, and batch size 5.

Table I shows an extensive description of the BCOADL-IDC model regarding Execution Time (EXET), depicting efficient performance across varying transaction (Tx) volumes and nodes. For 100 Txs, it achieved EXET of 19 s, 33 s, 144 s, and 291 s with 20 to 80 nodes. At 200 Txs, it achieved 28 s, 53 s, 423 s, and 619 s, while at 300 Txs, it achieved 59 s, 105 s, 629 s, and 965 s across the same node range.

TABLE I. EXECUTION TIME ANALYSIS OF THE BCOADL-IDC TECHNIQUE UNDER DISTINCT NODES

EXET (s)				
Number of Tx	**Nodes**			
	20	**40**	**60**	**80**
100	19	33	144	291
150	27	37	303	440
200	28	53	423	619
250	56	80	518	912
300	59	105	629	965

Table II presents an extensive comparison of the BCOADL-IDC method for Transaction Mining Time (TMT). For 5 Txs, the BCOADL-IDC method achieves the lowest transaction mining time (TMT) of 0.00033 s, compared to PoW, ePoW, and BHS-ALOHDL [27]. The results show that the BCOADL-IDC model achieves reasonable TMT values for all Tx values. The performance validation of the BCOADL-IDC method was determined using the ToN_IoT dataset [28].

TABLE II. TRANSACTION MINING TIME ANALYSIS OF BCOADL-IDC TECHNIQUE UNDER A DISTINCT NUMBER OF TX

TMT (s)				
Txs	**PoW**	**ePoW**	**BHS-ALOHDL**	**BCOADL-IDC**
5	0.00212	0.00134	0.00073	0.00033
10	0.00213	0.00137	0.00070	0.00036
15	0.00219	0.00140	0.00070	0.00033
20	0.00220	0.00140	0.00069	0.00035
25	0.00249	0.00139	0.00065	0.00032

TABLE III. DATASET DETAILS

Class	Sample Numbers
Normal	25000
Attack	25000
Overall samples	50000

Figure 2 displays the classifier outcomes of the BCOADL-IDC approach, indicating accurate classification into two classes.

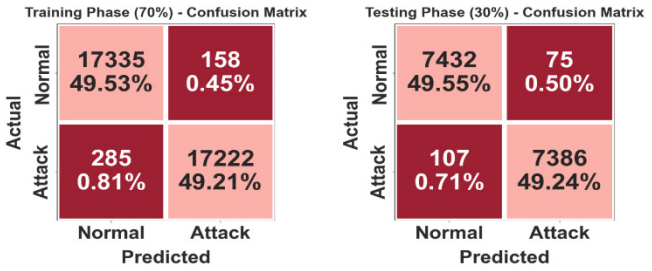

Fig. 2. Classifier outcomes of BCOADL-IDC technique - Confusion matrices.

Table IV compares the BCOADL-IDC technique with baseline models [27]. These findings show that the BiLSTM, Decision Tree (DT), and Naïve Bayes (NB) models performed poorly over other methods, while the Random Forest (RF) IDS model gained slightly improved results. However, the BCOADL-IDC model outperformed these baseline models with a maximum $accu_y$ of 98.79%, $prec_n$ of 98.79%, $reca_l$ of 98.79%, and F_{score} of 98.79%. The proposed model also achieved the lowest Computational Time (CT) of 4.02 s. These results highlight the better performance of the BCOADL-IDC approach.

TABLE IV. COMPARATIVE OUTCOME OF BCOADL-IDC TECHNIQUE WITH EXISTING APPROACHES

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	CT
BCOADL-IDC (Proposed)	98.79	98.79	98.79	98.79	4.02
IDS [27]	98.43	98.08	98.67	98.71	14.30
DT [27]	96.49	98.28	97.47	96.24	7.48
RF [27]	97.86	98.39	96.50	98.67	24.42
NB [27]	96.84	96.74	98.54	97.15	7.82
Bi-LSTM [27]	96.36	97.71	98.22	97.99	10.75

Table V summarizes the FLOPS, GPU usage, and inference time results compared to [29]. CNN anomaly detection approach with RF (CAA-RF) presents a balanced computation performance. LSTM and RNN exhibit higher computational demand, while some models provide mid-range efficiency. However, BCOADL-IDC is the most lightweight and fastest method in this comparison.

