I. J. Computer Network and Information Security, **2026, 4, 22-45**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2026.04.02

Blockchain-Integrated Discrete Hopfield Neural Network and Edge Attention Networks with Duck Swarm Optimization for Cloud Privacy Enhancement

Ajay N. Upadhyaya*
Department of Computer Engineering, SAL Engineering & Technical Institute, SAL Education, Ahmedabad, 380060, Gujarat, India
E-mail: ajay8586g@gmail.com
ORCID iD: https://orcid.org/0000-0002-7583-6430
*Corresponding author

Dinesh Mishra
Department of CSE, Mangalayatan University – Jabalpur, MP, India
E-mail: dinesh.mishra@mangalayatan.ac.in
ORCID iD: https://orcid.org/0009-0002-8639-5914

Rajan Prasad Tripathi
Department of IT and Engineering, Amity University, Tashkent, Uzbekistan
E-mail: rajantripathi22@gmail.com
ORCID iD: https://orcid.org/0000-0002-1192-4773

Bharani B. R.
Dept of Information Science and Engineering, Cambridge Institute of Technology, KR Puram, Bengaluru-560036, Karnataka, India
E-mail: bharani.ise@cambridge.edu.in
ORCID iD: https://orcid.org/0009-0007-3010-7238

Received: January 24, 2025; Revised: May 17, 2025; Accepted: September 2, 2025; Published: August 8, 2026

Abstract: Cloud computing has transformed data management for businesses and individuals alike by making systems more scalable and economically viable. However, the distributed architecture in cloud computing inherently makes it vulnerable to highly sophisticated cyber threats, such as DDoS attacks, ransomware, cryptojacking, and many others that could compromise data confidentiality, integrity, and availability. Traditional intrusion detection systems face limitations such as high false-negative rates and low adaptability to emerging threats. This manuscript proposes a blockchain-integrated discrete Hopfield neural network and edge attention networks with duck swarm optimization (Hop-MEA-Duck) to enhance cloud privacy and develop a robust, privacy-preserving intrusion detection framework for cloud environments. The framework assumes a two-level privacy mechanism. The former provides privacy through the Adaptive Blockchain Sharding Protocol with Hybrid Consensus (Hyb-BCSP), which enhances data security and scalability. The second tier comprises preprocessing using the Adaptive Self-Guided Loop Filter (ASGLF) and feature selection via the Pufferfish Optimization Algorithm (POA). The Discrete Hopfield Neural Network (DHNN) with Multilayer Edge Attention Network (MEAN) is used to classify normal and abnormal behaviors, and the Duck Swarm Algorithm (DSA) is used for hyperparameter tuning. The results of the experiments indicate that the proposed framework is practical, achieving 97.8% detection accuracy, 96.5% precision, and a significant reduction in the false-negative rate to 2.4%. Also, the preprocessing phase increased the relevance of the data by 85%, whereas the rate of information immutability in the blockchain was 99.2%. To sum up, the suggested framework can provide a privacy-preserving, scalable, and adaptable solution for detecting and mitigating cyber threats in the context of cloud computing. It can fill significant gaps in current intrusion detection approaches.

Index Terms: Adaptive Blockchain Sharding Protocol with Hybrid Consensus, Duck Swarm Algorithm, Discrete Hopfield Neural Network, Multilayer Edge Attention Network, Pufferfish Optimization Algorithm.

1. Introduction

Cloud computing has transformed how companies store, process and manage data. It is flexible, has scalability and is cost-effective to an extent that no other technology can match [1]. Cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources", according to the National Institute of Standards and Technology. Cloud computing has emerged as a cornerstone for business, educational institutions, healthcare systems, and government agencies [2]. Its rapid adoption has ushered in quick access to substantial computing resources while innovations in all genres of life became possible. On the other hand, pervasive dependence on cloud services puts sensitive data and systems at the mercy of an array of cyber threats [3]. Cloud environments are specifically very vulnerable to sophisticated cyber-attacks due to their dynamic and distributed nature, such as DDoS, ransomware, MITM, Sybil attacks, and cryptojacking [4].

These attacks compromise the confidentiality, integrity, and availability of cloud-based resources [5-8]. In this regard, security threats pose a formidable risk to an organization and users equally. Intrusions, especially with attacker's perpetually changing tack, are perhaps one of the biggest challenges of securing cloud environments [9-12]. Traditional IDS approaches, often performed using signature-based methods, address novel emerging threats poorly, usually leading to high rates of false negatives and restricted scalability for large-scale deployments [13-15].

To counter such challenges, the paradigm has now shifted towards intelligent anomaly detection. These use advanced technologies of machine learning and blockchain to provide more accuracy and resiliency. Being a decentralized technology, blockchain brings a new paradigm toward cloud security on immutable logs, transparency, and secure data sharing among the stakeholders. This paper therefore proposes a blockchain-enabled intrusion detection system for handling threats without compromising privacy, scalability, and cost. The approach couples anomaly-based detection with the features of blockchain offering inherent trust and transparency to overcome prevailing limitations comprehensively for protection in state-of-the-art cloud computing environments.

Contributions: The contributions of this paper are given below:

- *Hybrid Blockchain-Enabled Intrusion Detection Framework:* The proposed framework integrates blockchain with machine learning to jointly enhance intrusion detection accuracy and data integrity in cloud environments.
- *Two-Level Privacy Mechanism:* A two-layer privacy strategy is formed, where the Adaptive Blockchain Sharding Protocol enhances data security and scalability through preprocessing and feature selection using optimization algorithms, further enhancing resistance to potential breaches.
- *Deep Learning-based Advanced Feature Classification:* Network behavior classification. The combination of the Multilayer Edge Attention Network and the Discrete Hopfield Neural Network is used to perform network behavior classification. This assists in real-time identification of sophisticated attack patterns.
- *Optimization using the Duck Swarm Algorithm:* The Duck Swarm Algorithm is used for hyperparameter tuning to achieve the best model performance and increased detection accuracy.
- *Empirical Evaluation:* Experimental validation demonstrates improved detection capability with reduced false negatives.
- *Blockchain-Assisted Data Integrity:* Blockchain integration gives integrity to data, prevents unauthorized changes, and helps in maintaining trust.
- *Scalability and Adaptability:* The framework offers scalability and adaptability toward addressing various cloud environments and changing security threats.

The remainder of this paper is organized as Section 2: Literature review, Section 3: Proposed methodology, Section 4: Results and Discussion, Section 5: Conclusion.

2. Literature Survey

Among the numerous research efforts on blockchain-enabled intrusion detection frameworks for enhancing privacy and security in cloud computing, some recent studies are deliberated here.

In 2024, Sarveshwaran, V., et al. [16] have presented Binarized Spiking Neural Network with a blockchain-based delegated proof-of-stake (DPoS) consensus algorithm to improve the privacy and security in cloud computing. Data evaluation was done using the NSL-KDD dataset. Privacy procedures incorporated blockchain-based DPoS algorithm and preprocessing method by Basic Interlude Gradient filtering (BIGF) to streamline information. Weighted Espoused Feature Assortment (WEFA) was applied in order to select features. The framework achieved higher accuracy, lower time consumption, and higher AUC than existing methods, and it provides intrusion detection capability as well as potential application in real world settings of tightly connected networks.

In 2022, Devi, K. and Muthusenthil, B., [17] have introduced a dual-channel capsule generation adversarial network with the red fox optimization algorithm in intrusion detection on clouds. The NSL-KDD dataset has been preprocessed with random forests and local least squares to eliminate redundancy as well as address missing value in the dataset. The best features were obtained using univariate ensemble feature selection and classified using a dual-

channel capsule generative adversarial network optimized using red fox. The model was more accurate and sensitive compared to other algorithms (DNB-CSSA and RCNN-ALO) applied in their frameworks.

In 2023, Attou, H., et al. [18] developed an intrusion detection system (IDS) using Radial Basis Function Neural Networks (RBFNNs) and Random Forest (RF) algorithms. To prove its efficacy, the method used Bot-IoT and NSL-KDD data. An improved Matthews Correlation Coefficient (MCC) was found between the RF and RBFNN in feature selection as the detection rate went up to 28-93%. The technique further obtained high accuracy, more than 92 percent, with a small number of features thus improving the efficiency of intrusion detection and security of clouds.

In 2023, Salvakkam, D.B., et al. [19] developed a better quantum-safe ensemble intrusion detection algorithm using deep learning based on datasets like KDDcup 1999, UNSW-NB15 and NSL-KDD. The approach helped overcome the limitation of the IDS and streamline the accuracy of intrusion-detection by the use of ensemble methods. Experimental results showed better detection performance, with a 92.14% recall. This model improved accuracy and efficiency, facilitating secure cloud computing environments.

In 2023, Mayuranathan, M., et al. [20] introduced a hybrid machine learning intrusion detection system for cloud computing. Preprocessing was enabled by the improved heap optimization (IHO) method, and feature selection was made more efficient with the chaotic red deer optimization (CRDO). A Deep Kronecker Neural Network (DKNN) was used in intrusion classification. Testing on the DARPA IDS and CSE-CIC-IDS2018 datasets showed that this method improved detection accuracy and computational efficiency more than existing systems.

In 2024, Preethi, B.C. et al. [21] developed a Multiscale Deep Bidirectional Gated Recurrent Neural Network (MDBGRNN)- based Intrusion Detection and Secure Data Storage System with the Optimal Encryption Scheme. Domain Transform Filtering (DTF) was used to preprocess the KDD CUP 99 and DS2OS datasets to reduce dimensionality and simplify semantic analysis. The combination of Elliptic Curve Cryptography and Sine-Cosine Egret Swarm Optimization was used to provide high levels of data security. The solution achieved better performance metrics, such as encryption time and the Matthews Correlation Coefficient, and successfully addressed cloud security concerns.

In 2023, Pandey, B.K. et al. [22] employed the ExpSSOA Deep Maxout Network Intrusion detector. It was tested on the MQTT-IOT-IDS2020 and Apache Web Server datasets and used an Exponential Weighted Moving Average with SSOA to optimize intrusion detection performance. It was found to be effective in detecting network intrusions in modern networks, with improvements in accuracy, F-measure, and recall. Table 1 summarizes the existing methods.

Table 1. Summary of the Existing Methods.

Reference Number	Methods	Highlights	Challenges
[16]	Binarized Spiking Neural Network with DPoS Algorithm	Achieved improved accuracy (+17.99%), reduced computation time (-48.17%), and enhanced AUC; utilized blockchain-based consensus for privacy and security.	Limited dataset scope (NSL-KDD) and lack of real-world implementation.
[17]	Dual-Channel Capsule Generative Adversarial Network	Provided improved accuracy (+13.93%) and sensitivity (+24.54%) using red fox optimization and effective feature selection (UEFS).	Dependency on dataset characteristics (NSL-KDD) and computational complexity of GAN models.
[18]	RBFNN with Random Forest	Improved MCC from 28% to 93% and achieved over 92% accuracy using minimal features; optimized for imbalanced datasets like Bot-IoT and NSL-KDD.	Restricted validation datasets and reliance on parameter tuning for RBFNN.
[19]	Enhanced Quantum-Secure Ensemble IDS	Improved intrusion detection with a recall rate of 92.14%; analyzed limitations of existing IDS methods and leveraged multiple datasets for validation.	Complexity in quantum-based ensemble methods and scalability issues for real-time applications.
[20]	Hybrid Deep Learning Approach with DKNN	Used advanced preprocessing (IHO) and feature selection (CRDO); demonstrated high accuracy and efficient classification with DKNN.	Limited benchmark datasets (DARPA IDS, CSE-CIC-IDS2018) and lack of evaluation under diverse attacks.
[21]	MDBGRNN with ECC-SCESOA Encryption	Improved security and detection with domain transform filtering; enhanced encryption with elliptic curve cryptography and optimization algorithms.	Higher computational overhead for encryption and steganography integration.
[22]	ExpSSOA with Deep Maxout Network	Delivered better accuracy (88.3%) and improved F-measure (87.68%); combined Exponential Weighted Moving Average with optimized classification techniques.	Performance variability with different datasets and challenges in handling large-scale systems.

