Journal of Circuits, Systems, and Computers
Vol. 34, No. 11 (2025) 2550080 (24 pages)
© World Scientific Publishing Company
DOI: 10.1142/S021812662550080X

Prevention of Insider Attacks Using Block Chain with Hierarchical Auto-Associative Polynomial Convolutional Neural Network in Cloud Platform[*]

Arul Selvam Palanisamy[†]

Department of Computer Science and Engineering,
Hindusthan College of Engineering and Technology,
Valley Campus, Pollachi Highway,
Coimbatore 641 032, Tamil Nadu, India
profarulselvamp24351@gmail.com

Tamije Selvy Perumal

Department of Computer Science and Engineering,
Sri Krishna Institute of Technology, India

Received 10 February 2023
Accepted 2 September 2024
Published 22 January 2025

In recent years, blockchain (BC) technologies have been increasing for data secrecy, system reliability and safety. BC is vulnerable to cyberattacks despite its utility. According to the statistics, attacks are rare, which differs greatly from the average. The goal of BC attack detection is to discover insights, patterns and anomalies within massive data repositories, it may benefit from deep learning. In this paper, the Prevention of Insider Attacks using Blockchain with Hierarchical Auto-associative Polynomial Convolutional Neural Network in Cloud Platform (PIS-BCNN-CP) is proposed. Here, the node authentication is handled by the smart contract. The aim of authorizing a node is to confirm that only a particular node has the possibility to submit and recover the information. Then Hierarchical Auto-associative Polynomial Convolutional Neural Network (HAAPCNN) is proposed to detect the Insider Attacks as Malicious and Normal. Generally, HAAPCNN does not agree with any optimization strategies to determine the optimal parameters for guaranteeing the exact detection of insider attacks. Hence, the Bear Smell Search Algorithm (BSSA) is exploited to optimize the weight parameters of a HAAPCNN. The BC Enabled Secure Data Storage depends on Proof of Continuous Work (PoCW) consensus BC algorithm is used. The proposed system is implemented and evaluated using performance metrics. The results provide higher accuracy, and lower False Negative Rate when compared with existing state-of-the-art methods.

Keywords: Access control; blockchain; confidentiality; Hierarchical auto-associative polynomial convolutional neural network; ledger; insider attack; smart contract; security.

[*] This paper was recommended by Regional Editor Emre Salman.

[†] Corresponding author.

1. Introduction

There is widespread concern that cloud computing poses serious security risks. The assessment found that although 90% of those surveyed were confident in their capacity to access and safeguard their data on the cloud, just 58% of the overall populace and 86% of highly experienced executives were hooked on the potential of cloud technology.[1] A malicious insider can infiltrate the cloud by masquerading as a legitimate one, thus besmirching the entire cloud and potentially affecting all clients who utilize the infrastructure in a problem.[2] Cloud computing has to deal with security issues like data integrity, information corruption, confidentiality for the service provider, confidentiality for the user, and data protection.[3] The vulnerability due to the insider's intent is a major concern for any data center and related infrastructural facilities.[4] The term "insider threat" is often used to describe an authorized user who exploits their accessibility to the computing system.[5] In accordance with the CERT research, insider threats can come from former/current employees, service companies, or even other company associates who would have authorized significant exposure to a protected network, regulatory regime, or information and purposefully surpassed or mishandled that privilege in a way that adversely impacted the privacy and security, authenticity, or accessibility of the concerning data or information processes of the entire organization.[6] Identification of an insider threat is a complex problem. For example, take the case of a sacked worker who intends to launch a violent assault on his previous company.[7] Suppose they still retain access to the system (through a loophole she/he has developed previously).[8] In that case, they are still an internal threat, even if their access permissions should have been revoked and they were no longer seen as real participants.[9] Some dishonest insiders potentially have the privilege to access vital data and may steal the concepts of their organization and then exploit their access to the cloud's data centers to pilfer or delete sensitive material.[10,11] Security breaches, including data theft via compromised process/access and utility computing through the cloud, are also possible outcomes of such attacks.[12] However, locating such a covert entry is a complex problem to solve. For example, suppose an adversary gains accessibility via a loophole to a data center and/or cloud information management.[13] In that case, they may launch an assault on any of the most popular types of cloud services, including Platform/software/infrastructure as a service.[14,15] Consequently, the cloud computing approach might be used to offshore enormous portions of the core tech rather than just fixed offerings like the application or webpage hosting.[16,17] This research was motivated by the blockchain system's growing security challenge, which results from the system's many uses.[18–20] The most important component of these commonly used applications is the existence of insider threats, which signal that system insiders typically hold advantageous positions.[21,22] When insiders hold these positions, systems are more susceptible to abuse and attacks. The emphasis on abnormality identification is warranted since the best approach to safeguard any system is to identify threats to the system early on and prevent ongoing attacks on it. Other approaches have been put out, but they all share some of the same limitations.

The primary contributions of this paper are abbreviated as follows:

- Prevention of Insider Attacks using blockchain with Hierarchical Auto-associative Polynomial Convolutional Neural Network in Cloud Platform (PIS-BCNN-CP) is proposed.
- Here initially the node authentication is handled by the smart contract. The aim of authorizing a node is to confirm that only a particular node has the possibility to submit and recover the information.
- Then executed Hierarchical Auto-associative Polynomial Convolutional Neural Network (HAAPCNN)[23] to detect the Insider Attacks as Malicious and Normal.
- Generally, a HAAPCNN did not adopt any optimization models espoused for estimating the optimum parameters to promise exact recognition of insider attacks.
- Then, the proposed BSA[24] is considered to enhance the weight parameters of HAAPCNN.
- The blockchain permitting safe data storage depending upon Proof of Continuous Work (PoCW) consensus blockchain[25] is used.
- At last, the consequences are gained and stowed sequentially on blockchain to promise their immutability and dependability.

The remaining paper is arranged as follows: Sec. 2 describes the literature survey, Sec. 3 designates the proposed method, Sec. 4 verifies the results, and Sec. 5 concludes this paper.

2. Literature Survey

Amongst numerous research studies about insider attack recognition in cloud computing, certain recent researches are discussed here.

Kirupanithi *et al.*[26] suggested blockchain network Insider Attack Detection Using a Trusted Two-Way Intrusion Detection System. The node authentication was originally handled by the smart contract. The node must be authenticated to particular nodes that may submit and obtain the information. Then use a hierarchically weighted fuzzy algorithm to assess the transaction nodes' trustworthiness. Then, using a self-organized stacked network deep learning verifies the harmful transactions or activities have been performed on the submitted transaction. It provides higher specificity and minimal *f*-score.

Tukur *et al.*[27] suggested a cloud platform with edge-based blockchain-enabled anomaly identification of insider attacks. The method implemented the power of edge computing to lessen the delay as well as bandwidth supplies. By taking a closer process to the node, it enhances obtainability and also evades solitary points of fault. It uses certain features of sequence-based anomaly detection, at the same time assimilating the distributed edge to the blockchains, which provides smart contracts to recognize and correct the irregularities of incoming sensor data. It provides a minimum error rate conversely with a maximum false positive rate.

Alkadi *et al.*[28] presented the protection of cloud networks, using a deep blockchain framework enables collaborative intrusion detection. The deep blockchain framework to deliver cloud with privacy-based blockchain along smart contracts also secures distributed intrusion detection. UNSW-NB15, BoT-IoT data sets evaluate the intrusion detection approach that uses a bidirectional long short-term memory deep learning to deal with sequential network data. For the improvement of the privacy-based blockchain as well as smart contract approaches, the Ethereum library was applied. It provides higher precision with lower integrity.

