International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I7P131

Volume 73 Issue 7, 402-422, July 2025
© 2025 Seventh Sense Research Group®

Original Article

An Efficient Cryptanalysis Strategy for Enhancing Security Estimation and Reducing Resource Usage

K. Swanthana[1], S.S. Aravinth[2]

*[1,2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,
Andhra Pradesh, India.*

[1]Corresponding Author : k.swanthana@klh.edu.in

Received: 20 February 2025 Revised: 29 May 2025 Accepted: 04 June 2025 Published: 30 July 2025

Abstract - *The rapid expansion of blockchain technology has led to increased computational demands, necessitating efficient resource management to ensure optimal performance and security. Traditional resource allocation methods often struggle with high computation costs and inadequate security measures, leading to inefficient use of cloud resources and vulnerability to attacks. This paper presents a novel Chimp-based Optimized Recurrent Diffie-Hellman (CORDH) strategy designed to overcome these challenges by optimizing resource usage in blockchain networks. The proposed CORDH model leverages the Chimp Optimization Algorithm (COA) combined with recurrent frameworks to dynamically allocate computational resources such as Virtual Machines (VMs), CPU, and RAM. This methodology involves training on diverse datasets from healthcare, stock markets, and Network Traffic, followed by applying COA to allocate necessary resources, thus reducing computation costs efficiently. Additionally, the CORDH model enhances system security by implementing robust cryptanalysis strategies to counter brute force and DDoS attacks. Experimental results demonstrate that CORDH significantly improves resource utilization, enhances data security, and outperforms traditional methods in both computational efficiency and resilience to attacks.*

Keywords - *Cloud Environments, Resource Allocation, Chimp Optimization, Blockchain network, Cryptanalysis, Data security.*

1. Introduction

Blockchain is an advanced security technology used in distributed and decentralized ledger systems for data transfer via block structures [1]. It integrates data encryption, peer-to-peer frameworks, consensus mechanisms, and other techniques [2]. Ensuring privacy and secrecy on public blockchains while maintaining auditability and transparency remains challenging [3]. Privacy-preserving methods like ring signatures, zero-knowledge proofs, and secure multi-party computation help enable private transactions and safe data sharing [4, 5]. Selecting efficient consensus methods, such as Proof of Stake or Delegated Proof of Stake, can reduce energy and computational costs compared to Proof of Work [6]. Intelligent contract optimization and efficient data structures help lower gas fees and overhead [7, 8]. Sensitive data should be stored off-chain, using cryptographic summaries like hashes for validation [9, 10]. Compliance with regulations, including financial and data protection laws, increases complexity and computational demands [11, 12]. Smart contracts, while automating transactions, introduce vulnerabilities such as re-entrancy and arithmetic overflows, necessitating thorough auditing and testing [3, 5, 31]. Blockchain networks also face governance challenges, including hard forks and threats like 51% attacks, requiring ongoing community participation and computational investment [13, 14].

Previous approaches like game search optimization [15], federated learning [16], and hierarchical blockchain models [17] have been proposed, yet a comprehensive solution remains elusive. Thus, a robust blockchain-based security framework is essential to address ongoing threats and challenges. This study introduces a novel CORDH strategy to bridge this gap. Unlike prior approaches, CORDH dynamically optimizes resource allocation while simultaneously enhancing cryptographic robustness. The proposed model integrates the COA with recurrent neural frameworks to intelligently distribute computational resources across blockchain networks, significantly reducing cost and execution time. Simultaneously, it embeds advanced cryptanalysis to mitigate brute force and DDoS attacks. The novelty of this research lies in its dual focus-achieving computational efficiency through biologically inspired optimization and fortifying security via robust cryptographic countermeasures. Comparative evaluations using diverse datasets from healthcare, stock markets, and network traffic demonstrate that CORDH outperforms existing models in terms of execution speed, resource usage, and resistance to security threats. The paper is organized as follows: Section 2 reviews related studies, Section 3 outlines the system model and problem definition, Section 4 details the proposed approach, Section 5 evaluates performance and compares outcomes, and Section 6 presents the conclusion.

2. Literature Review

The integration of blockchain into various domains-such as healthcare, finance, and IoT-has led to an upsurge in data volumes and security requirements, consequently elevating the need for scalable and efficient security frameworks. However, despite its inherent properties of decentralization, immutability, and cryptographic protection, blockchain technology still faces substantial limitations in resource management, computational efficiency, and vulnerability to cyber threats.

Numerous privacy issues are affected by blockchain technology, and it is difficult to use the application in a blockchain environment. Therefore, Viswanadham et al. [18] have developed a hybridized Border collie-based rain optimization approach to address the security issues and manage the large-scale data. Here, the approach generates an optimal key for securely restoring and sanitizing data over the network. The developed model operates via the Ethereum network to address complex issues, facilitating original data analysis using the correct key. However, the new frameworks might face adoption challenges within the blockchain community.

To verify the applicability of the blockchain technology, an optimized byzantine fault tolerance algorithm was designed by Tao Liu et al. [19]. This technique is mainly used to control the target cost and technical faults over the blockchain models. Here, the blockchain is designed in chronological order to manage the consensus nodes and storage nodes. Also, the designed structure has two parts, such as a block header and a block body, to address the previous blocks' transaction history. Moreover, the hash functions are created using cryptographic techniques. Finally, the results show that enterprise networks are highly applicable for business integration. However, the implementation of the genetic algorithm is complex. In digital applications, public key generation algorithms are applied to improve the transmission process. Here, Halak et al. [20] have developed symmetric as well as asymmetric-based cryptographic algorithms to overcome the elliptic-based discrete logarithm and discrete logarithm issues. Here, IoT devices are integrated to assist in the attacks and manage cost-effective functions. Also, the energy costs among the symmetric and asymmetric groups are recorded for comprehensive evaluation. However, it has limited applicability to asymmetric cryptography.

Alam et al. [21] designed the blockchain-based Deep Reinforcement Learning (DRL) strategy to solve resource-intensive computational offloading problems. Moreover, this DRL model enhanced network performance through IOT devices and solved optimization problems. Also, this algorithm effectively defined the large-dimensional and dynamic properties.

IoT devices using MQTT lack built-in security and depend on resource-heavy TLS protocols. To solve this, Akshatha and Kumar [22] propose a blockchain-based approach using blockchain sharding for better scalability, performance, and reduced overhead. This approach utilizes Ethereum smart contracts for trust and privacy, along with a shard-based consensus mechanism to improve security with minimal computational overhead. A user-managed Proof-of-Access algorithm decentralizes data access control, reducing resource usage and demonstrating superior performance compared to TLS and other blockchain methods across different MQTT brokers.

The current issue in Ethereum is the high computational overhead and resource consumption for data verification, with traditional methods like Merkle Trees becoming inefficient for large datasets. To address this, Kuznetsov et al. [23] introduced an innovative method for aggregating Zero-Knowledge Proofs within Merkle Trees, enhancing efficiency and security, significantly reducing proof size and computational requirements. This system balances security and efficiency, validated by extensive tests on real Ethereum data, showing improved verification efficiency and economic viability. However, its development and deployment may incur higher initial costs and resources. Security challenges in IoT stem from the varied capabilities of devices and the dynamic environment, which render basic security measures inadequate against malware attacks. In response, Devi and Arunachalam [24] proposed a deep learning-based approach for detecting and preventing malware. Their method involves identifying attack nodes and predicting attacks using contextual features and a Deep LSTM classifier. For prevention, they employ the Improved Elliptic Curve Cryptography (IECC) algorithm with hybrid optimization techniques. The approach achieved 95% accuracy and 92% precision, with an IECC execution time of 6.02 seconds. However, implementing deep learning and cryptography can be complex and resource-intensive.

Table 1. Issues in current research

Sl.No.	Author	Methods	Advantage	Disadvantage
1.	Viswanadham, et al, [18]	hybridized Border collie-based rain optimization approach	It is effectively performed through the etheric network to solve the complex issues	The new frameworks might face adoption challenges within the blockchain community.
2.	Tao Liu, et al. [19]	optimized byzantine fault tolerance algorithm	It efficiently controls the target cost and technical faults over the blockchain models.	It is the complexity of implementing the genetic algorithm.

3.	Halak, et al. [20]	Symmetrical as well as asymmetric-based cryptographic algorithms	It assists in the attacks and manages the cost-effective functions	It has limited applicability to asymmetric cryptography.
4.	Alam, et al. [21]	Blockchain-based Deep Reinforcement Learning (DRL) strategy	Effectively defined the large-dimensional and dynamic properties.	Integrating and maintaining blockchain systems should be costly.
5.	Akshatha and Kumar, [22]	Blockchain-based approach	It reduces resource consumption and outperforms TLS.	It introduces additional complexity in system design and management.
6.	Kuznetsov et al. [23]	Zero-Knowledge Proofs within Merkle Trees	It improved verification efficiency and economic viability	Its development and deployment may incur higher initial costs and resources.
7.	Devi and Arunachalam [24]	Malware detection and prevention approach using DL	It achieved 95% accuracy and 92% precision, with an IECC execution time of 6.02 seconds.	Implementing deep learning and cryptography can be complex and resource-intensive.

A broader look at existing methods reveals persistent limitations:

- Security-centric models often lack efficient resource optimization mechanisms.
- Optimization-focused strategies tend to ignore dynamic security threats such as brute force or DDoS attacks.
- Hybrid solutions either rely on static resource allocations or introduce complexity that hinders real-time deployment.

Table 1 summarizes the advantages and limitations of the key studies. Notably, none of the reviewed works deliver a unified framework that concurrently ensures adaptive resource allocation, robust security, and resistance to real-time threats. Furthermore, techniques such as Federated Learning [16] and Hierarchical Blockchain Models [17] offer promising pathways to decentralization and scalability. However, they either rely heavily on edge devices with limited processing power or struggle to mitigate advanced cyberattacks effectively. There is a clear gap in existing research: a lack of lightweight, scalable, and attack-resilient cryptographic models that integrate intelligent resource optimization. To address these gaps, this paper proposes the CORDH framework. By combining COA with recurrent networks and cryptographic evaluation, CORDH offers a comprehensive solution that balances computational cost, dynamic resource allocation, and real-time security threat mitigation. Its novelty lies in its adaptability across domains and datasets, as demonstrated in this study through rigorous comparative analyses with state-of-the-art methods. The key contribution of the developed CORDH model is summarised in the section below;

- The novel CORDH strategy integrates blockchain with a COA to minimize resource usage and optimize computational costs in cloud environments. It also incorporates cryptanalysis to evaluate confidentiality and security during brute force and DDoS attacks.
- COA simulates chimpanzee behaviors to optimize virtual machine allocation in the cloud. It balances CPU, memory, and storage resource allocation while minimizing usage through iterative exploration, exploitation, and fitness evaluation processes.
- Brute force and DDoS attacks are launched during data sharing to test system security. A RNN is used for attack detection and mitigation, monitoring network traffic and identifying anomalies for proactive defence.
- The Diffie-Hellman algorithm is utilized for secure key generation and encryption, ensuring data confidentiality. Cryptanalysis evaluates the system's resistance to attacks, with minimal variations observed in performance before and after attacks due to robust encryption and mitigation strategies.

3. System Model and Problem Statement

Blockchain technology provides several benefits for securing the data, but also it has some limitations, especially for resource usage, computation cost, security assessment, etc. Figure 1 illustrates the system model architecture. Blockchain scalability is the compromise in performance for handling a large number of transactions. In some cases, a large amount of data is transmitted through the blockchain. Then, the computation cost can increase, leading to the highest resource usage and lower transaction time.

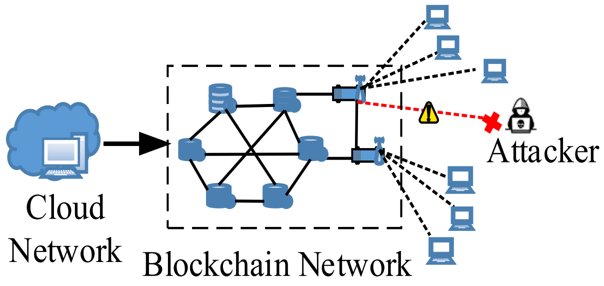


Fig. 1 System model architecture

Moreover, a large volume of data stored in the blockchain can increase storage and bandwidth requirements. This performance has contributed to higher energy consumption towards blockchain technology. Blockchain technology is susceptible to cryptanalysis attacks even if it uses strong cryptographic algorithms to encrypt data. Cryptographic flaws, brute force assaults, and advances in quantum computing can threaten blockchain network security. Cryptanalysis security estimation necessitates ongoing monitoring and a resource-intensive response to new threats. It takes a comprehensive strategy that takes advantage of developments in blockchain and cryptography to balance computing costs, resource consumption, and security considerations in order to overcome these constraints.