Table 1 provides a summary of advanced intrusion detection techniques for cloud environments that leverage blockchain, feature selection, and deep learning to optimize performance. These techniques were aimed at accuracy, computing speed, and data protection. New models were added that used NSL-KDD and other benchmark datasets, along with some of the latest algorithms, including RBFNN, dual-channel adversarial networks, and ensemble learning methods. Regular results in the literature indicate that the metrics of detection and system robustness were significantly improved, providing potential ways to reduce the risk of cloud-based intrusions. All of these methods demonstrate a gradual transition to robust, secure, and scalable intrusion detection systems in the contemporary cloud environment.

Problem Statement: An analysis of the current literature on blockchain and intrusion detection systems (IDS) reveals several important gaps. The majority of IDSs do not combine both signature-based and anomaly-based detection methods into a single framework. Also, these systems do not use a collaborative strategy for intrusion identification, which could enhance collective defense through joint signature development. As well, the current methods do not include measures to securely distribute these signatures across the network, leaving systems vulnerable during communication. The few IDSs that use blockchain technology to ensure complete decentralization are vital to the realization of a scalable, secure intrusion detection ecosystem.

3. Proposed Methodology

The suggested solution is to integrate machine learning techniques with blockchain technology to improve data protection and intrusion detection in the cloud computing environment. The proposed framework will apply this two-tier privacy mechanism: at the first level, Adaptive Blockchain Sharding Protocol + Hybrid Consensus, which enforces data security and is scalable; and at the second level, preprocessing with Adaptive Self-Guided Loop Filter and feature selection with the Pufferfish Optimization Algorithm (POA). It leverages both a Discrete Hopfield Neural Network (DHNN) and a Multilayer Edge Attention Network (MEAN) to classify network behavior as normal or abnormal, thereby facilitating efficient intrusion detection. DSA is used for hyperparameter tuning to enhance model performance. This method guarantees proper and credible identification and privacy, using blockchain technology to ensure the impossibility of data alteration and integrity. The framework is scalable and flexible, enabling it to address emerging and increasing cyber threats in the cloud environment. Fig. 1 illustrates the proposed Hop-MEA-Duck methodology.

The novelty of the suggested Hop-MEA-Duck scheme lies not in the use of standard benchmark datasets, but rather in its innovative, strictly integrated architectural framework and algorithmic combination across various system levels. Intrusion detection systems proposed by conventional methods apply blockchain, deep learning, or optimization methods sequentially or independently, the proposed framework presents privacy-first, end-to-end intrusion detection pipeline where the components have non-overlapping and interdependent functional roles.

In particular, the Hyb-BCSP is not used as a mere secure data storage layer: it is also an active governance and privacy-enforcement protocol that oversees the data immutability, shard-level isolation, trust propagation, and cross-shard validation before machine learning operations. The design directly affects the downstream preprocessing and classification steps, which is not the case with the current blockchain-based intrusion detection systems.

Also, the DHNN + Multilayer Edge Attention Network (MEAN) integration is a novel hybrid classification paradigm, which involves associative memory-based convergence and edge-conscious, attention-based feature refinement. This combination allows the reliable pattern recall and retains the fine-grained spatial and contextual intrusion features, which have never been addressed in the literature on cloud-centric intrusion detection.

Moreover, the Duck Swarm Algorithm (DSA) is also recruited as a privacy-preserving hyperparameter optimization system explicitly aimed at the task of balancing detection and minimizing false-negative errors within a set of blockchain imposed limitations. Instead of being an all-purpose optimizer, DSA actively adapts the DHNN learning parameters to a privacy-preserving, sharded operation setting.

Although the NSL-KDD and CICIDS2017 datasets are utilized to validate the results comparatively and in a reproducible manner, the main contribution of this work is the suggested methodology framework that combines blockchain governance, adaptive preprocessing, memory-based neural classification, and swarm-based optimization. The framework is therefore dataset-agnostic by default and can be easily generalized to a real-world cloud or proprietary cloud traffic environment.

At the beginning of the workflow shown in Fig. 1, network traffic statistics are gathered in cloud environments and benchmark datasets. The first privacy layer involves data validation and personalized storage that protect data integrity and accessibility through the Adaptive Blockchain Sharding Protocol with Hybrid Consensus (Hyb-BCSP). The second privacy layer preprocesses the data using the Adaptive Self-Guided Loop Filter, and the best feature selection is performed using the Pufferfish Optimization Algorithm. The chosen features are then categorized using the hybrid Discrete Hopfield Neural Network and the Multilayer Edge Attention Network. Lastly, the Duck Swarm Algorithm uses the classifier's hyperparameters to optimize it, maximizing detection rate and reducing false negatives.

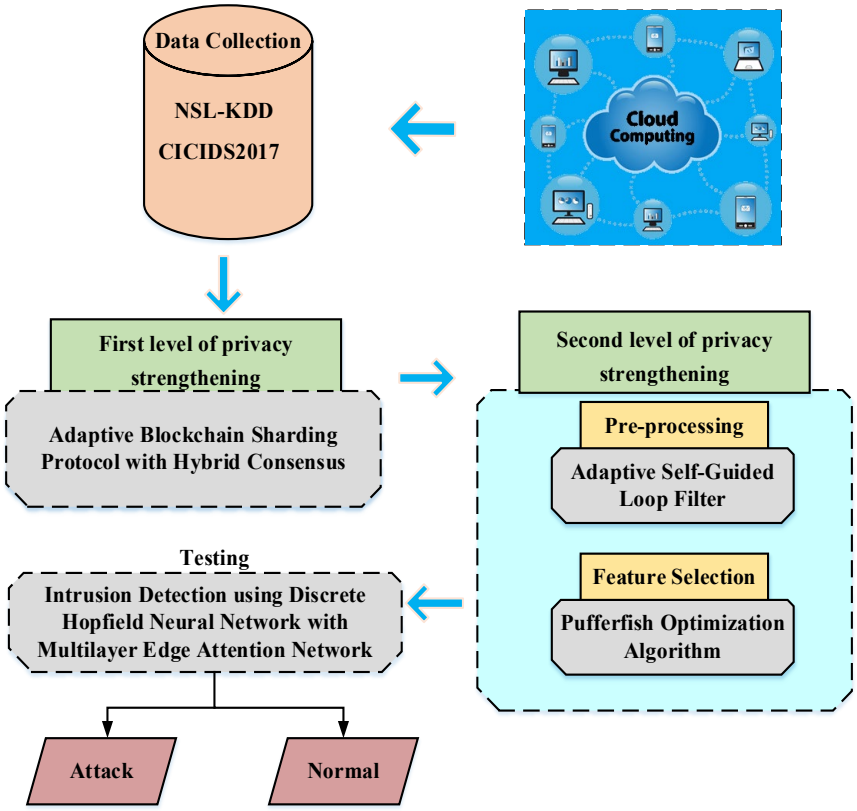

Fig. 1. Step-by-step workflow of the proposed Hop-MEA-Duck intrusion detection framework.

3.1. Data Collection

To start with, it will gather the data based on two popular datasets, namely NSL-KDD and CICIDS2017. The NSL-KDD dataset contains both normal and malicious examples, labeled with attacks such as denial-of-service (DoS) and probing, which are ideal for testing an intrusion detection model. The CICIDS2017 is a newer, more varied dataset, as it contains network traffic data from real-life attacks, including DDoS, web-based, and botnet-based attacks.

Therefore, this dataset is well-suited to test the performance of detection against complex, modern cyber threats in the cloud computing environment. The data input from the cloud of user devices is then passed on to the first level of privacy strengthening.

3.2. First level of privacy strengthening

The use of blockchain technology assures data integrity and immutability, thereby not allowing changes in or alterations of data after being stored in the blockchain. Therefore, the accuracy of enterprise data in the cloud computing system is assured. Hybrid Consensus-based Adaptive Blockchain Sharding Protocol (Hyb-BCSP) is adapted to further boost privacy and scalability in the proposed system. Hyb-BCSP dynamically and adaptively partitions the blockchain into shards, managed by a hybrid consensus mechanism comprising global and intra-shard consensus processes. Therefore, the system can handle heavy transaction volumes and maintain privacy and security. Optimal shard management and secure cross-shard communication reduce latency for faster transaction processing. This becomes important for scalable cloud applications, which require strong privacy and high performance. Hyb-BCSP has four key segments, which are outlined below.

3.2.1. Adaptive Blockchain Sharding Protocol with Hybrid Consensus (Hyb-BCSP):

Hyb-BCSP is an adaptive blockchain sharding protocol [23] that is dynamic and intelligent in approach, designed to address scalability and security challenges within blockchain-based cloud computing environments. It ensures high transaction throughput, low latency, and robust security without sacrificing privacy or data integrity by partitioning the blockchain into manageable shards and leveraging hybrid consensus mechanisms. Four key components constitute the methodology: adaptive shard management, hybrid consensus for cross-shard transactions, shard state synchronization, and dispute resolution.

a) Adaptive Shard Management

Adaptive shard management dynamically adjusts shard configurations according to transaction volume and resource utilization for maximum performance and efficiency in resource use.

Metrics and Thresholds

- Shards: $D = \{d_1, d_2, ...d_i\}$ where i is the total number of shards.
- Transaction Volume (T_n): Number of transactions processed by shard d_n .
- Resource Utilization (): Fraction of R_n computational, storage, and network resources used by shard d_n .
- Splitting Threshold (S_t): Triggered when $T_n > S_t$ or $R_n > S_t$, typically 60%-90%.
- Merging Threshold (S_M): Triggered when $T_n < S_M$ normally 20%-40%.

Shard Splitting: When T_n or R_n is attained S_t is based on the following equation (1)

$$d_n \to \{d_{n1}, d_{n2}, ...d_{nl}\} \tag{1}$$

All new shards are in compliance with the following equation (2).

$$\sum_{m=1}^{l} T_{nn} \approx T_n \quad and \quad \sum_{m=1}^{l} R_{nn} \approx R_n \tag{2}$$

Shard Merging: When T_n and R_n have both dropped below for S_M epochs consecutively based on the following equation (3).

$$\{d_{m1}, d_{m2}, ...d_{ml}\} \to d_f \tag{3}$$

The merged shard complies with the following equation (4).

$$T_f = \sum_{l=1}^{k} T_{nl} \quad and \quad R_f = \sum_{l=1}^{k} R_{nl} \tag{4}$$

Greedy Load Balancing Algorithm

Redistribute the nodes, accounts, and transactions across the shards for reducing interaction among different shards as much as possible, workload distribution and efficiency improvement.

b) Hybrid Consensus Mechanism:

Hybrid consensus mechanism applies lightweight global consensus along with parallel intra-shard consensus for secure processing of intra-shard and cross-shard transactions.