TABLE V. OUTCOME COMPARISON ON FLOPS, GPU USAGE, AND INFERENCE TIME

Approaches	FLOPS (G)	GPU (M)	Inference Time (s)
CAA	1,269,427,394	6574	15.27
CAA-RF	1,340,762,638	5863	12.88
LSTM	1,0120,866,978	7459	8.57
RNN	25,308,412,560	5560	16.69
VAE	141,332,137	6637	7.59
DBN	634,963,849	6563	12.01
BCOADL-IDC	969,586	856	3.09

V. CONCLUSION

This study presented a novel BCOADL-IDC that combines the BC architecture to ensure data integrity and immutability. The BCOADL-IDC technique involves min-max normalization, ABiRNN-based intrusion detection, and COA-based tuning. BC ensures the integrity, security, and transparency of the detection results. The comparative study of the BCOADL-IDC approach demonstrated superior accuracy (98.79%) over baseline models on the ToN_IoT dataset. The limitations include restricted scalability when applied to extremely large networks and potential threats in real-time processing under high data volumes. Future work will focus on optimizing computational efficiency and exploring adaptive mechanisms to improve detection accuracy in dynamic environments. In addition, integrating more advanced privacy-preserving models could further strengthen security.

REFERENCES

[1] A. A. Aliyu, M. Ibrahim, and S. Abdulkadir, "A Blockchain-Enhanced Deep Learning Approach for Intrusion Detection in Trusted Execution Environments," *Digital Technologies Research and Applications*, vol. 4, no. 1, pp. 135–157, June 2025, https://doi.org/10.54963/dtra.v4i1.962.

[2] M. Mounir, S. G. Sayed, and M. M. E. El-Dakroury, "Securing the Future: Real-Time intrusion Detection in IIoT Smart Grids through Innovative AI Solutions," *Journal of Cybersecurity and Information Management*, vol. 15, no. 2, pp. 208–244, 2025, https://doi.org/10.54216/JCIM.150216.

[3] S. R. Addula, A. K. Tyagi, K. Naithani, and S. Kumari, "Blockchain-Empowered Internet of Things (IoTs) Platforms for Automation in Various Sectors," in *Artificial Intelligence-Enabled Digital Twin for Smart Manufacturing*, John Wiley & Sons, 2024, pp. 443–477.

[4] Z. Wang, J. Wang, Y. Liu, X. Yang, F. Qi, and W. Song, "Privacy-Preserving Attribute-Based Access Control Scheme With Intrusion Detection and Policy Hiding for Data Sharing in VANET," *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 23348–23369, July 2024, https://doi.org/10.1109/JIOT.2024.3384753.

[5] R. K. Mahendran, A. Khan, F. Ullah, F. Ali, and A. A. AlZubi, "PRISM-IIoT: A holistic approach for privacy preservation in industrial IoT using advanced cryptography and blockchain-enabled auditability framework," *Alexandria Engineering Journal*, vol. 128, pp. 816–832, Sept. 2025, https://doi.org/10.1016/j.aej.2025.07.027.

[6] S. R. Khonde and V. Ulagamuthalvi, "Hybrid intrusion detection system using blockchain framework," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, June 2022, Art. no. 58, https://doi.org/10.1186/s13638-022-02089-4.

[7] N. Kumaran and S. J. S. Mohan, "BRDO: Blockchain Assisted Intrusion Detection Using Optimized Deep Stacked Network," *Cybernetics and Systems*, vol. 55, no. 8, pp. 2071–2092, Nov. 2024, https://doi.org/10.1080/01969722.2023.2175153.

[8] M. K. Nayak, K. Dehury, and D. Gountia, "A Hybrid Deep Learning and Blockchain Framework for Real-Time IoT DDoS Resilience," *SN Computer Science*, vol. 6, no. 7, Sept. 2025, Art. no. 863, https://doi.org/10.1007/s42979-025-04417-z.

[9] S. Hossain, S. M. Senouci, B. Brik, and A. Boualouache, "A privacy-preserving Self-Supervised Learning-based intrusion detection system for 5G-V2X networks," *Ad Hoc Networks*, vol. 166, Jan. 2025, Art. no. 103674, https://doi.org/10.1016/j.adhoc.2024.103674.

[10] J. Li *et al.*, "A Lightweight Intrusion Detection System with Dynamic Feature Fusion Federated Learning for Vehicular Network Security," *Sensors*, vol. 25, no. 15, July 2025, https://doi.org/10.3390/s25154622.

[11] V. Saravanan, M. Madiajagan, S. M. Rafee, P. Sanju, T. B. Rehman, and B. Pattanaik, "IoT-based blockchain intrusion detection using optimized recurrent neural network," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 31505–31526, Mar. 2024, https://doi.org/10.1007/s11042-023-16662-6.

[12] S. A. Alzakari *et al.*, "Heuristically enhanced multi-head attention based recurrent neural network for denial of wallet attacks detection on serverless computing environment," *Scientific Reports*, vol. 15, no. 1, Apr. 2025, Art. no. 13538, https://doi.org/10.1038/s41598-025-87636-x.

[13] A. A. Aliyu, J. Liu, and E. Gilliard, "A Decentralized and Self-Adaptive Intrusion Detection Approach Using Continuous Learning and Blockchain Technology," *Journal of Data Science and Intelligent Systems*, Oct. 2024, https://doi.org/10.47852/bonviewJDSIS42023803.