4. Proposed Methodology

Therefore, this paper is a novel Chimp-based Optimised Recurrent Diffie-Hellman (CORDH) for optimizing the resource usage of the designed blockchain. The architecture of the proposed CORDH model is illustrated in Figure 2. Furthermore, brute force and Denial-of-Service (DoS) attacks were conducted during the cryptanalysis to evaluate the blockchain design's confidentiality score. This work aims to optimize the blockchain computation cost in protecting the data by minimizing resource usage and security estimation using a cryptanalysis strategy.

Datasets such as healthcare, stock market, and network traffic data are processed using a recurrent framework for training. The CORDH strategy, based on a blockchain environment, is designed to optimize computation costs in the cloud using virtual machines and the Chimp optimization fitness function. Trained datasets are transferred between sources and destinations, with brute force and DDoS attacks launched during sharing. A cryptanalysis model evaluates confidentiality scores before and after applying the optimization. Performance is then compared to traditional methods to demonstrate effectiveness.

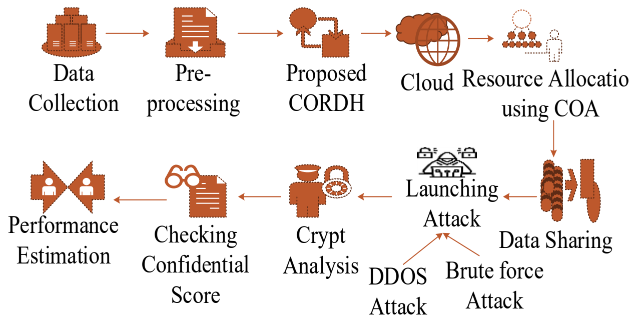


Fig. 2 Architecture of proposed CORDH model

4.1. Data Preprocessing

The data preprocessing module is tasked with gathering, cleaning, and transforming historical data related to resource utilization and user requests. This dataset, such as Healthcare data, stock market data, and network traffic data, is sourced from standard online platforms such as Kaggle. The initial step involves cleaning the data to eliminate noise and outliers, and filling in any missing values. After cleaning, the data is transformed into a format suitable for training recurrent models in the resource demand prediction module.

4.2. Design of Proposed CORDH Method

4.2.1. Cloud Environment Setup

Cloud computing, also known as utility-based computing, provides users with a variety of services, leading to an increased demand for computational resources. Evaluating and allocating resources such as virtual machines, CPU, RAM, and storage is crucial for cloud service providers to manage their virtualized distributed resources effectively and efficiently assign incoming tasks [25].

Efficient resource allocation in cloud computing is essential for conserving energy in a data-intensive environment. Given the substantial energy consumption of cloud data centers, effective resource management is crucial. Dynamic resource allocation techniques, which adjust resource provisioning in response to real-time workload changes, play a pivotal role in minimizing overprovisioning and ensuring that resources are allocated optimally as per demand. Therefore, an efficient resource allocation algorithm is necessary to manage resources effectively and reduce energy consumption in cloud environments.

In the cloud environment setup phase, the goal is to create a scalable and efficient infrastructure that can handle the computational demands of the CORDH model. This involves provisioning Virtual Machines (VMs) and other cloud resources. Let R be the total resources required for the blockchain operations. The cloud environment provides a set of virtual machines$\{V_{M1}, V_{M2}, \ldots\ldots\ldots V_{Mn}\}$ with varying capabilities. Each virtual machine (i) has a specific resource capacity $C_i$including CPU, memory, and storage [25]. The primary goal is to minimize the usage of these resources while ensuring all tasks are performed efficiently. This can be formulated as mentioned in Equations 1 and 2.

$$Minimize \sum_{i=1}^{n} C_i x_i \text{ n} \tag{1}$$

$$\sum_{i=1}^{n} C_i x_i \geq R \tag{2}$$

Where$x_i$is a binary variable that indicates whether a virtual machine (i) is selected if $(x_i = 1)$ else not $(x_i = 0)$.

4.2.2. Resource Allocation using Chimp Optimization

COA is inspired by the social behavior and hunting strategies of chimpanzees. It is used as a fitness function to find the optimal allocation of resources. COA was selected due to its superior ability to balance exploration and exploitation phases in high-dimensional optimization problems, which is essential for dynamic cloud resource allocation. Compared to traditional algorithms like PSO and

GA, COA simulates diverse chimpanzee behaviours-attack, chase, barrier, and drive-which enable more diverse and adaptive search patterns. This multi-agent role distribution allows COA to avoid local minima more effectively and converge faster in complex solution spaces. Moreover, unlike ACO, which can suffer from early stagnation and PSO, which is prone to premature convergence, COA demonstrates robustness in resource-intensive cloud scheduling scenarios. This makes COA a theoretically appealing candidate for blockchain environments with fluctuating workloads and security constraints. Each "chimp" in the algorithm represents a potential solution, i.e., a specific allocation of VMs, CPU, RAM, and storage to user-defined tasks. The goal is to find an optimal allocation that minimizes resource usage while ensuring the computational demands of the functions are met. The process can be broken down into several steps [26].

Exploration Stage

The exploration stage involves a global search to discover potential solutions and avoid premature convergence. This stage is based on the behavior of chimpanzees when they search for food. The procedure uses Equation 3 to identify possible solutions:

$$D = \left|a \cdot Z_{prey}(f) - b_f \cdot Z_{chimp}(f)\right| \tag{3}$$

$Z_{prey}(f)$is the position vector of the potential solution (resource allocation), $Z_{chimp}(f)$is the position vector of the chimp, $a$and b_f is a coefficient vector that $f$denotes the number of iterations. Equation 4 is employed to update the prey's position by incorporating the distance computed during the prior iteration.

$$\mathrm{Z_{prey}(f+1) = Z_{prey}(f) - c \cdot D} \tag{4}$$

The coefficients c, a, and b_f are calculated through Equations 5, 6 and 7.

$$c = 2 \cdot l_n \cdot r_1 - l_n \tag{5}$$

$$\mathrm{a = 2 \cdot r_2} \tag{6}$$

$$\mathrm{b_{f+1} = \lambda b_f(1 - b_f)} \tag{7}$$

Here, $l_n$it decreases nonlinearly from 2.5 to 0, then r_1 ,and r_2 is a random vector within [0, 1], and $b_f$is a chaotic vector based on logistic mapping.

Exploitation Stage

During this exploitation stage of the COA, the focus shifts to enhancing the current best solutions to locate local optima, inspired by chimpanzees exploiting a known food source. This phase involves generating new potential solutions around the current best ones by applying small perturbations. Thus, chimpanzees employed various strategies to locate prey, adjusting their search locations based on the movements of other chimpanzees. The distances for different chimp groups (attack, chase, block, drive) are calculated through Equations 8 and 9, and positions are updated accordingly.

The distances for different chimp groups (attack, chase, block, drive) are calculated, and their positions are accordingly.

$$D_{attack} = \left|a_1 \cdot Z_{attack}(f) - b_{f1} \cdot (f)\right|, \quad D_{chase} = \left|a_2 \cdot Z_{chase}(f) - b_{f2} \cdot (f)\right| \tag{8}$$

$$D_{block} = \left|a_3 \cdot Z_{block}(f) - b_{f3} \cdot (f)\right|, \quad D_{drive} = \left|a_4 \cdot Z_{drive}(f) - b_{f4} \cdot (f)\right| \tag{9}$$

Next, adjust the position of each group according to the computed distances. Equations 10 and 11 give the updated positions of the chimps belonging to the attacking$Z_1(f)$, chasing$Z_2(f)$, blocking$Z_3(f)$, and driving$Z_4(f)$ groups at $f$each iteration.

$$Z_1(f) = Z_{attack}(f) - c_1 \cdot D_{attack}, \quad Z_2(f) = Z_{chase}(f) - c_2 \cdot D_{chase} \tag{10}$$

$$Z_3(f) = Z_{block}(f) - c_3 \cdot D_{block}, \quad Z_4(f) = Z_{drive}(f) - c_4 \cdot D_{drive} \tag{11}$$

These positions are modified based on perturbations applied to their previous positions. A chimp's overall updated position is determined by averaging the four main groups, as represented in Equation 12.

$$\mathrm{Z(f+1) = \frac{Z_1(f) + Z_2(f) + Z_3(f) + Z_4(f)}{4}} \tag{12}$$

Where Z represents the current position vector of the chimpanzee. The variables Z_{attack}, Z_{chase}, Z_{block}, and Z_{drive}correspond to the position vectors of the attacker, chaser, barrier, and driver, respectively. The notation $Z(f+1)$denotes the updated position vector of the chimpanzee for the next time step.

In the final phase, the chimps launch their attack and end the hunt when the prey becomes motionless. This process is mathematically modeled by reducing the value of l_n. It is essential to recognize that the variable values are also limited by l_n. Specifically, $c$represents a random variable within the range$[-2l_n, 2l_n]$, and the value of $l_n$gradually decreases from 2.5 to 0 as iterations continue. In the final stage, chimpanzees initiate their attack and finish the hunt once the prey ceases movement. The mathematical model of this process involves a reduction in the variable l_n. It is important to note that another variable $c$is also constrained by l_n. This variable c is a random number within the range$[-2l_n, 2l_n]$, and its value decreases from 2.5 to 0 gradually over iterations. Additionally, similar to Equation 3, the coefficients c_i, a_i, b_{fi} (which i range from 1 to 4) are determined using Equations 13, 14 and 15.

$$c_i = 2 \cdot l_n \cdot r_{i1} - l_n \tag{13}$$

$$a_i = 2 \cdot r_{i2} \tag{14}$$

$$b_{fi+1} = \lambda b_{fi}(1 - b_{fi}) \tag{15}$$

This exploitation phase in COA focuses on refining the best solutions by adjusting their positions based on calculated distances, with the aim of finding local optima. This phase helps improve the quality of solutions iteratively.

Fitness Evaluation

The fitness of each solution in the COA is evaluated based on resource usage. The goal is to reduce overall resource consumption while ensuring that all resource constraints are met. This is evaluated using Equation 16 below.

$$Fitness(S) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot R_j \tag{16}$$

$S$is a solution, n is the number of tasks, $m$is the number of resources, x_{ij} is a binary variable indicating whether the resource $j$is allocated to a task i (1 if allocated, 0 otherwise), R_j and is the amount of resource$j$ used. After selecting the best solutions based on their fitness values. Repeat the exploration and exploitation phases until convergence or a maximum number of iterations is reached. After that, the required resources (VMs, CPU, RAM, and storage) are allocated to the tasks based on the optimal solution found by the algorithm.

4.2.3. Secure Data Sharing in the Blockchain Network

In the proposed CORDH methodology, the data-sharing process in the blockchain network ensures security, reliability, and efficiency through the integration of blockchain and the Diffie-Hellman algorithm. Initially, data is securely stored on the blockchain with cryptographic hashing for tamper-proof and immutable records. The Diffie-Hellman algorithm facilitates the generation of a shared secret key between the sender and receiver, enabling secure encryption of the data before uploading it to the blockchain. Encrypted data is distributed across decentralized nodes, and smart contracts manage access permissions, ensuring that only authorized users can retrieve the data. Upon request, the receiver uses the shared secret key to decrypt the data, maintaining its confidentiality and integrity. The system is fortified against security threats, such as brute force and DDoS attacks, through an RNN-based anomaly detection mechanism that monitors network activity. Additionally, cryptanalysis validates the process's resistance to attacks, ensuring high confidentiality and minimal performance impact. This secure and decentralized framework exemplifies an efficient and robust approach to data sharing within blockchain networks.

4.2.4. Attack Launching

Two types of attacks are simulated to evaluate the security of the system: brute force attacks and Distributed Denial of Service (DDoS) attacks. Both DDoS and brute force attacks were initiated to evaluate the model's performance in two phases: before and after the attacks. The minimal variation in results between these phases is attributed to the integration of an RNN, which continuously monitors the system and mitigates the impact of the attacks.

Brute Force Attack

A brute force attack is a method for decrypting encrypted data, such as passwords or PINs, through systematic trial and error. Attackers attempt every possible combination of characters until they discover the correct one.

DDoS Attack

A DDoS attack is a malicious attempt to overwhelm a targeted server, service, or network with an excessive amount of internet traffic, disrupting its normal operations. Unlike other cyberattacks that attempt to breach security measures, the main goal of a DDoS attack is to disrupt access to a website or server by flooding it with excessive traffic, making it unavailable to legitimate users.