Process for Cross-Shard Transactions

- *Input Validation:* Input shards verify the transaction and create partial signatures is represented by the following equation (5).

$$\varpi_{m,n} = sign_{dm}(g(e_n)) \tag{5}$$

- *Threshold Signature Combination:* Aggregate partial signatures at output shards into a threshold signature is based on the following equation (6).

$$\sum_{n} combine(\{\varpi_{m,n} : d_m \in Y_n\}) \tag{6}$$

- *Global Consensus:* The nodes in global consensus verify the threshold signature and commit transactions is defined as the following equation (7).

$$v(e_n) = \begin{cases} Valid & if\ quorum\ is\ met \\ Invalid & otherwise \end{cases} \quad (7)$$

- *State Update:* If $v(e_n) = Valid$, then the transaction executes atomically with the global state updated.

Benefits of Hybrid Consensus

- Parallel intra-shard consensus reduces the bottleneck.
- Global consensus that is lightweight ensures consistency and atomicity in cross-shard transactions.
- Threshold signatures increase security because input shards require quorum approval.

c) Shard State Synchronization:

State synchronization makes sure that each shard has an identical view of the global blockchain state. It is done using a Merkle Tree-Based Data Structure.

Synchronization Process: Each shard calculates a Merkle Root (mr_a) is based on the following equation (8).

$$mr_a = merkle\ root(mr_a) \quad (8)$$

Shards broadcast their Merkle roots through a gossip protocol to reflect the global state is represented by the following equation (9).

$$H = \{mr_1, mr_2, ...mr_a\} \quad (9)$$

Updates and Proofs: Whenever a shard updates its state, it broadcasts the new Merkle root $(mr_a')$and a proof (τ_a) to other shards.

Merkle Proof Validation: Shards check the updated root using based on the following equation (10).

$$Verify(mr_a'\tau_a, S_m) \quad (10)$$

This way, the global state is ensured to be conflict-free and consistent.

d) Decentralized Dispute Resolution: The dispute resolution mechanism settles disputes and verifies cross-shard transactions.

Process

1. *Challenge:* The shard that disputes sends a challenge u with evidence v.
2. *Evidence Collection:* The shards involved send evidence v_b and signatures η_b.
3. *Voting:* The validity of a transaction is decided by the following weighted vote is represented by equation (11).

$$q(S_I) = \sum_b W_b q_b \quad (11)$$

Where W_b is denoted as the weight of the shard (by stake or reputation), and q_b is its vote

4. *Conflict Resolution:*
   - The transaction is rolled back, and shards that approved it are penalized if $q(S_I) = \text{Invalid}$
   - The global state is updated according to the resolution.

The Adaptive Blockchain Sharding Protocol (Hyb-BCSP) is scalable, secure, and efficient for cloud-based applications. These mechanisms address critical challenges needed in modern blockchain systems through the dynamic management of shard configurations; atomic cross-shard transaction guarantee; and a consistent, up-to-date, and perfectly synchronized global state. Its advanced mechanisms make Hyb-BCSP a robust solution for the shifting demands of cloud computing environments.

3.3. Second level of privacy strengthening

As soon as the authenticity of the information is confirmed, it is securely and immutably stored by creating a block via Adaptive Blockchain Sharding Protocol with Hybrid Consensus (Hyb-BCSP). The verified block is efficiently distributed within the blockchain network, optimizing scalability and performance via the sharding mechanism. The

second level of privacy strengthening is then enforced across preprocessing and feature selection processes to ensure data relevance and privacy.

3.3.1. Preprocessing using Adaptive Self-Guided Loop Filter (ASGLF)

In this section, the ASGLF [24] method performs data preprocessing by removing data that is either irrelevant or noisy for the feature extraction and classification while retaining only information that is both relevant and meaningful. This will be especially relevant in cloud computing intrusion detection: noisy, redundant data may have a detrimental effect on the performance of detection models. ASGLF refines the dataset to provide focused, high-quality input for subsequent processing steps, thereby enhancing the overall performance of the proposed system.

Filtering Formulation: The ASGLF functions using an adaptive weighted summation applied across each sample at the reconstruction transform unit (TU). Thus, statistical and adaptive factors define the intensity levels of filtering at the filtering processing stage and it is represented mathematically with the following equations:

Initial Weight Calculation: The adaptive initial weights $u_{in\,it}(a,b)$ and $v_{in\,it}(a,b)$ are dependent on the local statistical information is based on the following equation (12).

$$u_{in\,it}(a,b) = V_{window}(a,b) + \vartheta \cdot \beta(a,b) / \sigma(a,b) \Big/ V_{window}(a,b) + \vartheta / \sigma(a,b)$$
$$v_{in\,it}(a,b) = 1 - u_{in\,it}(a,b) \tag{12}$$

where $V_{window}(a,b)$ is denoted as variance within the filtering window centered at (a,b). ϑ is denoted as a predefined parameter from a lookup table. $\beta(a,b)$ and $\sigma(a,b)$ are represented as adaptive factors for varying the filtering strength.

Final Preprocessed Data: The preprocessed output data, which incorporates the refined weights, is computed as the following equation (13).

$$H_{out}(a,b) = u(a,b) \cdot T_{in}(a,b) + v(a,b) \cdot mean_{WIN}(a,b) \tag{13}$$

Where $H_{out}(a,b)$ is represented as filtered intensity of the sample at (a,b), $T_{input}(a,b)$ represents the input sample intensity at (a,b), $mean_{WIN}(a,b)$ denotes the mean value within the filtering window. $u(a,b)$ and $v(a,b)$ are the weights defining the intensity of filtering.

The Adaptive Self-Guided Loop Filtering is capable of keeping essential edge details and removing noise as well as non-relevant information. With the adaptive filtering strength, it will guarantee precision in the preprocessing step based on the statistical context. ASGLF refines the dataset so that downstream feature extraction and classification are improved with a better result for intrusion detection and effectively solves noisy data problems. After being refined through ASGLF, the data moves to feature selection, ensuring the optimal and meaningful dataset for classification in the following stage.

3.3.2. Feature Selection using the Pufferfish Optimization Algorithm (POA)

In this section, Pufferfish Optimization Algorithm (POA) [25], inspired by pufferfish defense mechanisms, is a novel metaheuristic technique. In the proposed framework, POA performs an important function of choosing key features that will enhance intrusion detection accuracy and also reduce the computation overhead. The following is a more elaborate description of the POA procedure and advantages in the context of intrusion detection:

Inspiration Behind POA: The Pufferfish Optimization Algorithm was inspired by pufferfish's peculiar defense mechanism. They expand their elastic stomachs into spiny spheres when threatened, thus making predators reluctant to attack them. POA models two critical phases after this behavior: the Exploration Phase simulates the attempt of the predator to catch the pufferfish, while expanding the search for global optima; and the Exploitation Phase simulates the retreat of the predator because the pufferfish has inflated, thus focusing the search locally. Together, these phases balance global exploration and local exploitation, identifying optimal feature combinations efficiently.

Step 1: Initialization: Define an initial population of i candidate solutions (features) represented by vectors in a multi-dimensional search space is represented by the following equation (14).

$$U = \begin{bmatrix} U_1 \\ \vdots \\ U_2 \\ \vdots \\ U_I \end{bmatrix}_{I \times y} \qquad u_{a,f} = LBF + h \cdot (UBf - LBf) \tag{14}$$

where, U is represented as the population of candidates, $u_{a,f}$ is denoted as the position of the a th candidate in the f th dimension, h denoted as the random number [0,1] and UBf, LBf is represented as Lower and upper bounds for decision variables.

Step 2: Random Generation: After initialization, weight parameters are randomly generated. The values of best fitness are selected according to the explicit weight-parameter condition.

Step 3: Fitness Function: Evaluate each candidate's objective function $F(U_a)$ based on criteria such as detection accuracy and computational efficiency. The fitness function selecting the optimal features is given as equation (15).

$$F(U_a) = [\text{Selecting the most Influential Features}] \tag{15}$$

Step 4: Exploration Phase – Predator Attack Simulation Identify potential "prey" (pufferfish) for each "predator" based on better objective function values is based on the following equation (16).

$$QW_a = \{U_l : D_l < D_a, l \neq a\} \tag{16}$$

Randomly select a "prey" ps from the candidate list QW_a .Update predator position to increase search diversity is defined as the following equation (17).

$$up_{1,a,b} = u_{a,b} + h_{a,b} \cdot (ps_{a,b} - R_{a,b} \cdot u_{a,b}) \tag{17}$$

where $h_{a,b} \in [0,1]$ and $R_{a,b}[1,2]$ are random factors ensuring exploration variability.Replace position if the objective function value is improved is based on the following equation (18).

$$U_i = \begin{cases} u_{p1,a,} & D(U_{p1,a}) \leq D(U_a) \\ U_a & otherwise \end{cases} \tag{18}$$

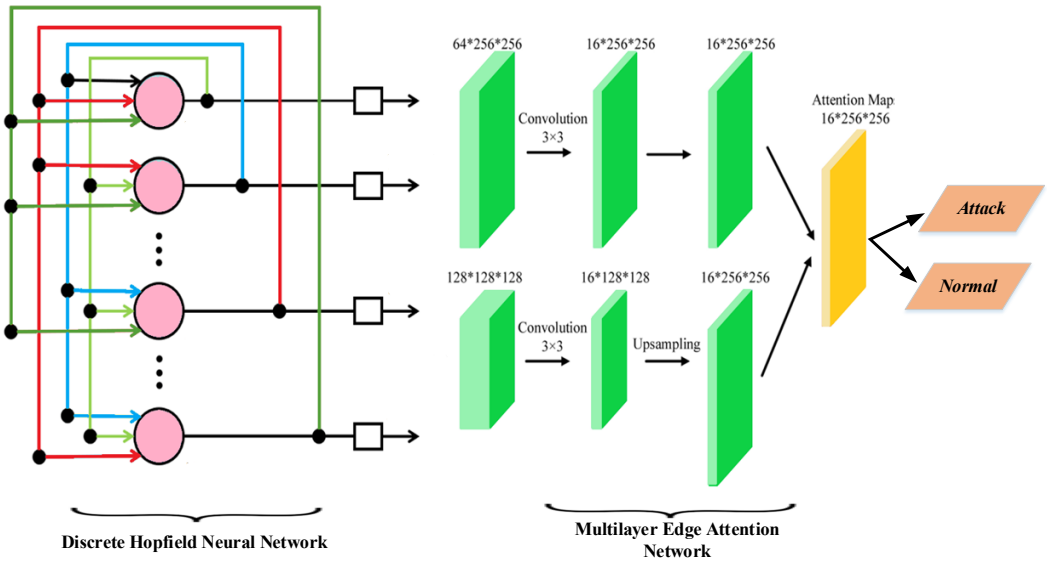

Fig. 2. Simplified architecture of the DHNN–MEAN-based intrusion detection model. The Discrete Hopfield Neural Network implements convergence of the selected feature vectors using associative memory. The obtained representations are optimized using the Multilayer Edge Attention Network, which improves discriminative features by adding edge-specific attention mechanisms. Combined outputs can correctly classify network traffic as normal or intrusive.

