Section: Research Paper
ISSN 2063-5346

GESTALT PATTERN MATCHED EXTREME LEARNT CRYPTOGRAPHIC BLOCKCHAIN FOR SECURE DATA COMMUNICATION IN CLOUD

[1]**P.M. Pazhani Selvam,** Reg. No: 18123152161002, Research Scholar, Computer Application, S.T.Hindu College Research Centre (Affliated with Manonmaniam Sundaranar University, Tirunelveli) , Nagercoil-629001

[2]**Dr. S.S. Sujatha,** Associate Professor, Dept. of M.C.A., S.T. Hindu College, Nagercoil-629001

[3]**Dr. K.K. Thanammal,** Assistant Professor, Dept. of Computer Science ., S.T. Hindu College, Nagercoil-629001

Email: [1]pmpselvam69@gmail.com, [2]sujaajai@gmail.com, [3]thanaravindran@gmail.com

DOI:10.48047/ecb/2022.11.11.66

ABSTRACT

Cloud computing is a network-based model where the data are accessed by the user anywhere and anytime. Cloud computing is a technology that provides resources and utility services according to user demand. Due to this demand, efficient cloud security methods are highly required, especially at the time of data communication for user authentication. • security is an important task to protect data and cloud resources from harmful activities. Therefore, a novel intrusion detection system is required in cloud networks to detect malicious activity and improve secure data transmission. In this paper, a novel technique called extreme learning based Gestalt Pattern Matched Extreme Learned Cryptographic Blockchain (GPMELCB) model is developed. The GPMELCB model includes three major processes namely user registration, block generation and validation, and secure data communication. First, the Cloud user registration process is performed and generates the user identity and the password. Then the user generates the block with the help of the Davis Mayer cryptographic compression function to generate the hash value of the data and stored it in Blockchain central server. The server performs block validation using Nelder–Mead Byzantine Fault Tolerance consensus algorithm. When the user wants to access the data from the server, the first verify their authenticity using a Gestalt pattern-matched Extreme Learning Classifier. The extreme learning classifier is used for identifying the authorized or unauthorized user (i.e. attacks). The server permits the authorized user to obtain the data and avoids the unauthorized user (i.e. attacks). In this way, secure data communication is carried out with higher data confidentiality and integrity. Experimental evaluation of the proposed GPMELCB model is carried out with respect to transaction latency, communication cost, data confidentiality, and data integrity with a different number of data and cloud users. The quantitatively analyzed results indicate that the

performance of the GPMELCB model increases data communication security with a higher confidentiality rate, integrity, and minimum latency as well as cost than the conventional methods.

Keywords: Cloud computing, secure data communication, Gestalt pattern matched Extreme Learning Classifier, blockchain , Nelder–Mead Byzantine Fault Tolerance consensus algorithm, Davis Mayer cryptographic compression function

1. INTRODUCTION

Cloud computing allows the users to distribute the configurable computing resources over a network quickly. Cloud computing is a network-based model wherein data are accessible to the user. With the fast development in cloud computing, more users store and share their applications and data on the cloud. The network security has been a key concern in cloud based environment to protect the data from malicious intruders.

A Homomorphic Block Ring Security System (HBRSS) was developed in [1] to guarantee the security during the process data transmission. Though the system minimizes the CPU utilization and execution time, the higher data confidentiality was not minimized. Bayesian game and blockchain-based auction model was developed in [2] for federated cloud services to solve he security concern. However, the designed model failed to reduce transaction costs and improve the contract security with possible attacks. A blockchain-based provenance system was developed in [3] for secure data-sharing ecosystem using smart contract. The designed system improves the data confidentiality, integrity, but the transaction latency was not minimized.

An Artificial Neural Networks (ANN) and encryption were introduced in [4] for secure cloud communication system using Fully Homomorphic Encryption. However, the security of the system was not improved at required level. A cloud-backed storage system was introduced in [5] for storing and

distribution of big data with higher secure, reliable, and efficient manner. But the system failed to store and distributes big critical datasets in an efficient way.

A Linear Elliptical Curve Digital Signature (LECDS) with Hyperledger blockchai was developed in [6] to maintain the privacy the sensitive data for authorized users. But it failed to implement the blockchain platform to guarantee security and performance metrics of Hyperledger.

A secure client framework was introduced in [7] for efficient and secure cloud environment along with the security constraint. But the higher data confidentiality was no archived. A blockchain data transfer method was designed in [8] based on homomorphic encryption to guarantee the security of data transmission. But the integrity of the data transmission was not enhanced. A distributed transmission method was developed in [9] with security to avoid the double attacks. But the time consumption of attack detection remained unaddressed. Hybridization of recurrent convolutional neural network and encryption method was developed in [10] for intrusion detection with secure data communication. However, the cryptographic hash-based algorithm was not applied to improve the data integrity for data communication.

1.1Contributions of the paper

In order to overcome the existing issues, a novel GPMELCB model is introduced with the following novel contributions.

- To improve the security of data transmission in the cloud, a novel GPMELCB model is introduced based on the user registration, block generation and validation, secure data transmission
- To minimize the transaction latency, GPMELCB model uses the Davis Mayer cryptographic compression function for block generation with the user data. After that, the Nelder–Mead BFT consensus algorithm is applied to validate the block. Based on , only validated block is added to chain of distributed network.

- To increase data integrity, Davis Mayer cryptographic compression function generates the hash value for each user data and stored in the blockchain central server to avoid the data alteration by the attackers.
- To increase data confidentiality, Gestalt pattern matched extreme learning classifier is introduced in GPMELCB model for identifying the authorized or unauthorized. The server permits to access the data for authorized user and denied the access for unauthorized user (i.e. attacks)
- Finally, comprehensive experiment evaluations are carried out to estimate the performance of the GPMELCB model and other cryptographic techniques along with the various metrics.

1.1. Organization of paper

The rest of the paper is organized into different sections as follows. Section 2 reviews the related works of security in cloud. Section 3 provides a brief description of the proposed GPMELCB model with a neat architecture diagram. Section 4 describes the experimentation with the dataset description. In section 5, the performance results of the proposed GPMELCB model and existing methods are analyzed with different metrics. At last, Section 6 concludes the paper.

2. RELATED WORKS

Blockchain -based user authentication method was developed in [11] by using improvised public-key cryptosystem to enhance the security. But the designed method failed to identify the attacked and non-attacked data. An integration of cloud computing with blockchain was introduced in [12] to improve the data integrity for all homomorphic encryption schemes. But the designed scheme failed to provide information about which data records have been attacked.

A ciphertext-policy attribute-based method was developed in [13] with keyword search to improve the data sharing for encrypted cloud data. However,

the latency aware data sharing was not performed. A novel provable data possession method was developed in [14] for efficient privacy preserving cloud storage. But the method was not efficient to attain improved data confidentiality. A secure authentication protocol was designed in [15] for a cloud-assisted system with access control using blockchain. But the data integrity was major concern.

A new security-by-design framework was developed in [16] for cloud computing environment. But an efficient blockchain cryptographic technique was not applied to further enhance the security of data transmission.Online Secure Communication Model for Cloud (OSC-MC) was introduced in [17] by identifying and terminating malicious threads and improving the security.A new two-factor secure authentication method was developed in [18] for resisting the different types of attacks. A Non-Deterministic Cryptographic Scheme (NCS) was developed in [19] to improve the security strength with minimum execution time. A new blockchain-assisted framework was developed in [20] for effective data distribution and retrieval using cloud platforms. However, the data privacy percentage and the performance of data sharing and retrieval of the system were not improved.

3. PROPOSAL METHODOLOGY

Cloud computing is a most prominent storage and computing technology to provide the pool of computing resources. Cloud implements all security problems of its parts due to its complex framework. During the large amounts of complex structured data transactions, security is a significant task to preserve the data from harmful activities. Therefore, the secure communication in a cloud system is affected by different kinds of attacks. For data communication and processing, a novel security technique is essential for protecting data from various kinds of attacks. To overcome the security threat, a novel technique called GPMELCB model is introduced.