4.2.5. Attack Detection and Mitigation

A Recurrent Neural Network (RNN) is utilized to oversee the system, identifying and counteracting potential attacks. This network is trained on typical login and traffic patterns to recognize anomalies, such as those resembling brute force attempts or unusual traffic spikes indicative of DDoS attacks. The RNN consistently monitors login attempts and network traffic, identifying irregular patterns and flagging suspicious activities.

4.2.6. Crypto Analysis

After launching the attack, Crypt analysis is performed. It encompasses both encryption and decryption processes. The proposed model generates a key using the Diffie-Hellman algorithm, which facilitates secure key exchange over an unsecured communication channel. Its primary function is to allow two parties, who may lack prior knowledge of each other, to establish a shared secret key collaboratively. This shared key can then be used to encrypt and decrypt messages between the parties, ensuring confidentiality [27]. The key used for data encryption is derived from the file number and public variables. Equation 17 presents the formula for generating this key.

$$\mathrm{G_k} = \mathrm{t^h} \bmod \mathrm{u} \tag{17}$$

In the encryption process described by Equation 18, the generated key $G_k$is used to encrypt the data, with $t$and $u$representing public variables and $h$denoting the file number.

$$E_n^*(O_d) = G_k \times O_d = C_{text}^* \tag{18}$$

In the encryption process, the original data O_d is transformed into ciphertext by applying a multiplication operation using a generated key. Here, E_n^*represents the encryption function, and C_{text}^*denotes the resulting ciphertext. Once data is encrypted, the system issues a private key to the user. This key is used to decrypt the ciphertext and recover the original data. This decryption process retrieves the initial input data. DecryptionD_y is represented through Equation 19.

$$O_d = D_y(C_{text}^*, G_k) = {C_{text}^*}/{G_k} \tag{19}$$

After that, the confidential score of the system, both before and after applying the Chimp optimization model, is measured.

This step ensures that the data remains secure and confidential. Figure 3 represents the flowchart of the processed CORDH method, and the algorithm proposed is illustrated in the Table 2.

Table 2. Algorithm for the proposed CORDH model

```
Start
{
  Data preprocessing
   {
     //cleaning and transforming the collected dataset to eliminate noise and outliers, and fill in any missing values
   }
Proposed CORDH module ()
 {
  Cloud Environment Setup
  {
  Int $V_{M1}, V_{M2}, Vg_{Mn}$
$\{V_{M1}, V_{M2}, \ldots\ldots\ldots V_{Mn}\}$                    // using eqn.(1)
// Set of VMs with varying resource capabilities including CPU,   memory, and storage.
  }
 Resource Allocation
 {
 Int $a, b_f, c, D, f, f+1, Z_{prey}, Z_{chimp}$
$D = |a \cdot Z_{prey}(f) - b_f \cdot Z_{chimp}(f)|$ //using eqn. (3) //The global search equation is used to calculate the potential solutions
$Z_{prey}(f+1) = Z_{prey}(f) - c \cdot D$          // using eqn.(4)
// Updating the prey's position by calculating the distance during the prior iteration          $D_{attack} = |a_1 \cdot Z_{attack}(f) - b_{f1} \cdot (f)|, D_{chase} = |a_2 \cdot Z_{chase}(f) - b_{f2} \cdot (f)|$
                                                                                                                                  //
using eqn. (8)
$D_{block} = |a_3 \cdot Z_{block}(f) - b_{f3} \cdot (f)|, D_{drive} = |a_4 \cdot Z_{drive}(f) - b_{f4} \cdot (f)|$
                                                                                  // using eqn.(9)
 // The distances for different chimp groups (attack, chase, block, drive) are calculated
 $Z_1(f) = Z_{attack}(f) - c_1 \cdot D_{attack}$ ,
 $Z_2(f) = Z_{chase}(f) - c_2 \cdot D_{chase}$                       // using eqn.(10)
 $Z_3(f) = Z_{block}(f) - c_3 \cdot D_{block}$ ,
 $Z_4(f) = Z_{drive}(f) - c_4 \cdot D_{drive}$                       // using eqn.(11)
// Current position vector of each chimp group
        $Z(f+1) = \frac{Z_1(f)+Z_2(f)+Z_3(f)+Z_4(f)}{4}$
                                                  // using eqn.(12)
 //Update the position of each group based on the calculated distance to generate new potential solutions
Fitness $(S) = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot R_j$    // using eqn.(13)
 //The best solutions are selected based on their fitness values to allocate the required resources.
  }
 Secure Data Sharing in the Network
```

```
  {
   Attack Launching
    {
     //The brute force and DDOS attack is launched to evaluate the security of the system
    }
   Attack Detection and Mitigation
    {
     // The RNN is trained to consistently monitor the system, identifying and mitigating the attack
    }
   Crypto analysis
    {
     Int $G_k, t^h, u, E_n^*, O_d, C_{text}^*$
   $G_k = t^h \bmod u$                    // using eqn.(17)
    // key is generated
   $E_n^*(O_d) = G_k \times O_d = C_{text}^*$        // using eqn.(18)
   // After generating the key, the data is encrypted with the secret key. $O_d = D_y(C_{text}^*, G_k) = C_{text}^*/G_k$ // using eqn.(19)
   // Finally, the user decrypts the ciphertext using the secret key after verifying.
  }
  }
Stop
}
```

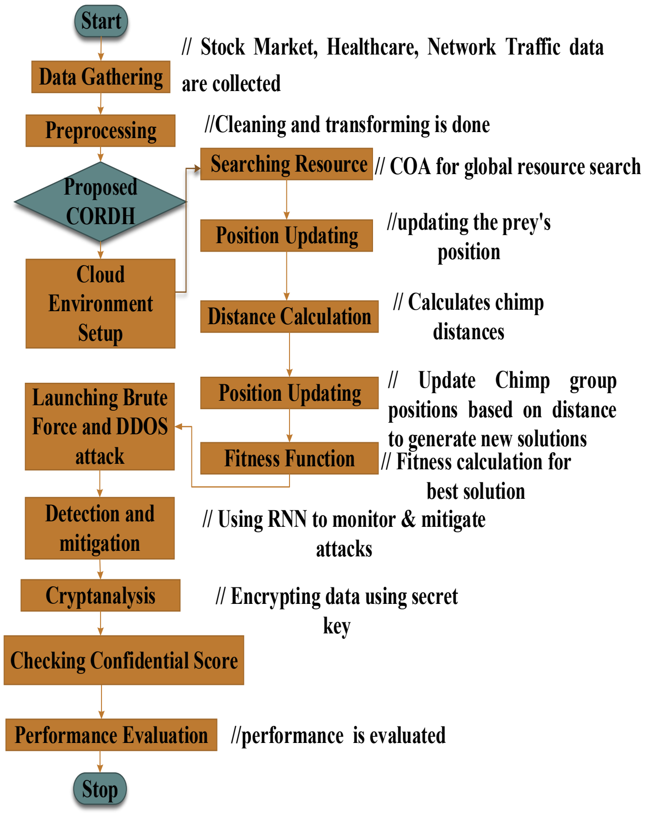

Start
Data Gathering
// Stock Market, Healthcare, Network Traffic data are collected
Preprocessing
//Cleaning and transforming is done
Proposed CORDH
Searching Resource
// COA for global resource search
Position Updating
//updating the prey's position
Cloud Environment Setup
Distance Calculation
// Calculates chimp distances
Position Updating
// Update Chimp group positions based on distance to generate new solutions
Launching Brute Force and DDOS attack
Fitness Function
// Fitness calculation for best solution
Detection and mitigation
// Using RNN to monitor & mitigate attacks
Cryptanalysis
// Encrypting data using secret key
Checking Confidential Score
Performance Evaluation
//performance is evaluated
Stop

Fig. 3 Flowchart of proposed CORDH

5. Result and Discussion

A new method called CORDH has been introduced to enhance the security of datasets from the fields of network traffic, healthcare, and the stock market using a hybrid Blockchain system, developed within a Python environment. This method offers robust data protection by fusing recurrent neural networks with sophisticated cryptography algorithms, optimize resource utilization, and reducing blockchain computation costs. The strategy also includes a cryptanalysis-based evaluation of security. The experimental setup for the CORDH technique is described in Table 3.

Table 3. Experimental setup

Parameter	Description
Platform	Python version 3.11
Operating System (OS)	Window 10
Version	3.10
Data Set	RBLBANK.NS.csv Network traffic.csv. patientMonitoring.csv
RAM	256 GB
Processor	Intel Xeon D 1653N
Library	Cryptography (Fernet), PyNaCl, PyOpenSSL
System type	64-bit OS
Clock rate	4.2 GHz
Storage	20 TB with SSD
Number of cores	16
Dataset size	RBLBANK.NS.csv- 20.7 MB (21,792,075 bytes) Network traffic.csv -1.77 GB PatientMonitoring.csv -18.0 MB (18,924,951 bytes)
Attack launched	DDoS, Brute Force

5.1. Case Study

This case study demonstrates the effectiveness of the CORDH system in enhancing security and optimizing resource usage within blockchain networks. A variety of datasets, such as network traffic data, healthcare data, and stock market data, were used to assess the CORDH model's effectiveness across several domains. The primary objective was to address the inefficiencies and security vulnerabilities in traditional resource allocation methods by leveraging advanced optimization and cryptographic techniques.

The development and testing of the CORDH system aimed to enhance security and optimize resource utilization in blockchain networks. Using data from network traffic, healthcare, and the stock market, the system exhibited notable improvements in both computational efficiency and security. The architecture of the model integrates cloud-based VMs, CPUs, and RAM, which are optimized through the COA to ensure dynamic and efficient resource allocation. Data preprocessing involves cleaning and transforming the datasets to remove noise and outliers, preparing them for training recurrent models to predict resource demands accurately.

For secure data sharing, the CORDH system employs blockchain technology to provide decentralized management and robust cryptographic methods to maintain data confidentiality. The system's resilience was validated by simulating DDoS and brute force attacks, with an RNN effectively monitoring and mitigating these threats. Performance metrics, including accuracy, recall, confidentiality rate, and error rate, were evaluated before and after the attacks. Cryptanalysis further assessed the system's security using Diffie-Hellman key exchanges for secure encryption and decryption.

The results demonstrated high accuracy and recall, low error rates, and strong confidentiality, indicating the system's reliability and security. Additionally, the model significantly reduced resource usage, as evidenced by efficient CPU and RAM utilization, and minimized execution, encryption, and decryption times, confirming the CORDH system's effectiveness in real-time applications.

5.1.1. Dataset Description

Stock Market Data

To address stock market volatility, Machine Learning, specifically RNNs, is employed for forecasting Bank Nifty stocks over time. RNNs are effective in capturing temporal patterns, leading to more accurate predictions over a three-year span. Data is gathered using two methods: real-time retrieval via the Yahoo Finance ticker and a six-month historical dataset of Bank Nifty stocks. This combination provides both up-to-date information and historical context, improving forecasting accuracy amid market fluctuations. https://www.kaggle.com/code/ridhijhamb/stock-price-prediction-multivariate-time-series/input

Healthcare Data

The IoT-enabled ICU is designed with two beds, each equipped with nine patient sensors and a Bedx-Control-Unit, all developed using the IoT-Flock tool. This setup allows for continuous real-time monitoring of vital signs, improving healthcare responsiveness. https://www.kaggle.com/datasets/faisalmalik/iot-healthcare-security-dataset

Network Traffic Data

The dataset, sourced from Kaggle, contains 3,577,296 network traffic instances collected at Universidad Del Cauca, Colombia, over multiple days in 2017. It includes 87 features such as IP addresses, ports, interarrival times, and protocols, analyzed using CICFlowmeter and ntopng, with both numeric and categorical attributes. By providing detailed flow statistics, this dataset enhances network traffic classification capabilities, allowing for the identification of specific applications such as Facebook and YouTube rather than just broad traffic types. This serves as a crucial tool for developing and accessing machine learning models focused on detecting network anomalies and improving network security. https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps

For the statistical analysis, a T-test was performed to compare the performance of the proposed CORDH models (Stock Market, Health Care, and Network Traffic) with existing approaches. The key metrics compared include anomaly detection accuracy, energy efficiency, and latency, as shown in Table 4. The T-tests assess if the differences between the proposed models and the benchmarks are statistically significant.

Table 4. Statistical comparison of proposed CORDH

	Anomaly Detection Accuracy (%)	**Energy Efficiency (%)**	**Latency (ms)**
Comparison	Proposed CORDH vs BioBlock	Proposed CORDH vs AI-SecIoT	Proposed CORDH vs BB-IoT
T-Statistic	2.35	1.92	-2.82
P-Value	0.024	0.057	0.007
Confidence Interval	(0.23, 3.89)	(-0.16, 4.02)	(-9.15, -1.42)
Interpretation	Statistically significant improvement	No significant difference; it is slightly better	Significantly better latency

Anomaly Detection Accuracy: The Proposed CORDH (Stock Market) outperforms BioBlock with a T-statistic of

2.35 and a P-value of 0.024. The confidence interval (0.23, 3.89) confirms the statistical significance, showing the proposed model is significantly better.

