Multimedia Tools and Applications (2024) 83:31505–31526
https://doi.org/10.1007/s11042-023-16662-6

IoT-based blockchain intrusion detection using optimized recurrent neural network

V. Saravanan[1] · M Madiajagan[2] · Shaik Mohammad Rafee[3] · P Sanju[4] · Tasneem Bano Rehman[5] · Balachandra Pattanaik[6]

Received: 23 November 2022 / Revised: 30 June 2023 / Accepted: 23 August 2023 /
Published online: 16 September 2023

Abstract
In recent years, Intrusion Detection Systems (IDS) monitor the computer network system by collecting and analyzing data or information by identifying the behavior of the user or predicting the attacks by the automatic response. So, in this paper, the Blockchain-based African Buffalo (BbAB) scheme with Recurrent Neural Network (RNN) model is proposed for detecting the intrusion by enhancing security. Furthermore, normal and malware user datasets are collected and trained in the system and the dataset is encrypted using Identity Based Encryption (IBE). The encrypted data are securely stored in the blockchain in the cloud. Hereafter, Recurrent Neural Network (RNN) was employed to detect the intrusion in a cloud environment. African buffalo optimization was used in the RNN prediction phase for continuous monitoring of intrusion. Finally, the performance results of the developed technique are compared with other conventional models in terms of accuracy, precision, recall, F1-score, and detection rate. The outperformance of the designed model attains better accuracy of 99.87% and high recall of 99.92%.it shows the efficiency of the designed model to protect data and security in cloud computing.

Keywords Cloud computing · Intrusion detection system · Deep learning · Identity-based encryption · African buffalo optimization · Blockchain

✉ V. Saravanan
saravanan.vace675@gmail.com

[1] Department of Computer Science, College of Engineering and Technolgy, Dambi Dollo University, Dambi Dollo, Oromia Region, Ethiopia

[2] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

[3] Department of EEE, Sasi Institute of Technology & Engineering, Tadepalligudem, Andhra Pradesh, India

[4] Computer Science and Engineering, University College of Engineering Tindivanam, UCET, Tindivanam, India

[5] Department of Computer Science and Engineering, SAGE University Bhopal, Madhya Pradesh, India

[6] Department of Electrical and Computer Engineering, College of Engineering and Technology, Wollega University, Nekemte, Ethiopia

1 Introduction

Nowadays, the Internet of Things (IoT) is measured as a connected system based on the permitted protocols that may exchange information between several devices through the Internet [1]. The recent advancement of IoT contains sensors and devices that lead the developing fields of modern communication and computing technologies [2]. Currently, IoT is more comfortable and benefits human life at the expenditure of security [3]. Moreover, it is the most attractive target framework for cybercriminals and intrusion [4]. The most challenging task in the IoT environment is intrusion detection [5]. As well, it is the combination of certain layers containing the network layer which transfers the data packets among the hosts [6]. However, it is vulnerable and complex also leads to numerous security problems. Still, certain security techniques are developed for addressing security problems [7]. Moreover, the detection of intrusion is equivalent to the classification issue that is identifying the attacks such as Denial of Service (DOS) [8]. In any case, transporting gentle information to distributed and open public cloud storage facility acts as a security threat such as confidentiality, accessibility, and trustworthiness [9]. Frequently, cloud computing is web-based computing that stores the data or information offered from the software, infrastructure, devices, stages, and various resources [10]. Additionally, the Intrusion Detection System (IDS) monitors the computer network system by collecting and analyzing data or information by identifying the behavior of the user or predicting the attacks by an automatic response [5]. Additionally, network attacks concern mobile users and cloud providers also access the cloud distance through a routine activity [11]. Furthermore, security maintenance and intrusion detection is shown in Fig. 1.

Especially, blockchain is a decentralized and distributed system that shared immutable information among the nodes without an intermediator [12]. Additionally, the blocks in the blockchain are connected and stored in the distributed ledger [13]. All the blocks contain hash, data, timestamp, difficulty, nonce, previous hash, the public key, and private key [14]. For the identification or creation time of the block, the timestamp is used and a hash is generated to secure the data by hashing algorithm [15]. A nonce is denoted as an arbitrary value for ensuring the uniqueness of each block in the blockchain and also securing the block from attacks [16]. The references of the previous hash are the block hash that is immediately ahead of the current block [17]. Furthermore, private and public keys are used to encrypt and decrypt healthcare data [18]. Finally, the difficulty is a value that is used to apply hash value prefixes as zeros. It adjusts the time according to the essentials of the communication process [19].

Furthermore, blockchain is characterized by three kinds such as public, consortium, and private blockchain [20]. Everybody has participated in the public blockchain and

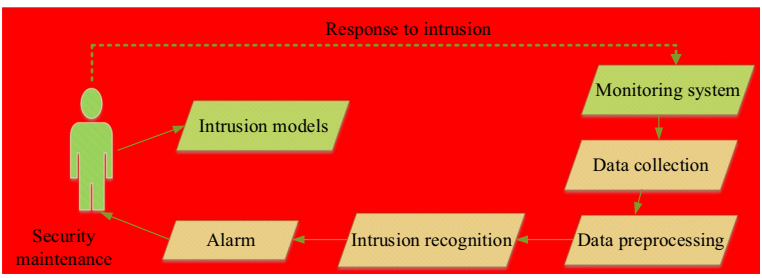

Fig. 1 Detecting intrusion and security maintenance

consortium blockchain is used among businesses [21]. Likewise, a private blockchain is used for strict data access organization as well as the entire node cannot contribute to the blockchain [22]. Consequently, blockchain is used in several fields like agriculture, healthcare, banking, cloud environment, and so on [23]. The most critical and challenging task in IDS is detecting attacks with high accuracy [24]. Many techniques are developed to enhance the performance of intrusion detection such as Deep Learning (DL), Recurrent Neural Networks (RNN), Long Short term Memory (LSTM), Machine Learning (ML), and so on but still have issues with vanishing gradient problems, overfitting, data complexity, absence of communication compatibility, error, security, and attack issues. So, develop an optimization-based deep learning model with cryptographic technique and blockchain to enhance the security of the cloud environment and predict attacks present in that environment. The main motive of the designed model is to enhance security by encrypting the data using IBE and securely stored in the cloud using blockchain. Additionally, African buffalo optimization was updated in the RNN phase for continuous monitoring of intrusion or attacks in the cloud environment. The designed model accurately detects the intrusion by the threshold values of each user and secures the cloud data by encryption. The main objective and key contribution of the designed model are detailed below,