Section: Research Paper
ISSN 2063-5346

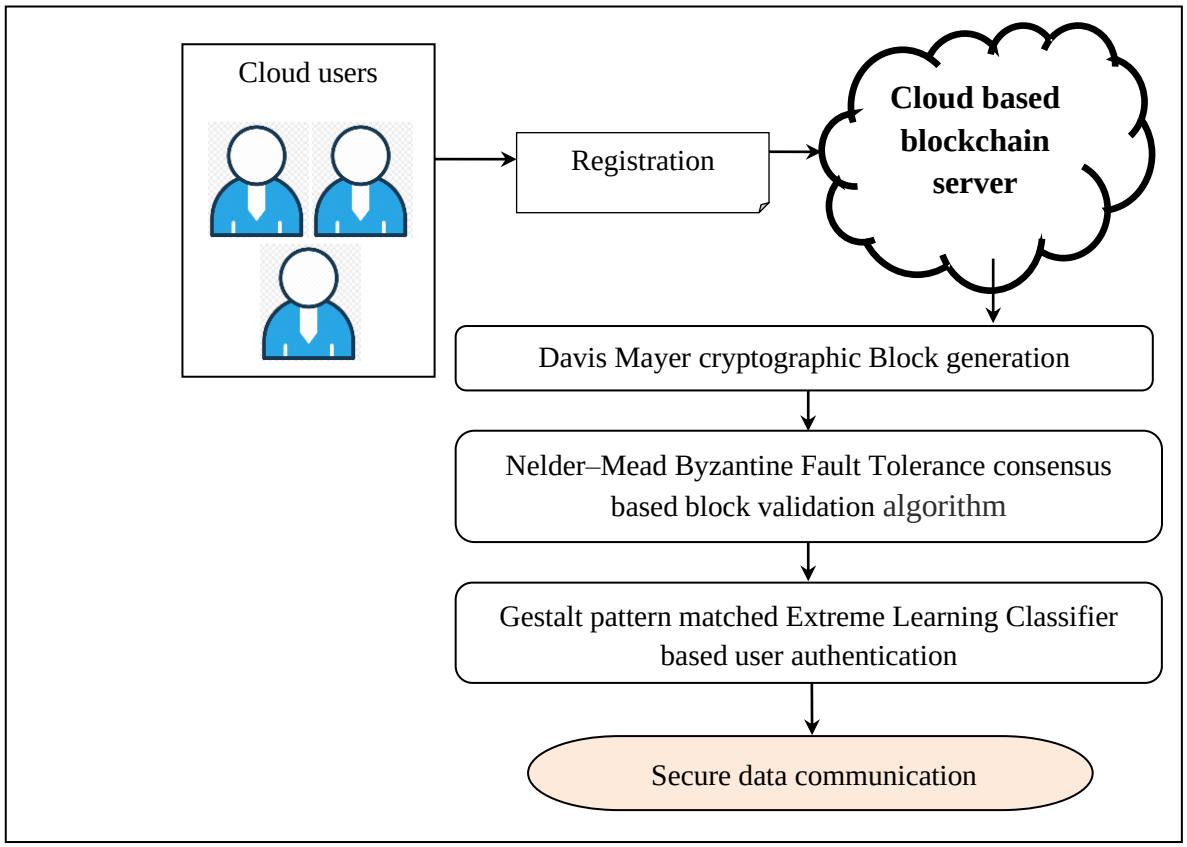

Figure 1 Architecture of proposed GPMELCB model

Figure 1 illustrates the architecture diagram of the proposed GPMELCB model to provide efficient secure data transmission in the cloud. The cloud computing architecture includes two entities such as users or client and cloud server. The cloud users '$U = \{Cu_1, Cu_2, \dots, u_m\}$ who dynamically generates the large volume of data $DP_1, DP_2, \dots, DP_q$.

In the cloud computing environment, blockchain is regarded as a service called Blockchain as a Service (BaaS) and to provide a decentralized storage architecture which is robustly opposed to data modifications. The blockchain cloud architecture uses decentralized architecture and store the data with different types cryptographic technique in particular databases. When cloud providers combined into blockchain technology, increases the data integrity. In addition, blockchain on the cloud offers faster data organization since, with cloud blockchain architecture, there are several copies of the same data stored on numerous computer nodes. Finally, the communication between the users in

Section: Research Paper
ISSN 2063-5346

cloud needs to be secure for protecting the sensitive data from the unauthorized entity by using Extreme learning machine.

The proposed GPMELCB model includes three major processes namely cloud data owner registration, block generation and validation, and secure communication between the user and cloud server. Initially, the cloud data owner registration process is carried out in cloud computing environment. Next, for each registered cloud data owners, the Block Generation and Block Validation model is designed. Finally, a secure communication process is designed by means of Gestalt pattern matched extreme learning classifier. The different process of GPMELCB model is described in the following subsections.

3.1 Cloud user registration

Blockchain is defined as a decentralized, distributed and cryptographic technique that allows to store and access the data. One of the most significant features of Blockchain is to avoid any changes or alters the data once they have been added to the Blockchain resulting it improved the data integrity.

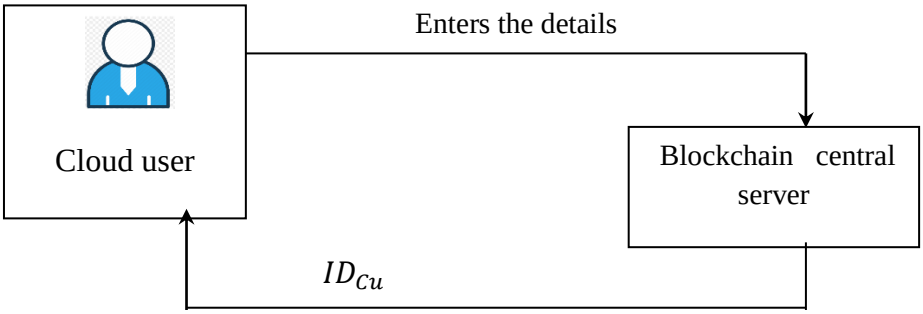

Figure 2 User registration

Figure 2 illustrates a user registration based on blockchain. User stores the details into the blockchain server for accessing the different services such as uploading or downloading the data. After the registration, every cloud data user has to register their details itself on the blockchain. All the users who want to upload or download data to register identity information in the blockchain. The blockchain server generates the unique identity information for different

identities of users. The blockchain server stores the user's registration details in their database.

$$Q \rightarrow \{ID_{Cu}, PS_{Cu}\} \quad (1)$$

Where, Q denotes an identity generator, ID_{Cu} represents the unique identity of cloud user, PS_{Cu} denotes a password. Each operation of the user is stored in the blockchain as a transaction. The proposed blockchain plays an important role for providing the confidentiality, authentication of an identity, integrity, and storage of user operation records. It also reduces the malicious operation of users.

Algorithm 1: User registration and identity generation
Input: Dataset 'DS', cloud data user '$Cu = Cu_1, Cu_2, \dots, Cu_n$',Data '$DP = DP_1, DP_2, \dots, DP_q$', blockchain central Server 'CS'
Output: User registration
Begin **Step 1:** **For** each cloud user 'Cu' **Step 2**: Send their information to blockchain central Server 'CS' **Step 3:** Central Server 'CS' sends the unique identity and password to user **Step 4**: **end for** **Step 5:** **Return** registered cloud data owners **End**

Algorithm 1 given above illustrates the step by step process of user registration and identity generation to improve the secure transaction in cloud based blockchain technology. For each cloud data user who participates in the secure communication process, registration is first performed. Then the clod server generates the unique identity and password to each registered user.

3.2 Block generation and validation

After the registration, the registered user information is stored into the block and tnitiate the transaction in the block format. First, block is generated for each registered user to construct the chain.

Section: Research Paper
ISSN 2063-5346

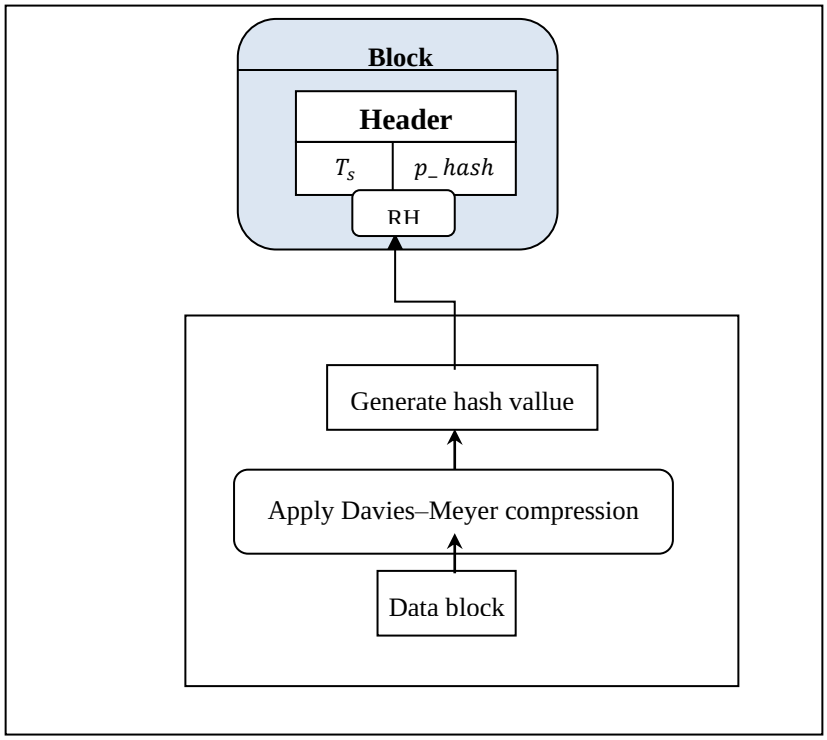

Figure 3 Construction of Davis Mayer cryptographic Block generation

Figure 3 demonstrates the generation of block for each registered user. Each block has a block header, timestamp (T_s), root hash (RH), and a hash of the previous block ($p\_ hash$). Each transaction comprises the data and each block has a block header. A time step refers to the time when the block was created. The root hash (RH) value is generated using Davis Mayer cryptographic compression function to enhance the data integrity. As shown in the block generation, the data block has a user data.