Energy Efficiency: Proposed CORDH (Health Care) shows a slight improvement over AI-SecIoT, but the T-test results with a P-value of 0.057 indicate that the difference is not statistically significant, even though the confidence interval suggests a possible improvement.

Latency: Proposed CORDH (Network Traffic) has a significantly lower latency compared to BB-IoT (P-value = 0.007, Confidence Interval = (-9.15, -1.42)), confirming the proposed method's superiority.

Training parameters are given in Table 5.

Table 5. CORDH training details

Parameter	Value
Layer Count	3 (LSTM layers)
Activation Functions	ReLU (hidden), Sigmoid (output)
Training Epochs	50
False Positive Rate (FPR)	~5%

The selection of the COA over other metaheuristic algorithms such as PSO, GA, and ACO is grounded in theoretical advantages and empirical evidence. Unlike PSO, which is prone to premature convergence, or GA, which requires complex parameter tuning and crossover operations, COA models diverse social behaviours-attack, chase, barrier, and drive-that enhance its exploration-exploitation balance in dynamic environments. This makes COA more suitable for real-time blockchain-based resource scheduling where solution landscapes are constantly changing. Figure 4 compares energy consumption (in KWH) across various optimization algorithms and the proposed CORDH approach over 25 iterations. Generally, MCWOA, WOA, CA, GWO, ALO, and GA show higher energy consumption, ranging from 0.13 to 0.17 KWH. In contrast, the proposed CORDH variants (Stock Market, Healthcare, and Network Traffic) consistently exhibit lower energy consumption. Notably, CORDH (Network Traffic) consistently has the lowest consumption, reaching 0.10 KWH at iterations 15 and 25, indicating its superior energy efficiency compared to the other methods. Figure 5 presents latency (in ms) for various algorithms and the proposed CORDH across increasing tasks (100 to 400). Across all task counts, MCWOA, WOA, CA, GWO, ALO, and GA generally exhibit higher latency compared to the proposed CORDH variants. Specifically, CORDH (Network Traffic) consistently demonstrates the lowest latency, ranging from 7.40 ms for 100 tasks to 8.70 ms for 400 tasks. The other CORDH variants (Stock Market and Healthcare) also show lower latency than the comparative algorithms, indicating the efficiency of the proposed approach in managing task execution time as the workload increases.

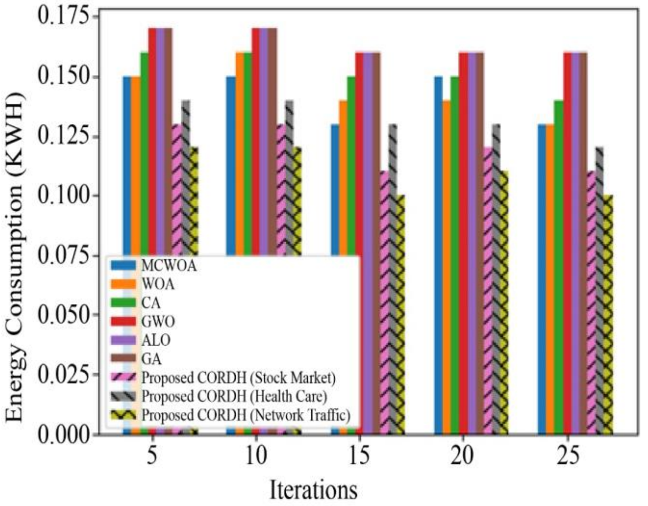


Fig. 4 Comparison of energy consumption with optimization

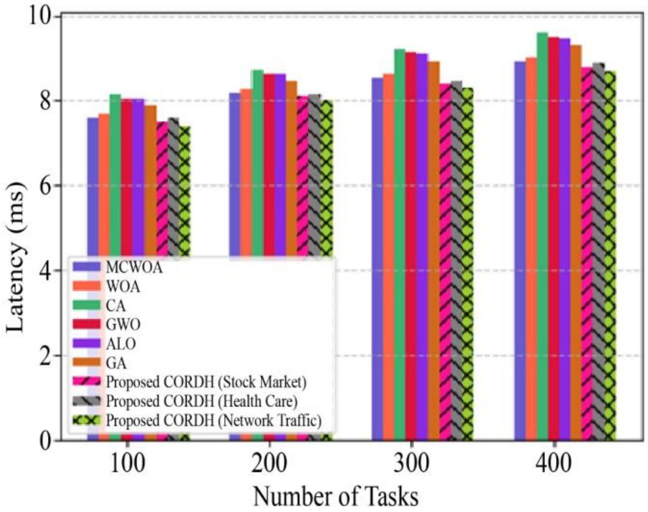


Fig. 5 Latency comparison with optimization

6. Performance Estimation

The proposed CORDH strategy features an efficient cryptanalysis approach that enhances data security while minimizing resource usage in a blockchain environment. This method's performance is evaluated using metrics such as accuracy, recall, error rate, confidentiality score, security level, search rate, processing time, and resource consumption.

6.1. Accuracy

Accuracy is a key metric that evaluates the alignment between a model's predictions and the true outcomes. Within the CORDH strategy, accuracy is assessed by measuring the model's effectiveness in precisely forecasting resource demand, optimizing resource allocation, and ensuring the proper functioning of blockchain operations. A high accuracy level signifies that the model's predictions closely match actual resource demands, optimizing computational resource utilization and reducing waste. This is essential for sustaining the efficiency and effectiveness of the CORDH model within a blockchain framework.

$$A_{cc} = \frac{t^p + t^n}{t^p + t^n + f^p + f^n} \quad (20)$$

AccuracyA_{cc} is calculated using Equation 20, which is calculated by dividing the total of true positives$t^p$and true negativest^n by the sum of true positives, true negatives, false positivesf^p, and false negativesf^n.

6.2. Recall

Recall is a critical performance metric that evaluates a model's ability to detect relevant instances among all truly relevant cases accurately. For the proposed CORDH strategy, recall is vital in measuring the model's effectiveness in detecting and managing security threats like brute force and DDoS attacks. A high recall rate signifies the model's proficiency in identifying the most genuine threats, thereby reducing false negatives and enhancing system security by recognizing and addressing attack attempts. In the resource allocation, a high recall rate ensures the model accurately identifies and allocates the necessary resources, ensuring all essential tasks receive adequate support.

$$R_e = \frac{t^p}{t^p + f^n} \quad (21)$$

Recall R_e is calculated using Equation 21 by dividing true positives by the sum of true positivest^p and true negativest^n. Figure 6 shows stock market accuracy, recall, error rate, and search rate.

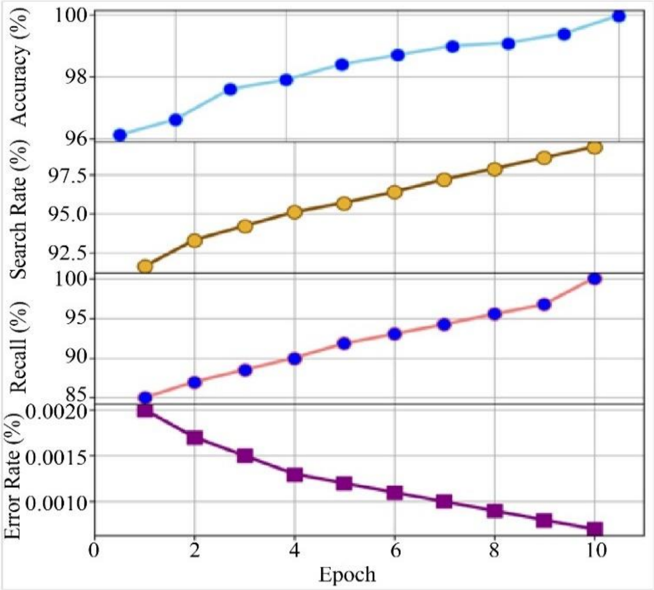


Fig. 6 Stock market proposed metrics (accuracy, recall, error rate, and search rate)

6.3. Error Rate

The error rate in Equation 22 R_{Error}quantifies the percentage of incorrect predictions or decisionsW_p made by a model compared to the total number of predictions or decisionsT_p. In the CORDH strategy, the error rate is used to evaluate the frequency of incorrect resource allocations or failures in detecting security threats. A lower error rate indicates higher accuracy and reliability of the model. In resource management, a low error rate ensures that resources are allocated correctly and efficiently, reducing wastage and improving performance. In security, a low error rate implies that the model effectively distinguishes between genuine threats and benign activities, thereby reducing false positives and ensuring that legitimate operations are not disrupted. Monitoring and minimizing the error rate is crucial for maintaining the overall effectiveness and robustness of the CORDH model in a blockchain environment.

$$\mathrm{R_{Error}} = \frac{\mathrm{W_p}}{\mathrm{T_p}} \times 100\% \quad (22)$$

6.4. Confidential Score of Before and After Applying Optimization

The confidential score evaluates the system's ability to maintain data confidentiality and integrity. It is measured before and after applying the COA to ensure that the data remains secure from unauthorized access and tampering attempts. Before optimization, the confidential score establishes a baseline for the system's security measures. After applying COA, the confidential score should ideally remain stable or improve, indicating that the system's security enhancements, including resource optimization and cryptanalysis strategies, are effective in safeguarding sensitive data. It is calculated using the equation below.

$$\mathrm{C_s} = \frac{\mathrm{C_{SD}}}{\mathrm{T_D}} \quad (23)$$

Where C_s the confidential score C_{SD} represents the number of correctly shared data samples being evaluated for their security and confidentiality, and T_D the total number of data samples. Figure 7 shows CORDH stock market metrics (confidential score, processing time, RAM and CPU usage).

6.5. Security Level

Security level measures the system's resilience against cyber threats, specifically brute force and DDoS attacks, which are simulated as part of the evaluation process. Brute force attacks test the strength of encryption and password policies, and DDoS assaults evaluate the system's capacity to manage large traffic volumes and preserve availability. A high-security level indicates that the CORDH strategy, along with the RNN-based attack monitoring and mitigation system, effectively detects, prevents, and mitigates these types of attacks, ensuring the system's robustness against potential threats.

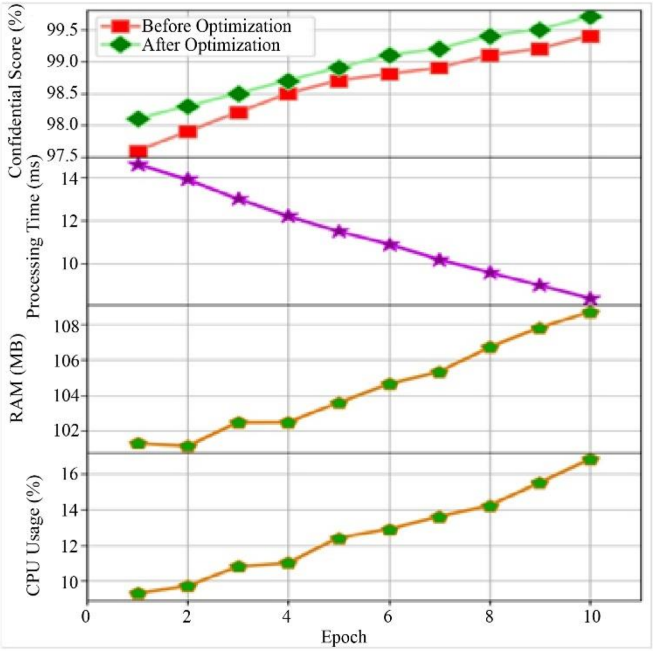


Fig. 7 Stock market Proposed metrics (confidential score, processing time, RAM and CPU usage)

6.6. Search Rate

The convergence curve of the search rate over 10 epochs for the proposed CORDH model reflects the efficiency and speed at which the model allocates necessary resources, such as VMs, CPU, and RAM. As the training progresses, the search rate steadily increases, indicating that the model is becoming more adept at identifying and provisioning the optimal resources. This trend demonstrates the model's ability to refine its resource allocation strategy over time, leading to enhanced computational efficiency and reduced latency in blockchain operations.

The consistent upward trajectory of the search rate highlights the effectiveness of the COA in improving the responsiveness and agility of the CORDH model in dynamic and changing environments. Figures 8 and 9 give BRDH Network traffic metrics (accuracy, recall, search and error rate, confidential score, processing time, RAM and CPU usage).