Deep *et al.*[29] presented preventing insider threats in a cloud context using distributed databases as a service. By offering a strong framework in admittance regulator appliances created to stop inside dangers utilizing a distributed architecture-based authentication mechanism, a Blockchain-dependent authentication mechanism, also monitoring insider activity mechanism, the research attempts to close the gap. The framework gives the ability to manage dispersed insider activity tracking for Database as a Service (DBaaS) and authentication mechanisms. It provides higher accuracy with minimum security.

Hu *et al.*[30] presented tracking insider attacker based on blockchain traceability scheme for insider threats. To prevent insider attackers, it involves first creating internal threat architecture in the internal network at different angles. After that, consider the challenge it is to find attackers and gather proof in cases when there is an insider danger. After that, transaction structure, data structure, consensus algorithm, block structure, data storage algorithm, and query algorithm for the blockchain tracing system are built which employs differential privacy to safeguard user privacy. It reached greater accuracy, but more error rate.

Awadallah and Samsudin[31] presented the use of blockchain in Cloud computing to improve relational database safety. It introduces the improved cloud relational database (RDB) structure, dubbed blockchain over cloud-RDB, also depending upon blockchain mechanism. That allows the client recognition to prevent indelicate RDB manipulation by self-verification progression. Agile BC-based RDB and secure BC-based RDB were the two solutions that were suggested as ways to enhance cloud-RDB. Using the Byzantine Fault Tolerance consensus basis, both were divided across various cloud service providers. It provides a minimum error rate with minimum security.

Gayathri *et al.*[32] presented Insider Threat Analysis with a Hybrid Deep Learning utilizing SPCAGAN Augmentation. A hybrid model was presented for insider threat assessments in terms of deep learning. Using benchmark datasets, extensive experiments were assessed for anomaly finding, adversarial strength, data synthesis and synthetic data quality assessments. The suggested SPCAGAN method avoiding issues were collapsing and converging types more quickly compared to earlier generative adversarial network design when used to generate synthetic data. It provides higher specificity with minimum accuracy.

Based on the literature survey presented above, the following advantages and disadvantages are observed by the previous methodologies used.

The research works presented focus on addressing insider attack recognition in cloud computing environments through a variety of innovative approaches. These include utilizing smart contracts and deep learning models for node authentication and transaction verification, integrating privacy-based blockchain frameworks with intrusion detection systems, and developing distributed architecture-based frameworks for insider threat management. Advantages of these approaches include improved precision, scalability, and privacy protection. However, challenges such as complexity in implementation, resource-intensive requirements for deep learning models, and potential overhead in maintaining distributed architectures and privacy mechanisms exist. Additionally, efforts towards preventing insider attackers from escaping through blockchain tracing systems with differential privacy and securing cloud relational databases using blockchain mechanisms offer promising solutions but may require careful consideration of performance impacts and coordination among multiple service providers. Furthermore, hybrid models combining deep learning techniques for insider threat assessments demonstrate competitive performance but may necessitate meticulous parameter tuning and handling of complex datasets.

3. Proposed Methodology

In this work, the Prevention of Insider Attacks using Blockchain with Hierarchical Auto-associative Polynomial Convolutional Neural Network in Cloud Platform (PIS-BCNN-CP) is proposed. The block diagram of the proposed PIS-BCNN-CP approach is shown in Fig. 1. The explanation about the prevention of Insider Attacks in Blockchain in the cloud platform is given below.

Figure 1 illustrates the integrated process of Secure Data Storage and insider attack detection. Initially, a smart contract ensures secure node authentication is given to regulate information submission and retrieval. Then HAAPCNN identifies insider attacks, distinguishing between malicious and normal behaviors. The Bear Smell Search Algorithm plays a key role in enhancing the accuracy of detection by fine-tuning the weight parameters of HAAPCNN. Additionally, blockchain technology, particularly the PoCW consensus algorithm is employed to guarantee secure storage of data. This comprehensive strategy combines smart contract authentication, advanced neural network detection, BSA optimization, and blockchain-powered data security, strengthening the system's ability to fend off insider threats.

3.1. *Data acquisition*

Data of malicious attacks from the nodes is taken in the typical reputation mode. There is just a low threshold value established, and a node is deemed malevolent if its

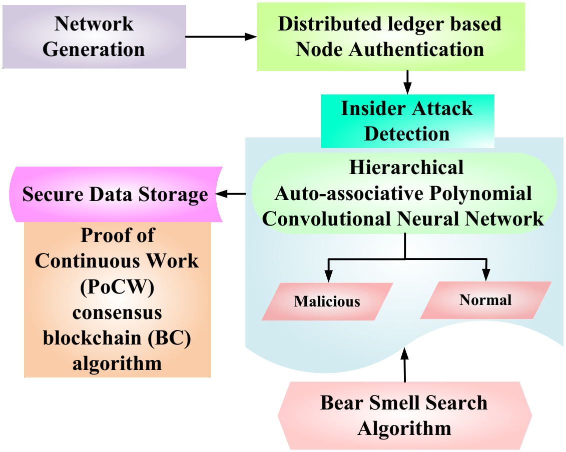
Network Generation
Distributed ledger based Node Authentication
Insider Attack Detection
Hierarchical Auto-associative Polynomial Convolutional Neural Network
Malicious
Normal
Secure Data Storage
Proof of Continuous Work (PoCW) consensus blockchain (BC) algorithm
Bear Smell Search Algorithm

Fig. 1. Block diagram for PIS-BCNN-CP Methodology.

reputation value is below this level. A dual threshold value for G and E in BCNN is set. Less reputation interval ranging is 0 to G, medium reputation interval as G to E and E to 1 for high reputation interval. If the node has less reputation interval, it is considered a malicious attack. One may classify the node in high reputation interval as a typical attack. To obtain data on malicious attacks within the described network topology it uses a formula for $G = (V, E)$ where $V = \{V_1, \ldots, V_M\}$ is collection of entire node along $|U| = M$, E is arbitrary set of vertices. The communication path is denoted through the group of edges $E \subseteq V \times V$ wherein$(V_i, V_j) \in E$. There is any communiqué connection (V_i, V_j) among Vi and Vj, then V_i and V_j has two way communication amidst each other. The node i neighbor set is determined by Eq. (1):

$$M_i = \{V_i \in V : V_i, V_j \in\}, \quad \forall j \in \{1, 2, \ldots, M\}. \tag{1}$$

To node I for updating their local state information, it communicates to their instantaneous neighbor $j \in M_i$. Assume that this consensus processing is activated in entire L discrete instances without sacrificing generality. Consider $b_i^k(0) = b_i^k$ as the beginning condition of node I normally, wherein b_i^k is a separate random variable that remains stable throughout time. The recursion evaluates the degree of similarity and direction amongst the current reputation matrix and benchmark node trust matrix when the node's reputation value falls within the medium interval represented in Eq. (2):

$$b^k(r+1) = q^k(r)b^k(r), \quad r = 1, 2, \ldots, R \tag{2}$$

where $q^l(r) \in R^{M\times M}$ implies weight matrix at 1 instance time r; $b^k(r) = (b_M^k(r))^r \in R^M$ implies random vector signifying the states of every node in rth iteration. We no longer write $Q(r)$ with superscript 1 because that prefix is needless in attack prevention. The node I selects the node j randomly and its states are synchronized such that they are identical to mean in rth time frame. The weight matrix is described in Eq. (3):

$$Q_{ij}r = I - \frac{(e_i - e_j)(e_i - e_j)^r}{2} \tag{3}$$

here, $e_i = [0, \dots, 0, 1, 0, \dots, 0]^r$ is a $M \times 1$ unit vector along rth component equivalent to one. Through describing $[X]_{ij} = X_{ij}$ and $\sum = \text{diag}([\sum_1, \dots, \sum_M])$ as a diagonal matrix $\sum_j = \sum_{j=1}^{M}(X_{ij} + X_{ji})$, the estimated weight matrix is described in Eq. (4):

$$\overline{Q} = E[Q(r)] = I - \frac{1}{2M}\sum + \frac{X + X^r}{2M} \tag{4}$$

The threshold of node tolerable trust span is X_{ij}. If there is positive node reputation sequence similarity $\lambda_2(\overline{Q}) < 1$, it is deemed as normal and updated $i \in V$. If there is negative similarity $\lambda_2(\overline{Q}) > 1$, the node enters the observation period. If the node's reputation value drops after half an observation cycle $(I/2)$, this is considered malevolent. Using this malicious attack data are gathered from the nodes for classifying the input as malicious and normal attack.