2 Objective

- To design an optimization-based recurrent neural network for continuous monitoring of intrusion in a cloud environment
- To enhance the security of the cloud environment design a cryptographic-based blockchain technique
- To attain better experimental results using African buffalo optimization and deep learning

Key contribution

- Initially, the malware executable detection dataset was collected from the net source and trained to the Python frame.
- Then design BbAB-RNN with suitable parameters for detecting the intrusion and securing the data.
- Moreover, IBE encryption is utilized to encrypt data into cipher text and securely store it in the blockchain.
- Consequently, RNN was employed with African Buffalo Optimization (ABO) for detecting the intrusion by continuous monitoring.
- Subsequently, the key metrics are calculated with other existing mechanisms in terms of accuracy, precision, F1-score, recall, and execution time to prove the efficiency of the designed model.

The main advantages of the designed model are high reliability, improve security, less execution time, detection attack, less error rate, high accuracy, continuous monitoring of attacks, and communication compatibility.

The organization of the research article is summarized as follows: Section 2 discussed related works of attack detection using blockchain and Section 3 details the system model and problem statement. Similarly, Section 4 elaborated on the process of the proposed methodology and Section 5 described the results and discussions. Finally, the conclusion about the designed model is discussed in Section 6.

3 Related works

3.1 Some literature surveys based on intrusion detection are detailed below

Chao Liang et al [20] developed a hybrid placement approach with blockchain, DL, and multi-agent systems. It includes data collection, data analysis, data management, and data response. Moreover, the NSL-KDD dataset is tested and trained in the system and also attained results that validate the efficiency of the developed model for detecting attacks. Thus the attained results are suitable for detecting intrusion in an IoT environment but obtain vanishing gradient problems.

Generally, blockchain-based intrusion detection enhances the detection and data privacy also ensemble learning simplifies malicious events and confirms data privacy. Osama et al [21] developed a deep blockchain system for offering the security of blockchain-based IoT network. Here, bidirectional LSTM is employed in DL for dealing and assessing sequential data. This framework is potentially used for the decision support system that contributes to the cloud providers and users. However, it has overfitting and complex problems.

Aruna and Thilagam [22] proposed an intrusion detection-based CNN framework using ant lion optimization. Here, the designed model identified the attacks present in the cloud and also classified them successfully. Moreover, the experimental outcomes enhance the classification models by enhancing the detection rate and high accuracy. It attains 94% classification accuracy and 0.0012 rates of error but the absence of communication compatibility.

The set of smart devices is called an IoT network that contains home appliances, sensors, vehicles, computers, etc which are interconnected by the global internet. Derhab et al [23] proposed DL based IDS framework for improving the performance of the detection system. Also, features engineering models are used for feature transformation and feature space reduction. Additionally, the Bot-IoT dataset is used for testing and training processes but it has attained security problems.

Hamed et al [24] developed a potential RNN for detecting malware using IoT applications. It analyses the operation codes of IoT applications and the dataset contains 270 benign and 281 malware files. By analyzing the LSTM configurations, the designed model attains 98.18% accuracy when compared to other models but it has the issues of rigid architecture, controlling data, and lack of device security.

Arwa Aldweesh et al [28] proposed a DL-based intrusion detection framework for detecting intrusion in the cloud. Also intended for intrusion detection is Temporal Convolution Neural Network (TCNN). To address unbalanced datasets, TCNN is paired with Synthetic Minority Oversampling (SMO) model and Nominal Continuous (NC) technique According to experimental findings, TCNN successfully balances effectiveness and efficiency. In terms of training time, it has a performance that is fairly comparable to CNN. However, the cost of computing is substantial.

To address the difficulties of handling resources in the IIoT, Premkumar, et al [29] created a neural network system and fuzzy logic framework. The IoT's use of the neuro-fuzzy method aids in the development of controlling intelligence systems. The majority of calculations required to address resource management concerns in IIoT networks could be implemented using neural networks and fuzzy sets. On a sensor node, TinyOS is used to simulate and create a real-time experimentation network. But require more execution time.

A unique Poor and Rich Optimisation (RPO) using DL Model for Blockchain-based Intrusion Detection model was created by Romany [30]. The created model's feature selection

procedure uses the Adaptive Harmony Search (AHS) technique. The attention-based bi-directional gated RNN model is utilized for intrusion detection and classification. Furthermore, the CPS ecosystem uses blockchain technology to improve security. The performance of the designed model is validated with prevailing models to prove efficiency. However, it has a computational complexity issue.

A hybrid RNN framework is presented by Asma, et al [31] for recognizing a collection of incursions in the context of the IoT. After gathering a set of data, an RNN is utilized to estimate the various individual invasions. Based on a decomposition method, each of these intrusions is subsequently utilized to identify a group of outliers. The outcomes demonstrate the advantages of the suggested framework and its unmistakable superiority to cutting-edge methods. But the detection rate is less when comparing other models.

4 System model and problem definition

The Internet of Things (IoT) based attack detection framework is designed with a multi-agent system and blockchain. It discovers the attacks present in the network by continuous monitoring of IoT devices. Moreover, the basic system model and problem definition are shown in Fig. 2. It contains five phases such as data collection, data processing, analysis and detection, response, and blockchain. Initially, data are collected using a communication