6.7. Processing Time

Processing time P_T measures the total time taken by the CORDH strategy to execute various tasks, including data preprocessing, model training, optimization, data transfer, and attack monitoring and mitigation. Lower processing time indicates higher efficiency in resource management and allocation, ensuring that tasks are completed within acceptable time frames. It is crucial to evaluate the overall performance and operational efficiency of the CORDH model in handling complex blockchain computations and maintaining real-time responsiveness. It is calculated by dividing the completion time by the total data-sharing time, and then multiplying the result by 100 and is given in the equation.

$$P_T = \frac{T_{ST}(s)}{T_T(s)} \tag{24}$$

Where T_{ST} denotes the single task completion time, $s$stands for tasks per second, and T_T is the total amount of time needed to do the assignment.

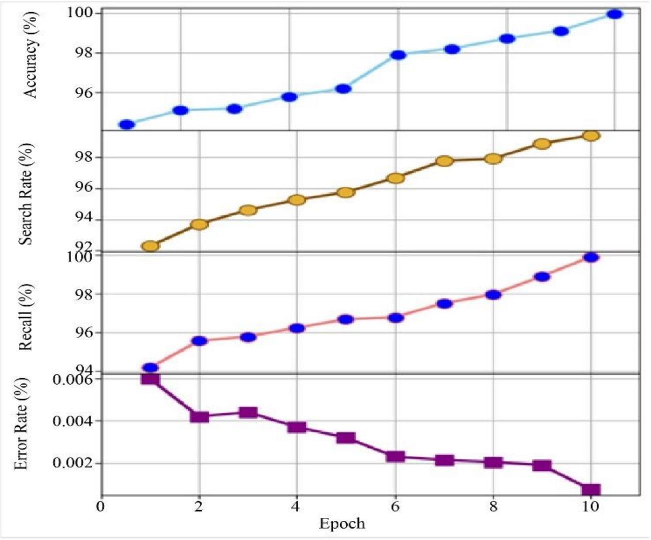


Fig. 8 Network traffic proposed metrics (accuracy, recall, error rate, and search rate)

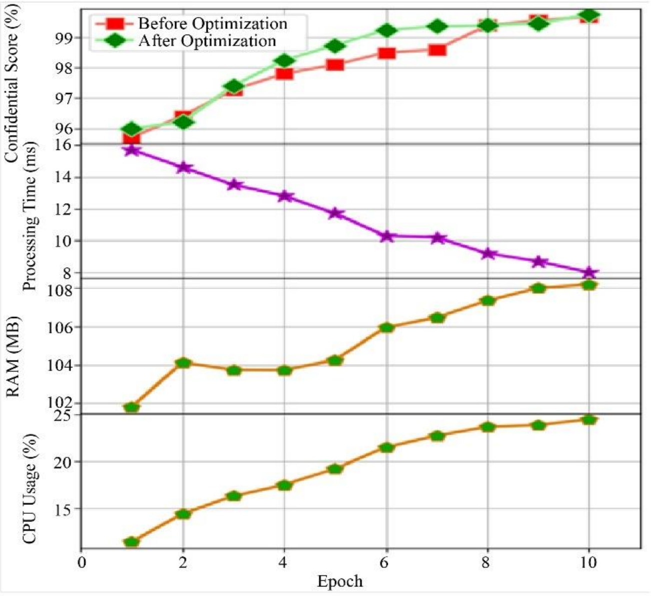


Fig. 9 Network traffic proposed metrics (confidential score, processing time, RAM and CPU usage)

Figures 10 and 11 give BRDH Network traffic metrics (accuracy, recall, search and error rate, confidential score, processing time, RAM and CPU usage).

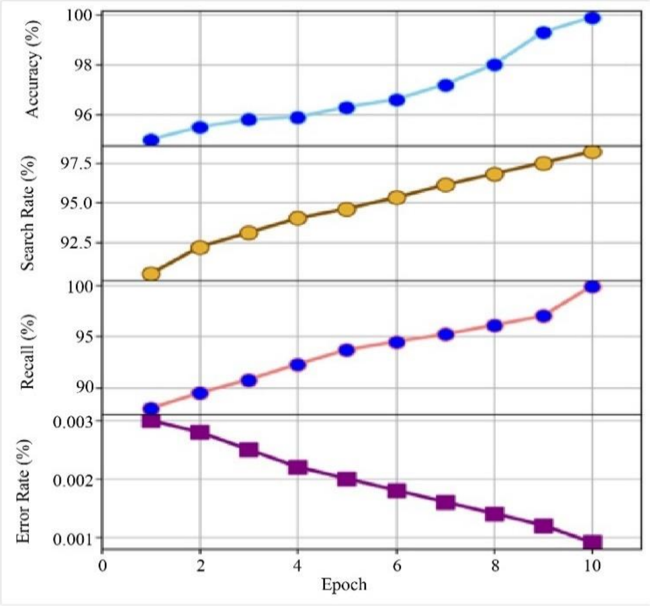


Fig. 10 Health care BRDH metrics (accuracy, recall, error rate, and search rate)

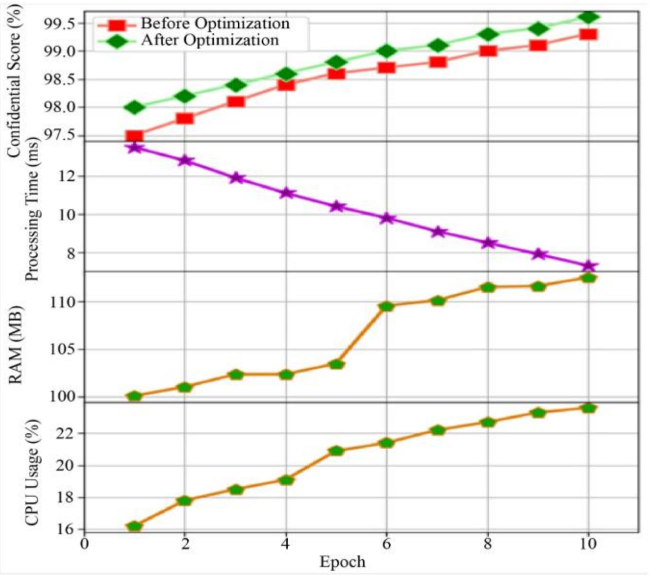


Fig. 11 Healthcare BRDH metrics (confidential score, processing time, RAM and CPU usage)

6.8. Resource Usage

Resource usage measures the amount of computational resources, including CPU and RAM, utilized by the CORDH strategy. The goal is to minimize resource usage while meeting the computational demands of blockchain operations. Efficient resource usage ensures cost-effectiveness and sustainability by reducing operational costs associated with running cloud-based services. This metric is critical for assessing the economic viability and environmental impact of the CORDH model, in addition to its performance and security benefits.

7. Comparative Analysis

The developed model's efficacy was assessed by comparing its metrics, such as accuracy, confidentiality score, security level, search rate, processing time, and resource consumption, with those of other existing models. Moreover, the existing techniques like a Lightweight Network Intrusion Detection System (LN-ID), Passban IDS (P-ID), and Hybrid Chain IDS (HC-ID) [28], Encrypted Scheme based on Curve Integration (ESCI), Deep-based Sensitive Aware Elliptic Curve Cryptography with Harmony Search optimization (DECC_HO), Educational Records Secure Storage and Sharing (ERSS), Deep Belief-based Diffie Hellman (DBDH) [27], Radial basis function (RF), Realguard, Kitsune, LSTM [29], Rivest-Shamir-Adleman- Simplified Swarm Optimization based Additional chain AC (RSA-SA), Elliptic Curve Cryptography (ECC) [30], Binary Particle Swarm Optimization and Grey Wolf Optimizer (BPSOGWO), Binary GWO (BGWO), Binary Whale Optimization Algorithm (BWOA) [31]. The performance analysis considers six optimization algorithms: Modified Chaotic Whale Optimization Algorithm (MCWOA), Whale Optimization Algorithm (WOA), Cuckoo Algorithm (CA), Grey Wolf Optimizer (GWO), Ant Lion Optimizer (ALO), and Genetic Algorithm (GA), each evaluated for their energy efficiency across multiple iterations [26], Bioinspired Blockchain (BioBlock), BB-IoT (Blockchain-Based IoT), BioAI-IoT (Bio-inspired AI for IoT), AI-BCIoT (Artificial Intelligence with Blockchain in IoT), and AI-SecIoT (AI-Enhanced Secure IoT) [33]. Recurrent neural Diffie-Hellman (RNDH), buffalo-based recurrent Diffie-Hellman (BRDH), Fruit fly-based load balancing recurrent Diffie-Hellman (FLBRDH).

These methods were selected based on their relevance in optimizing blockchain security, reducing computational overhead, and improving attack resilience. By comparing CORDH with these advanced methods, the study aims to establish its superiority in accuracy, error rate, resource usage, and security effectiveness under adversarial conditions. Each of these referenced methods represents recent advances in cryptography and resource management, making them ideal benchmarks for evaluating the proposed CORDH framework.

7.1. Comparison of the Proposed CORDH Method in Terms of Accuracy

The comparison of accuracy among different methods shows notable differences; the comparison values of accuracy are shown in Table 6. LN-ID achieves 82% accuracy, while P-ID slightly improves to 83.8%. HC-ID performs better with 94.8% accuracy. The proposed CORDH method significantly outperforms these, achieving 99.99% accuracy with the Stock Market Dataset, 99.90% with the Medical Dataset, and 99.95% with the Network traffic Dataset. This demonstrates the suggested CORDH method's superior and reliable performance over a range of datasets. The comparison of accuracy with the existing methods is visualized in Figure 12 below.

Table 6. Comparison of accuracy and recall

Methods	Accuracy (%)	Recall (%)
LN-ID [29]	82	73
P-ID [29]	83.8	75.4
HC-ID [29]	94.8	89
RNDH[33]	99.21	99.29
BRDH -S[34]	99.89	99.82
BRDH -H[34]	99.82	99.86
BRDH -N[34]	99.88	99.89
FLBRDH -S[35]	99.97	99.95
FLBRDH -H[35]	99.98	99.96
FLBRDH -N[35]	99.96	99.94
Proposed CORDH -S	99.99	99.99
Proposed CORDH -H	99.95	99.90
Proposed CORDH -N	99.80	99.9

Stock Market Dataset-S, Healthcare Dataset-H, Network traffic Dataset-N.

7.2. Comparison of the Proposed CORDH Method in Terms of Recall

Figure 12 compares recall rates of various methods in detecting relevant instances across different datasets, and the comparison values are shown in Table 6. LN-ID and P-ID methods show lower recall rates at 73% and 75.4%, respectively. HC-ID improves to 89%. RNDH and BRDH methods achieve nearly perfect recall rates of around 99.29% to 99.94%.

FLBRDH further enhances recall, reaching up to 99.98%. The proposed CORDH method excels with the highest recall rates: 99.99% for the stock market, 99.92% for healthcare, and 99.90% for network traffic datasets, demonstrating its superior effectiveness in identifying relevant instances.