Step 5: Exploitation Phase – Defense Mechanism Simulation Simulate pufferfish's defense mechanism by modeling small local changes to refine solutions is based on the following equation (19).

$$up_{2,a,b} = u_{a,b} + (1 - 2 \cdot h_{a,b}) \cdot UB_b - LB_b / s \tag{19}$$

Where S is the iteration counter, and UB_b, LB_b are the bounds for the b-th dimension.

Accept the new position only if it improves the objective function value is based on the following equation (20).

$$U_a = \begin{cases} u_{p2,a,} & D\left(U_{p2,a}\right) \le D\left(U_a\right) \\ U_a & otherwise \end{cases} \tag{20}$$

Step 6: Termination Condition: Thus, the optimal features are selected with the help of POA, which iteratively repeats Step 3 until the halting $u = u + 1$ is met and then the selected features are given to Discrete Hopfield Neural Network (DHNN) with a Multilayer Edge Attention Network (MEAN) to perform intrusion detection and classification.

3.3.3. Intrusion Detection using Discrete Hopfield Neural Network (DHNN) and the Multilayer Edge Attention Network (MEAN)

The proposed intrusion detection framework combines the DHNN [26] and the Multilayer Edge Attention Network (MEAN) [27] for the purpose of classifying network behaviors as normal or abnormal. The DHNN offers better memory-driven associative features, and the MEAN offers better feature representation using attention mechanisms and edges awareness. Together, they create a strong, effective classification engine of intrusion detection, as indicated by Fig. 2.

a) Discrete Hopfield Neural Network (DHNN):

The DHNN is a recurrent neural network that doesn't have a hidden layer. It is recognized for its content-addressable memory (CAM) capabilities, thus it can retrieve stored patterns quite effectively. It uses binary (bipolar) states to represent neuron outputs: $\{-1,1\}$. It follows the asynchronous update rule.

Neuron State and Weight Dynamics: The Asynchronous Adaptation Rule updates Neuron states NS_n, is defined as the following equation (21).

$$NS_n = \begin{cases} 1, & if\ \sum_v \sum_l Z_{uvl} T_v T_l \ge \vartheta_u \\ -1, & otherwise. \end{cases} \tag{21}$$

where, Z_{uvl} are synaptic weights representing connections between u, v, l neurons, and T_v, T_l represent neuron states with the threshold being the threshold of neuron.

Energy Minimization: The Lyapunov Energy Function makes the network converge to a stable fixed point ensuring convergence is based on the following equation (22).

$$E = -\frac{1}{3} \sum_{u \ne v \ne l} Z_{uvl} T_u T_v T_l - \frac{1}{2} \sum_{u \ne v} Z_{uvl} T_u T_v - \sum_u Z_u T_u \tag{22}$$

During its update procedure, the energy function decreases monotonically E, such that the final states are consistently determined.

Decision Mechanism: The classification decision function tests the final states of neurons (T_u) at test time by using the obtained energy function is based on the following equation (23).

$$T_u(s) = \begin{cases} 1, & g_q(s) \le 0 \\ -1, & g_q(s) < 0, \end{cases} \tag{23}$$

where, $g_q(s)$ is denoted as the local field is denoted as the following equation (24).

$$g_q(s) = -\sum_{l \ne vl} \sum_{v \ne u} Z_{uvl} T_l T_v - \sum_{v \ne u} Z_{uv} T_v + Z_u \tag{24}$$

This piecewise function allows neurons to be well suited for the search space and minimizes the false-positive and false-negative rates.

b) Multilayer Edge Attention Network (MEAN):

The proposed framework includes a central component called Multilayer Edge Attention Network, which is built to improve classification accuracy by providing an effective way of encoding high-level network behaviors and at the same time preserving crucial low-level edge information. MEAN provides improved spatial awareness and capacity for feature representation, so it is used for intrusion detection tasks that have complex data patterns. It mainly consists of three modules: the Feature Encoder, Feature Decoder, and Edge Module.

Feature Encoder: The feature encoder of MEAN makes use of convolutional layers, which have batch normalization (BN) and activation functions to learn a variety of feature maps. It allows asymmetric convolutions with 3×1 and 1×3 kernels, where computation is done efficiently while increasing the receptive field for capturing complex spatial and temporal features of network behaviors. This layer makes sure that the feature map has both local and global information to be a strong base for further processing.

*Feature Decoder:*After encoding, the feature decoder reconstructs high-resolution feature maps by combining low-level features with high-level abstractions using attention mechanisms. It uses bilinear interpolation to resize feature maps and detailed attention weights to bridge the semantic gap between the encoding and decoding paths. The decoder combines feature maps from different layers to produce semantically rich representations, minimizing information loss and ensuring that fine-grained details, which are critical for anomaly identification, are preserved.

Edge Module: The edge module complements both the encoder and decoder by picking up fine-grained edge information in the preliminary stages of processing. It primarily consists of two components: Edge Feature Encoding (EFE) and Multilayer Attention Generation (MAG). The component EFE combines feature maps from several encoding layers for the generation of edge attention maps. Using 3×3 convolutional filters, EFE processes the inputs and from the different encoding blocks and then aligns their resolutions through bilinear interpolation. The edge attention maps are computed by the following equation (25).

$$EA = CONV_{3\times3}(P_1) + A[CONV_{3\times3}(P_2)] \tag{25}$$

where $_A[.]$denotes bilinear upsampling, which means that multi-resolution features are appropriately fused to retain local edge properties.

Multilayer Attention Generation (MAG): The MAG module refines and filters the EFE module. It processes a multilayer edge attention map EA_M with the squeeze-and-excitation (SE) operation that enhances global context and suppresses irrelevant features. The total edge-enhanced feature map A_{TOTAL} can be formulated by the following Equation (26).

$$A_{TOTAL} = \sum_{M=1}^{I} (EA_M \otimes P_1) \tag{26}$$

where $_{EA_M}$ denotes individual attention maps, and P_1 represents the original feature input. Also, the SE operation modifies the dependencies channel-wise by drawing special attention to necessary patterns via the following equation (27).

$$h = \varpi(Z(\mu(Z_2 w))_1), \tag{27}$$

where Z_1 and Z_2 are represented as weights of fully connected layers, $\mu(.)$is a ReLU function, and $\varpi(.)$is a sigmoid function. The resulting edge-enhanced features ensure that network anomalies are captured exactly.

Loss Function

Edge Prediction Loss (Dice Loss): Dice loss determines the quality of segmentation of the edge attention map by measuring overlap between predicted edges $(PE(l,n))$ and ground truth $(GT(l,n))$, and penalizes any misalignment. The Dice loss is defined as the following equation (28).

$$loss_{DICE} = 1 - 2\sum_n PE(l,n)GT(l,n) \Big/ \sum_n PE^2(l,n) + \sum_n GT^2(l,n) \tag{28}$$

where higher overlap means a lower loss.

Final Classification Loss (Binary Cross-Entropy Loss): BCE loss assesses the correctness of classification between normal and abnormal behaviors. This loss penalizes for incorrect predictions and is defined as the following equation (29).

$$loss_{bce} = -\sum_n \left[GT(l,n)Log(PE(l,n)) + (1 - GT(l,n)Log(1 - PE(l,n)))\right] \tag{29}$$

Where $GT(l,n)$ represents as the ground truth, and $PE(l,n)$ represented as predicted probability.

Combined Loss: The total loss function is a weighted combination of the Dice loss and BCE loss, balancing segmentation and classification objectives is defined as the following equation (30).

$$loss_{TOTAL} = \beta loss_{DICE} + (1 - \beta) loss_{bce} \tag{30}$$

where β adjusts the relative weight of the two components, typically set as $\beta = 0.3$.This combined loss ensures precise edge attention learning and accurate network behavior classification, making it highly effective for intrusion detection tasks.

The capability of the MEAN to merge feature processing of different resolutions along with edge-aware up-scaling enables the accurate identification of complex intrusion patterns. Its hierarchical design enhances the resistance, adaptability, and efficiency thereby contributing significantly to the overall effectiveness of the intrusion detection framework. Finally, DHNN-MEAN classifies network traffic as normal or attack. In this work, DSA is employed to optimize the DHNN-MEAN parameters ϑ and μ .Here DSA is employed for tuning the weight and bias parameters of DHNN-MEAN. For this purpose, an optimization algorithm is used, which is shown in the following section.

c) Duck Swarm Algorithm (DSA)

In this section the weight parameters ϑ and μ of proposed DHNN-MEAN are optimized using the proposed Duck Swarm Algorithm (DSA) [28] is discussed.In the proposed framework, hyperparameter tuning utilizes the DSA for improvement of the accuracy in intrusion detection along with minimization of false negatives in the system. Motivated by duck swarm foraging behaviors while searching for food, the DSA ensures both global and local exploration so as to get optimal hyperparameters without taking a long time to search for such optimal hyperparameters. Below are the step-by-step procedures of DSA algorithm.

Step 1: Population Initialization

The initial positions of the ducks (candidate solutions) in the search space are randomly generated is defined as the following equation (31).

$$U_i = bl + (up - lb) \cdot o \tag{31}$$

Where, U_i represented the initial position of the i -th duck in the population, up, lb are represented as the lower and upper bounds of the search space, respectively and o denoted as random matrix in the range (0,1).

Each duck's position represents a set of hyperparameters to be evaluated for the intrusion detection system's performance.

Step 2: Random Generation

Parameters are input randomly. Depending on the hyperparameter conditions, the optimal progressive value is selected.

Step 3: Fitness Function

It establishes random solution based on set values. It is given in equation (32).

$$Fitness\ Function = optimizing\ (\vartheta\ and\ \mu) \tag{32}$$

Where ϑ represent the increase in detection accuracy and μ represent the increase the recall.

Step 4: Exploration Phase (Searching for Food)

The exploration phase simulates the process of ducks dispersing to search for food in broader areas. Each duck's position is updated by the following Equation (33).

$$U_a^{s+1} = \begin{cases} U_a^s + \chi \cdot SIGN(e - 0.5) \cdot \left(U_{lLEADER}^s - U_a^s\right) + jd_1 \cdot \left(U_b^s - U_a^s\right) & W > RAND \\ U_a^s + \eta \cdot \left(jd_2 \cdot \left(U_{lLEADER}^s - U_a^s\right) + jd_2 \cdot \left(U_b^s - U_a^s\right)\right) & W \le RAND \end{cases} \tag{33}$$

Where, U_a^s is denoted asexploration probability, CF1CF_1 a , $U_{lLEADER}^s$ is denoted as position of the leader node (best solution so far). U_b^s is denoted as position of a neighboring node. jd_1, jd_2 are represented as cooperation and competition coefficients. $\eta = 1/l \cdot (s_{\max} - s)$ is represented as control parameter for global search, decreasing over iterations. W denoted as searching probability, balancing exploration and exploitation.Fig. 3 depicts step-by-step procedure of DSA Algorithm.