Then the input data block is given to the Davies–Meyer one way compression function. The compression function receives the input data and it divided into number of chunks (b_i)

$$b_i = b_1, b_2, b_3, \ldots . b_k \tag{2}$$

Section: Research Paper
ISSN 2063-5346

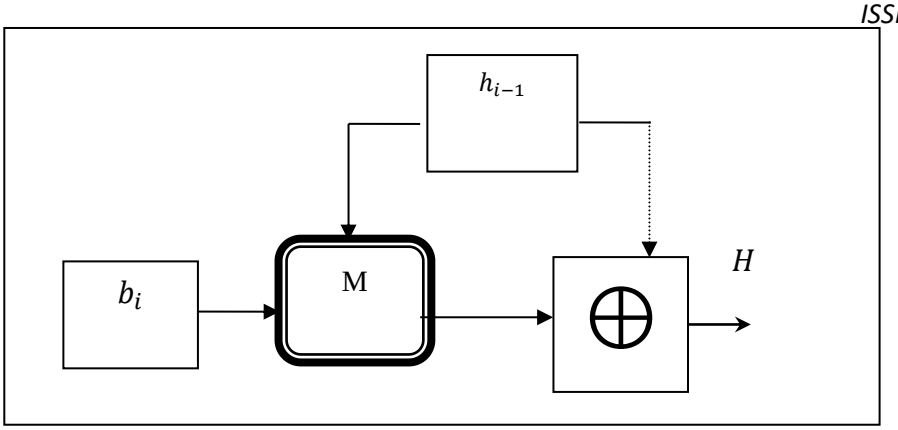

Figure 4 One way compression function

Figure 4 illustrates the block diagram of Davis Mayer compression function that obtains the input data chunk 'b_i' and the previous hash value (h_{i-1}) is initially preset to '0'. From figure 4, 'M' represent a chunk cipher and the data chunk (b_i) as the key to a chunk cipher and XORed with the previous hash value. The final hash is generated by the below mathematical function,

$$H = [M_{b_i}(h_{i-1}) \oplus h_{i-1}] \quad (3)$$

Where, H denotes a final hash value. Similarly, the other data block output is generated. The hash of one chunk $H(b_1)$ is not similar to another chunk $H(b_2)$.

$$H(b_1) \neq H(B_2) \quad (4)$$

Finally, the output is taken from the final compression function. The final output hash is generated. This helps to avoid the data that are not altered by the unauthorized user resulting it increases the data integrity rate.

3.2.1 Block validation

Once the blocks are generated, the validation is performed to construct the chain. In other words, the validated block is added to the chain. The validation is significant process before the data transaction. This ensures that only valid generated blocks are broadcasted on the network. The independent validation also ensures that users who access the data legally. The proposed technique uses the Nelder–Mead Byzantine Fault Tolerance consensus algorithm to perform the block validation. In blockchain construction, the consensus algorithm is a most

significant method that directs the trust among the blocks. However, the conventional algorithms suffer from low throughput, high delay, and vulnerability to different types of attacks. Therefore, the Nelder–Mead BFT consensus algorithm is a developed.

Nelder–Mead BFT consensus algorithm introduces a dynamic weighting mechanism for consensus nodes, which guarantee the security of blockchain system by identifying the influence of malicious nodes. The Nelder–Mead BFT consensus algorithm used for validating whether a node or block is true or not. In the blockchain, central server manages all blocks in the chain.

In the request phase, the user sends its request message to the central server. Let us consider the user 'Cu' initiate the block 'B' sends its request message 'R_q' to the blockchain central server 'CS'.

$$Cu \overset{R_q}{\rightarrow} CS \tag{5}$$

Central server distributes the request message to other blocks 'B_i' in the distributed network.

$$CS \overset{R_q}{\rightarrow} \sum_{i=1}^{n} B_i \tag{6}$$

For each block, sends the response the feedback information about the new block to be validated in the consensus process. The feedback information of a block includes the behavior information and the evidence.

Server maintains an initial weight and a processing weight for each block in the blockchain system. The processing weight of a block represents the degree of trust and the level of activity of the block sent through the feedback information in the consensus method. The central server decides whether a block participates in the consensus process or not along with the threshold. It is computed as given below,

$$T\ (B_i) = \frac{\vartheta_{pro}}{\vartheta_{Ini}} \tag{7}$$

Section: Research Paper
ISSN 2063-5346

Where, $T(B_i)$ threshold value for the block 'B_i', ϑ_{pro} denotes a processing weight, ϑ_{Ini} represents the initial weight. The processing weight value of each block is dynamically altered. When a block is more active or truthfully conducts the consensus activities, its weight value gets increased. If the block activity reduces when the maliciously interferes with the consensus process and hence its processing weight gets reduced.

Then the cloud server verifies that the obtained threshold value either maximum or minimum by applying Nelder mead method. It is a numerical technique used to determine the minimum or maximum of an objective function (i.e. weight) in a multidimensional space by sorting the threshold value.

$$T_1(B_i) \leq T_2(B_i) \leq T_3(B_i) \leq \cdots \ T_m(B_i) \tag{8}$$

After sorting, the server finds the threshold value higher than the other value is selected and it added into the blockchain for consensus. Otherwise, the block with lesser threshold is said to be a faulty nodes and malicious nodes and it not included into the blockchain.

$$F = \begin{cases} if \arg max \ T(B_i) \ ; block \ is \ valid \\ otherwise \qquad ; block \ is \ not \ valid \end{cases} \tag{9}$$

Where, F denotes an output of the Nelder mead method, $\arg max$ denotes an argument of maximum function, $T(B_i)$ denotes a threshold value of the block. If the threshold value of the block is higher, then the block is valid and it added to blockchain. Otherwise, block is not valid and it did not added o the blockchain.

Section: Research Paper
ISSN 2063-5346

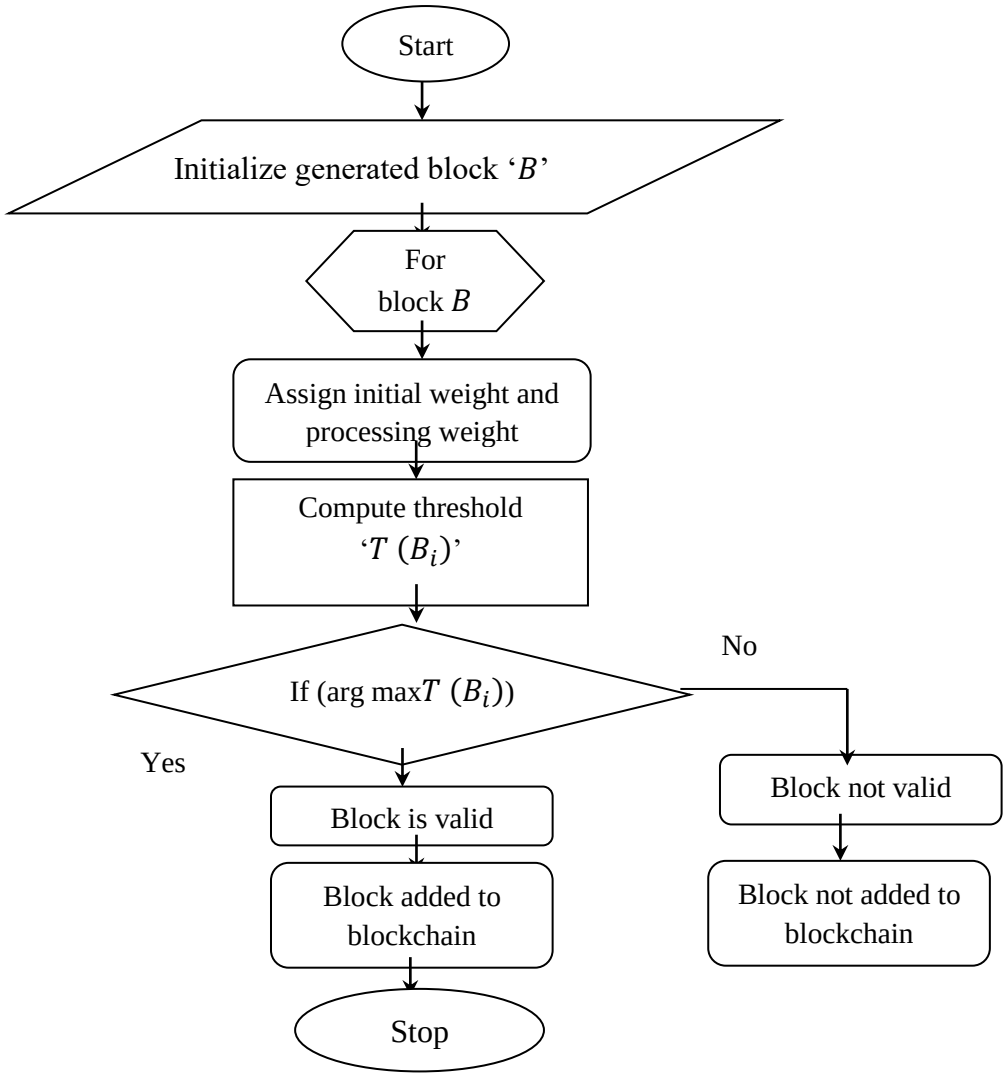

Figure 5 Flow chart of the Nelder–Mead BFT consensus algorithm based block validation

Figure 5 illustrates the flow process of the Nelder–Mead BFT consensus algorithm based block validation to improve the security of data transmission. The valid block is added into the chain of the distributed network as show in figure 6.