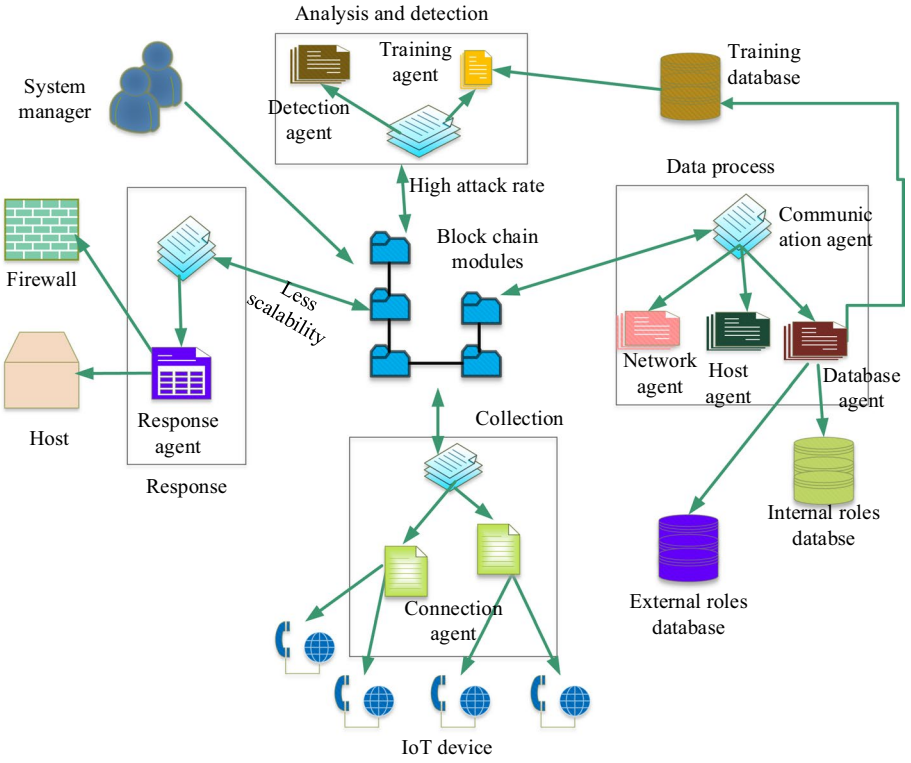

Fig. 2 System model and problem definition

and collection agent that is responsible to gather the data from IoT devices [25]. The data process module is utilized to perform a preprocessing process such as data integration, missing values, and data validation. Hereafter, the detection and analysis module detects unlabelled data and malicious patterns using labelled data. The response module produces a data visualization chart based on the user response. Finally, blockchain stores and records the actions in the cloud system. It secures the data from attacks but attains some limitations like less scalability, less integrity, and a high attack rate.

The most challenging problems in the IoT-based blockchain technique are attack detection because of less scalability, less reliability, and high communication delay. Some techniques require more time for detecting attacks also the error rate is high. Furthermore, the energy consumption rate is high because of time complexity; to overcome these issues design a blockchain-based optimized RNN strategy for enhancing the performance of intrusion detection in a cloud environment. It continuously monitors the attacks and secures the data using blockchain.

5 Proposed methodology

To detect the intrusion in the cloud environment Blockchain-based African Buffalo (BbAB) scheme with Recurrent Neural Network (RNN). That enhances the security of the cloud by encrypting data stored in the blockchain. Furthermore, Identity-Based Encryption (IBE) is utilized to encrypt the data and the encrypted data is securely stored in the database with separate blocks. The fitness of African Buffalo Optimization (ABO) continuously monitors the attacks present in the network and then detects the attack based on the threshold value. Moreover, IoT-based normal and malware datasets are updated to the designed cloud system. The architecture of the proposed methodology is illustrated in Fig. 3.

Hereafter, normal and malware user dataset are sent to IBE for securing the data by converting plain text into ciphertext. By using the session key, data are encrypted and the encrypted data is sent to the block using a private key. Moreover, blocks are transferred to the blockchain for securing the data. Then update the AB fitness in the detection phase for continuous monitoring of intrusion present in the network. Finally, the designed model secures the data and also detects the intrusion by their threshold value. The novelty of the designed model is to create an optimization-based cryptographic and deep learning model enhancing the security of the cloud environment and detecting malware present in the cloud using African buffalo fitness. The encrypted data are securely stored in the blockchain with several blocks which improve the security of data from malware. The designed model attains high accuracy and high recall using ABO, but the existing model's experimental outcomes are less, also developed model takes low execution time when compared to other conventional models. It overcomes the issues of computational complexity, security issues, vanishing gradient issues, and overfitting.

5.1 Dataset description

In this research Malware Executable Detection (MED) dataset (https://www.kaggle.com/datasets/piyushrumao/malware-executable-detection?select=uci_malware_detection.csv) was taken from the Kaggle website dataset and updated in the Python system. Features from both non-malicious and malicious Windows executable files are included in the

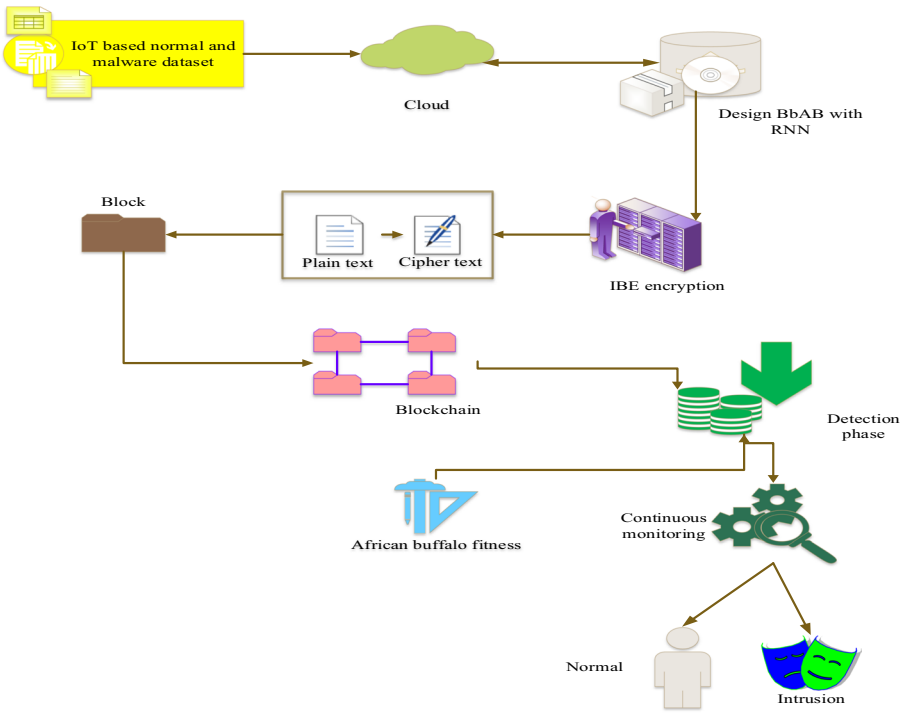

Fig. 3 Proposed methodology

dataset. This training file was produced using hybrid features from binary hexadecimal and DLL calls. The file contains 373 samples in total, 301 of which are malicious files and the remaining 72 are not. Malware samples make up a larger portion of the dataset than normal samples. From 531 features are shown, and a label field indicates if the file is malicious or not. It was impossible to describe the binary hexadecimal feature. This holds for any DLL calls that are present in it. Out of the 531 features, some can be discarded because they are incredibly unimportant. The real value of the dynamic file's field label would indicate whether or not it was malware. Initially, the input dataset was updated to the designed technique to encrypt and secure the data from malware. The training process used 80% of the dataset and the testing process used 20% of the dataset.