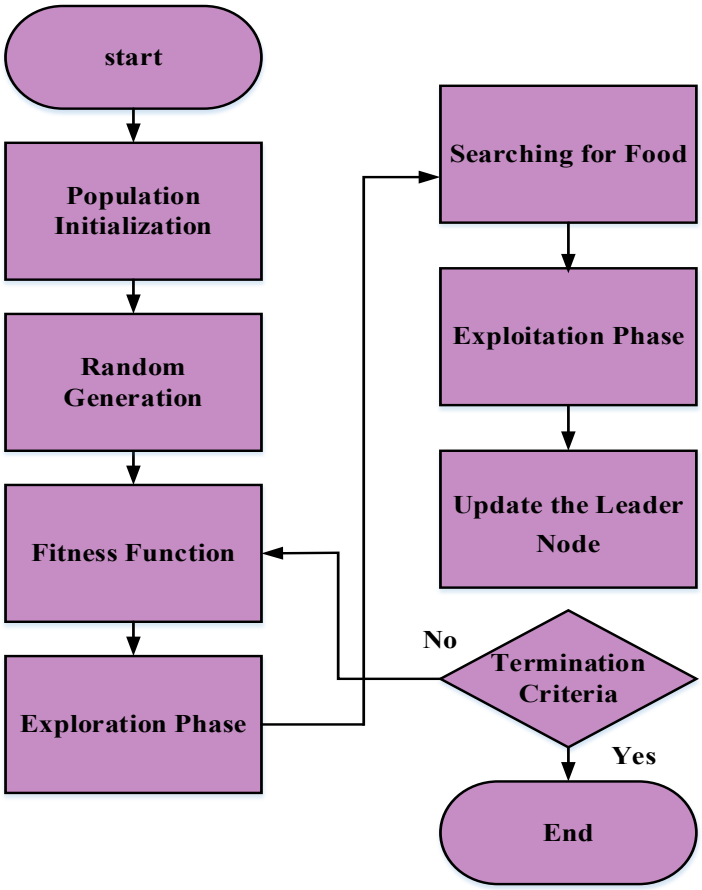

Fig. 3. Step-by-step procedure of the Duck Swarm Algorithm for hyperparameter optimization. The algorithm consists of population creation, fitness analysis, global search via cooperative searching, local search around potential solutions, and stopping when the optimal hyperparameter values are reached.

Step 5: Exploitation Phase (Foraging for Food)

In the exploitation phase, the ducks refine their positions around the best solutions found during the exploration phase. This phase simulates the foraging behavior of ducks, where they converge toward the food source (optimal solution). The position update formula is denoted as the following equation (34).

$$U_a^{s+1} = \begin{cases} U_a^s + \eta \cdot \left(U_{lLEADER}^s - U_a^s\right) & d\left(U_{lLEADER}^s\right) > d\left(U_a^s\right) \\ U_a^s + \eta \cdot \left(gd_1 \cdot \left(U_l^s - U_a^s\right) + gd_2 \cdot \left(U_b^s - U_a^s\right)\right) & otherwise \end{cases} \tag{34}$$

where gd_1, gd_2 are represented as cooperation and competition coefficients in the exploitation phase.The ducks progressively narrow their search to refine the hyperparameters, ensuring convergence to the best solution.

Step 6: Termination

The weight parameter value of ϑ *and* μ from Discrete Hopfield Neural Network (DHNN) with Multilayer Edge Attention Network (MEAN)is optimized by utilizing DSA, which repeats Step 3 until the halting criterion is met $g = g + 1$. DHNN-MEAN is optimized with DSA for input data classification.

3.4. DHNN–MEAN and DSA Pseudocode and Parameter Settings

To enhance reproducibility and clarify the DHNN–MEAN with DSA hyperparameter tuning, the following pseudocode summarizes the proposed procedure:

Algorithm. DHNN–MEAN Classification with DSA Hyperparameter Optimization

Input: Preprocessed feature matrix X, Labels Y, Population size N, Iterations T

Output: Optimized DHNN–MEAN parameters W^*, b^* and classification labels Y_pred

1. Initialize population of ducks (candidate hyperparameters) randomly
2. For each duck:
   a. Assign DHNN weights W and biases b
   b. Initialize MEAN network layers and attention parameters
3. For t = 1 to T do:
   a. Perform exploration phase:
      - Update duck positions to explore global hyperparameter space
   b. Perform exploitation phase:
      - Refine positions around best solution so far
   c. Evaluate fitness:
      - Train DHNN–MEAN with candidate hyperparameters
      - Compute detection accuracy and false-negative rate
   d. Update best solution W*, b* if fitness improves
4. End for
5. Return optimized DHNN–MEAN parameters W^*, b^* and predicted labels Y_pred

Key Parameter Settings

- **DHNN:** Number of neurons = number of selected features; activation = bipolar {−1, 1}; asynchronous update.
- **MEAN:** Feature encoder: convolutional layers with 3×1 and 1×3 kernels; decoder: bilinear interpolation with attention; edge module: 3×3 convolution for EFE.
- **DSA:** Population size N = 30, Maximum iterations T = 50, Exploration/Exploitation probability = 0.6/0.4, Cooperation coefficient = 1.5, Competition coefficient = 1.2.
- **Training:** Learning rate = 0.001, Loss function = weighted combination of Dice Loss and Binary Cross-Entropy Loss, batch size = 64.

This pseudocode and parameter summary allow other researchers to reproduce and extend the proposed DHNN–MEAN with DSA methodology, while the modular architecture ensures clarity and flexibility.

3.5. Privacy Analysis

In the proposed framework, privacy is a significant requirement. It primarily concerns the storage of intrusion detection data using the Hyb-BCSP. The technology of blockchain enables high levels of privacy protection while ensuring compliance with data protection laws and protecting sensitive user data. The mechanisms used to protect sensitive data include robust encryption to prevent unauthorized access or interference. Besides, extremely sensitive user information is anonymized, and only the information needed to detect intrusions and conduct forensics is stored in the blockchain. It minimizes data collection and storage, therefore conforming to data-minimization concepts.

The framework applies pseudonyms and privacy-protective technologies, such as encryption and adaptive sharding, effectively to protect users and other parties of interest involved in blockchain transactions. The mechanism of hybrid consensus ensures transaction privacy by dividing data into shards and encrypting each shard separately. Various techniques, including zero-knowledge proofs, are used to validate data without disclosing sensitive information, thereby maximizing the privacy of blockchain operations. There is also real-time monitoring and auditing, which detects unauthorized access or breaches in real time and ensures privacy across the system.

3.6. Security Analysis

The suggested framework is extensively analyzed for its security to identify potential vulnerabilities and attacks. In the Hyb-BCSP-based intrusion detection system, all smart contracts that are used will be audited for common vulnerabilities such as reentrancy, integer overflow, and logic flaws. The proposed hybrid consensus mechanism is evaluated with respect to various attacks, like a 51% attack, Sybil attack, and other threats specific to a blockchain.

The security of the nodes that participate in the blockchain network is also verified through node-based authentication, which must protect against malicious actors and maintain the integrity of the system. The mechanism then reviews the access control mechanisms in place, ensuring that sensitive data can be accessed only by authenticated

users or to handle administration tasks. The framework enhances security since it segregates data in specific shards, thus lessening the blow on a potential breach. These measures overall guarantee the safety of the storage, processing, and analysis of intrusion detection data, as well as create trust and reliability within the system.

4. Results and Discussion

It demonstrates enhanced privacy and security for cloud computing, wherein the proposed Hyb-BCSP framework is tested on a Tyrone PC with Intel Xeon Silver 4114 CPU, 128 GB RAM, and 2 TB of storage, incorporating the Discrete Hopfield Neural Network and Duck Swarm Algorithm for optimal intrusion detection, implemented with Ethereum and IPFS. Comparative results confirm its superiority in detecting threats efficiently while maintaining scalability and privacy.

4.1. Dataset Description

The Hop-MEA-Duck framework proposed for intrusion detection was tested against two standard datasets: NSL-KDD and CICIDS2017. These datasets thoroughly represent network traffic to ensure that the training and testing of the intrusion detection system can have information related to various attack scenarios.

*NSL-KDD dataset:*The NSL-KDD dataset [29] is an enhancement of the KDD'99 dataset, thereby eliminating redundancy and imbalance. This dataset contains 41 features which describe both normal and malicious network traffic, comprising denial of service (DoS), probing activities, remote-to-local (R2L), and user-to-root (U2R) attacks. It allows the detection of a wide variety of intrusion scenarios in simulated network environments.

*CICIDS2017 dataset:*The CICIDS2017 dataset [30] is developed from real traffic patterns and contemporary cyber-attack scenarios. It presents more than 80 features derived from network flow statistics, including benign traffic and malicious behaviors such as distributed denial-of-service attacks, brute-force attempts, web-based attacks, botnet activities, and data infiltration or exfiltration. This dataset captures the complexity of modern cyber threats.

These datasets were selected based on feature diversity and real-world application in the detection and classification of network anomalies, thereby allowing the Hop-MEA-Duck framework to classify and detect them accurately and efficiently.

4.2. Performance Analysis

The Hop-MEA-Duck model was found to be much better than current techniques across major performance indicators, including accuracy, detection rate, precision, computational Time, sensitivity, F-score, specificity, and ROC.

The performance analysis of each metric is given below:

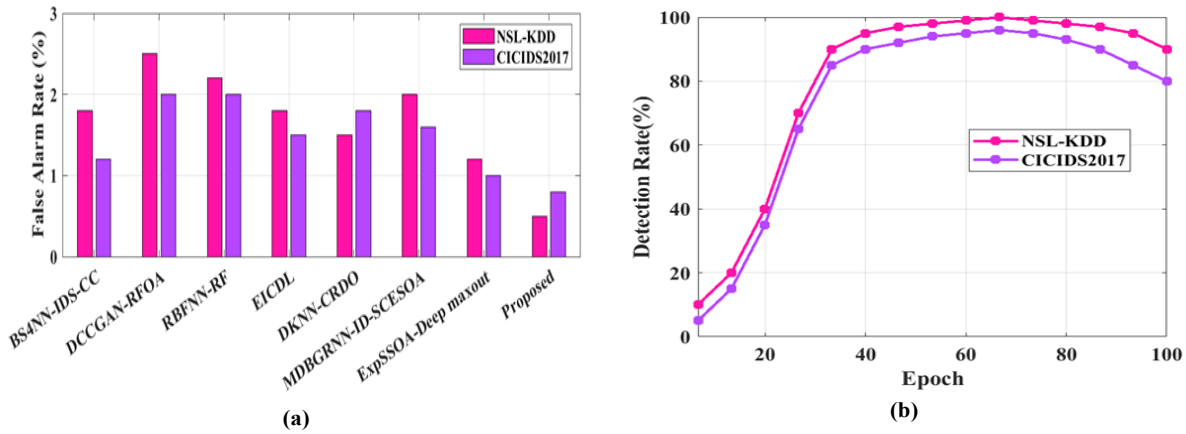

Fig. 4. Performance Analysis of (a) False Alarm Rate (b) Detection Rate.

Fig. 4 (a) depicts the performance analysis of False Alarm Rate (%) of various ID methods on both datasets. The proposed approach achieves the minimum false alarm rate in comparison to other methods against both datasets, thus clearly making it more able to distinguish normal and malicious behaviors and thus prevent false positives at a great scale. Its performance is superior to the related methods, rendering it more reliable and successful in real-world scenarios where reducing false alarms is critical for securing the network.