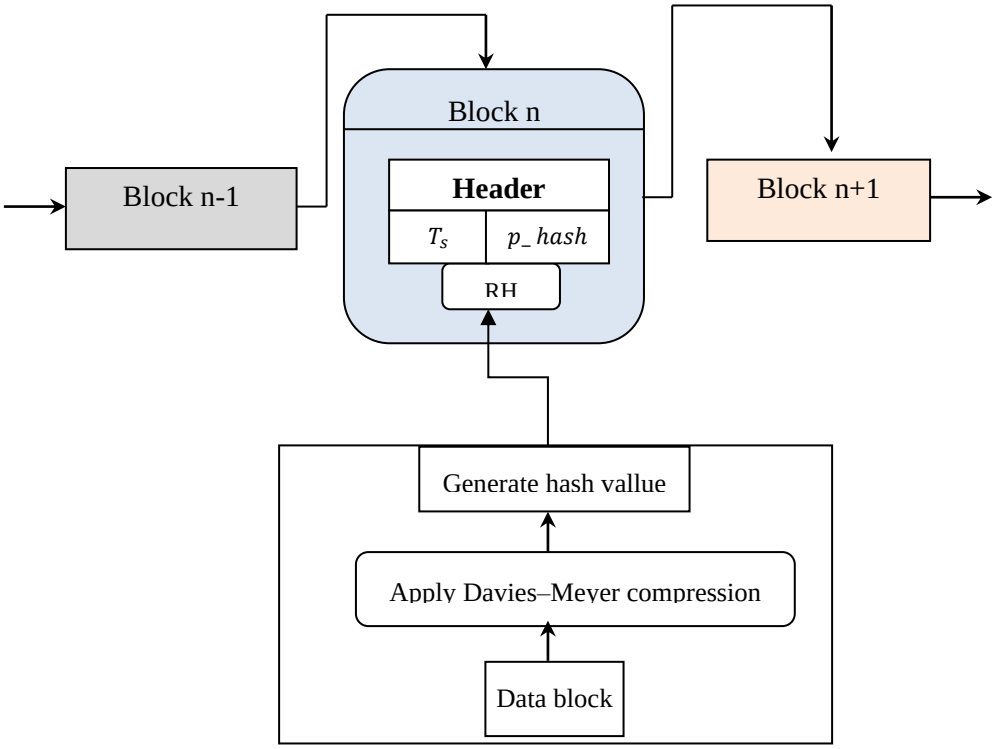

Figure 6 illustrates the construction of blockchain based on the Nelder–Mead BFT consensus algorithm.

The valid block added into the existing chain to perform secure data transaction. Otherwise, the invalid block does not included into the distributed blockchain.

Algorithm 2 given above illustrates step by step process of transaction block generation and validation. For each cloud data in the secure communication process, generate hash value and construct the block to ensure integrity. After that, the block is validated using Nelder–Mead BFT consensus algorithm. With this, the block is validated based on the weight value. The valid block is added into blockchain and other invalid block is not added to the chain in the distributed network. In this way, blockchain and validation process is said to be performed.

Algorithm: 2 Block generation and validation

Input: Dataset 'DS', cloud data user '$Cu = Cu_1, Cu_2, \dots, Cu_n$',Data '$DP = DP_1, DP_2, \dots, DP_q$', blockchain central Server 'CS'

Output: Obtain validated block

Begin
//Transaction block generation
Step 1: **For** each registered cloud user
Step 2 Generate data in the form of block
Step 3: **For each transaction 't'**
Step 4: Divide into data 'DP ' into message chunks $b_1, b_2, b_3, \dots b_k$ using (2)
Step 5: **for each** chunk b
Step 6: Generate root hash value 'H' using (3)
Step 7: **End for**
Step 8: Obtain the final hash
Step 9: **End for**
Step 10: **End for**
Step 11: Return generated blocks
//Transaction block validation
Step 12: for each generated blocks
Step 13: 'Cu' initiate the block 'B' sends its request 'R_q' to the central server 'CS' using (5)
Step 14: CS distributes the request to other blocks 'B_i' using (6)
Step 15: **For each** block in network
Step 16: Sends the feedback information **to** CS
Step 17: CS assigns initial weight and processing weight
Step 18: CS compute threshold using (7)
Step 19: Sorting the threshold value using (8)
Step 20: $\boldsymbol{if}$ (arg max$T$ (B_i)) **then**
Step 21: Block is valid
Step 22: Added to blockhain
Step 23: **else**
Step 24: Block is not valid
Step 25: Not added to blockhain
Step 26: End if
Step 27: End for
Step 28: End for
End

3.3 Gestalt pattern matched Extreme Learning Classifier based secure data transmission

Finally, secure communication between cloud data owners and cloud users are performed through the authentication process with the help of Gestalt pattern matched Extreme Learning Classifier. It defines that Extreme learning classifier

Section: Research Paper
ISSN 2063-5346

identifies the authorized cloud users and unauthorized cloud users. The blockchain central server allows only the authorized cloud users access or view the data and denied the access for unauthorized cloud users. Therefore, the proposed technique uses the Extreme learning classifier for secure communication model to attain higher data confidently rate.

Contrary to the conventional deep learning algorithm, the extreme learning classifier is a feed-forward neural network that provides a straightforward solution that did not perform any iterative process. Extreme learning classifier is also very fast to perform data analysis and provide a linear output. Therefore, the proposed technique uses the extreme learning classifier for accurate user authentication with minimum time.

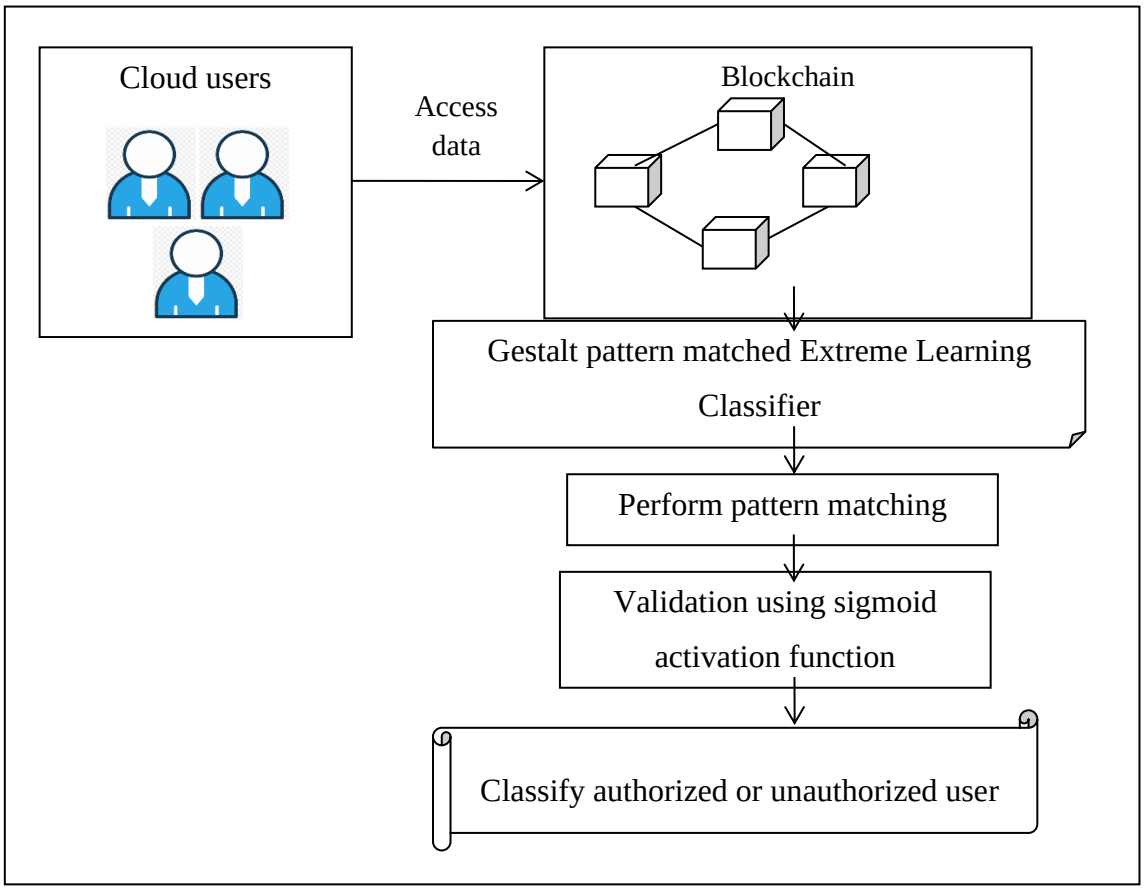

Figure 7 Flow process of Gestalt pattern matched Extreme Learning Classifier based user authentication

Section: Research Paper
ISSN 2063-5346

Figure 7 illustrates the flow process of Gestalt pattern matched Extreme Learning Classifier based user authentication for secure data communication. User request an access the data stored in a blockchain central server. Then blockchain server uses the Extreme Learning Classifier to identify the authorized user with the help of the smart contract for access control in the decentralized network. A smart contract is a self-executing contract that exists between two parties based on the terms of the agreement or certain rules that are encoded as a set of lines of code stored in the blockchain. The smart-contracts make the interaction between the users through the communications called transactions. These contracts stored in a blockchain automatically perform legally relevant events according to the terms of certain rules without believing external third parties. The rule defines only authorized user access the data from the server.