5.2 Encryption using Identity-Based Encryption (IBE)

The IoT-based collected dataset is updated to the designed model and they are updated to the cloud. Furthermore, the IBE scheme is employed to covert the original information or pain text into ciphertext, and the encryption process is processed using private and public keys. IBE's potential weaknesses are cryptographic errors, key escrow issues, poor performance, using an outdated, less safe algorithm, configuration files with a hard-coded password, and incorrect administration of cryptographic keys. There is not enough randomness for cryptographic operations because of the lack of security after encryption. But developed model used blockchain technology to secure the encrypted data, so the performance of the cloud security is enhanced. Especially, data owners securely transmit the data to an untrusted cloud server.

The transmitted data are encrypted and stored in a ciphertext format in the blockchain. Additionally, the blocks in the blockchain are connected and stored in the distributed ledger. All the blocks contain hash, data, timestamp, difficulty, nonce, previous hash, the public key, and the private key. For the identification or creation time of the block, the timestamp is used. It is only accessible by the permitted users by the data owners. The IBE contains three processes such as key generation, data encryption, and data decryption.

- Identity-Based Encryption

The normal and malware datasets are considered as input and return the secret value x_d of the user details and the session key asSx_d. Thus the generated secret value and session key are not disclosed. Then generate the random number k for the identification of intrusion that offers security to the designed technique. Thus the generated k is choosing to identify the corresponding private key P_{ky}^{r} which is obtained using Eq. (1).

$$K(g_e) = \left(m_k \alpha \left(j' \prod_{i \in k} P_i \right), Sx_d \right) \tag{1}$$

Let, P_i is represented as i^{th} bit of user identity, $m_k\alpha$ is denoted as a master key, $j^{'}$ is termed as a public parameter and Sx_d is represented as a session key.

To convert, the plain text into ciphertext, user identity information $I_d$and the master key $m_k\alpha$ is used. The plain text is denoted asT_p, and T_c is considered ciphertext, and $A_{pk}^{d}$is considered as a public key. The conversion of plain text into ciphertext is processed using Eqn. (2).

$$D_{en} = I_d \left(T_p (m_k \alpha) A_{pk}^{d} \left(j' \prod_{i \in k} P_i \right), T_c \right) \tag{2}$$

If the public key A_{pk}^{d} is incorrect means attaining encryption failure. Moreover, IBE encryption secures the data from unauthorized access and the encrypted data are securely stored in the blockchain using the private keys.

The decryption of data from the ciphertext needs a private key P_{ky}^{r} that is used to decrypt the ciphertext into plaintext. Moreover, the decryption of the designed model is processed using Eq. (3).

$$D_{de} = I_d \left(T_c (m_k \alpha) P_{ky}^{r}, \frac{T_c (m_k) \left(j' \prod_{i \in k} P_i \right)}{T_c (m_k \alpha) \left(j' \prod_{i \in k} P_i \right)}, T_p \right) \tag{3}$$

The encrypted data are sent to the blocks using the private key, and then the blocks are transferred to the blockchain. The designed model to identify intrusion is detailed in the algorithm.1.

Design BbAB with RNN framework for detecting intrusion in a cloud environment

Input: *normal and malware user details*

Output: *detect an intrusion*

Start

{

Initialize dataset

// normal and malware user

Design BbAB with RNN

Update the dataset

// to secure the data from attacks

Identity-based encryption

//Encrypt the data for enhancing security

Key generation

// Generate the session key

For all j=1

{

session key, Sx_d

}

End for

Data encryption

//plain text into ciphertext

Generate public key, A_{pk}^{d}

Encrypt the data using Eq. (2)

Data decryption

//ciphertext into plain text

Generate private key, P_{ky}^{r}

Decrypt the data using Eq. (3)

Encrypt data sent to block

Block transfer the data into the blockchain using key

Detection phase

// Predict intrusion in a cloud environment

Update AB fitness

Calculate threshold value

if $(t_r(v) \le 0.1$

{

Intrusion

}

else** if* $(t_r(v)) \ge 0.1$)***

{

normal

}

End if

Detect intrusion present in the network

Enhance the Security of the data

}

End

All the blocks in the blockchain involve a hash function and the user details are hashed by the provided key generation. Thus the key generator generates the keys required for creating the blocks in the blockchain. Finally, secure the user details in the blockchain with secret keys. The detailed flow of IBE is shown in Fig. 4.

- Detection phase

In the detection phase, update the fitness function of African buffalo in RNN it contains input, hidden, and output layers for accurately detecting the intrusion in the cloud environment. The process of fitness in AB is identifying the food and an attacker using the 'waa' (alert) and 'maa' (alarm) sounds. Thus the developed technique searches and identifies the optimal solution and increases the appropriate number of random solutions for detecting intrusion. It consists of two stages such as exploration and exploitation. The tracking of intrusion in fog computing was performed using Eq. (4),

$$D(ph) = \begin{cases} 1, \ if \ \sum_{s=1}^{p} D_{de} \times P_i \times t_r(v) \times A_b(t) + C_m\left(P_i\right) > 0 \\ 0, \qquad\qquad otherwise \end{cases} \tag{4}$$

Here, $t_r(v)$ is considered as the threshold value of each user, C_m is represents the continuous monitoring element and $A_b(t)$ is denoted as the fitness function of AB which identifies and searches the attacks in the network using their searching behavior. In the detection phase, $t_r(v)$and C_m are the parameters play an important role to detect the intrusion. C_m continuous monitor the attacks present in the network based on the $t_r(v)$. While the threshold value is $t_r(v) \leq 0.1$ means present intrusion but the threshold value is $\boldsymbol{if}(t_r(v)) \geq 0.1$ means normal. The developed model easily detects the intrusion using these parameters and minimizes the time. To get the optimum result, the specific threshold ranges were fixed then during the process the fitness module can be iterated again and again till the finest values are reached Based on the threshold value of the normal user, identify the intrusion in the network Moreover, a developed framework to identify and detect the intrusion also enhances the performance of detection accuracy. The workflow of the designed model is shown in Fig. 5.