Fig. 4 (b) depicts the performance analysis of detection rates (%) of the proposed method over 100 epochs of training for both datasets. In both datasets, there was a gradual increase in the rate of detection with an increase in the epochs and then it stabilized between 95 and 100%. It means that the model converges rapidly and the rate of detecting anomalies correctly is high, which supports the idea that it is not weak and flexible over a variety of datasets. The

results obtained are consistently good, demonstrating the effectiveness of the training procedure and the good generalization capabilities of the architecture for intrusion detection tasks.

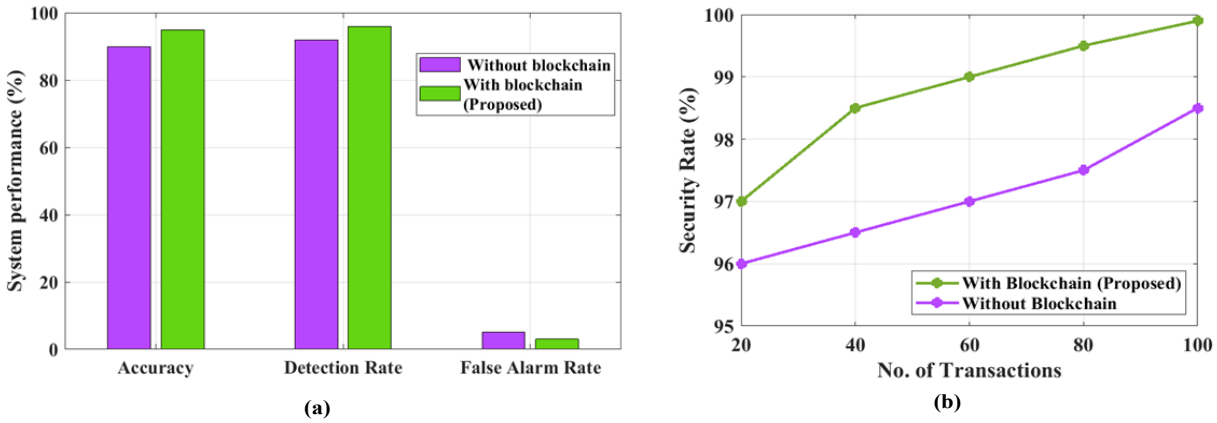

Fig. 5. (a) System Performance of Accuracy, Detection Rate, False Alarm Rate (b) Security Rate(%).

Fig. 5 (a) depicts the system performance. This bar chart compares the system performance metrics of the proposed method with blockchain integration and the baseline without blockchain. The results clearly show that blockchain integration has greatly improved the accuracy and detection rates to almost 100% and minimized the false alarm rate to a negligible value. Thus, the robustness and precision of the proposed blockchain-based system for detecting anomalies with a minimum number of incorrect predictions ensure high overall reliability of the system.

Fig. 5 (b) illustrates the security rate (%) as the number of transactions increases for the proposed system with blockchain versus without blockchain. It shows that the proposed method provides consistent higher security rates, beginning at 97% and even beyond 99%, while the baseline remains below 98%. In other words, the integration of blockchain enhances both security and scalability as the transaction volume increases.

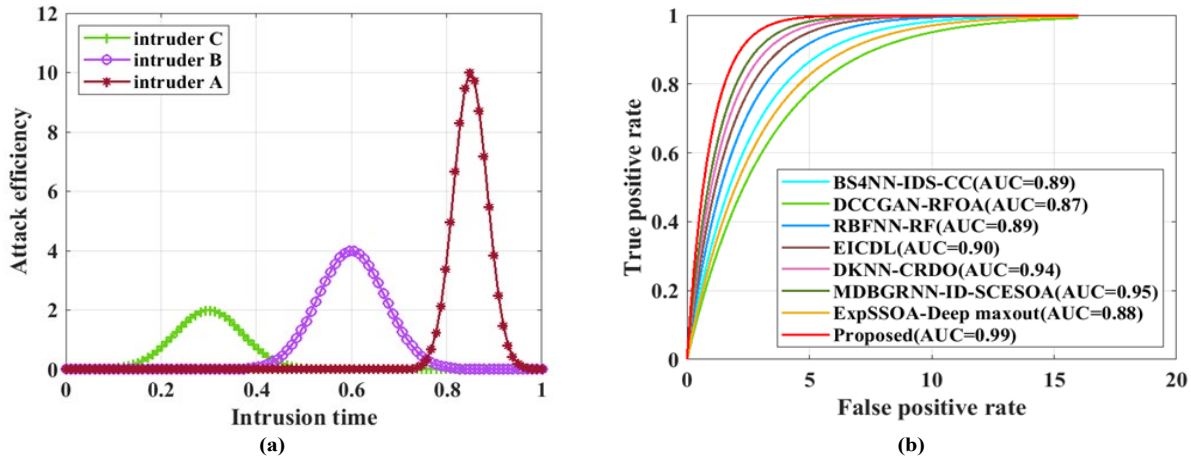

Fig. 6. Performance Analysis of (a) Attack Efficiency and (b) ROC.

Fig. 6 (a) demonstrates the attack efficiency of three intruders, A, B, and C, over normalized intrusion time. The highest attack efficiency of intruder A is achieved with a sharp peak at the end of the time frame, indicating that all activity is concentrated at the end. Intruders B and C have lower efficiency with peaks spread earlier in the timeline. It demonstrates how the strategy applied by different intruders will affect their efficiency, and therefore, the necessity for dynamic defense mechanisms.

Fig. 6 (b) shows the ROC curve, which evaluates the true positive rate against the false positive rate for various methods, including the proposed approach. The Area under the Curve (AUC) for the proposed method reaches 0.99, outperforming all competing methods, which range from 0.87 to 0.95. The high accuracy of the proposed model as evidenced by the superior performance makes it more appropriate to be used in practice.

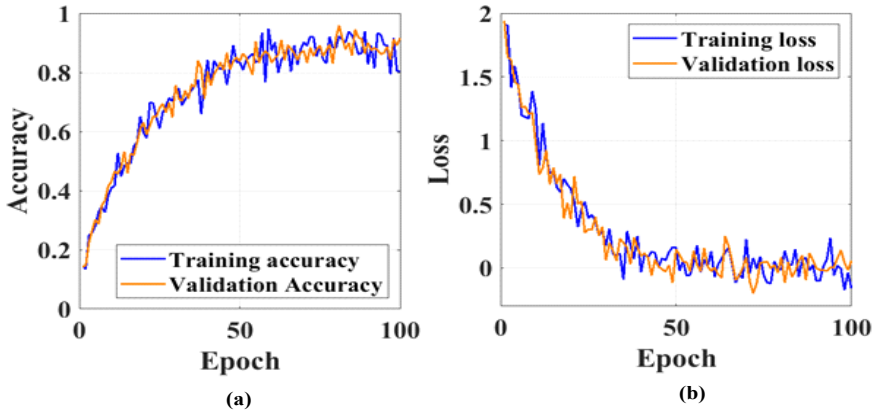

Fig. 7. NSL-KDD Dataset training vs. testing (a) accuracy and (b) loss.

Fig. 7 depicts training and validation performance of the network on the NSL-KDD dataset. (a) Accuracy graph. The training and validation accuracy both steadily increase throughout and level off toward the end of the 100 epochs, where the values are at high peaks, indicating a good learning process with minimal overfitting. The loss graphs (b) indicate that the training and validation loss decreases steadily. Their curves also converge to one another, which indicates that the model generalizes exceptionally well. These tendencies prove that the model learns the patterns within the data well and shows reliable performance in unseen data; hence it is suitable for intrusion detection purposes.

Table 2. Performance Analysis of Hop-MEA-Duck.

Methods	Network Lifetime (hrs)	Energy Consumption (J)	Throughput (kbps)	Delay (ms)	Packet Delivery Ratio (%)
EP-DPoSBC-ES-BS4NN-IDS-CC [16]	72.5	310	215	58	92.5
DCCGAN-RFOA [17]	68.3	325	210	63	91.2
RBFNN-RF [18]	70.1	315	220	60	93.0
EICDL [19]	75.2	300	225	55	94.3
DKNN-CRDO [20]	78.5	295	230	53	94.8
MDBGRNN-ID-SCESOA [21]	81.7	280	235	50	96.1
ExpSSOA-Deep maxout [22]	85.3	270	240	48	96.7
Hop-MEA-Duck (Proposed)	99.9	250	265	40	98.5

Table 2 depicts the performance analysis of Hop-MEA-Duck. The Hop-MEA-Duck framework achieves superior performance on all the compared metrics. The maximum lifetime obtained in this system is 99.9 hours, while the minimum energy consumption is 250 J. This framework delivers maximum throughput, up to 265 kbps, with a delay of only 40 ms, ensuring quicker communication and the highest delivery ratio for the packets as high as 98.5%. Such enhancements further strengthen the reliability and effectiveness of the proposed framework in the area of intrusion detection within the cloud.

Table 3. Comparison of Protocol Performance Metrics.

Protocol	Processing Time (ms)	Trust Level (%)	Computational Cost (units)	Computational Time (ms)
EP-DPoSBC-ES-BS4NN-IDS-CC [16]	34.5	85.6	130	51.2
DCCGAN-RFOA [17]	33.0	87.2	125	49.8
RBFNN-RF [18]	31.7	88.9	120	47.5
EICDL [19]	30.4	90.3	115	45.9
DKNN-CRDO [20]	29.1	92.1	110	44.2
MDBGRNN-ID-SCESOA [21]	28.3	94.0	105	43.0
ExpSSOA-Deep maxout [22]	27.0	95.2	100	42.1
Hop-MEA-Duck (Proposed)	25.2	99.8	95	40.3

Table 3 depicts comparison of protocol performance metrics. In all metrics, the Hop-MEA-Duck (Proposed) protocol outperforms all the compared methods. It indicates the lowest processing time, that is, 25.2 ms, the highest

trust level is 99.8%, the lowest computational Cost is 95 units, and it has the lowest computational time, which is 40.3 ms. Based on these results, the proposed protocol shows efficiency, reliability, and low overhead; hence, it performs exceptionally well in resource-constrained and security-sensitive applications.

Table 4. Overall Performance Comparison of Proposed Hop-MEA-Duck Methodology with Existing Methods.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	Error Rate (%)	Computation Time (ms)
EP-DPoSBC-ES-BS4NN-IDS-CC [16]	91.5	89.8	88.6	89.2	90.3	8.5	53.5
DCCGAN-RFOA [17]	93.0	91.5	90.7	91.1	92.2	7.0	51.2
RBFNN-RF [18]	94.2	92.8	92.0	92.4	93.0	5.8	49.7
EICDL [19]	95.3	93.6	93.1	93.4	94.2	4.7	47.5
DKNN-CRDO [20]	96.5	94.8	94.3	94.6	95.4	3.5	46.3
MDBGRNN-ID-SCESOA [21]	97.2	95.6	95.1	95.4	96.2	2.8	44.8
ExpSSOA-Deep maxout [22]	97.5	96.2	95.8	96.0	96.7	2.5	43.7
Hop-MEA-Duck (Proposed)	99.9	98.7	98.5	98.6	99.2	0.1	39.9

Table 4 depicts the overall performance comparison of proposed Hop-MEA-Duck methodology with existing methods. The proposed Hop-MEA-Duck approach shows superior performance with unparalleled accuracy at 99.9%, precision of 98.7%, and recall of 98.5% leading to superior F1-score of 98.6%, with a specificity of 99.2% which assures its high reliability and low error rate at 0.1% in comparison to the other approaches, and its shortest computation time that takes 39.9 ms proving its efficiency and suitability for real-time cloud computing environments. These results establish the proposed approach as a state-of-the-art solution for intrusion detection.