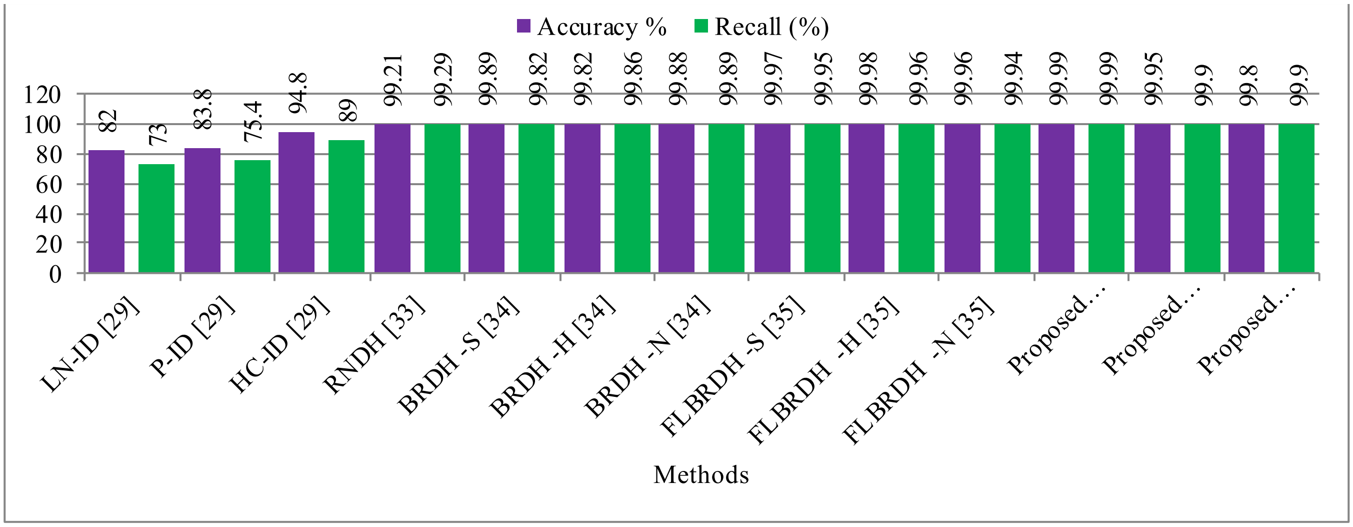


Fig. 12 Comparison of accuracy and recall

7.3. Comparison of the Proposed CORDH Method in Terms of Error Rate

The comparison of error rates among various methods highlights the superior performance of the proposed CORDH strategy across different datasets, which is visualized in Figure 13. And the comparison values of error rate are shown in Table 7. ESCI and DECC-HO exhibit relatively high error rates at 2.4% and 4%, respectively, indicating lower accuracy in identifying relevant instances. ERSS shows improvement with an error rate of 1.1%, while DBDH and RNDH demonstrate further enhancements, achieving error rates of 0.04% and 0.007%. The BRDH method applied to the Stock Market, Healthcare, and Network traffic datasets yields even lower error rates of 0.0015%, 0.0014%, and 0.0013%, respectively. Notably, the proposed CORDH strategy surpasses all these methods, attaining the lowest error rates of 0.0007% for the Stock Market dataset, 0.0009% for the Healthcare dataset, and 0.00074% for the Network traffic dataset. This demonstrates the proposed CORDH strategy's exceptional accuracy and reliability in various applications, significantly reducing the likelihood of errors compared to existing methods.

Table 7. Comparison values of processing time, error and confidential rate

Methods	Confidential rate (%)	Processing time (ms)	Error Rate (%)
ESCI [28]	79	30.00	2.4
DECC-HO[28]	76	50.00	4
DBDH [28]	95	12.00	0.04
ERSS [28]	84	15.00	1.1
RNDH[33]	-	-	0.007
BRDH -H[34]	-	-	0.0015
BRDH -S[34]	-	-	0.0018
BRDH -N[34]	-	-	0.0020
Proposed CORDH -S	99.70	8.40	0.0007
Proposed CORDH -H	99.60	7.30	0.00074
Proposed CORDH -N	99.75	8.00	0.0008

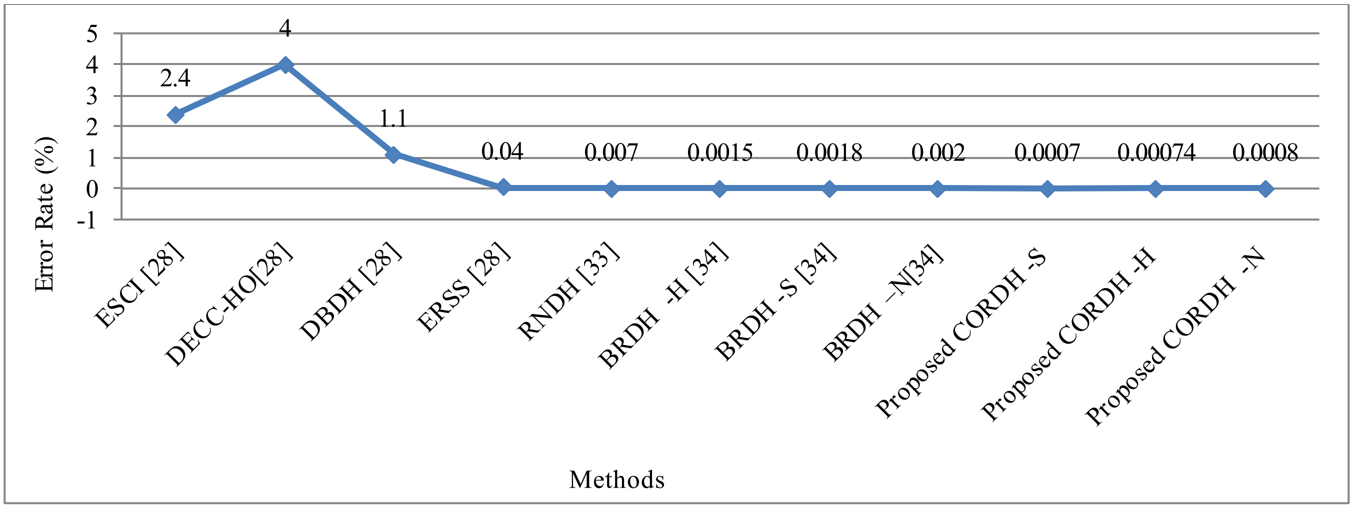


Fig. 13 Comparison of error rate

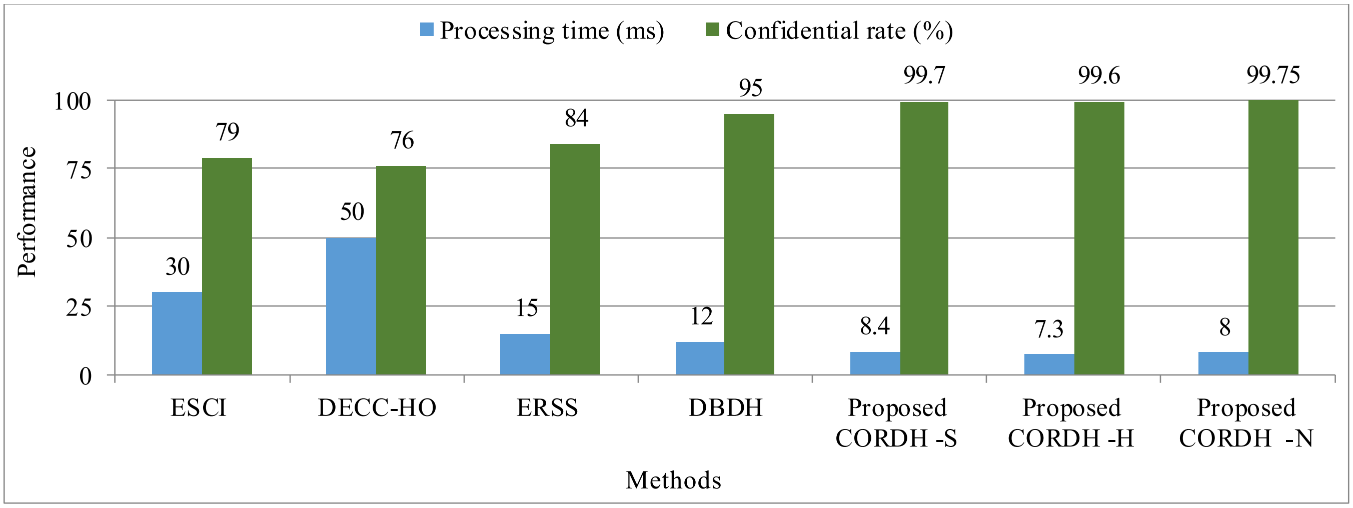


Fig. 14 Processing time and confidential rate comparison

7.4. Comparison of the Proposed CORDH Method in Terms of Confidential Rate

The comparison of Confidential Scores reveals the superior performance of the proposed CORDH method across different datasets. The comparison graph of the confidential score is represented in Figure 14, and its comparison values are shown in Table 7. Existing methods like DBDH (95%), ERSS (84%), ESCI (79%), and DECC-HO (76%) fall short compared to CORDH. For the stock market dataset, CORDH scores 99.40% before optimization and 99.70% after. For the medical dataset, it achieves 99.30% before optimization and 99.60% after. The network traffic dataset shows similar results with 99.68% before and 99.75% after optimization. These results highlight CORDH's robustness and effectiveness in ensuring data confidentiality, especially when optimized.

7.5. Comparison of the Proposed CORDH Method in Terms of Processing Time

The comparison of processing times across different methods highlights the efficiency of the proposed CORDH approach, which is shown in Figure 14. Table 7 provides the processing time comparison values. It takes 30 milliseconds to process using the ESCI technique, 50 milliseconds for DECC-HO, 15 milliseconds for ERSS, and 12 milliseconds for DBDH. In contrast, the proposed CORDH method significantly outperforms these methods with processing times of 8.4 milliseconds for the stock market dataset, 7.3 milliseconds for the medical dataset, and 8.0 milliseconds for the network traffic dataset. This demonstrates the superior performance of CORDH, particularly in handling diverse datasets with greater efficiency.

7.6. Comparison of the Proposed CORDH Method in Terms of Resource Usage

The comparison of CPU usage percentages among different methods reveals significant differences in their computational efficiency. The comparison graph of CPU usage is shown in Figure 15, and its comparison values are given in Table 8. Realguard and Kitsune exhibit moderate CPU usage at 36% and 33.8%, respectively. RF shows a much higher CPU consumption at 76.8%, indicating a heavy computational load. LSTM also has relatively high usage at 47.6%. In contrast, the proposed CORDH method demonstrates substantially lower CPU usage across different datasets: 16.80% for the Stock Market Dataset, 23.60% for the Medical Dataset, and 24.50% for the Network traffic dataset. This indicates that the proposed CORDH method is more efficient and requires less computational power than the other methods.

The comparison of the RAM usage of various methods for processing different datasets, highlighting the efficiency of the proposed CORDH method across multiple domains, is visualized in Figure 15. The comparison values of the RAM usage are given in Table 8. Realguard, Kitsune, RF, and LSTM exhibit higher RAM consumption, with RF being the highest at 180.3 MB. In contrast, the proposed CORDH method demonstrates lower memory usage across all datasets, with the stock market dataset at 108.70 MB, the medical dataset at 112.50 MB, and the network traffic dataset at 108.20 MB. This indicates that the proposed CORDH method is more efficient in terms of RAM usage compared to other methods, making it a more resource-effective solution for handling diverse data types.

Table 8. Comparison values of CPU and RAM usage

Methods	RAM (MB)	CPU (%)
Rearguard [30]	114.5	36
Kitsune [30]	156.3	33.8
RF [30]	180.3	76.8
LSTM [30]	143.1	47.6
Proposed CORDH -S	108.70	16.80
Proposed CORDH- H	112.50	23.60
Proposed CORDH -N	108.20	24.50

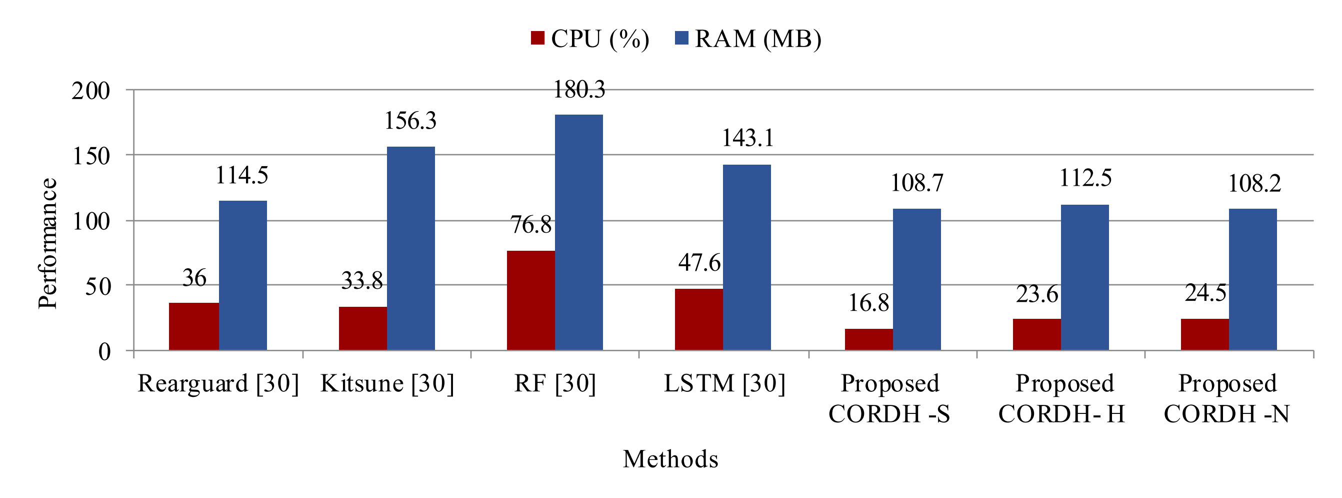


Fig 15. Comparison of resource usage

Table 9. Comparison values of security levels

Methods	Security Level (%)
RSA-SA [31]	91.00
ECC-SA [31]	92.80
Proposed CORDH -S	98.00
Proposed CORDH -H	94.90
Proposed CORDH -N	97.60

7.7. Comparison of the Proposed CORDH Method in Terms of Security Level

The comparison of the security levels among different methods shows varied levels of security, and the comparison graph of the security level is shown in Figure 16; then, the comparison values are given in Table 9. The RSA-SA method achieves a security level of 91%, while the ECC-SA method slightly outperforms it with a security level of 92.8%. The proposed CORDH method demonstrates superior security performance across different datasets: it achieves 98.00% with the Stock Market Dataset, 94.90% with the Medical Dataset, and 97.60% with the Network traffic Dataset. These results indicate that the proposed CORDH method provides a higher security level compared to traditional methods like RSA-SA and ECC-SA, particularly when applied to specific datasets.