6 Results and discussions

The designed paradigm was implemented in a Python framework and the normal and malware user datasets are collected from the net source and trained in the Python system. Initially, normal and malware user dataset are tested and trained in the system then they are updated to the designed model. Moreover, data encryption is processed using IBE and encrypting the data with the generated public key. Afterward, convert the plain text into ciphertext that is transferred to the blocks using the private key. Then the blocks are transferred to the blockchain to secure the data. Furthermore, update the AB fitness in the RNN to detect the intrusion by continuous monitoring of the network. Finally, attained performance results are compared with other conventional models. The details of hardware and software specifications are detailed in Table 1.

6.1 Performance metrics

The attained results of the proposed BbAB-RNN model are validated with other relevant techniques in terms of accuracy, precision, recall, execution time, and F-measure. Moreover, the designed model efficiency is compared with various conventional techniques such as

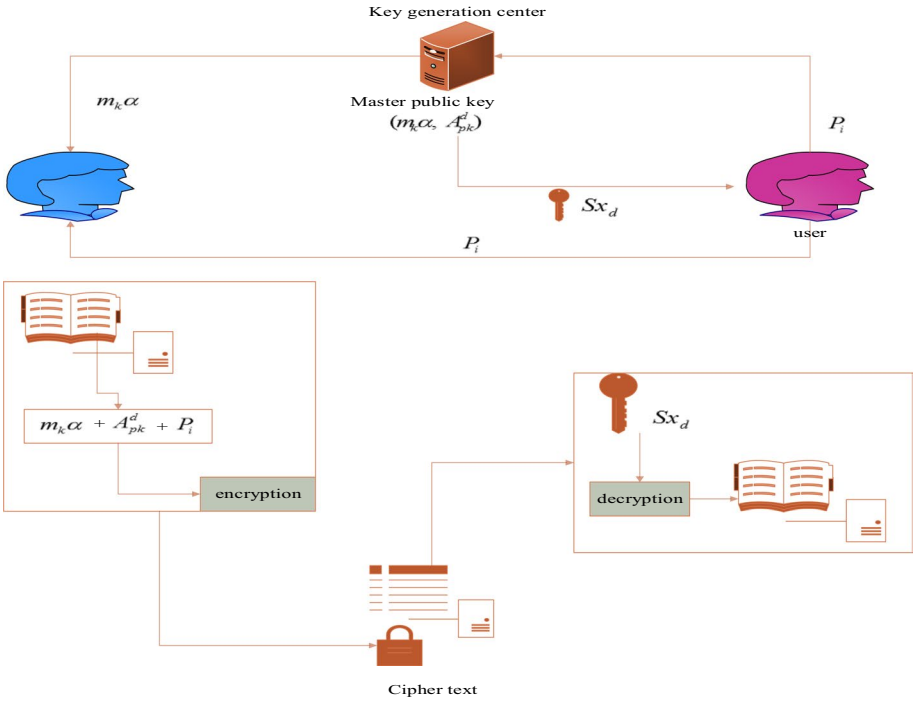

Fig. 4 Process of IBE in the developed model

Blockchain and Multi-Agent Systems (BMAS) for IDS [24], Network-based Cloud Computing (NbCC) [26], and Temporal Convolution Neural Networks (TCNN) for IDS [27].

6.1.1 Accuracy

Accuracy is the degree to which the quantity of calculation, measurement, or specification follows the standard value or correct value as well as the state or quality of being precise or correct. Moreover, accuracy is the degree of closeness to the true value that indicates the effectiveness of the designed model for detecting intrusion. The closeness of the arrangement between measurement results and true value results is accuracy and the measurement of accuracy is obtained using Eq. (5).

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{5}$$

Let, T_p is denoted as true positives of correctly detected and classified positive intrusion, T_n is considered as the true negatives of correctly detected and classified negative intrusion, F_p is represented as false positives of incorrectly detected and classified positive intrusion, and F_n is denoted as false negatives of incorrectly detected and classified negative intrusion. The comparison of the accuracy with existing techniques is detailed in the Table 2.

The accuracy of the proposed BbAB-RNN model is calculated and validated using prevailing methods like BMAS, NbCC, and TCNN approaches. Initially, the BMAS technique gained 98.27% accuracy, the NbCC replica achieved 93.83% accuracy, and the TCNN technique gained 98.96% accuracy for 10 epochs. Moreover, the BMAS technique gained 97%

accuracy, the NbCC replica achieved 88.5% accuracy, and the TCNN technique gained 96.6% accuracy for 20 epochs. Furthermore, the BMAS technique gained 95.7% accuracy, the NbCC replica achieved 86.12% accuracy, and the TCNN technique gained 94% accuracy for 30 epochs. In addition, the BMAS technique gained 93% accuracy, the NbCC replica achieved 83% accuracy, and the TCNN technique gained 92.5% accuracy for 40 epochs. Likewise, the BMAS technique gained 92.3% accuracy, the NbCC replica achieved 81.3% accuracy, and the TCNN technique gained 90% accuracy for 50 epochs. Thus the comparison of accuracy is detailed in Fig. 6.

The developed BbAB-RNN approach has obtained 99.87% for 10 epochs, 99.63% for 20 epochs, 99.12% for 30 epochs, 98.86% for 40 epochs, and 98.32% for 50 epochs. While comparing other models developed technique attained better accuracy which shows the efficiency and security of the designed model.