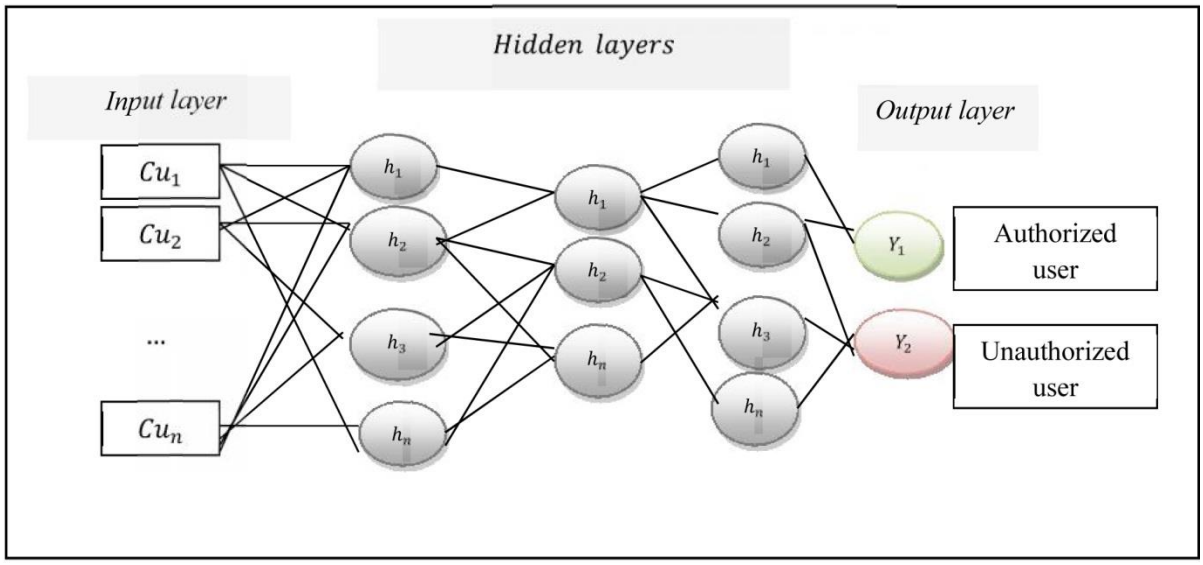

Figure 8 Structures of multiple hidden layered extreme learning classifiers

Figure 8 illustrates the structure of extreme learning machines a type of feed-forward neural network for user authentication with a multiple layers of hidden nodes 'h'. The extreme learning machines include single input layer and output layer, multiple hidden layers. In contrast with the conventional deep learning method, extreme learning machines has two features such as input weights, biases that are randomly initialized and another one is output layer

weights solved as the least squares problem. As revealed in the figure 8, let us consider that extreme learning machines includes a training set as $\{Cu, Y\}$ where Cu denotes a cloud users and a label or output 'Y' representing the two classes Y_1 and Y_2 respectively.

First, the proposed classifier receives 'n' training data as input ($Cu_i = Cu_1, Cu_2, \dots, Cu_n$), and classifier randomly set a weight ' $q_1, q_2, \dots, q_m$' associated with the input layer and added bias 'V' in the hidden layer that stored the value is '1'. The input layer receives the input only but it did not execute any computations, whereas the output layer is linear with no transformation function and no bias. The generated input weights are fixed and did not alters up to entire process.

$$X(t) = \sum_{i=1}^{n} [Cu_i * q_{ij}] + V \tag{10}$$

Where, the activity of neurons at the input layer '$X(t)$' denotes that the weighted input Cu_i, weight and bias, 'α_{ij}' denotes a weight between the j^{th} input layer neuron and the i^{th} hidden layer neuron, V denotes a bias.

The input is transferred into the first hidden layer where the certain computation process is performed. In that layer, user sent request to access the data stored in a blockchain central server. The user generates the hash value for their data 'H_s'.

The generated hash value is transferred into the next hidden layer where the blockchain central server uses the Gestalt pattern matching coefficient to perform the hash verification. The Gestalt pattern matching is a statistical method used to measure the relationship between the two hash values.

$$MC_P = \ 2 * \left[\frac{M_{Patterns}}{N_{Patterns}}\right] \tag{11}$$

Where, MC_P denotes a Gestalt pattern matching coefficient, $M_{Patterns}$ denotes a number of matched patterns between newly generated hash and already

stored hash at the time of block generation , $N_{Patterns}$ denotes a total number of patterns in already stored hash. The coefficient returns a value from 0 to 1. The output of the pattern matching coefficient is given to next hidden layer where sigmoid activation function is applied for user authentication.

$$\omega = \left[\frac{1}{1+\exp(-MC_P)}\right] \tag{12}$$

$$\omega = \begin{cases} 1, & Pattern\ matched \\ 0, & Pattern\ notmatched \end{cases} \tag{13}$$

The activation function 'ω' returns the output '1' indicates that the two hash values are matched (i.e. pattern matched), '0' denotes two hash values are not matched (i.e. pattern not matched). Based on activation function results, user is classed into authorized or unauthorized. If the two patterns are matched correctly, then the user is classified into the authorized user. If the patterns are not matched, then the user is classified into the unauthorized user. Therefore, the output of the hidden layer is given below.

$$Z = \sum_{i=1}^{L} q_{ij}\ \omega\ (q_{jk}\ a_o + V) \tag{14}$$

Where, Z represents the output of hidden layer, ω indicates an activation function, 'q_{jk}' denotes the j^{th} hidden layer neuron and k^{th} output layer neuron, q_{ij} denotes a weight between input and hidden layer, a_o denotes an output of previous hidden layer, L denotes a number of hidden units. Finally the accurate classification results are obtained at output layer of the extreme learning machine**.** The blockchain server grants the access to authorized user and avoids the unauthorized users (attacks). This helps to improve the confidentiality rate.

Algorithm 3 given above illustrates the algorithmic process of authorized user identification using a Gestalt pattern matched extreme learning classifier to improve the confidentiality rate. The proposed learning classifier consists of many layers to process the given input. The number of cloud users is given to the input layer. In that layer, the random weights and bias are assigned. In hidden

layer 1, user wants to access the data and they generate the hash value. Then the Gestalt pattern matching coefficient is applied to match the hash value. Finally, the activation neurons in the hidden node find authorized or unauthorized users (i.e. attacks). Finally, authorized or unauthorized users classification results at the output layer. The server permits to access the data to the authorized user resulting it enhance the confidentiality rate.

// Algorithm 3: Gestalt pattern matched Extreme Learning Classifier
Input: cloud users $Cu_i = Cu_1, Cu_2, \dots, Cu_n,$ blockchain central server CS **Output:** Increase the confidentiality rate

Begin
Step 1: Number of cloud users $Cu_i = Cu_1, Cu_2, \dots, Cu_n$ as input at the input layer
Step 2: For each user// **[hidden layer 1]**
Step 3: Send access request to blockchain server
Step 4: Generate hash value for data 'H_s' and sent to server
Step 5: end for
Step 6: CS verifies the hash value using (11) // **[hidden layer 2]**
Step 7: **Apply** sigmoid activation function using (12) // **[hidden layer 3]**
Step 8: $\boldsymbol{if}$ $(\omega = 1)$ **then**
Step 9: ω returns 'pattern matched '
Step 10: user is said to be authorized
Step 11: else
Step 12: ω returns 'pattern not matched '
Step 13: user is said to be unauthorized
Step 14: end if
Step 15: Obtain the classification results at the output layer
End

Section: Research Paper
ISSN 2063-5346

4. EXPERIMENTAL SETUP

Experimental evaluation of proposed GPMELCB model and existing System HBRSS [1] Bayesian game and blockchain-based auction model [2] and blockchain-based provenance system [3] are implemented using Java language with CloudSim simulator. For the experimental consideration, the CIDDS (Coburg Intrusion Detection Data Sets)-001 dataset taken from https://github.com/markusring/CIDDS for secure data transmission on the cloud server. The dataset is used for cloud-based attack detection during the data communication model. The dataset includes a traffic data that was recorded in the external server environment over a period of four weeks. The dataset includes 16 attributes for intrusion detection.

Table 2 Attribute information

S.No	Attributes	Description
1	Date first seen	Start time flow first seen
2	Duration	Duration of the flow
3	Proto	Transport Protocol (e.g. ICMP, TCP, or UDP
4	Src IP Addr	Source IP Address
5	Src Pt	Source Port
6	Dst IP Addr	Destination IP Address
7	Dst Pt	Destination Port
8	Packets	Number of transmitted packets
9	Bytes	Number of transmitted bytes
10	Flows	packet data flows
11	flag	Concatenation of all TCP Flags
12	class	Class label (normal, attacker, victim, suspicious or unknown
13	Tos	Type of Service
14	attackType	Type of Attack (portScan, dos, bruteForce)
15	attackID	Unique attack id. All flows which belong to the Same attack carries the same attack id.
16	AttackDescription	Provides additional information about the set attack parameters (e.g. the number of attempted password guesses for SSH-Brute-Force attacks)

Section: Research Paper
ISSN 2063-5346

3. PRFORMANCE RESULTS AND DISCUSSION

The comparative result analysis of proposed GPMELCB model and existing HBRSS [1] Bayesian game and blockchain-based auction model [2] and blockchain-based provenance system [3] are discussed in this section with different parameters transaction latency, communication cost, data confidentiality and data integrity with a different number of data and cloud users. The comparative performance analyses are done with the help of either table or graphical representation.