7.8. Comparison of Anomaly Detection Accuracy, Latency and Energy Efficiency

The performance of various approaches was evaluated based on Anomaly Detection Accuracy, Energy Efficiency, and Latency. BioBlock achieved the highest accuracy at 95.2% but had moderate energy efficiency (91.2%) and latency (15.2 ms). BB-IoT showed lower performance across all metrics. BioAI-IoT and AI-BCIoT had comparable results, with accuracy around 91-91.5% and energy efficiency in the low 80s. In comparison, AI-SecIoT offered a slightly better accuracy of 93% and energy efficiency of 85.5% with a latency of 23 ms. The proposed CORDH variants (Stock Market, Healthcare, and Network traffic Dataset) demonstrated a good balance, with accuracy ranging from 92.0% to 92.7%, energy efficiency between 86.0% and 88.5%, and consistently low latency around 16-17 ms.

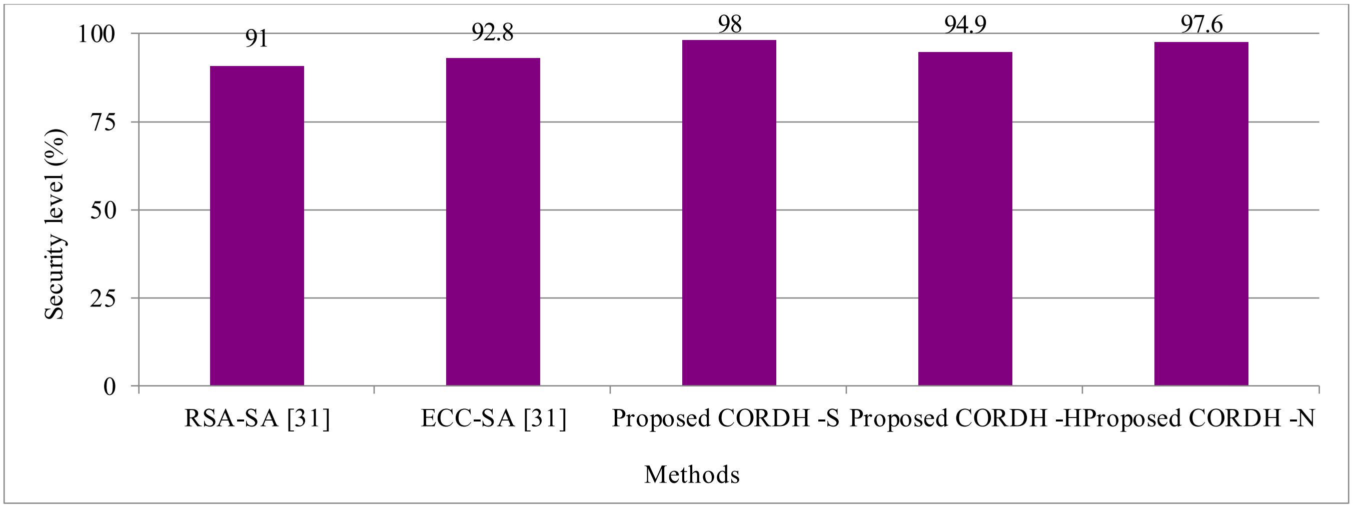


Fig. 16 Comparison of security level

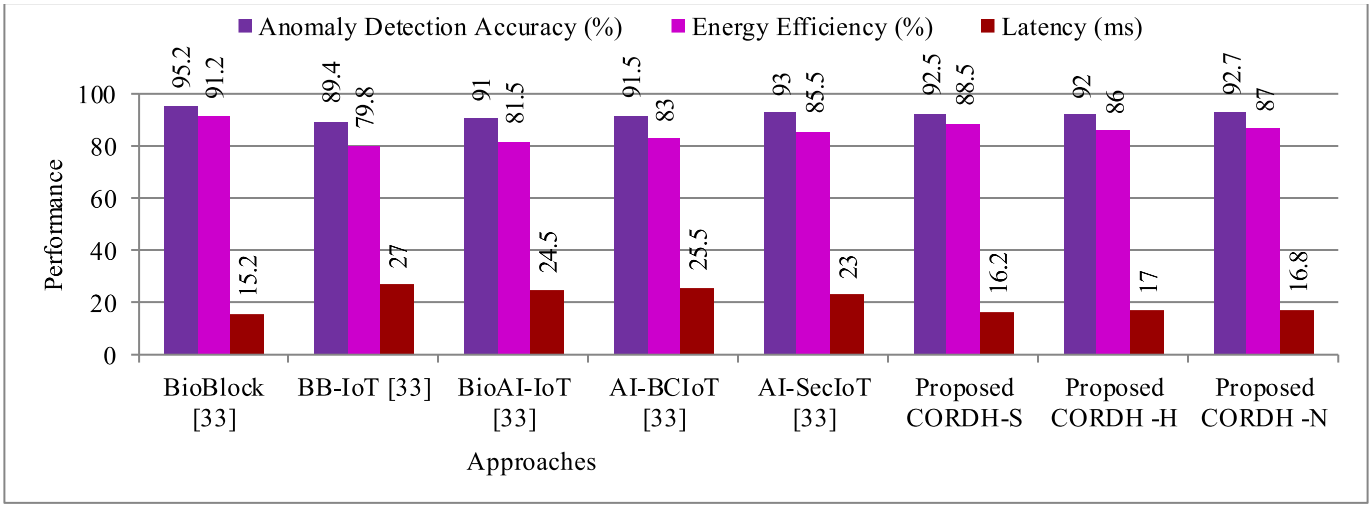


Fig. 17 The comparative graph on anomaly detection accuracy, energy efficiency and latency

Table 10. Comparison of anomaly detection accuracy, energy efficiency and latency

Approach	Anomaly Detection Accuracy (%)	Energy Efficiency (%)	Latency (ms)
BioBlock [33]	95.2	91.2	15.2
BB-IoT [33]	89.4	79.8	27.0
BioAI-IoT [33]	91.0	81.5	24.5
AI-BCIoT [33]	91.5	83.0	25.5
AI-SecIoT [33]	93.0	85.5	23.0
Proposed CORDH-S	92.5	88.5	16.2
Proposed CORDH -H	92.0	86.0	17.0
Proposed CORDH -N	92.7	87.0	16.8

Figure 17 shows a comparative graph on Anomaly Detection Accuracy, energy efficiency, and latency, and its readings are provided in Table 10.

7.9. Comparison of Search Rate

The suggested CORDH model performs better across a variety of datasets, as seen by the comparison of search rate percentages between techniques given in Table 11. The BPSOGWO method achieves a search rate of 93.75%, BGWO reaches 81.25%, and BWOA attains 87.5%, indicating strong but varying effectiveness in optimization tasks.

However, the proposed CORDH model significantly outperforms these methods, achieving a search rate of 99.30% on the Stock Market dataset, 98.20% on the Healthcare dataset, and 99.37% on the Network Traffic dataset, as shown in Figure 18.

Table 11. Comparison values of search rate

Methods	Search rate %
BPSOGWO [32]	93.75
BGWO [32]	81.25
BWOA [32]	87.5
Proposed CORDH -S	99.30
Proposed CORDH -H	98.20
Proposed CORDH -N	99.37

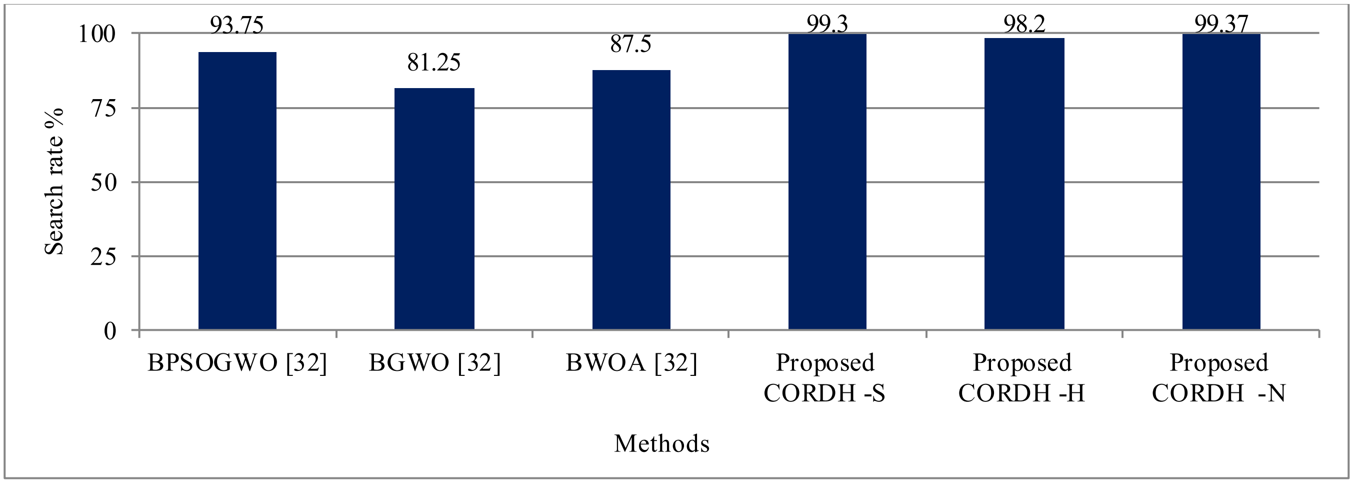


Fig. 18 Comparison of search rate

Table 12. Overall performance values of the proposed CORDH method

Model	Accuracy (%)	Recall (%)	Error Rate	Confidence Rate (%)
PSBA	99.79	98.99	0.0020	99.80
PSAA-B	99.50	99.80	0.0028	98.50
PSAA-D	99.40	99.70	0.0025	98.80
PHBA	99.96	99.97	0.00039	99.75
PHAA-D	98.86	98.95	0.0020	98.94
PHAA-B	98.91	99.26	0.0022	99.41
PNBA	99.54	99.65	0.00091	99.78
PNAA-D	99.25	99.25	0.0040	98.54
PNAA-B	99.41	99.37	0.0052	98.61