3.2. *Distributed ledger-based node authentication*

In this section, the primary criteria for accurately identifying intrusions should be focused on each node's authentication trust, traffic volume and node activity levels. The goal of the proposed model is to make sure that potential new customers don't endanger the cloud organization when trying to enter the market. The cloud and the client comprise the two tiers on which the system runs. Before entering the cloud market, the new node must successfully complete the two authentication processes that have been recommended. The cloud verifies the validity of each client's certificate. A distinct certificate is given to each customer in the market by the cloud. Each certificate contains a distinct logical identity, media access control (MAC) address, public and private keys, and MAC address. Each market client must get in touch with the cloud service provider (CSP) to confirm certificates for other nodes because the CSP is the one who issues certificates for each node. To avoid malicious attacks, CSPs run virus as well as hacking tool scans before allowing a new node into the cloud market. The CSP will only accept certificates from CSPs that are on a specific list, thus the new node must give it to them. If the certificate was not issued by a reliable certificate authority, a new one is given to the node. The client transmits the new certificate, the MAC address, and the node ID in an encrypted message using the public key. The definition of authentication's evaluation value is deliberated in Eq. (5):

$$Q_r = \begin{cases} 1, & \text{authenticate node} \\ 0, & o, z \end{cases} \tag{5}$$

when utilizing a distributed ledger to assess each node's credibility, there are a number of elements to consider. Such measurements include things like response time, throughput, accessibility, and success rate. These properties and their values are kept on file by the CSP. You may be sure that you're always using the most recent parameters available because these settings are updated following each market occurrence. The overall length of time needed to complete the service request is the node's response time. Here is a way to determine how quickly the ith node responds in Eq. (6):

$$R_j = \frac{R_s - R_{\min}}{R_{\max} - R_{\min}}, \tag{6}$$

where $R_{\max}$ and $R_{\min}$ signify maximal and minimal response times in the area, R_s signifies node average response time. The node seems like a greater R_i score. Throughput is determined for ith node using Eq. (7)

$$H_j = \frac{H_s - H_{\min}}{H_{\max} - H_{\min}} \tag{7}$$

where $H_{\max}$ and $H_{\min}$ represent the maximal and minimal attainable throughputs in the area, and H_s indicates average node throughput. The minimal node's H_i value is to be invader. The term "availability" describes the length of time a system is completely functional when it is required to be. The CSP would attempt to resolve the issue in order to keep the node operational. However, a node's repeated failures imply that it is acting purposefully. As a result, the valuation of availability in an evaluation may be determined using Eq. (8):

$$V_j = 1 - \frac{V_0 - V_{\min}}{V_{\max} - V_{\min}}, \tag{8}$$

where V_0 implicates the sum of ith node's fault events, $V_{\max}$ refers to maximal restarts receiving through neighbors, $V_{\min}$ represents the smallest. The percentage of requests that are actually fulfilled is referred to as the success rate. As a result, the evaluation value for the success rate can be determined as follows Eq. (9):

$$W_j = 1 - \frac{W_0 - W_{\min}}{W_{\max} - W_{\min}}, \tag{9}$$

where W_0 signifies total requests processing via ith node, $T_{\max}$ the maximal count of requests processing from neighbors, and $T_{\min}$ as minimal. Higher contention rates happen when a competitor's inner node in the cloud market floods the network with more traffic than it can handle, preventing customers from using cloud services. One later node marks the connection as fragmented after multiple unsuccessful attempts to relay a packet by that node; the routing algorithm then starts looking for an alternative route. Cloud market packets can't be sent until another route is found. As a result, packet loss increases and throughput drastically decreases. The amount of unnecessary traffic that is injected into the system causes a rise in network latency and a decrease in service quality. Based on the research's conclusions, a trust score was given to each node in Eq. (10):

$$Q_r = \begin{cases} 1, & j \in 0 \\ 0, & o, z \end{cases}, \tag{10}$$

where Q_r specifies nodal trust score, and o specifies set of trust distance. Node ranking is calculated by Eq. (11):

$$L_j = \sum_{R \to \infty} Q_r \tag{11}$$

Consider L_j implies node ranked matrix.

$$M_j = \frac{L_j - L_{\min}}{L_{\max} - L_{\min}} \tag{12}$$

where M_j refers to ith node's activity level, $L_{\max}$ and $L_{\min}$ implicate maximal and minimal counts of activities, and j implies a set of every feasible activity. The following discussion pertains to insider attack detection utilizing HAAPCNN.

3.3. *Inside attack detection using HAAPCNN*

The theories and foundations of the suggested HAAPCNN framework are presented in this section. HAAPCNN is used as IDS for a blockchain-based cloud network to find threats. It can be viewed as a potent deep neural network that processes sequence data by looping internally stored information. The intrusion detection issues have temporary patterns in the users' manner, making the handling of the temporal sequence more pertinent. This would enhance the discovery of anomalies or outliers, which can be challenging to determine while trusting at spatial domain. Processed sequenced lists of crypto records are done so using the internal state of the HAAPCNN. A number of time steps are taken to process the input sequence, and to create a hidden state, associated memory is renovated. HAPCNN-basis insider attack detection comprises bandwidth and resource demand. An attempt is made to integrate a region-based concept and a polynomial layer with the neural networks in HAPCNN derived from HAPNet. Considering that convolutional neural networks inherently incorporate region notion in their structure, the influence of a polynomial layer was extremely intriguing and the architecture design of HAAPCNN is shown in Fig. 2.

Without pre-trained networks, this study evaluates how bandwidth affects resource demand utilizing a traditional simple structure. Using Eq. (13), the convolutional polynomial is added to the original picture or the pooling layer is

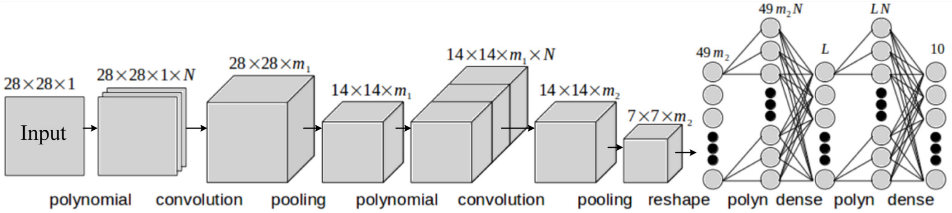

Fig. 2. Architecture design for HAAPCNN.

changed prior to convolution:

$$e_{\rho 1}(a,b,c,n) = e_{\rho 0}(a,b,c)^n. \tag{13}$$

For $n = 1, 2, \dots, N$ forward pass is continual along the polynomial layer presented $\rho = 1, 2, \dots, \rho$ on every iteration. Here e has three dimensions because e is formed by merging $e_{\rho 0}(a,b,c)^n$ with c axis, then e as $A \times B \times (C \times N)$. The weight makes forward pass convolution on $e_{\rho 0}$ which contains similar depth that originates prior to c. The weight links the attack recognition in the layer with length h to the layer prior to its length l with fully connected layers. When a polynomial is entered, the process deepens across the dimension displayed as the fourth dimension for transparency. To specify that the depth of weight implies N times larger, the weight varied from $w(i,j,c)$ to $w(i,j,c,n)$. Any process is employed to c or l variable at fully connected layers n where n implies second dimension. The attack convolutional process is exhibited in Eq. (14):

$$e_{\rho 2}(a,b,m_\rho) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \sum_{c=-1}^{C} \sum_{n=1}^{N} e_{\rho 1(a+i,b+j,c,n) w(i,j,c,n) + y_{m_\rho}}. \tag{14}$$