6.1.2 Recall

Recall is the capability of correctly identifying and detecting intrusion present in the network. The recall is utilized to calculate the number of true positives that are predicted precisely. Also, it is the probability of detecting intrusion using IoT which is measured using Eq. (6). The comparison of the recall with existing techniques is detailed in Table 3.

$$S_N = \frac{T_p}{T_p + F_n} \tag{6}$$

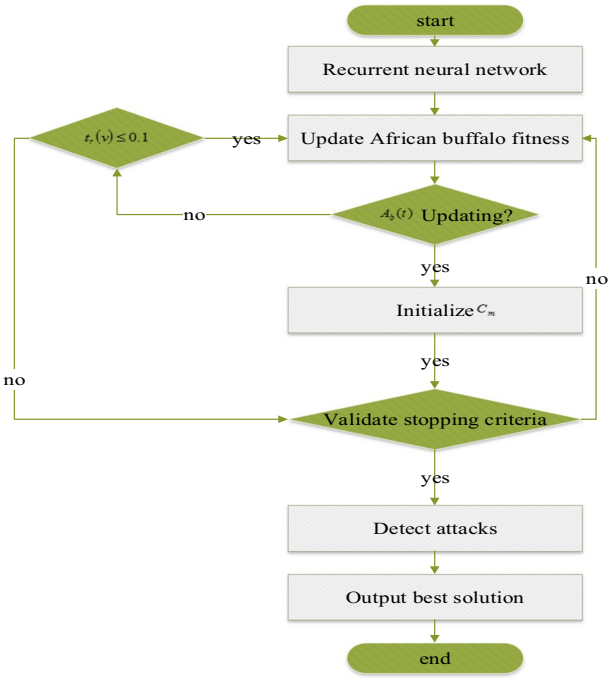

Fig. 5 Process of the AB fitness in RNN to detect intrusion

Table 1 Software and hardware description

Hardware and software	Description
Processor	Intel Xeon E5-2667 v4" 3.20 GHz x16
Labelling Software	labelling
Memory	128 GB
Gpu	NVIDIA Tesla M60
Library	Tensor flow
Operation System	Windows Server 2016 x 64
Programming Language	Python

The recall of the proposed BbAB-RNN model is calculated and validated using prevailing methods like BMAS, NbCC, and TCNN approaches. Initially, the BMAS technique gained 84.14% recall, the NbCC replica achieved 80.48% recall, and the TCNN technique gained 98.51% recall for 10 epochs. Moreover, the BMAS technique gained 82.3% recall, the NbCC replica achieved 78% recall, and the TCNN technique gained 96% recall for 20 epochs. Furthermore, the BMAS technique gained 80% recall, the NbCC replica achieved 76.35% recall, and the TCNN technique gained 93.76% recall for 30 epochs. In addition, the BMAS technique gained 78.7% recall, the NbCC replica achieved 72% recall, and the TCNN technique gained 91.5% recall for 40 epochs. Likewise, the BMAS technique gained 76.3% recall, the NbCC replica achieved 70.76% recall, and the TCNN technique gained 89.42% recall for 50 epochs. Thus the comparison of recall is detailed in Fig. 7.

The developed BbAB-RNN approach has obtained 99.92% for 10 epochs, 99.66% for 20 epochs, 99.25% for 30 epochs, 99.02% for 40 epochs, and 98.84% for 50 epochs. While comparing other models developed technique attained better recall.

6.1.3 Precision

Generally, precision is the amount to which a device or development will repeat the same value. This process has been evaluated for identifying and detecting the number of correct positive estimates which are aligned with the overall positive estimates. Also, precision is the proportion of intrusion detection, which is computed using Eq. (7). The comparison of the precision with existing techniques is detailed in Table 4.

$$P = \frac{T_p}{T_p + F_p} \tag{7}$$

Table 2 Validation of the accuracy

No. of. epochs	Accuracy (%)			
	BMAS	NbCC	TCNN	Proposed
10	98.27	93.83	98.96	99.87
20	97	88.5	96.6	99.63
30	95.7	86.12	94	99.12
40	93	83	92.5	98.86
50	92.3	81.3	90	98.32

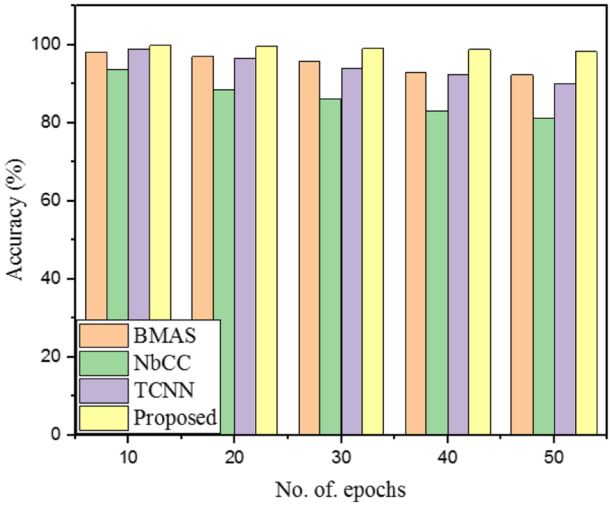

Fig. 6 Comparison of accuracy

The precision of the proposed BbAB-RNN model is calculated and validated using prevailing methods like BMAS, NbCC, and TCNN approaches. Initially, the BMAS technique gained 83.53% precision, the NbCC replica achieved 98.9% precision, and the TCNN technique gained 84.02% precision for 10 epochs. Moreover, the BMAS technique gained 81.5% precision, the NbCC replica achieved 97% precision, and the TCNN technique gained 82.5% precision for 20 epochs. Furthermore, the BMAS technique gained 79.3% precision, the NbCC replica achieved 95.4% precision, and the TCNN technique gained 80.2% precision for 30 epochs. In addition, the BMAS technique gained 77.6% precision, the NbCC replica achieved 93.5% precision, and the TCNN technique gained 78% precision for 40 epochs. Likewise, the BMAS technique gained 75% precision, the NbCC replica achieved 91% precision, and the TCNN technique gained 76.6% precision for 50 epochs. Thus the comparison of precision is detailed in Fig. 8.

The developed BbAB-RNN approach has obtained 99.66% for 10 epochs, 99.32% for 20 epochs, 99.12% for 30 epochs, 98.88% for 40 epochs, and 98.65% for 50 epochs. While comparing other models developed technique attained better precision.