[14] N. Sharma and P. G. Shambharkar, "Multi-attention DeepCRNN: an efficient and explainable intrusion detection framework for Internet of Medical Things environments," *Knowledge and Information Systems*, vol. 67, no. 7, pp. 5783–5849, July 2025, https://doi.org/10.1007/s10115-025-02402-9.

[15] M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, "A Self-Attention-Based Deep Convolutional Neural Networks for IIoT Networks Intrusion Detection," *IEEE Access*, vol. 12, pp. 45762–45772, 2024, https://doi.org/10.1109/ACCESS.2024.3380816.

[16] A. Al Mazroa, F. R. Albogamy, M. Khairi Ishak, and S. M. Mostafa, "Boosting Cyberattack Detection Using Binary Metaheuristics With Deep Learning on Cyber-Physical System Environment," *IEEE Access*, vol. 13, pp. 11280–11294, 2025, https://doi.org/10.1109/ACCESS.2025.3526258.

[17] M. Anoop, L. W. Mary, A. J. Wilson, and W. S. Kiran, "Optimized graph transformer with molecule attention network based multi class attack detection framework for enhancing privacy and security in WSN," *Multimedia Tools and Applications*, vol. 84, no. 15, pp. 14273–14304, May 2025, https://doi.org/10.1007/s11042-024-19516-x.

[18] R. Shanmugavelu and V. Ravi, "Enhancing Security in Healthcare Frameworks using Optimal Deep Learning-based Attack Detection and Classification for Medical Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 21197–21202, Apr. 2025, https://doi.org/10.48084/etasr.9741.

[19] D. Torre, A. Chennamaneni, J. Jo, G. Vyas, and B. Sabrsula, "Toward Enhancing Privacy Preservation of a Federated Learning CNN Intrusion Detection System in IoT: Method and Empirical Study," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 2, Jan. 2025, Art. no. 53, https://doi.org/10.1145/3695998.

[20] N. Sun, W. Wang, Y. Tong, and K. Liu, "Blockchain based federated learning for intrusion detection for Internet of Things," *Frontiers of Computer Science*, vol. 18, no. 5, Dec. 2023, Art. no. 185328, https://doi.org/10.1007/s11704-023-3026-8.

[21] S. Jiao *et al.*, "Foundation-Model-Based Federated Learning for Intrusion Detection in Drone-Aided Industrial IoT," *IEEE Internet of Things Journal*, vol. 12, no. 22, Aug. 2025, Art. no. 46889–46901, https://doi.org/10.1109/JIOT.2025.3597349.

[22] J. A. Alzubi, O. A. Alzubi, I. Qiqieh, and A. Singh, "A Blended Deep Learning Intrusion Detection Framework for Consumable Edge-Centric IoMT Industry," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 2049–2057, Oct. 2024, https://doi.org/10.1109/TCE.2024.3350231.

[23] K. S. Sarin, R. E. Kolomnikov, M. O. Svetlakov, and I. A. Hodashinsky, "Fuzzy Min-Max Classifier in Cybersecurity Applications," *Automatic Documentation and Mathematical Linguistics*, vol. 58, no. 5, pp. 299–309, Oct. 2024, https://doi.org/10.3103/S0005105524700250.

[24] N. Girubagari and T. N. Ravi, "Parallel ABILSTM and CBIGRU Ensemble Network Intrusion Detection System," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 1, pp. 93–107, Feb. 2024, https://doi.org/10.22266/ijies2024.0229.10.

[25] K. Sarathkumar, P. Sudhakar, and A. C. Kanmani, "Enhancing intrusion detection using coati optimization algorithm with deep learning on vehicular Adhoc networks," *International Journal of Information Technology*, vol. 16, no. 5, pp. 3009–3018, June 2024, https://doi.org/10.1007/s41870-024-01827-9.

[26] S. R. Khonde and V. Ulagamuthalvi, "Blockchain: Secured Solution for Signature Transfer in Distributed Intrusion Detection System," *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 37–51, 2022, https://doi.org/10.32604/csse.2022.017130.

[27] H. Alamro *et al.*, "Modeling of Blockchain Assisted Intrusion Detection on IoT Healthcare System Using Ant Lion Optimizer With Hybrid Deep Learning," *IEEE Access*, vol. 11, pp. 82199–82207, 2023, https://doi.org/10.1109/ACCESS.2023.3299589.

[28] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, "Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China, Dec. 2020, pp. 848–855, https://doi.org/10.1109/TrustCom50675.2020.00114.

[29] S. Jia *et al.*, "CAA-RF: An Anomaly Detection Algorithm for Computing Power Blockchain Networks," *Applied Sciences*, vol. 15, no. 11, May 2025, https://doi.org/10.3390/app15115804.