The fully connected layer is labeled in Eq. (15):

$$h_{\phi 1}(l,n) = h_{\phi 0}(l)^n \tag{15}$$

Update the fully connected layers by Eq. (16):

$$h_{\phi 2}(k) = \sum_{l=0}^{L-1} \sum_{n=1}^{N} h_{\phi 1}(l,n) w(l,n,k) + y_k. \tag{16}$$

After each of the Q iterations in the convolutional portion, a critical design choice is made; before reconfiguring the attack, one last polynomial layer may be added. Cloud-based insider attack is an index of servers for work allotment. The substitution of reconfiguring attack operations while utilizing an extra polynomial layer is articulated in Eq. (17).

$$e(Q+1)1(a,b,c,n) \rightarrow h_{\phi 0}(l) \tag{17}$$

when considering the backpropagation of the polynomial layer as a modified previous layer. A typical CNN has the same type of connections as its front layer at full energy cost. Since the two layers are only locally connected and do not overlap, back-propagating the error and loss from the polynomial layer to the previous layer in the fully connected area resembles non-overlapping average pooling. The bandwidth loss is transmitted back through the polynomial layer utilizing Eq. (18):

$$f_{\phi \text{prepoly}}(l) = \sum_{n=1}^{N} f_{\phi \text{prepoly}}(l,n) \tag{18}$$

where $f_{\phi \text{prepoly}}(l) \alpha f_{\phi - 1 \text{prepoly}}(l)$ are altered by insider attack activation and pooling functions, maximizing algorithm diversification. High-value $f_{\phi \text{prepoly}}(l)$ maximizes the mobility of all mass in the search space. The ideal solution space is found by assuming greater values$f_{\phi\text{prepoly}}(l)$. At last, exactly predict the insider attack. Any changes made to the contract will be reversed by the IDS upon detecting abnormal behavior. There is a chance to avoid irreparable loss due to vulnerability. Generally, a Hierarchical Auto-associative Polynomial Convolutional Neural Network could not agree on any optimization models to determine the ideal parameters. Therefore, BSSA is preferred to increase the weight parameters of the Hierarchical Auto-associative Polynomial Convolutional Neural Network.

In this work, the Bear Smell Search Algorithm (BSSA) is employed to enhance the better parameters of HAAPCNN classifier, also tuning the weight and bias parameters of HAAPCNN. Nevertheless, these explorations share their uncommon weakness concerning the reiteration period, then there is no subterfuge-accumulated popular examination. So, the BSSA is chosen in this work because this one contains self-development, also taking a slow iteration period compared to other tuning models, like grid, manual, and random investigations to find better weight parameter $e_{\rho 0}$ of HAAPCNN. The stepwise procedure of BSSA is delimited as follows.

3.4. *Stepwise processing of bear smell search algorithm (BSSA)*

The stepwise processing is delimited to reach optimal values of HAAPCNN utilizing BSSA. Firstly, BSSA makes the consistently dispersed populace optimize the optimal parameters of HAAPCNN. The BSSA is gaining attention as a nature-inspired optimization tool due to its powerful search operators, combining broad exploration akin to a bear's detection of odors from vast distances with focused exploitation resembling a bear closing in on prey. Inspired by the neural network of a bear's olfactory bulb, BSSA enhances exploration and exploitation capabilities. Compared to evolutionary algorithms, BSSA, a meta-heuristic algorithm, offers potential advantages such as achieving a balance between exploration and exploitation crucial for complex problems and the possibility of outperforming established algorithms in enhancing the hyperparameters of neural network. The best solution is stimulated to apply BSSA method then the associated flowchart is depicted in Fig. 3. The stepwise processing of BSSA is deliberated beneath.

***Step 1*:** Initialization

Initialize the populace of BSSA as f and maximum iteration ω for enhancing the weight parameter values $e_{\rho 0}$ of HAAPCNN. This is exhibited in Eq. (19):

$$R_i = [rc_i^1, rc_i^2 \ldots rc_i^j \ldots rc_i^k], \tag{19}$$

where r represents the breathing function, k represents the constant value for exhalation time, i denotes the direction of odors and j denotes the cycle of breathing.

***Step 2*:** Random generation

The input parameters are randomly produced afterward initializing. Therefore, optimum fitness is nominated in regard to their explicit hyperparameter circumstance.

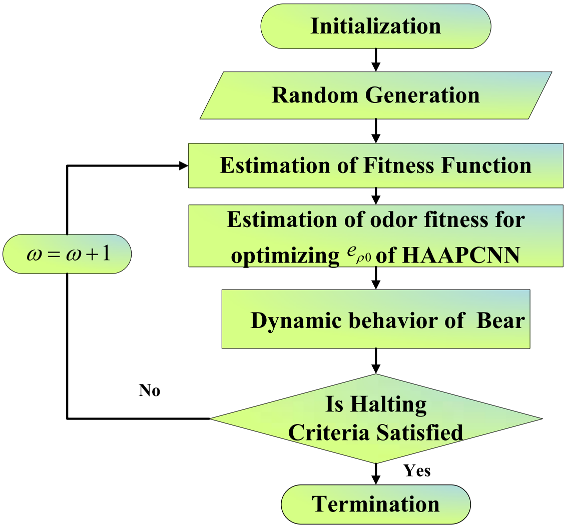
Initialization
Random Generation
Estimation of Fitness Function
Estimation of odor fitness for optimizing $e_{\rho 0}$ of HAAPCNN
Dynamic behavior of Bear
Is Halting Criteria Satisfied
No
Yes
$\omega = \omega + 1$
Termination

Fig. 3. Flowchart of BSSA for enhancing the weight parameter of HAAPCNN.

Step 3: Estimation of Fitness

Function The fitness function is estimated with parameter optimization to enhance the weight factor $e_{\rho 0}$ of HAAPCNN revealed in Eq. (20)

$$\text{Fitness Function} = \text{optimizing}[e_{\rho 0}]. \tag{20}$$

Step 4: Estimation of odor fitness for optimizing $e_{\rho 0}$ of HAAPCNN

Estimation of odor fitness is counterfeit in terms of Pearson correlation. So that opinion supports tolerating the selected preeminent mode to the following location which supports HAAPCNN for the detection of insider attack in a cloud environment. The probability odor components are labeled in Eq. (21):

$$\text{POC}_a = \frac{R_a}{\max(R_a)}. \tag{21}$$

Let POC represent Probability odor components, and R_a denotes threshold value and arrays odor length depending upon the middling rate of odor's data.

Step 5: Dynamic behavior of the bear

BSSA mimics together dynamic manners of abiding on the basis of sense of smell appliance and the tactic endure moves on the pursuit of food in thousand miles beyond. The neural architecture of the vertebrate forebrain, the olfactory bulb, allows for strong exploration and exploitation leading to optimization. This is expressed in Eq. (22):

$$P_a(R_a) = \frac{1}{n}\sum_{a=1}^{n} e(R_{ax}^{y}) = \left\{ \begin{matrix} 1, & s_1 \le R_{ax}^{y} \\ 0, & s_1 \ge R_{ax}^{y} \end{matrix} \right\} \tag{22}$$

where e denotes the local solution of each odor, x and y indicate the local and global solutions and P_a represents the dissimilarity between two odors.

***Step 6*:** Termination Condition

The weight parameter $e_{\rho 0}$ of HAAPCNN is optimized under BSSA repeating iterative step 3 until fulfilling the halting condition $\omega = \omega + 1$. In the end, HAAPCNN with BSSA detects the insider attack with greater accuracy by diminishing the error rate. The detailed discussion about blockchain-enabled secure data storage is given below.