6.1.4 F1-score

It is the degree of the test accuracy and is well-defined as the weight of the harmonic mean of recall and precision test. The calculation is based on the precision and recall

Table 3 Validation of recall

No. of. epochs	Recall (%)			
	BMAS	NbCC	TCNN	Proposed
10	84.14	80.48	98.51	99.92
20	82.3	78	96	99.66
30	80	76.35	93.76	99.25
40	78.7	72	91.5	99.02
50	76.3	70.76	89.42	98.84

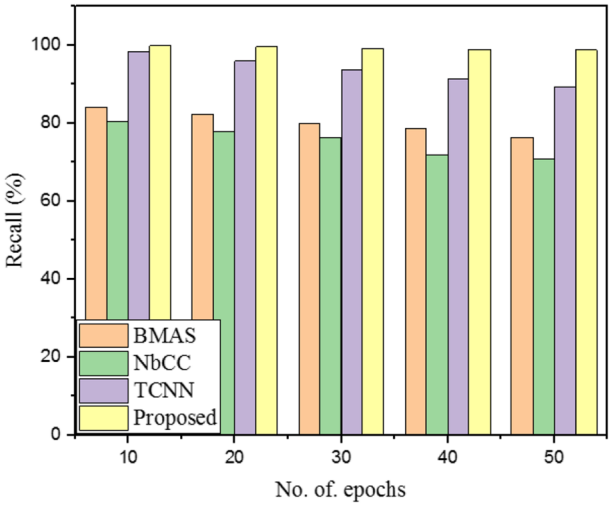

Fig. 7 Comparison of recall

measurements for recognizing the efficiency of detection intrusions which is measured using Eq. (8). The comparison of the F1-score with existing techniques is detailed in Table 5.

$$F - \boldsymbol{measure} = \frac{2 * T_p}{\left(2 * T_p + F_p + F_n\right)} \tag{8}$$

The F1-score of the proposed BbAB-RNN model is calculated and validated using prevailing methods like BMAS, NbCC, and TCNN approaches. Initially, the BMAS technique gained an 83.94% F1 score, the NbCC replica achieved an 89.18% F1 score, and the TCNN technique gained an 88.53% F1 score for 10 epochs. Moreover, the BMAS technique gained 81.6% F1-score, the NbCC replica achieved 86.6% F1-score, and the TCNN technique gained 86.4% F1-score for 20 epochs. Furthermore, the BMAS technique gained a 78% F1 score, the NbCC replica achieved an 81.1% F1 score, and the TCNN technique gained an 84.3% F1 score for 30 epochs. In addition, the BMAS technique gained a 75.7% F1 score, the NbCC replica achieved a 79.7% F1 score, and the TCNN technique gained an 81.34% F1 score for 40 epochs. Likewise, the BMAS technique gained a 72.8% F1 score, the NbCC replica achieved a 77.4% F1 score, and the TCNN technique gained a 79% F1 score for 50 epochs. Thus the comparison of the F1 score is detailed in Fig. 9.

Table 4 Validation of Precision

No. of. epochs	Precision (%)			
	BMAS	NbCC	TCNN	Proposed
10	83.53	98.9	84.02	99.66
20	81.5	97	82.5	99.32
30	79.3	95.4	80.2	99.12
40	77.6	93.5	78	98.88
50	75	91	76.6	98.65

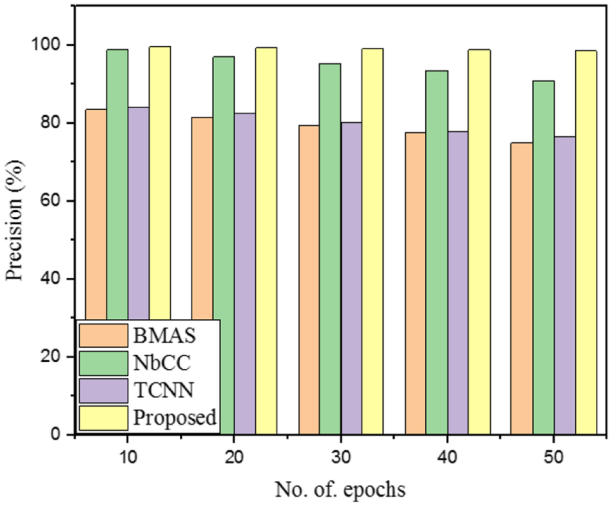

Fig. 8 Comparison of precision

The developed BbAB-RNN approach has obtained 98.38% for 10 epochs, 98.14% for 20 epochs, 97.88% for 30 epochs, 97.65% for 40 epochs, and 97.27% for 50 epochs. While comparing other models developed technique attained a better F1 score.

6.1.5 Execution time

Execution time is the ratio of the quantity of completing time to the quantity of total data-sharing time which is multiplied by 100. The complete time for detecting the intrusion is represented as the execution time. Additionally, the measurement of execution time is calculated using Eq. (9). The comparison of the execution time with existing techniques is detailed in Table 6.

$$\boldsymbol{Execution\_time} = \frac{Ct(t_1)}{T_r(t_1)} \times 100 \tag{9}$$

Let, Ct is denoted as the single task completion time and T_r is represented as the total time essential to complete the task. Moreover, t_1 is represented as a task per second.

The precision of the proposed BbAB-RNN model is calculated and validated using prevailing methods like BMAS, NbCC, and TCNN approaches. Initially, the BMAS technique gained 33s precision, the NbCC replica achieved 19s precision, and the TCNN technique

Table 5 Validation of F1-score

No. of. epochs	F1-score (%)			
	BMAS	NbCC	TCNN	Proposed
10	83.94	89.18	88.53	98.38
20	81.6	86.6	86.4	98.14
30	78	81.1	84.3	97.88
40	75.7	79.7	81.34	97.65
50	72.8	77.4	79	97.27

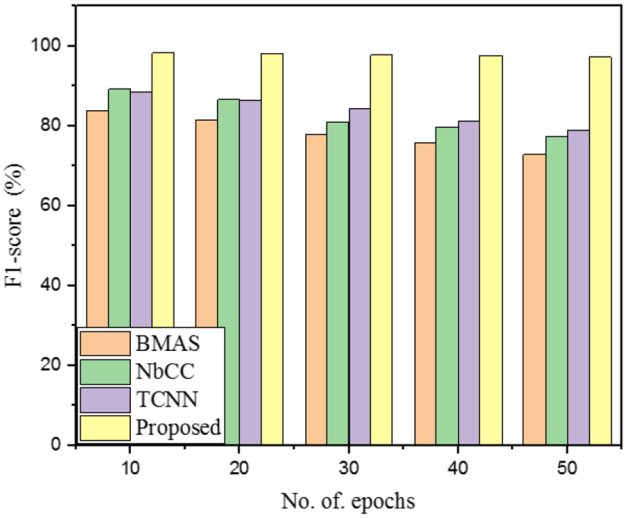

Fig. 9 Comparison of F1-score

gained 29s precision for 10 epochs. Moreover, the BMAS technique gained 36s precision, the NbCC replica achieved 22s precision, and the TCNN technique gained 34s precision for 20 epochs. Furthermore, the BMAS technique gained 39s precision, the NbCC replica achieved 26s precision, and the TCNN technique gained 37s precision for 30 epochs. In addition, the BMAS technique gained 42s precision, the NbCC replica achieved 29s precision, and the TCNN technique gained 40s precision for 40 epochs. Likewise, the BMAS technique gained 46s precision, the NbCC replica achieved 32s precision, and the TCNN technique gained 43s precision for 50 epochs. Thus the comparison of execution time is detailed in Fig. 10.