Transaction latency: Latency in a blockchain is computed as amount of time consumed for a particular platform to respond to each transaction. The transaction latency is measured as difference between the time of the block generation and actual time when the transaction broadcasted.

$$TL = t_{BInc} - t_{TB} \tag{15}$$

Where, 'TL' indicates a transaction latency measured based on the time the generation of blocks were included 't_{BInc}' and the actual time when the transaction was broadcasted 't_{TB}'. It is measured in terms of seconds (s).

Computation time: It is defined as an amount of time taken by the algorithm to perform secure data communication. The computation time is mathematically calculated as follows,

$$CT = \sum_{i=1}^{n} D_i * T(secure\ data\ transmission) \tag{16}$$

From (16), CT denotes Computation time,D_i denotes a number of data, T denotes a time for secure data transmission. Computation time is measured interms of milliseconds (ms).

Data confidentiality rate: It is measured based on the cloud data as accessed by the authorized cloud users with respect to the total number of cloud users' data. Data confidentiality rate is mathematically expressed as given below.

Section: Research Paper
ISSN 2063-5346

$$DCR = \frac{U_D[Auth]}{U_D} * 100 \quad (17)$$

From the above equation (17), the data confidentiality rate 'DCR' is measured based on data accessed by authorized cloud user '$U_D[Auth]$)' and the number of cloud user data 'U_D'. It is measured in terms of percentage (%).

Data Integrity rate: It is measured as the ratio of number of data that are not altered by the unauthorized access (i.e. attacks) to the total number of data. The data integrity rate is measured as given below,

$$DIR = \sum_{i=1}^{n} \frac{DNA}{U_D} * 100 \quad (18)$$

In (18), DIR represent the data integrity rate which is measured as number of data 'U_D' and number of data not altered 'DNA'. The data integrity rate is measured in percentage (%).

Table 2 Tabulation of transaction latency

Number of cloud user data	Transaction latency (s)			
	GPMELCB	**HBRSS**	Bayesian game and blockchain-based auction model	Blockchain-based provenance system
1000	60	80	75	70
2000	63.5	82	80	72.6
3000	68.36	85.6	83	75.2
4000	70.1	88.7	85.52	78
5000	75.65	92.6	89.62	80
6000	81.2	96	93.4	84.56
7000	83.69	97.2	95.5	88.65
8000	86.21	110.6	105.6	90.48
9000	90.63	124.6	115.2	97.65
10000	95.14	130	123.5	112

Section: Research Paper
ISSN 2063-5346

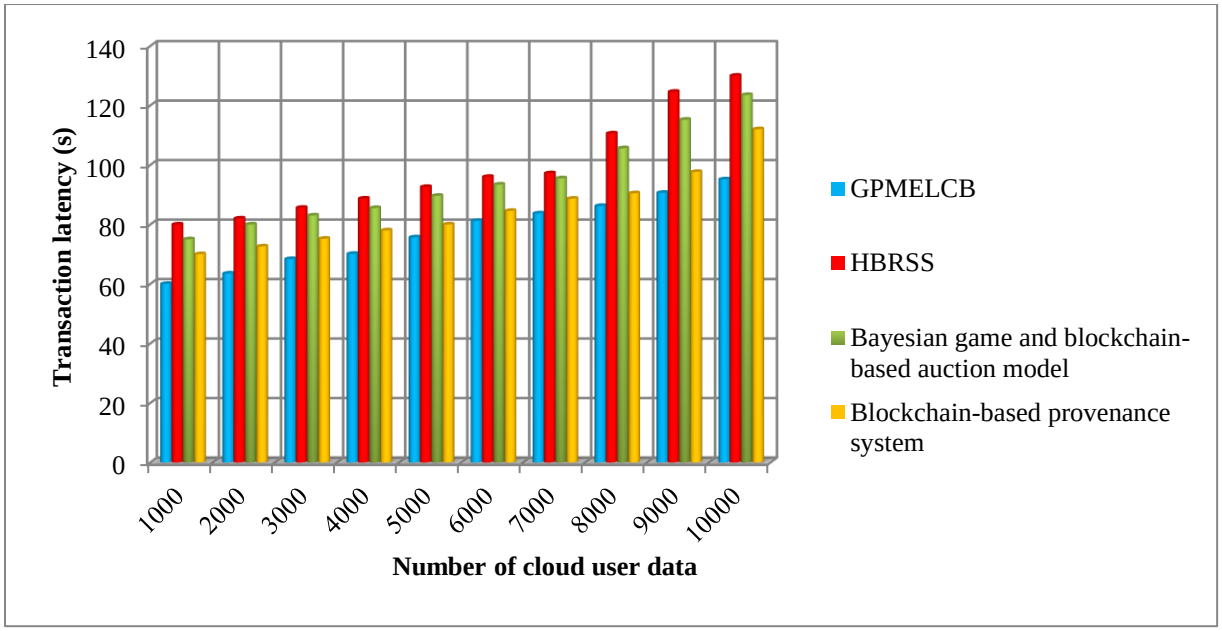

Figure 9 Graphical comparison of Transaction latency

Figure 9 depicts the performance of transaction latency for blocks distribution for each user data. The transaction latency is computed based on the number of user data varied from 1000 to 10000. Among four methods, the transaction latency of the GPMELCB model is found to be minimized than the existing [1] [2] and [3]. By increasing the number of cloud user data from 1000 to 10000, the transaction latency of all three methods gets increased. But the GPMELCB model is significantly better than the [1] [2] [3]. The reason for the low transaction latency by the GPMELCB model uses the Nelder–Mead Byzantine Fault Tolerance consensus algorithm to perform block validation. In blockchain construction, the consensus algorithm is a most significant method that directs the trust among the blocks. The result help to minimize the malicious cloud blocks for creating additional transactions in the cloud computing environments to perform transaction and hence consume lesser latency. The average of the ten comparison results indicates that the transaction latency for blocks distribution is found to be reduced by 21%, 18% and 9% when compared to existing methods.

Section: Research Paper
ISSN 2063-5346

Table 3 Tabulation of Execution time

Number of cloud user data	Execution time (ms)			
	GPMELCB	HBRSS	Bayesian game and blockchain-based auction model	Blockchain-based provenance system
1000	110	128	125	121
2000	124	140	136	130
3000	135	165	150	144
4000	148	168	160	152
5000	160	185	180	170
6000	174	228	213	180
7000	224	252	238	231
8000	232	272	256	240
9000	279	306	297	288
10000	320	350	340	330

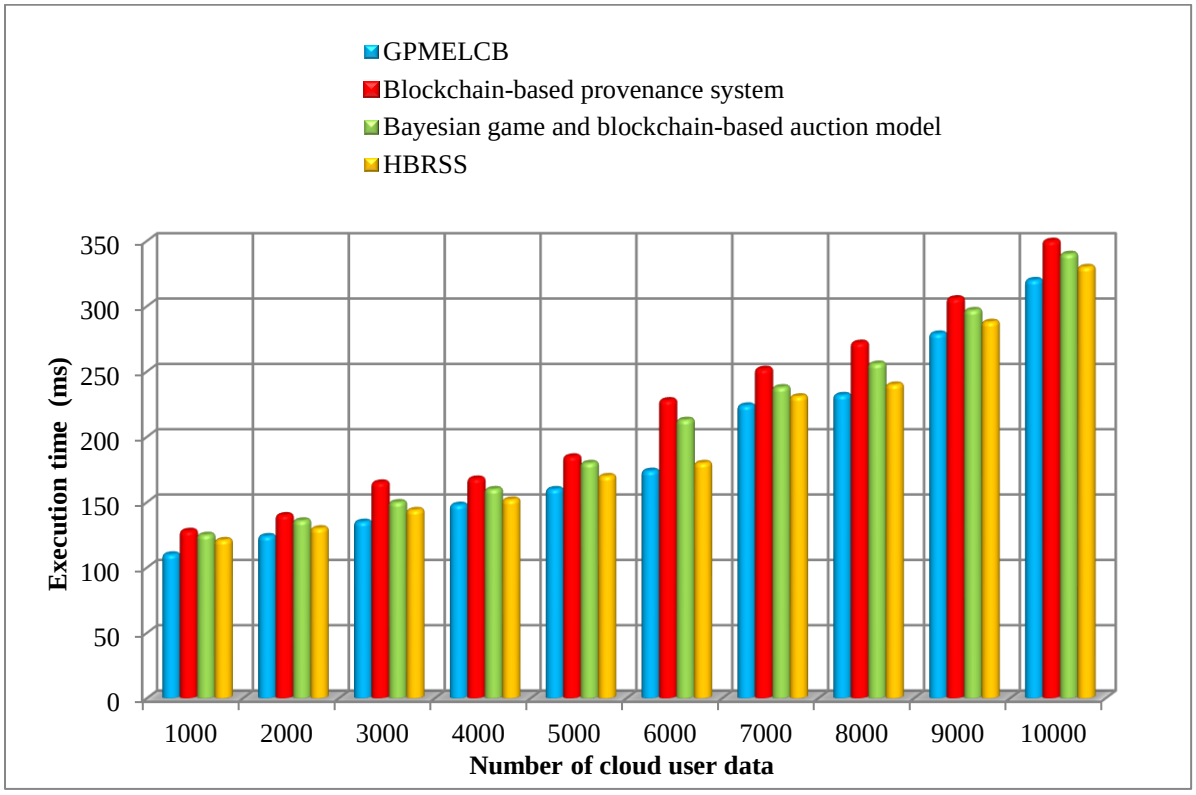

Figure 10 Graphical comparison of execution time

Section: Research Paper
ISSN 2063-5346

Table 3 and figure 10 given above illustrates the performance results of the execution time for secure data communication. The time consumption is calculated using four different methods GPMELCB model and existing HBRSS [1] Bayesian game and blockchain-based auction model [2] and blockchain-based provenance system [3] shown in figure 10. The execution time for secure data communication is found to be increased while increasing the number of user data. The observed overall results indicate that the execution time of GPMELCB model is relatively minimized than the existing methods. Let us consider the 1000 user data for calculating the time to perform secure data transmission.