3.5. *Blockchain-enabled secure data storage*

This is employed to secure the attack data utilizing BC basis consensus approach. In this case, the data storage process is carried out in accordance with the storage servers and BC model. Subsequently, the abridged data is saved in BC transaction objects. The original attack data is kept on the BC storage server, which can improve data security. The explanation about PoCW is specified below.

3.5.1. *Blockchain-based proof of continuous work (PoCW) consensus framework*

PoCW is an effective way to increase security compared to traditional blockchain techniques. PoCW comprises a chain of proof transactions and incentives for data storage.

- **Chain the Proof Transactions**

Chain structure is taken into consideration in PoCW for organizing the data proofs. In this case, using the last transaction references which are updated by the same miner who can construct the first transaction required to finish the registration process — the memory nodes are allocated to the proof transaction. It adapts the actual mode of decrypted data proof transaction by combining the reference fields as articulated in Eq. (23)

$$Txn_{\text{proof}} = (hash, idx, ciphertext_{\text{off } x}, R, \pi) \tag{23}$$

where $hash$ specifies block hash to locate prior transaction, idx specifies transaction list index, R represents random value, π signifies storage proof.

The miners, long transaction chains and stored data are reliable based on the observation. Only two parameters are utilized to adjust the mining challenges. Consider the length of proof transaction (p_L) and the count of allocated information (num). The calculation of the mining struggle is expressed in Eq. (24)

$$H(di) < (p_L * num + 1) * d \tag{24}$$

here d implies block distance. The miners in BC are not in charge of operations and storage. However, they can choose to define during the proving procedure. Miners make the mistake of persistently acquiescing evidence that varies counter's variable and number to affect mining trouble.

To lessen the accumulation of (p_L) maximize without limitation, the BC creation is regulated through a prior miner. Whenever a miner adds a new block to BC, the counter is updated and visible to all. The BC maintains track of the state of the data set at a higher level and the mining set at each block height. Following the successful integration of a miner's new block, the BC clears the miner's counter.

- **Data Storage**

It deems the profit from miners. To a single miner, income contains rent payments as of the data owner and honors from effectual mining. The larger miner that joins in the proposed consensus makes more money than before since transactions with the help of two consensus techniques that have comparable returns as well as expenses. A miner forfeits a significant cumulative mining gain if it leaves midway through the storage contribution. Thus, the proposed approach produces stable participants when miners anticipate their rationality. Removing the lag between two successive resource transactions makes the transaction chain more complex. To this end, the proposed PoCW-based blockchain scheme increased the safety of data storage which ensures its immutability and reliability.

4. Results and Discussions

The simulation performance of Prevention of Insider Attacks using Blockchain with HAAPCNN in the Cloud Platform is discussed. A Blockchain-dependent PoCW consensus approach is activated in Ethereum and Solidity programming language (version 0.6.50) and installed in the Interplanetary file system (version 0.4.19). Moreover, the mentioned metrics are estimated. The obtained results of the PIS-BCNN-CP is compared to the existing systems, such as PIA-BC-HWFA-SOSN,[26] PIA-BC-EBC[27] PIA-BC-Bi-LSTM.[28]

4.1. *Performance metrics*

This is scrutinized to confirm the performance of the PIS-BCNN-CP technique. For that, the below confusion matrix is needed.

True Positive (T_P): Normal attack identified as normal.
True Negative (T_N): Malicious attack is identified as malicious.
False Positive (F_P): Malicious attack identified as normal.
False Negative (F_N): Normal attack identified as malicious.

4.1.1. *Accuracy*

This is defined as the rate of successful predictions to the whole proceedings in the dataset using Eq. (25)

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{25}$$

4.1.2. *Precision*

The ability of classifiers to determine normal attack devoid of any states by using Eq. (26):

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{26}$$

4.1.3. *Sensitivity*

The quantity of real positives exactly predicted and scaled by Eq. (27):

$$\text{Sensitivity} = \frac{T_P}{F_N + F_P} \tag{27}$$

4.1.4. *Specificity*

It is also known as TN rate computed through Eq. (28):

$$\text{Specificity} = \frac{F_N}{F_P + F_N} \tag{28}$$

4.1.5. *F-score*

It is determined using Eq. (29):

$$F\text{-Score} = \frac{2T_P}{2(T_P + F_P + F_N)} \tag{29}$$

Table 1 tabulates the assessment of the proposed and existing models in blockchain schemes. The Verifiable limit shows the ability to evaluate and monitor every transaction by using timestamp archives. Utilizing BC technology, the PIS-BCNN-CP strategy enhances transaction verifiability. In the Cloud computing network, security, confidence, and secrecy are regarded as the three major interactions. The processing time of the proposed PIS-BCNN-CP is better and attains the Scalability Fortification.

4.2. *Simulation results*

Figures 4–14 exhibit the simulation outcomes of the Prevention of Insider Attacks using Block Chain with HAAPCNN in Cloud Platform. The performance metrics are

Table 1. Assessment of the proposed and existing methods in the blockchain system.

Models	Verifiability limit	Safety, trust, concealment	Processing period	Scalability fortification
PIA-BC-HWFA-SOSN	No	No	less	No
PIA-BC-EBC	No	No	less	No
PIA-BC-Bi-LSTM	No	No	less	No
PIS-BCNN-CP (Proposed)	Yes	Yes	high	Yes

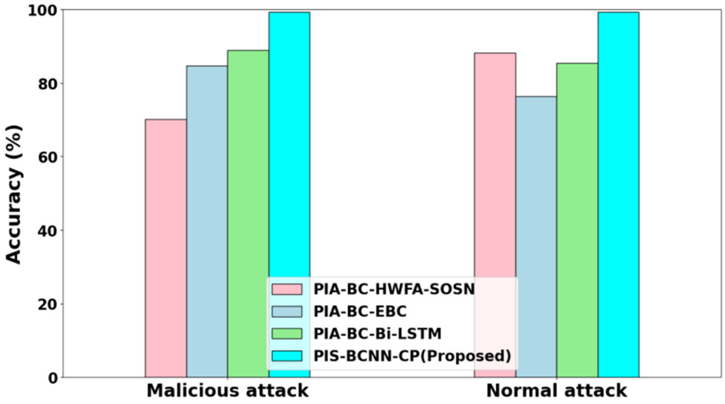

Fig. 4. Performance analysis of accuracy.

analyzed with existing PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM models, respectively.

In Fig. 4, an accuracy analysis is presented comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results demonstrate significant improvements in accuracy achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP outperforms existing methods by 34.73%, 29.94% and 22.33%, while for normal attacks, it achieves 24.86%, 16.97% and 33.86% higher accuracy. These results suggest the superiority of the PIS-BCNN-CP in accurately detecting both malicious and normal attacks, highlighting its potential effectiveness in enhancing security in blockchain systems.

Figure 5 portrays the sensitivity analysis. The results indicate notable improvements in sensitivity achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP outperforms existing methods by 26.98%, 38.08% and

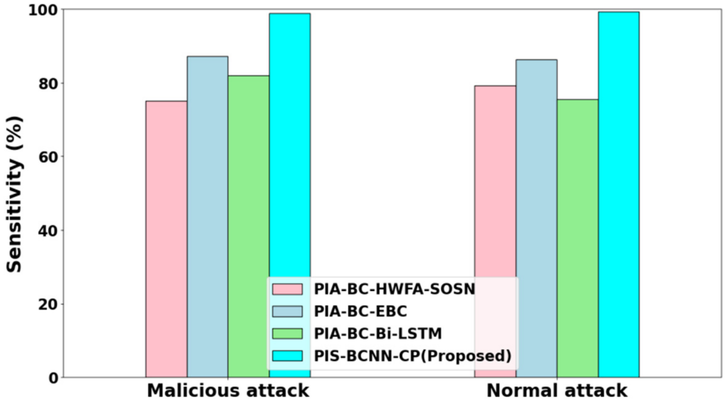

Fig. 5. Sensitivity analysis.