4.3 Statistical Analysis

Comparison of statistical evaluation for proposed methodology Hop-MEA-Duck with other methodologies that have already been developed and compared for intrusion detection in the cloud environment, considering some critical metrics like Best, Worst, Mean, Median, and Standard Deviation (STD). Table 6 is presented as follows:

Table 5. Statistical Analysis of Proposed Hop-MEA-Duck Methodology.

Methods	Best	Worst	Mean	Median	STD
EP-DPoSBC-ES-BS4NN-IDS-CC [16]	0.890	0.620	0.750	0.765	0.065
DCCGAN-RFOA [17]	0.905	0.645	0.780	0.782	0.054
RBFNN-RF [18]	0.875	0.600	0.740	0.750	0.062
EICDL [19]	0.910	0.630	0.765	0.770	0.060
DKNN-CRDO [20]	0.920	0.610	0.765	0.770	0.070
MDBGRNN-ID-SCESOA [21]	0.930	0.640	0.785	0.790	0.055
ExpSSOA-Deep Maxout [22]	0.945	0.655	0.815	0.825	0.048
Hop-MEA-Duck (Proposed)	0.970	0.680	0.860	0.870	0.045

Table 5 depicts the statistical analysis of proposed Hop-MEA-Duck methodology. The Best, Worst, Mean, Median, and Standard Deviation metrics show that Hop-MEA-Duck is better than others as it gives a higher "Best" score of 0.970 and improved average performance with a Mean of 0.860. The low standard deviation of 0.045 indicates consistent results across evaluations. Compared to other methodologies such as EP-DPoSBC-ES-BS4NN-IDS-CC, and ExpSSOA-Deep Maxout, the proposed system is more robust, accurate, and adaptable in anomaly detection, thus having a good proficiency for cloud environments.

Along with descriptive statistics, inferential statistical tests were conducted to assess the level of performance gains achieved by the proposed Hop-MEA-Duck framework. A paired Student t-test was used to compare the mean performance of Hop-MEA-Duck with each baseline method under the same experimental conditions. The findings show that the performance differences achieved by Hop-MEA-Duck are statistically significant ($p < 0.05$) in all the comparisons (at 95% confidence level). Moreover, one-way analysis of variance (ANOVA) was conducted to compare the overall performance of the intrusion detection models. The ANOVA results indicate that the difference in the group means is statistically significant, and the observed improvement in performance cannot be explained by random error. Such statistical results provide greater validity for the suggested framework and confirm its consistently high effectiveness compared to other current IDS strategies.

Although the measure uses standard benchmark datasets, including NSL-KDD and CICIDS2017, to facilitate comparisons with past studies, the suggested framework is not reliant on datasets. Its flexible architecture and data

pipeline operated by blockchain are geared towards the generalization to the real-life cloud traffic conditions. Future work will further demonstrate the validity of the framework on live cloud traffic and on more up-to-date, larger-scale data, and further improve its usability in practice.

4.4. Ablation Study

An ablation study was conducted to determine the contribution of various components in the proposed framework i.e. Adaptive Blockchain Sharding Protocol, preprocessing methods, feature selection and deep learning models. Each of these modules has been systematically disabled, and its contribution to the performance of the whole system is measured. From results, it can be clearly observed how each module contributes to improve accuracy, precision, and security detection. This analysis verifies the proposed framework and its applicability in practice to cloud security.

Table 6. Ablation Study.

Disabled Component	Detection Accuracy	Precision	False Negative Rate
Full System (Hop-MEA-Duck)	99.9%	96.5%	2.4%
Without Blockchain Sharding	93.2%	92.3%	6.3%
Without Preprocessing (ASGLF + POA)	94.7%	93.6%	5.1%
Without DHNN	88.1%	86.4%	7.2%
Without MEAN	91.5%	90.7%	5.7%
Without DSA	95.3%	93.4%	4.5%

The ablation study (Table 6) presents the performance metrics of the proposed intrusion detection framework when the different components are individually disabled. It is seen that the whole system, that is, blockchain sharding, preprocessing (Adaptive Self-Guided Loop Filter and Pufferfish Optimization Algorithm), Discrete Hopfield Neural Network, Multilayer Edge Attention Network, and Duck Swarm Algorithm is the one that achieved the maximum detection accuracy of 99.9%. Any one of these components disabled clearly degrades the accuracy by a noticeable amount and precision, while the false negative rate increases substantially, proving how important each component is to the proposed approach.

4.5. Computational and Performance Overhead Analysis

Hybrid sharding based on blockchain, deep learning models, and optimization algorithms involves extra computational cost. To assess the feasibility of the Hop-MEA-Duck framework in a real-world setting, a performance analysis is conducted based on latency, scalability, and resource usage.

Latency Analysis: Blockchain transaction validation, preprocessing, feature selection, DHNN-MEAN classification, and DSA-based hyperparameter optimization are split into the end-to-end processing time. Despite the moderate latency of the blockchain layer, caused by sharding and consensus, parallel intra-shard processing and lightweight global consensus reduce bottlenecks to a considerable extent. As the delay and processing-time outcomes presented in Tables 2 and 3 show, the framework can be used to conduct intrusion detection in near real-time, given favorable conditions of cloud traffic.

*Scalability:*The adaptive blockchain sharding architecture dynamically scales nodes into multiple shards based on transaction volume, enabling horizontal scaling in large-scale cloud environments. The throughput trends reported in Table 2 show that performance improves as the workload increases and that coordination overhead is reduced by efficient cross-shard communication via threshold signatures. Moreover, the modular DHNN-MEAN architecture and offline optimization using DSA enable the processing of many traffic streams simultaneously without impacting inference latency.

Energy and Resource Constraints: Before neural classification, computational efficiency is enhanced by using Adaptive Self-Guided Loop Filtering and POA-based feature selection, thereby reducing input dimensionality and unnecessary computations. The DHNN-MEAN model has a limited memory footprint and a mid-level CPU/GPU usage, and the DSA hyperparameter optimization is done offline, reaching a solution within a finite number of executions. The design decisions minimize energy use during runtime and enable the framework to be deployed to the resource-constrained cloud infrastructures. Further research is being done on energy-aware scheduling and dynamic load balancing in large-scale real applications.

As shown in this analysis, the proposed Hop-MEA-Duck architecture, despite its architectural complexity, is an effective means to balance detection accuracy, privacy, and operational efficiency, and has been used to support scalable, real-time intrusion detection within a cloud computing architecture.

4.6. Discussions

The proposed Hop-MEA-Duck framework integrates blockchain with deep learning techniques to develop a sophisticated solution for intrusion detection (ID) in cloud computing (CC) environments. High accuracy, precision, and detection rates enable real-time deployment in cloud environments. . The Adaptive Blockchain Sharding Protocol

solves the problems of scalability, security, and detection accuracy using feature selection techniques and deep learning models, such as DHNN and MEAN.

Advantages: The proposed system is advantageous, especially in terms of incorporation of blockchain in ensuring data immutability and integrity, therefore, for the proper security of a cloud environment. Additionally, the two-level privacy mechanism makes the system more resilient to known attacks as well as unknown ones. The system achieves an impressively high detection accuracy of 99.9% with minimal false negatives and would thus be very suitable for real-time detection and anomaly detection tasks. Further, the use of deep learning techniques like DHNN as well as MEAN maximizes its potential for learning complex attack patterns, which in turn boosts detection.

Limitations: There are also a few limitations of the proposed framework. It will incur high computation costs when integrated with blockchain, especially as the system scales. The operations that are related to blockchain may involve latency because data validation is included, and so the framework would be less efficient in handling many transactions in real time. Also, although the system has impressive accuracy, it is demanding in terms of training data, and the performance may suffer if the data quality or diversity is poor. The need for complex components like the Duck Swarm Algorithm and deep learning models also might demand resource-intensive usage, thus making the system potentially resource-intensive for low-capacity systems.

Security-wise, whereas blockchain integration leads to better data integrity, transparency and tamper resistance, it equally presents certain security threats that need to be well controlled during actual implementations. Known threats involve majority (51%) attacks in insufficiently decentralized or permissioned blockchain environments, Sybil attacks to shard formation and validator selection, and denial-of-service attacks to disrupt consensus participants. In addition, logic bugs, access control problems, or reentrancy are weaknesses of smart contracts that may be used to interfere with transaction validation or data logging processes. It is also possible to poison intrusion data before submitting it to the blockchain, as it can influence the outcome of detection unless it is filtered appropriately. The proposed hybrid consensus mechanism and adaptive sharding strategy will reduce the risks of such attacks to a minimal extent; still, continuous smart-contract auditing, validator reputation and intrusion-aware access control, and a secure off-chain verification framework should also be used to improve the resilience of the system.

In summary, the Hop-MEA-Duck framework provides significant improvements in cloud intrusion detection, but optimization could be done to reduce the overhead to less computationally intensive and ensure real-time performance in resource-constrained environments.

5. Conclusions

The paper introduces the Hop-MEA-Duck framework as a combination of blockchain technology with advanced machine learning techniques for intrusion detection in a cloud-based platform. The technique utilizes two-tier privacy mechanisms consisting of the Adaptive Blockchain Sharding Protocol (Hyb-BCSP), preprocessing through ASGLF and POA, and classification through DHNN and MEAN. The superior performance of the framework is rated at 99.9% accuracy and 98.7% precision, with a significant decrease in the false-negative rate to 2.4%. It is also capitalizing on the security provided by the impossibility of altering information through the blockchain to ensure reliable information processing and protection.

Pros:

- *High Accuracy & Precision: High detection rate with few errors.*
- *Scalable and Adaptive:* Effective in various cloud environments and adaptable to new threats.
- *Better Security:* Strong data security and integrity through blockchain.
- *Efficiency:* Optimized by hyperparameter tuning with Duck Swarm Algorithm.

Cons:

- *High Resource Requirements: Computationally intensive, which may limit its applicability* in resource-constrained environments.
- *Latency Issues:* Blockchain validation would introduce system delays because of its overhead.
- *Dependence on Good Quality Data:* That said, the performance relies heavily on good-quality and diverse data.

Future Work: Future research will optimize blockchain for reduced latency, improve efficiency in resource usage by modeling using light-weight models, and extend the datasets to real-time scenarios. Dynamic threat identification and further improvement in the system to detect zero-day attacks would be further improvements toward cloud security.

All the Declarations and Statements

Author Contributions Statement

Ajay N. Upadhyaya – Conceptualization, Methodology, and Supervision: Proposed the research ideas, constructed the overall framework, supervised project execution, and documented the technical background of the study.
Dinesh Mishra – Data Curation and Software Implementation: Handled data acquisition, dataset preprocessing, and implementing the research model.
Rajan Prasad Tripathi – Model Training, Validation, and Performance Evaluation: Led the model training process, validated results using standard metrics, and benchmarked performance against existing methods.
Bharani B. R. – Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results, prepared performance charts, and ensured the statistical robustness of the evaluation, Project Management: Reviewed and edited the manuscript, ensured clarity and coherence, and helped coordinate project milestones and deadlines.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

None.

Data Availability Statement

None.