The execution time was observed '$110ms$' using GPMELCB model. However, the time consumption of existing [1], and [2] [3] was found to be $128ms$,$125ms$ and $121ms$ respectively. The observed results indicate that the GPMELCB model minimizes the execution time. After obtaining the ten results, the overall time consumption of the GPMELCB model is compared to the results of existing methods. The average of ten results indicates that the proposed GPMELCB model decreases the time consumption by 23%, 18% and 11% as compared to the [1] and [2] [3] respectively. This is because of applying the Davis Mayer cryptographic compression based block generation and Nelder–Mead Byzantine Fault Tolerance consensus algorithm based block validation. When the user wants to access the data from the server, the server first verifies their authenticity using Gestalt pattern matched Extreme Learning Classifier. This process enhances the security of data transmission with minimum time.

Section: Research Paper
ISSN 2063-5346

Table 4 Tabulation of Data confidentiality rate

Number of cloud user data	Data confidentiality rate (%)			
	GPMELCB	**HBRSS**	Bayesian game and blockchain-based auction model	Blockchain-based provenance system
1000	98.1	89	91.8	93.5
2000	97.75	87.5	90	92
3000	97.23	86.53	89.5	91.86
4000	96.62	86.12	88.12	91.4
5000	96.24	85.3	88.1	91.26
6000	96.18	85.08	88.08	90.86
7000	95.65	84.92	87.17	90.28
8000	94.86	84.35	87.06	89.03
9000	93.95	83.6	87	88.97
10000	93.21	81.56	86.95	88.12

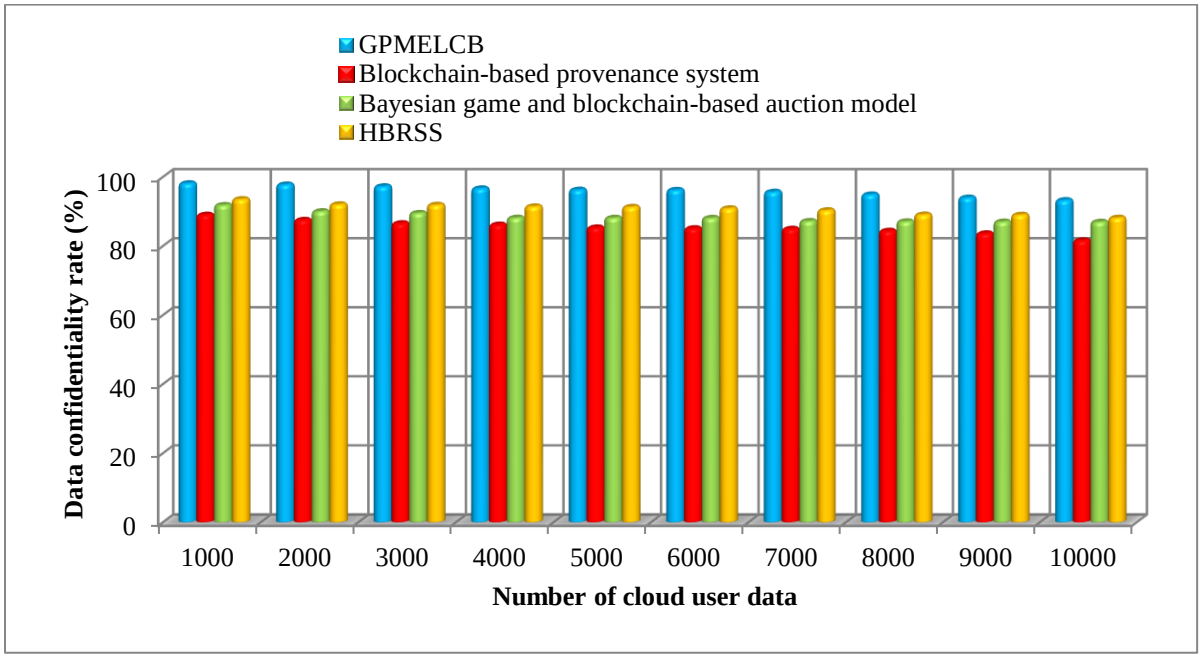

Figure 11 Graphical comparison of data confidentiality rate

Figure 11 indicates the performance comparison of the data confidentiality rate versus number of user data ranging from 1000 to 10000 by applying four methods namely GPMELCB model and existing HBRSS [1] Bayesian game and

blockchain-based auction model [2] and blockchain-based provenance system [3]. As shown in the graph, user data taken in the horizontal axis and the performance outcomes of data confidentiality rate was observed in vertical axis. Among three different methods, the proposed GPMELCB model increases the performance of data confidentiality than the existing methods. This is proved through statistical evaluation. Let us consider 1000 data taken as input in the first iteration for calculating the data confidentiality rate. By applying the GPMELCB model, the data confidentiality rate was observed to be 98.1% whereas the data confidentiality rate of existing [1] and [2] [3] was found to be 89%, 91.8%, 93.5% respectively. Similarly, different performance results are observed for each method. At last, the overall results of GPMELCB model are compared to the existing methods. The average of ten comparison results shows that the performance of the data confidentiality rate of gets increased by 12%, 7% and 6% when compared to [1] [2] and [3] respectively. This is because of the proposed GPMELCB model uses the Gestalt pattern matched extreme learning classifier to identify the authorized or unauthorized user during the data communication. During the data communication, the block chain server verifies the hash value. Based on the verification, the authorized and unauthorized users are identified. The blockchain server grants the access to the authorized user and denied the access to unauthorized users. This helps to improve the data confidentiality rate.

Table 5 and figure 12 given above illustrate the performance of data integrity rate for 10000 distinct user data. In order to calculate the data integrity rate, the numbers of data are taken in the ranges from 1000 to 10000. From the above graphical results, the GPMELCB technique outperforms well to achieve higher data integrity rate when compared to existing results. Let us consider the experiment with 1000 user data, the performance of data integrity rate was found to be 97.7% using GPMELCB model. By applying [1] [2] [3], the data integrity

Section: Research Paper
ISSN 2063-5346

rate was found to be 87.4% 90.7% and 92.8% respectively. Likewise, different performance results are observed for each method.

Table 5 Tabulation of Data integrity rate

Number of cloud user data	Data integrity rate (%)			
	GPMELCB	HBRSS	Bayesian game and blockchain-based auction model	Blockchain-based provenance system
1000	97.7	87.4	90.7	92.8
2000	96.85	85.5	89	91.5
3000	96.33	85.33	88	89.5
4000	95.25	85.12	86.3	88.52
5000	95.16	84.2	85.12	88.2
6000	95.08	84.16	85	88.08
7000	94.8	83.78	84.64	87.17
8000	94.56	82.65	84.06	86.56
9000	92.27	82.33	83.62	86.44
10000	91.45	80.46	81.25	86.12