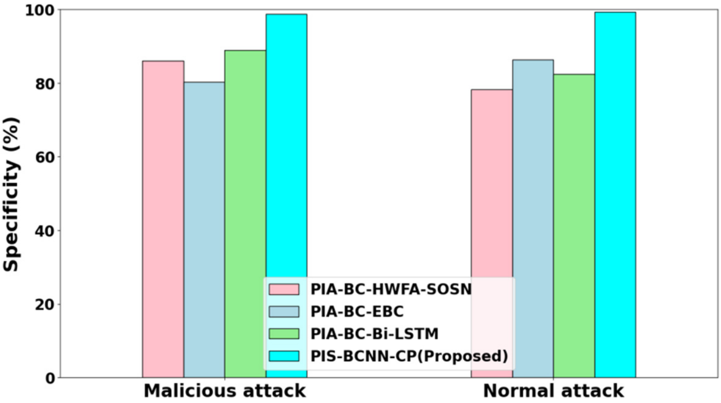

Fig. 6. Specificity analysis.

36.77%, while for normal attacks, it achieves 34.73%, 29.94% and 22.33% higher sensitivity. These results underscore the efficiency of the PIS-BCNN-CP in accurately detecting both malicious and normal attacks, suggesting its potential for enhancing security in blockchain systems.

In Fig. 6, specificity analysis is presented, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results reveal significant enhancements in specificity achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP outperforms existing methods by 24.86%, 16.97% and 33.86%, while for normal attacks, it achieves 26.98%, 38.08% and 36.77% higher specificity. These outcomes underscore the proficiency of PIS-BCNN-CP in accurately distinguishing between malicious and normal activities, indicating its potential to bolster security measures in blockchain systems.

In Fig. 7, precision analysis is presented, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results indicate substantial improvements in precision achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP outperforms existing methods by 25.75%, 35.76% and 24.65%, while for normal attacks, it achieves 24.75%, 25.64% and 31.54% higher precision. These outcomes underscore the superior precision of the PIS-BCNN-CP in accurately identifying both malicious and normal activities, demonstrating its potential to enhance security measures in blockchain systems.

In Fig. 8, F-Score analysis is depicted, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results illustrate significant improvements in F-Score achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP outperforms existing methods by 33.76%, 27.98% and 32.06%, while for normal attacks, it achieves 37.86%, 25.06% and 34.96% higher F-Score. These results

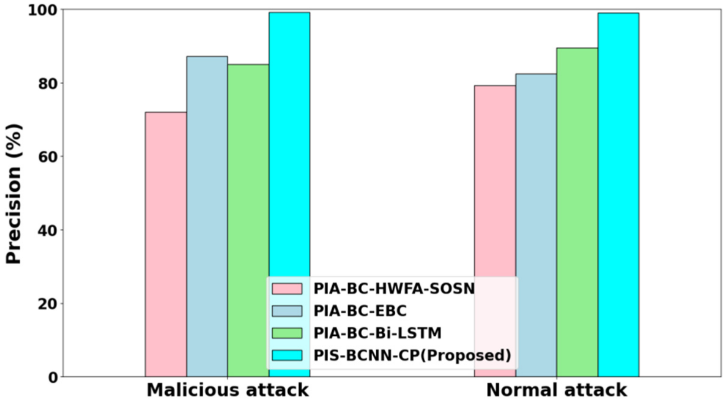

Fig. 7. Precision analysis.

underscore the superior performance of the PIS-BCNN-CP in accurately detecting both malicious and normal activities, indicating its potential to enhance security measures in blockchain systems.

In Fig. 9, FNR analysis is presented, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results demonstrate notable reductions in FNR achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP exhibits 26.95%, 25.64% and 20.65% lower FNR, while for normal attacks, it achieves 24.56%, 34.56% and 54.86% lower FNR. These outputs underscore the superior ability of the PIS-BCNN-CP to accurately detect both malicious and normal activities, indicating its potential to significantly improve security measures in blockchain systems.

In Fig. 10, the security analysis is presented, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN,

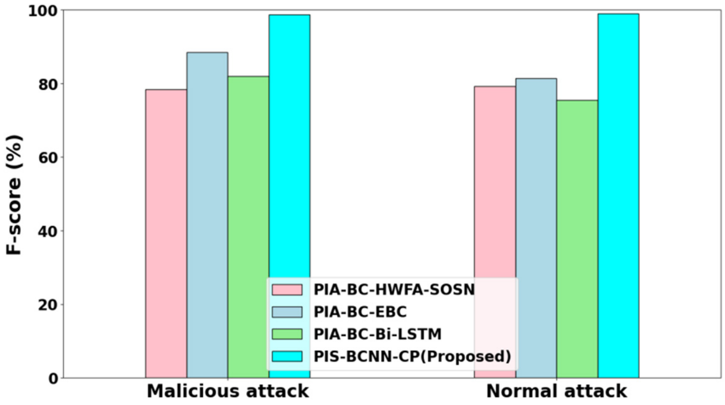

Fig. 8. F-Score analysis.

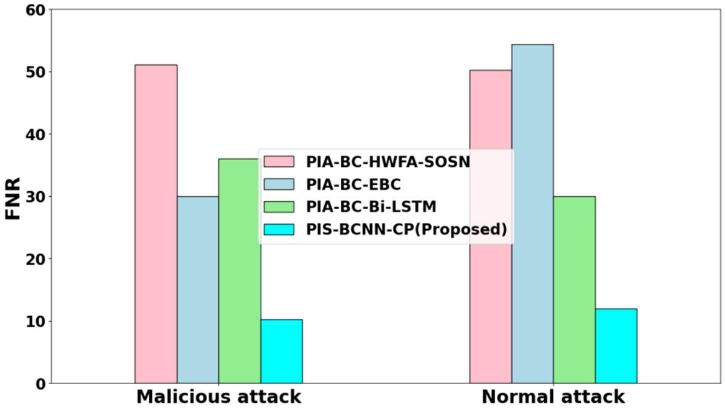

Fig. 9. False Negative Rate (FNR) analysis.

PIA-BC-EBC and PIA-BC-Bi-LSTM. The results demonstrate significant improvements in security achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP provides 36.06%, 29.65% and 26.75% higher security, while for normal attacks, it achieves 26.44%, 38.95% and 25.86% higher security. These outcomes underscore the enhanced security capabilities of the PIS-BCNN-CP in effectively detecting both malicious and normal activities, indicating its potential to bolster security measures in blockchain systems.

In Fig. 11, integrity analysis is presented, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results reveal substantial improvements in integrity achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP provides 36.94%, 29.05% and 44.86% higher integrity, while for normal attacks, it achieves 25.86%, 22.85% and 34.64% higher integrity compared to the same models. These findings underscore the enhanced integrity assurance capabilities of the PIS-BCNN-CP model in effectively detecting both malicious

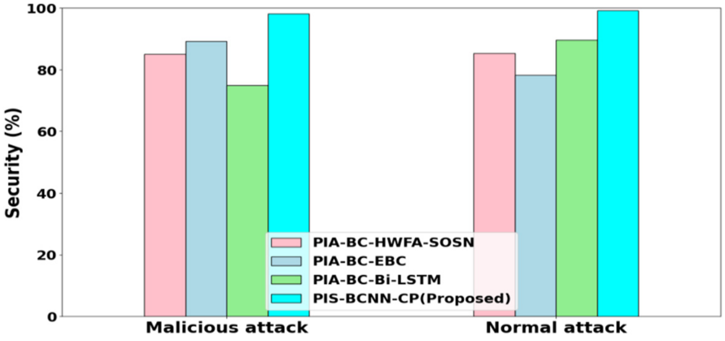

Fig. 10. Security analysis.