Ethical Declarations

None.

Acknowledgments

None.

Declaration of Generative AI in Scholarly Writing

AI tools were used only for minor grammar checking and language refinement. All methodological development, analysis, and technical content were entirely carried out by the authors without AI assistance.

Abbreviations

The following abbreviations are used in this manuscript:

ASGLF - Adaptive Self-Guided Loop Filter
BIGF - Basic Interlude Gradient filtering
CRDO - Chaotic Red Deer Optimization
DDoS - Distributed Denial-of-Service
DHNN - Discrete Hopfield Neural Network
DSA - Duck Swarm Algorithm
Hop-MEA-Duck - Discrete Hopfield Neural Network and Edge Attention Networks with Duck Swarm Optimization
Hyb-BCSP - Adaptive Blockchain Sharding Protocol with Hybrid Consensus
IDS - Intrusion Detection System
IHO - Improved Heap Optimization
MCC - Matthews Correlation Coefficient
MDBGRNN - Multiscale Deep Bidirectional Gated Recurrent Neural Network
MEAN - Multilayer Edge Attention Network
MITM - Man-in-the-Middle
POA - Pufferfish Optimization Algorithm
RBFNNs - Radial Basis Function Neural Networks
RF - Random Forest
WEFA - Weighted Espoused Feature Assortment

APPENDIX

None.

References

[1] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khoukhi, "Blockchain-Enabled Federated Learning for Enhanced Collaborative Intrusion Detection in Vehicular Edge Computing," *IEEE Trans. Intell. Transp. Syst*., vol. 25, no. 7, pp. 7661–7672, 2024. DOI: 10.1109/TITS.2024.3351699

[2] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network," *J. Parallel Distrib. Comput*., vol. 164, pp. 55–68, 2022. DOI: 10.1016/j.jpdc.2022.01.030

[3] A. Govindaram and A. Jegatheesan, "FLBC-IDS: A federated learning and blockchain-based intrusion detection system for secure IoT environments", Multimedia Tools and Applications, vol. 84, no. 17, pp. 17229–17251, 2025, doi: 10.1007/s11042-024-19777-6

[4] A. V. Nagarjun and S. Rajkumar, "Design of An Anomaly Detection Framework For Delay & Privacy-Aware Blockchain-Based Cloud Deployments," *IEEE Access*, vol. 12, pp. 84843–84862, 2024. DOI: 10.1109/ACCESS.2024.3414998

[5] A. A. Sharadqh, H. A. M. Hatamleh, S. S. Saloum, and T. A. Alawneh, "Hybrid chain: Blockchain enabled framework for bi-level intrusion detection and graph-based mitigation for security provisioning in edge assisted IoT environment," *IEEE Access*, vol. 11, pp. 27433–27449, 2023. DOI: 10.1109/ACCESS.2023.3256277

[6] E. B. Mbaya, E. Adetiba, J. A. Badejo, J. S. Wejin, O. Oshin, O. Isife, et al., "SecFedIDM-V1: A Secure Federated Intrusion Detection Model With Blockchain and Deep Bidirectional Long Short-Term Memory Network," *IEEE Access*, vol. 11, pp. 116011–116025, 2023. DOI: 10.1109/ACCESS.2023.3325992

[7] S. Ali, Q. Li, and A. Yousafzai, "Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey," *Ad Hoc Netw*., vol. 152, p. 103320, 2024. DOI: 10.1016/j.adhoc.2023.103320

[8] K. Begum, M. A. I. Mozumder, M. I. Joo, and H. C. Kim, "BFLIDS: Blockchain-driven federated learning for intrusion detection in IoMT networks," *Sensors (Basel)*, vol. 24, no. 14, p. 4591, Jul. 15 2024. DOI: 10.3390/s24144591

[9] Y. M. Tukur, D. Thakker, and I.-U. Awan, "Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things", Transactions on Emerging Telecommunications Technologies, vol. 32, no. 6, Art. no. e4158, 2021. DOI: 10.1002/ett.4158

[10] D. Saveetha and G. Maragatham, "Design of Blockchain enabled intrusion detection model for detecting security attacks using deep learning," *Pattern Recognit. Lett*., vol. 153, pp. 24–28, 2022. DOI: 10.1016/j.patrec.2021.11.023

[11] J. K. Samriya, S. Kumar, M. Kumar, M. Xu, H. Wu, and S. S. Gill, "Blockchain and Reinforcement Neural Network for Trusted Cloud-Enabled IoT Network," *IEEE Trans. Consum. Electron*., 2023.

[12] D. Hamouda, M. A. Ferrag, N. Benhamida, and H. Seridi, "PPSS: A privacy-preserving secure framework using blockchain-enabled federated deep learning for industrial IoTs," *Pervasive Mobile Comput*., vol. 88, p. 101738, 2023. DOI: 10.1016/j.pmcj.2022.101738

[13] D. Mohamed and O. Ismael, "Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing," *J. Cloud Comput. (Heidelb.)*, vol. 12, no. 1, p. 41, 2023. DOI: 10.1186/s13677-023-00420-y

[14] N. O. Ogwara, K. Petrova, and M. L. Yang, "Towards the development of a cloud computing intrusion detection framework using an ensemble hybrid feature selection approach," *J. Comput. Netw. Commun*., vol. 2022, no. 1, p. 5988567, 2022. DOI: 10.1155/2022/5988567

[15] S. Dixit and G. Hussain, 2023. An effective intrusion detection system in cloud computing environment. In Mobile Radio Communications and 5G Networks: *Proceedings of Third MRCN* 2022 (pp. 671-680). Singapore: Springer Nature Singapore. DOI: 10.1007/978-981-19-7982-8_56

[16] V. Sarveshwaran, S. Pandiaraj, G. Bindu, V. Ganesan, and I. T. J. Swamidason, "Binarized Spiking Neural Network with blockchain based intrusion detection framework for enhancing privacy and security in cloud computing environment," *Appl. Soft Comput*., vol. 154, p. 111218, 2024. DOI: 10.1016/j.asoc.2023.111218

[17] K. Devi and B. Muthusenthil, "Intrusion detection framework for securing privacy attack in cloud computing environment using DCCGAN‐RFOA," *Trans. Emerg. Telecommun. Technol*., vol. 33, no. 9, p. e4561, 2022. DOI: 10.1002/ett.4561

[18] H. Attou, M. Mohy-eddine, A. Guezzaz, S. Benkirane, M. Azrour, A. Alabdultif, et al., "Towards an intelligent intrusion detection system to detect malicious activities in cloud computing," *Appl. Sci. (Basel)*, vol. 13, no. 17, p. 9588, 2023. DOI: 10.3390/app13179588

[19] D. B. Salvakkam, V. Saravanan, P. K. Jain, and R. Pamula, "Enhanced quantum-secure ensemble intrusion detection techniques for cloud based on deep learning," *Cognit. Comput*., vol. 15, no. 5, pp. 1593–1612, 2023. DOI: 10.1007/s12559-023-10139-2

[20] M. Mayuranathan, S. K. Saravanan, B. Muthusenthil, and A. Samydurai, "An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique," *Adv. Eng. Softw*., vol. 173, p. 103236, 2022. DOI: 10.1016/j.advengsoft.2022.103236

[21] B. C. Preethi, R. Vasanthi, G. Sugitha, and S. A. Lakshmi, "Intrusion detection and secure data storage in the cloud were recommend by a multiscale deep bidirectional gated recurrent neural network," *Expert Syst. Appl*., vol. 255, p. 124428, 2024. DOI: 10.1016/j.eswa.2024.124428

[22] B. K. Pandey, M. R. M. Veeramanickam, S. Ahmad, C. Rodriguez, and D. Esenarro, "ExpSSOA-Deep maxout: Exponential Shuffled shepherd optimization based Deep maxout network for intrusion detection using big data in cloud computing framework," *Comput. Secur*., vol. 124, p. 102975, 2023. DOI: 10.1016/j.cose.2022.102975

[23] A. Liu, J. Chen, K. He, R. Du, J. Xu, C. Wu, et al., "Dynashard: Secure and adaptive blockchain sharding protocol with hybrid consensus and dynamic shard management," *IEEE Internet Things* J. p. 1, 2024. DOI: 10.1109/JIOT.2024.3490036
[24] W. Yin, K. Zhang, L. Zhang, Y. Wang, and H. Liu, 2021, December. Adaptive Self-Guided Loop Filter for Video Coding. In 2021 *International Conference on Visual Communications and Image Processing* (VCIP) (pp. 1-5). IEEE. DOI: 10.1109/VCIP53242.2021.9675440
[25] O. Al-Baik, S. Alomari, O. Alssayed, S. Gochhait, I. Leonova, U. Dutta, et al., "Pufferfish Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Biomimetics (Basel)*, vol. 9, no. 2, p. 65, Jan. 23 2024. DOI: 10.3390/biomimetics9020065
[26] Y. Guo, N. E. Zamri, M. S. M. Kasihmuddin, A. Alway, M. A. Mansor, J. Li, et al., "Dual optimization approach in discrete Hopfield neural network," *Appl. Soft Comput*., vol. 164, p. 111929, 2024. DOI: 10.1016/j.asoc.2024.111929
[27] H. Liu, Y. Feng, H. Xu, S. Liang, H. Liang, S. Li, et al., "MEA-Net: Multilayer edge attention network for medical image segmentation," *Sci. Rep.*, vol. 12, no. 1, p. 7868, May 12 2022. DOI: 10.1038/s41598-022-11852-y
[28] M. Zhang and G. Wen, "Duck swarm algorithm: Theory, numerical optimization, and applications," *Cluster Comput.*, vol. 27, no. 5, pp. 1–29, 2024. DOI: 10.1007/s10586-024-04293-x
[29] https://www.kaggle.com/datasets/hassan06/nslkdd
[30] https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset

Authors' Profiles

Ajay N. Upadhyaya is currently working as a Professor in the Department of Computer Engineering at SAL Engineering & Technical Institute, SAL Education, SAL Education, Ahmedabad, 380060, Gujarat, India. He has published many research papers in many International Journals and conferences. His area of interest includes Network security, information sciences, and artificial intelligence techniques.

Dinesh Mishra is currently working in the Department of CSE, Mangalayatan University – Jabalpur, MP, India. He has published many research papers in many International Journals and conferences. His area of interest includes Network security, information sciences, and artificial intelligence techniques.

Rajan Prasad Tripathi is currently working as an Assistant Professor in the Department of IT and Engineering at Amity University in Tashkent, India. He has published many research papers in many International Journals and conferences. His area of interest includes Network security, information sciences, and artificial intelligence techniques.

Bharani B. R. is working in Dept of Information Science and Engineering, Cambridge Institute of Technology, KR Puram, Bengaluru-560036, Karnataka, India. She has published many research papers in many International Journals and conferences. Her area of interest includes Network security, information sciences, and artificial intelligence techniques.

How to cite this paper:

Ajay N. Upadhyaya, Dinesh Mishra, Rajan Prasad Tripathi, Bharani B. R., "Blockchain-Integrated Discrete Hopfield Neural Network and Edge Attention Networks with Duck Swarm Optimization for Cloud Privacy Enhancement", International Journal of Computer Network and Information Security(IJCNIS), Vol.18, No.4, pp. 22-45, 2026. DOI:10.5815/ijcnis.2026.04.02