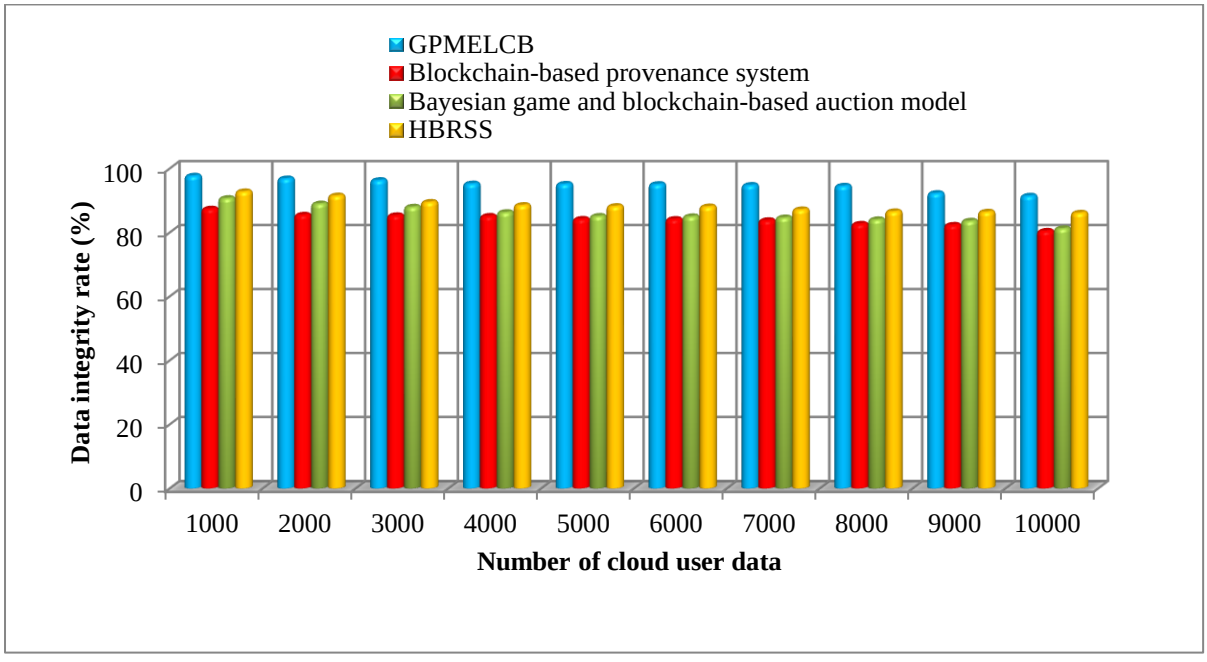

Figure 12 Performance Comparison of Data integrity rate

The overall observed results of GPMELCB model is compared to existing [1] [2] [3]. The comparison results evidently proves that the performance of data integrity rate GPMELCB model is increased by 13%,115 and 7% when compared

to [1] [2] [3] respectively. The reason behind the improvement was owing to the application of Davis Mayer cryptographic compression function. The one way compression is used to generate hash value of the user data and stored into the Blockchain central server. Once it stored into the server, the hash value are not altered by any intruders. As a result, the integrity rate of the user data gts improved.

6. CONCLUSION

In this paper, a new GPMELCB model is developed to ensure the security of data transmission in the cloud computing environment. The GPMELCB model initially performs the user registration process for secure data transmission. After that, Davis Mayer cryptographic compression function is applied to generate a hash value for each user data and stored in the server. The Nelder–Mead Byzantine Fault Tolerance consensus algorithm is employed for validating the generated block and it is added to the blockchain network. Finally, the security of data communication is performed through the authorized user by means of a Gestalt pattern-matched extreme learning classifier. The authorized user receives the data with higher confidentiality and integrity. The comprehensive performance of the GPMELCB model is carried out with different metrics. The results showed that the proposed GPMELCB model provides better performance with an improvement of data confidentiality rate, and integrity and minimizes the execution time as well as transaction latency when compared to the existing works.

REFERENCES

[1] Hui Xie, Zhengyuan Zhang, Qi Zhang, Shengjun Wei, Changzhen Hu, “HBRSS: Providing high-secure data communication and manipulation in insecure cloud environments”, Computer Communications, Elsevier, Volume 174, 2021, Pages 1-12. https://doi.org/10.1016/j.comcom.2021.03.018

[2] Zeshun Shi, Huan Zhou, Cees de Laat, Zhiming Zhao, "A Bayesian game-enhanced auction model for federated cloud services using blockchain", Future Generation Computer Systems, Elsevier, Volume 136, 2022, Pages 49-66. https://doi.org/10.1016/j.future.2022.05.017

[3] Emmanuel Boateng Sifah , Qi Xia , Kwame Opuni-Boachie Obour Agyekum , Hu Xia , Abla Smahi , and Jianbin Gao , "A Blockchain Approach to Ensuring Provenance to Outsourced Cloud Data in a Sharing Ecosystem", IEEE Systems Journal , Volume 16, Issue 1, 2022, Pages 1673 – 1684. **DOI:** 10.1109/JSYST.2021.3068224

[4] Muhammad Usman Sana, Zhanli Li, Fawad Javaid, Hannan Bin Liaqat, and Muhammad Usman Ali, **"**Enhanced Security in Cloud Computing Using Neural Network and Encryption", IEEE Access , Volume 9, 2021, Pages 145785 – 145799. **DOI:** 10.1109/ACCESS.2021.3122938

[5] Ricardo Mendes, Tiago Oliveira, Vinicius Cogo, Nuno Neves, AlyssonBessani, "Charon: A Secure Cloud-of-Clouds System for Storing and Sharing Big Data", IEEE Transactions on Cloud Computing, Volume 9, Issue 4, 2021, Pages 1349 – 1361. **DOI:** 10.1109/TCC.2019.2916856

[6] B. Sowmiya, E. Poovammal, Kadiyala Ramana, Saurabh Singh, Byungun Yoon, "Linear Elliptical Curve Digital Signature (LECDS) With Blockchain Approach for Enhanced Security on Cloud Server", IEEE Access , Volume 9, 2021, Pages 138245 – 138253. **DOI:** 10.1109/ACCESS.2021.3115238

[7] Dheresh Soni ,Deepak Srivastava , Ashutosh Bhatt , Ambika Aggarwal , Sunil Kumar , and Mohd Asif Shah , "An Empirical Client Cloud Environment to Secure Data Communication with Alert Protocol", Mathematical Problems in Engineering, Hindawi, Volume 2022, September 2022, Pages 1-14. https://doi.org/10.1155/2022/4696649

[8] Sheng Peng , Zhiming Cai ,Wenjian Liu, Wennan Wang, Guang Li, Yutin Sun, and Linkai Zhu, "Blockchain Data Secure Transmission Method Based on Homomorphic Encryption", Computational Intelligence and Neuroscience, Hindawi, Volume 2022, 0 April 2022, Pages 1-9. https://doi.org/10.1155/2022/3406228

[9] Ying Huang and Jing Lei, "Distributed secure transmission for covert communication under multi-user network", IET Communications, Volume 16, Issue 16 , Pages 1901-1911. https://doi.org/10.1049/cmu2.12445

[10] Varun Prabhakaran and Ashokkumar Kulandasamy, "Integration of recurrent convolutional network and optimal encryption scheme for intrusion detection with secure data storage in the cloud", computational intelligence, Wiley, Volume 37, Issue 1, 2021, Pages 344-370. https://doi.org/10.1111/coin.12408

[11] Samy IAA and Safish Mary, "Secure Data Transmission in Cloud Computing Using Std-rsa With Eslurnn Data Classification and Blockchain Based User Authentication System", Research Square, 2022, Pages 1-21. DOI: 10.21203/rs.3.rs-1724672/v1

[12] Ruba Awadallah, Azman Samsudin, Je Sen The, Mishal Almazrooie, "An Integrated Architecture for Maintaining Security in Cloud Computing Based on Blockchain", IEEE Access, Volume 9, 2021, Pages 69513 – 69526. **DOI:** 10.1109/ACCESS.2021.3077123

[14] Jia Yu and Rong Hao, "SEPDP: Secure and Efficient Privacy Preserving Provable Data Possession in Cloud Storage", IEEE Transactions on Services Computing, Volume 14, Issue 6, 2021, Pages 2090 – 2092. **DOI:** 10.1109/TSC.2019.2912379

[15] Seunghwan Son, Joonyoung Lee, Myeonghyun Kim, Sungjin Yu, Ashok Kumar Das, and Youngho Park, "Design of Secure Authentication Protocol

for Cloud-Assisted Telecare Medical Information System Using Blockchain", IEEE Access , Volume 8, 2020, Pages 192177 – 192191. **DOI:** 10.1109/ACCESS.2020.3032680

[16] Feras M. Awaysheh , Mohammad N. Aladwan, Mamoun Alazab , Sadi Alawadi, José C. Cabaleiro , and Tomás F. Pena, "Security by Design for Big Data Frameworks Over Cloud Computing", IEEE Transactions on Engineering Management , Volume 69, Issue 6, 2022, Pages 3676 – 3693. **DOI:** 10.1109/TEM.2020.3045661

[17] Nazatul Haque Sultan, Nesrine Kaaniche, Maryline Laurent, Ferdous Ahmed Barbhuiya, "Authorized Keyword Search over Outsourced Encrypted Data in Cloud Environment", IEEE Transactions on Cloud Computing, Volume 10, Issue 1, 2022, Pages 216 – 233. **DOI:** 10.1109/TCC.2019.2931896

[18] Sandeep kaur ,Gaganpreet kaur, and Mohammad Shabaz, "A Secure Two-Factor Authentication Framework in Cloud Computing", Security and Communication Networks, Hindawi, Volume 2022, March 2022, Pages 1-9. https://doi.org/10.1155/2022/7540891

[19] John Kwao Dawson, Frimpong Twum, James Benjamin Hayfron Acquah, Yaw Marfo Missah, "Ensuring confidentiality and privacy of cloud data using a non-deterministic cryptographic scheme", PLoS ONE, Volume 18, Issue 2, Pages 1-19. https://doi.org/10.1371/journal.pone.0274628

[20] Yogesh M Gajmal and R. Udayakumar, "Privacy and Utility-Assisted Data Protection Strategy for Secure Data Sharing and Retrieval in Cloud System", Information Security Journal: A Global Perspective, Volume 31, Issue 4, 2022, Pages 1-15. https://doi.org/10.1080/19393555.2021.1933270