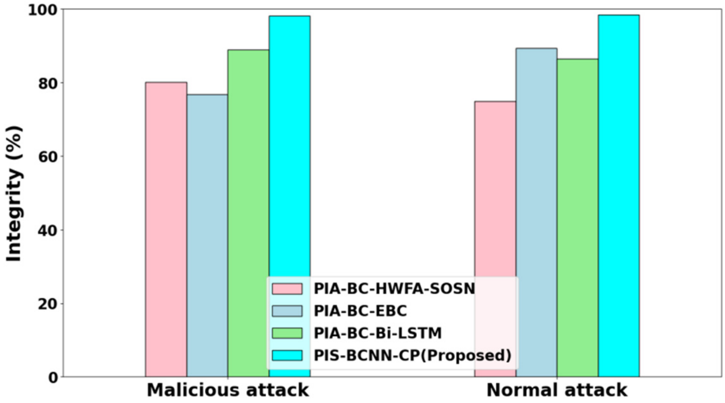

Fig. 11. Integrity analysis.

and normal activities, indicating its potential to strengthen integrity measures in blockchain systems.

In Fig. 12, loss function analysis is depicted, comparing the performance of the proposed PIS-BCNN-CP method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results demonstrate significant improvements in loss function achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP provides 25.23%, 29.47% and 22.86% lower loss function, while for normal attacks, it achieves 20.65%, 22.34% and 28.76% lower loss function. These outcomes underscore the superior performance of the PIS-BCNN-CP in minimizing loss during the training process, indicating its potential for more effective learning and optimization, and ultimately leading to improved detection of both malicious and normal activities in blockchain systems.

In Fig. 13, inference speed analysis is presented, comparing the performance of the proposed Prevention of Insider Attacks using Blockchain with the Hierarchical Auto-associative Polynomial Convolutional Neural Network in Cloud Platform (PIS-BCNN-CP) method with existing models: PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. The results demonstrate significant improvements in

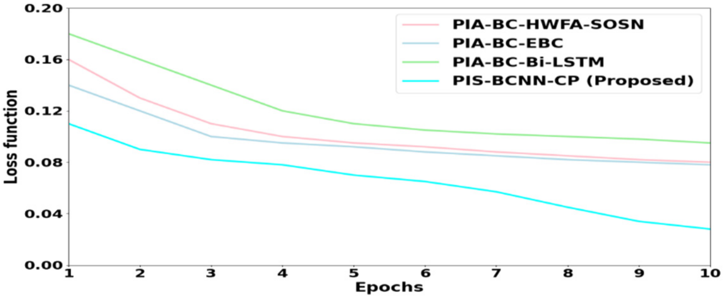

Fig. 12. Loss function analysis.

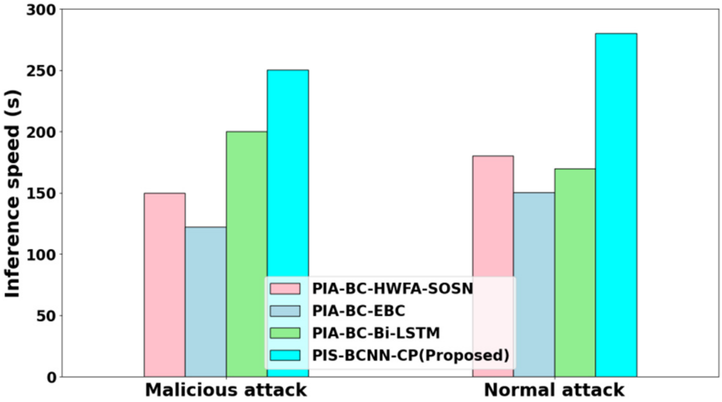

Fig. 13. Inference speed analysis.

inference speed achieved by the PIS-BCNN-CP approach. Specifically, for malicious attacks, PIS-BCNN-CP provides 22.34%, 21.57% and 23.98% higher inference speed, while for normal attacks, it achieves 25.76%, 24.45% and 26.86% higher inference speed. These outcomes underscore the superior efficiency of the PIS-BCNN-CP in processing inference tasks, indicating its potential for faster and more responsive detection of both malicious and normal activities in blockchain systems.

Figure 14 depicts the ROC curve for insider attack identification in cloud computing. The ROC of the PIS-BCNN-CP method demonstrates 21.34%, 31.85% and 32.09% higher Area Under the Curve (AUC) than existing PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM models, respectively. These findings highlight the superior discriminatory power of the PIS-BCNN-CP model in distinguishing between true positive and false positive rates, indicating its potential for more accurate and reliable insider attack detection in cloud computing environments.

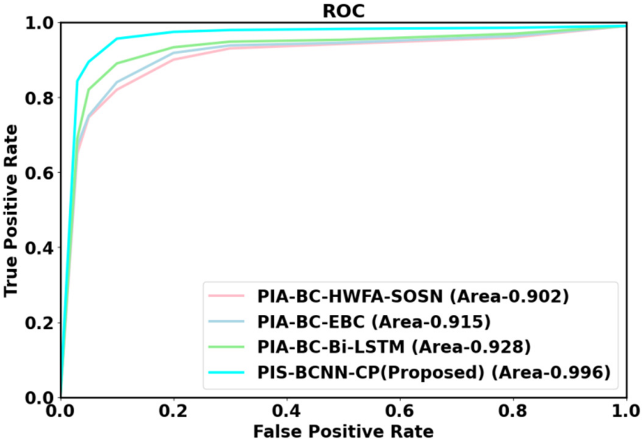

Fig. 14. ROC curve for insider attack detection in cloud computing environment.

Table 2. Ablation study of the proposed PIS-BCNN-CP method.

	PIS-BCNN-CP (with optimization)		PIS-BCNN-CP (without optimization)	
Methods (%)	Malicious	Normal	Malicious	Normal
Accuracy	28.5	26.4	12.2	10.2
Loss Function	23.1	29.4	10.1	14.7
Inference Speed	24.1	29.92	11.2	18.5
False Negative Rate	25.6	27.51	20.8	18.9

4.3. *Ablation study*

Table 2 represents the ablation study of PIS-BCNN-CP method, revealing the impact of optimization on its performance metrics. With optimization, the accuracy of detecting malicious instances significantly improves from 12.2% to 28.5% and for normal instances, it increases from 10.2% to 26.4%. Optimization leads to decreases in loss function values for both malicious (from 10.1% to 23.1%) and normal instances (from 14.7% to 29.4%). Additionally, inference speed increases with optimization, rising from 11.2% to 24.1% for malicious instances and from 18.5% to 29.92% for normal instances. Furthermore, optimization reduces the false negative rate, dropping from 20.8% to 25.6% for malicious instances and from 18.9% to 27.51% for normal instances. Overall, optimization positively impacts various aspects of the model's performance, including accuracy, loss function, inference speed, and false negative rate, highlighting its importance in enhancing the PIS-BCNN-CP method's effectiveness.

4.4. *Discussion*

The results presented in the figures and tables underscore the effectiveness of the PIS-BCNN-CP model in maximizing security in blockchain systems. The performance metrics analysis reveals significant improvements in accuracy, sensitivity, specificity, precision, F-Score, security, loss function, integrity, Inference speed and ROC curve compared to existing models such as PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM. Notably, the PIS-BCNN-CP model demonstrates superior performance across various aspects, including detection accuracy for both malicious and normal attacks, as well as reduced false negative rates and higher security and integrity assurance. Additionally, the ROC analysis indicates enhanced discriminatory power for insider attack detection in cloud computing environments. These findings highlight the potential of the PIS-BCNN-CP model to significantly enhance security measures in blockchain systems, providing a robust framework for detecting and preventing insider attacks while maintaining integrity and scalability. Moreover, the ablation study emphasizes the importance of optimizing the coefficient $e_{\rho 0}$to strike a balance between insider attack detection quality and the original task's performance, further enhancing the model's practical applicability.