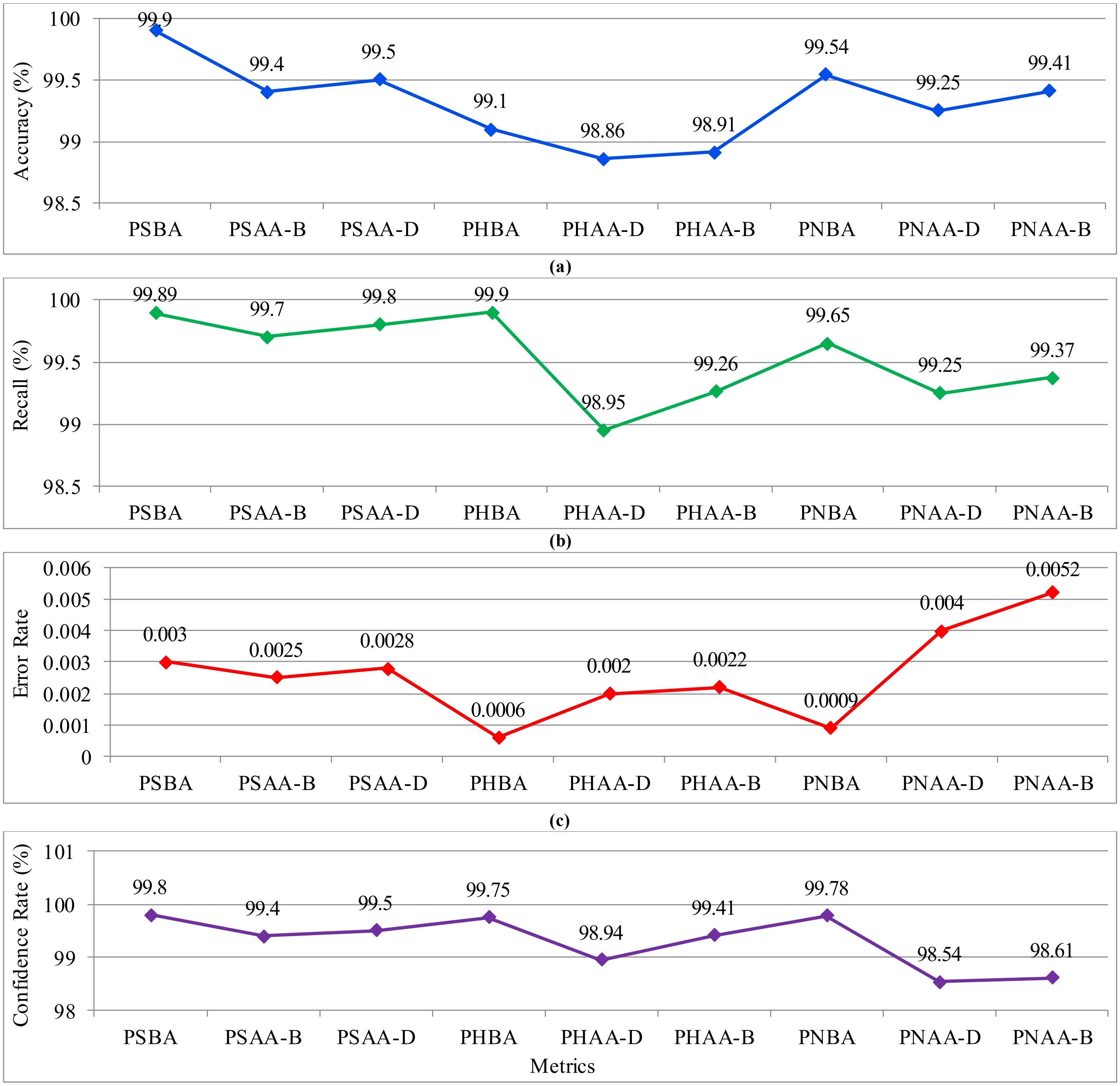
7.10. Overall Effectiveness of the Suggested CORDH

The system's performance is assessed both before and after the introduction of DDoS and brute force attacks, across all three datasets. The performance results are detailed in Table 12. Figure 19 depicts the effectiveness of the proposed Chimp-based Optimized Recurrent Diffie-Hellman (CORDH) method in mitigating these attacks, ensuring continuous and secure transmission. Additionally, the CORDH method optimizes blockchain computation costs by minimizing resource usage and enhancing security through an effective cryptanalysis strategy. Where,

- PSBA stands for Proposed CORDH Stock Market before the attack.
- PSAA-D stands for Proposed CORDH Stock Market After Attack DDoS.
- PSAA-B stands for Suggested CORDH Stock Market After Attack Brute Force.
- PHBA stands for Suggested CORDH Healthcare Before Attack.
- PHAA-D stands for Suggested CORDH Healthcare After Attack DDoS.
- PHAA-B stands for Suggested CORDH Healthcare After Attack Brute Force.
- PNBA stands for Suggested CORDH Network Traffic Before Attack.
- PNAA-D stands for Suggested CORDH Network Traffic After Attack DdoS.

7.11. Clarification and Contribution Justification

Existing blockchain security and resource optimization methods face significant drawbacks, including high computational overhead, limited scalability, complex encryption mechanisms, and inadequate cyberattack resilience. To address these challenges, the proposed CORDH strategy is evaluated using comprehensive experiments across healthcare, the stock market, and network traffic datasets. The COA significantly improves the exploration-exploitation balance in high-dimensional solution spaces, enabling more precise and dynamic resource allocation (e.g., VM, CPU, RAM) while reducing computational overhead. The results demonstrate CORDH's effectiveness. In terms of resource consumption, CORDH significantly reduced CPU usage to 16.8% (Stock Market dataset), compared to 76.8% for RF and 47.6% for LSTM, while RAM usage dropped to 108.7 MB versus RF's 180.3 MB. Regarding accuracy, CORDH achieved 99.99% with stock market data and 99.95% with healthcare data, far surpassing the 82%–83.8% accuracy of LN-ID and P-ID. Error rates were also reduced to 0.0007%, compared to 4% in DECC-HO. Additionally, to address the lack of real-time intrusion detection in prior methods, CORDH incorporates an RNN-based monitoring system that demonstrated resilience to brute force and DDoS attacks, with accuracy drops of only 0.39% post-attack (e.g., 99.79% to 99.4% for stock market data). This dual-functionality approach not only provides computational efficiency but also enhances security, making CORDH a comprehensive and scalable solution for blockchain environments. Furthermore, statistical validation via T-tests confirms significant improvements in anomaly detection accuracy and latency, supporting the method's effectiveness and reliability across diverse application domains, including healthcare, stock markets, and network traffic systems.



(d)

Fig. 19 Overall performance of the proposed CORDH method

8. Conclusion

The proposed CORDH method effectively addresses key issues in blockchain resource management and security. Traditional approaches often result in inefficiencies like high computational costs and poor resource utilization, while lacking robust security measures against brute force and DDoS attacks. The CORDH model uses COA and recurrent frameworks to dynamically allocate resources more efficiently, reducing both resource usage and computation costs while enhancing security. This method optimizes the allocation of virtual machines, with CPU and RAM, ensuring cost-effective and adequate resource provisioning for blockchain operations. Experimental results validated this with a maximum accuracy of 99.99%, a minimum error rate of 0.0007%, and a confidentiality score exceeding 99.7%. Additionally, CPU usage was reduced to 16.8% (compared to 76.8% in RF and 47.6% in LSTM), and RAM usage dropped to 108.7 MB. The model also achieved low processing time (as low as 7.3 ms), high-security levels (up to 98%), and strong robustness under DDoS and brute force attacks, with negligible accuracy loss (e.g., 99.79% to 99.4%). Future work will enhance the CORDH model's adaptability and scalability, integrating advanced machine learning techniques and expanding its application to diverse datasets. Continuous improvements in security measures will also be pursued to counter evolving cyber threats, ensuring the model remains robust and resilient.

References

[1] Moumita Das, Xingyu Tao, and Jack C.P. Cheng, "BIM Security: A Critical Review and Recommendations Using Encryption Strategy and Blockchain," *Automation in Construction*, vol. 126, pp. 1-57, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ehab Zaghloul et al., "Bitcoin and Blockchain: Security and Privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10288-10313, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xiongfei Zhao et al., "Minimizing Block Incentive Volatility through Verkle Tree-Based Dynamic Transaction Storage," *Decision Support Systems*, vol. 180, pp. 1-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xin Wang et al., "Blockchain-Enabled Decentralized Edge Intelligence for Trustworthy 6G Consumer Electronics," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1214-1225, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Safak Kayikci, and Taghi M. Khoshgoftaar, "Blockchain Meets Machine Learning: A Survey," *Journal of Big Data*, vol. 11, no. 1, pp. 1-29, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Haiao Li, Lina Ge, and Lei Tian, "Survey: Federated Learning Data Security and Privacy-Preserving in Edge-Internet of Things," *Artificial Intelligence Review*, vol. 57, no. 5, pp. 1-38, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] K. Sasikumar, and Sivakumar Nagarajan, "Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing," *IEEE Access*, vol. 12, pp. 52325-52351, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Muhammad Noman Sohail et al., "Optimizing Industrial IoT Data Security through Blockchain-Enabled Incentive-Driven Game Theoretic Approach for Data Sharing," *IEEE Access*, vol. 12, pp. 51176-51192, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Tri Nguyen, Huong Nguyen, and Tuan Nguyen Gia, "Exploring the Integration of Edge Computing and Blockchain IoT: Principles, Architectures, Security, and Applications," *Journal of Network and Computer Applications*, vol. 226, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Paraskevas Koukaras et al., "Integrating Blockchain in Smart Grids for Enhanced Demand Response: Challenges, Strategies, and Future Directions," *Energies*, vol. 17, no. 5, pp. 1-32, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Volodymyr Nakonechnyi et al., "Blockchain Implementation in the Protection System of Banking System during Online Banking Operations," *2024 35th Conference of Open Innovations Association (FRUCT)*, Tampere, Finland, pp. 492-500, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zhili Zhou et al., "Blockchain-Based Secure and Efficient Secret Image Sharing with Outsourcing Computation in Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 423-435, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Shams Forruque Ahmed et al., "Insights into Internet of Medical Things (IoMT), Data Fusion, Security Issues and Potential Solutions," *Information Fusion*, vol. 102, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Jiaxiang Zhang et al., "Adaptive Resource Allocation for Blockchain-Based Federated Learning in Internet of Things," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10621-10635, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Basim Aljabhan, and Muath A. Obaidat, "Privacy-Preserving Blockchain Framework for Supply Chain Management: Perceptive Craving Game Search Optimization (PCGSO)," *Sustainability*, vol. 15, no. 8, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Selvarajan Shitharth et al., "Federated Learning Optimization: A Computational Blockchain Process with Offloading Analysis to Enhance Security," *Egyptian Informatics Journal*, vol. 24, no. 4, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Qingqing Xie, Fan Dong, and Xia Feng, "HLOChain: A Hierarchical Blockchain Framework with Lightweight Consensus and Optimized Storage for IoT," *Security and Communication Networks*, vol. 2023, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yedida Venkata Rama Subramanya Viswanadham, and Kayalvizhi Jayavel, "A Framework for Data Privacy Preserving in Supply Chain Management Using Hybrid Meta-Heuristic Algorithm with Ethereum Blockchain Technology," *Electronics*, vol. 12, no. 6, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Tao Liu, Yi Yuan, and Zhongyang Yu, "An Intelligent Optimization Control Method for Enterprise Cost Under Blockchain Environment," *IEEE Access*, vol. 11, pp. 3597-3606, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Basel Halak, Yildiran Yilmaz, and Daniel Shiu, "Comparative Analysis of Energy Costs of Asymmetric vs Symmetric Encryption-Based Security Applications," *IEEE Access*, vol. 10, pp. 76707-76719, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Tanweer Alam, Arif Ullah, and Mohamed Benaida, "Deep Reinforcement Learning Approach for Computation Offloading in Blockchain-Enabled Communications Systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 9959-9972, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] P.S. Akshatha, and S.M. Dilip Kumar, "MQTT and Blockchain Sharding: An Approach to User-Controlled Data Access with Improved Security and Efficiency," *Blockchain: Research and Applications*, vol. 4, no. 4, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Oleksandr Kuznetsov et al., "Enhanced Security and Efficiency in Blockchain with Aggregated Zero-Knowledge Proof Mechanisms," *IEEE Access*, vol. 12, pp. 49228-49248, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] R. Aiyshwariya Devi, and A.R. Arunachalam, "Enhancement of IoT Device Security Using an Improved Elliptic Curve Cryptography Algorithm and Malware Detection Utilizing Deep LSTM," *High-Confidence Computing*, vol. 3, no. 2, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Saurabh Singhal et al., "Energy Efficient Resource Allocation in Cloud Environment Using Metaheuristic Algorithm," *IEEE Access*, vol. 11, pp. 126135-126146, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Chirag Chandrashekar et al., "MCWOA Scheduler: Modified Chimp-Whale Optimization Algorithm for Task Scheduling in Cloud Computing," *Computers, Materials & Continua*, vol. 78, no. 2, pp. 2593-2616, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Zhe Sun et al., "A Data Attack Detection Framework for Cryptography-Based Secure Aggregation Methods in 6G Intelligent Applications," *Electronics*, vol. 13, no. 11, pp. 1-22, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Akanksha Goel, and S. Neduncheliyan, "An Intelligent Blockchain Strategy for Decentralized Healthcare Framework," *Peer-to-Peer Networking and Applications*, vol. 16, no. 2, pp. 846-857, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ahmed A.M. Sharadqh et al., "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. [CrossRef] [Google Scholar] [Publisher Link]

[30] Xuan-Ha Nguyen et al., "Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways," *Sensors*, vol. 22, no. 2, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] A. Mullai, and K. Mani, "Enhancing the Security in RSA and Elliptic Curve Cryptography Based on Addition Chain Using Simplified Swarm Optimization and Particle Swarm Optimization for Mobile Devices," *International Journal of Information Technology*, vol. 13, no. 2, pp. 551-564, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Rizk M. Rizk-Allah et al., "On the Cryptanalysis of a Simplified AES Using a Hybrid Binary Grey Wolf Optimization," *Mathematics*, vol. 11, no. 18, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Abdul Rehman, and Omar Alharbi, "Bioinspired Blockchain Framework for Secure and Scalable Wireless Sensor Network Integration in Fog-Cloud Ecosystems," *Computers*, vol. 14, no. 1, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]