5. Conclusion

Prevention of Insider Attacks using Blockchain with Hierarchical Auto-associative Polynomial Convolutional Neural Network in Cloud Platform (PIS-BCNN-CP) is successfully implemented in this paper. The proposed PIS-BCNN-CP method is activated in Ethereum and Solidity programming language (version 0.6.50).; its efficiency is evaluated under performance metrics. The performance of the PIS-BCNN-CP technique achieves 12.566%, 12.075% and 15.993% better accuracy, 15.86%, 15.26% and 16.25% better sensitivity analyzed to the existing PIA-BC-HWFA-SOSN, PIA-BC-EBC and PIA-BC-Bi-LSTM models, respectively.

References

1. W. Meng, W. Li, L. T. Yang and P. Li, Enhancing challenge-based collaborative intrusion detection networks against insider attacks using blockchain, *Int. J. Inf. Secur.* **1** (2020) 279–290.
2. G. Deep, J. Sidhu and R. Mohana, Access management of user and cyber-physical device in DBaaS according to Indian IT laws using blockchain, *Scalable Comput.: Pract. Exp.* **2** (2020) 407–424.
3. A. Ali, A. Khan, M. Ahmed and G. Jeon, BCALS: Blockchain-based secure log management system for cloud computing, *Trans. Emerg. Telecommun. Technol.* **33** (2022) e4272.
4. A. Ali, M. Ahmed, A. Khan, A. Anjum, M. Ilyas and M. Helfert, VisTAS: blockchain-based visible and trusted remote authentication system, *PeerJ. Comput. Sci.* **7** (2021) e516.
5. L. He, A. J. Valocchi and C. A. Duarte, A transient global-local generalized FEM for parabolic and hyperbolic PDEs with multi-space/time scales, *J. Comput. Phys.* **488** (2023) 112179.
6. A. Ali, M. A. Almaiah, F. Hajjej, M. F. Pasha, O. H. Fang, R. Khan, J. Teo and M. Zakarya, An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network, *Sensors* **22** (2022) 572.
7. R. Sahay, G. Geethakumari and B. Mitra, A novel blockchain based framework to secure IoT-LLNs against routing attacks, *Computing* **102** (2020) 2445–2470.
8. Q. Zheng, P. Zhao, D. Zhang and H. Wang, MR-DCAE: Manifold regularization-based deep convolutional autoencoder for unauthorized broadcasting identification, *Int. J. Intell. Syst.* **36** (2021) 7204–7238.
9. L. Lin, H. Yang, J. Zhan and X. Lv, VNGuarder: An internal threat detection approach for virtual network in cloud computing environment, *Secur. Commun. Netw.* **2022** (2022).
10. L. He, A. J. Valocchi and C. A. Duarte, An adaptive global–local generalized FEM for multiscale advection–diffusion problems, *Comput. Methods Appl. Mech. Eng.* **418** (2024) 116548.
11. J. Gong and N. J. Navimipour, An in-depth and systematic literature review on the blockchain-based approaches for cloud computing, *Cluster Comput.* **25** (2022) 383–400.
12. W. Li, Y. Wang, W. Meng, J. Li and C. Su, BlockCSDN: Towards blockchain-based collaborative intrusion detection in software defined networking, *IEICE Trans. Inform. Syst.* **105** (2022) 272–279.
13. A. Mitra, B. Bera, A. K. Das, S. S. Jamal and I. You, Impact on blockchain-based AI/ML-enabled big data analytics for Cognitive Internet of Things environment, *Comput. Commun.* **197** (2023) 173–185.

14. Q. Zheng, P. Zhao, H. Wang, A. Elhanashi and S. Saponara, Fine-grained modulation classification using multi-scale radio transformer with dual-channel representation, *IEEE Commun. Lett.* **26** (2022) 1298–1302.
15. M. N. Al-Mhiqani, R. Ahmad, Z. ZainalAbidin, W. Yassin, A. Hassan, K. H. Abdulkareem, N. S. Ali and Z. Yunos, A review of insider threat detection: Classification, machine learning techniques, datasets, open challenges and recommendations, *Appl. Sci.* **10** (2020) 5208.
16. Q. Zheng, X. Tian, Z. Yu, H. Wang, A. Elhanashi and S. Saponara, DL-PR: Generalized automatic modulation classification method based on deep learning with priori regularization, *Eng. Appl. Artif. Intell.* **122** (2023) 106082.
17. J. A. Alzubi, O. A. Alzubi, A. Singh and M. Ramachandran, Cloud-IIoT-based electronic health record privacy-preserving by CNN and blockchain-enabled federated learning, *IEEE Trans. Indust. Inform.* **19** (2022) 1080–1087.
18. S. Gurusamy and R. Selvaraj, Resource allocation with efficient task scheduling in cloud computing using hierarchical auto-associative polynomial convolutional neural network, *Exp. Syst. Appl.* **249** (2024) 123554.
19. P. S. Rani and S. B. Priya, A block chain-based approach using proof of continuous work consensus algorithm to secure the educational records, *Peer-to-Peer Netw. Appl.* **16** (2023) 2456–2473.
20. M. Sindhuja, S. Vidhya, B. S. Jayasri and F. H. Shajin, Multi-objective cluster head using self-attention based progressive generative adversarial network for secured data aggregation, *Ad Hoc Netw.* **140** (2023) 103037.
21. M. Vollmer, Infrared thermal imaging, *Computer Vision: A Reference Guide* (Springer International Publishing, Cham, 2020), pp. 1–4.
22. Z. Teng, B. Pang, C. Du and Z. Li, Malicious node identification strategy with environmental parameters, *IEEE Access* **8** (2020) 149522–149530.
23. P. K. Martell, Hierarchical Auto-Associative Polynomial Convolutional Neural Networks, Doctoral Dissertation, University of Dayton (2017).
24. A. Ghasemi-Marzbali, A novel nature-inspired meta-heuristic algorithm for optimization: Bear smell search algorithm, *Soft Comput.* **24** (2020) 13003–13035.
25. H. Yin, Z. Zhang, J. He, L. Ma, L. Zhu, M. Li and B. Khoussainov, Proof of continuous work for reliable data storage overpermissionless blockchain, *IEEE Int. Things J.* **9** (2021) 7866–7875.
26. D. N. Kirupanithi, D. A. Antonidoss and G. Subathra, Detection of insider attacks in block chain network using the trusted two way intrusion detection system, arXiv:2211.03138.
27. Y. M. Tukur, D. Thakker and I. U. Awan, Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things, *Trans. Emerg. Telecommun. Technol.* **32** (2021) e4158.
28. O. Alkadi, N. Moustafa, B. Turnbull and K. K. R. Choo, A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks, *IEEE Int. Things J.* **8** (2020) 9463–9472.
29. G. Deep, J. Sidhu and R. Mohana, Insider threat prevention in distributed database as a service cloud environment, *Comput. Indust. Eng.* **169** (2022) 108278.
30. T. Hu, B. Xin, X. Liu, T. Chen, K. Ding and X. Zhang, Tracking the insider attacker: A blockchain traceability system for insider threats, *Sensors* **20** (2020) 5297.
31. R. Awadallah and A. Samsudin, Using blockchain in cloud computing to enhance relational database security, *IEEE Access* **9** (2021) 137353–137366.
32. R. G. Gayathri, A. Sajjanhar and Y. Xiang, Hybrid deep learning model using SPCA-GAN augmentation for insider threat analysis, arXiv:2203.02855.