The developed BbAB-RNN approach has obtained 8s for 10 epochs, 10s for 20 epochs, 12s for 30 epochs, 14s for 40 epochs, and 15s for 50 epochs. While comparing other models developed technique attained less execution time to detect intrusion.

6.2 Discussion

The proposed model of BbAB-RNN has shown good performance by proposed model of BbAB-RNN has shown good performance by attaining the best results in accuracy, recall, F1 score, precision, and execution time. Thus, the developed scheme encrypts the data

Table 6 Validation of execution time

No. of. epochs	Execution time (ms)			
	BMAS	NbCC	TCNN	Proposed
10	33	19	29	8
20	36	22	34	10
30	39	26	37	12
40	42	29	40	14
50	46	32	43	15

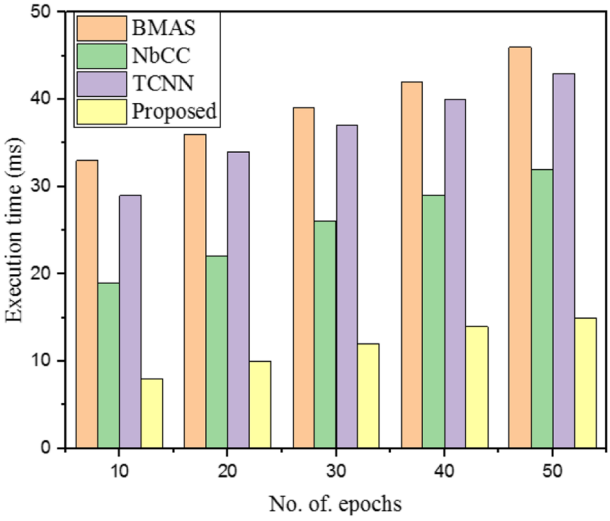

Fig. 10 Comparison of execution time

using IBE and securely stores it in the blockchain using private keys. Finally, detect the intrusion present in the cloud environment based on the threshold value of normal and malware users using ABO. Thus the developed BbAB-RNN technique enhances the performance of detecting intrusion. The overall performance of the designed model with data sizes is detailed in Table 7.

The designed model attain better experimental results which gained high accuracy of 99.23%, high precision of 99.43%, a high recall of 99.64%, and a high F1-score of 98.17% for using 20 bytes. The developed model efficiently locates suspicious activity and issues notifications when it does. To raise the alert and deliver notifications, it maintains an eye on routers, firewalls, important servers, and files. It provides centralized management for the attack correlation. The designed model's scalability safeguards the cloud environment from viruses and infiltration. Velocity, variety, volume, the absence of skilled resources, obtaining valuable knowledge, carrying extensive data, the ambiguity of handling data, information storage, quick retrieval, security, processing, identifying and resolving problems with data quality, expanding big data structures, assessing and choosing big data technologies, and immediate insights are potential challenges and opportunities for scaling up the system to handle larger datasets or more network traffic.

Table 7 Overall performance of the developed model with data sizes

Data sizes (bytes)	Overall performance			
	Accuracy	precision	recall	F1-score
20	98.8	98.76	98.75	98
40	99	98.92	99.24	98.04
60	99.23	99.43	99.64	98.17
80	99.87	99.66	99.92	98.38

7 Conclusions

Cloud computing increases the approach of processing and collecting a huge quantity of data but malicious attacks and intrusion (attacks) cause the development of cloud computing. To overcome this issue proposed BbAB-RNN for accurate detection of intrusion in a cloud environment. Moreover, the IBE scheme is employed to decrypt the data and securely stored it in the blockchain. Here, the intrusion was found by the threshold value of a normal user by continuously monitoring the attacks present in the network using the fitness of AB. Finally, predict the intrusion present in the cloud environment. Finally, attained results of the designed model are validated with BMAS, NbCC, and TCNN approaches. The proposed technique attained 99.87% accuracy which is 4%, 3.5%, and 2% higher than BMAS, NbCC, and TCNN models. Additionally, the developed model gained a recall rate is 99.92% which is 5.2%, 6%, and 3% higher than BMAS, NbCC, and TCNN models. Likewise, the developed model gained a precision rate of 99.66% which is 3.2%, 7%, and 5.4% higher than the BMAS, NbCC, and TCNN models. While comparing other models, the designed model requires less execution time of 8ms to encrypt the data and detect the attacks with a short time duration. Thus the designed model can detect intrusion and secure the data in the cloud environment. In the future, hybrid optimization with an improved deep learning model with enhances the performance of security and overcome data complexity issue in a cloud environment. Also, optimization with a proxy re-encryption scheme will provide better security than other cryptographic models.

Code availability Not applicable

Funding Not applicable

Data availability https://www.kaggle.com/datasets/piyushrumao/malware-executable-detection?select=uci_malware_detection.csv

Declarations

Human and Animal Rights This article does not contain any studies with human or animal subjects performed by any of the authors.

Informed Consent Informed consent does not apply as this was a retrospective review with no identifying patient information.

Consent to participate Not applicable

Consent for publication Not applicable

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Perwej Y et al (2019) The internet of things (IoT) and its application domains. Int J Comput Appl 182(49):36–49
2. Rath M, Pattanayak B (2018) Technological improvement in modern health care applications using Internet of Things (IoT) and proposal of novel health care approach. International Journal of Human Rights in Healthcare

3. Khoa TA et al (2020) Designing efficient smart home management with IoT smart lighting: a case study. Wireless Communications and Mobile Computing 2020
4. Al Makdi K, Sheldon FT, Hussein AA (2020) Trusted Security Model for IDS Using Deep Learning. 2020 3rd International Conference on Signal Processing and Information Security (ICSPIS). IEEE
5. Anthi E et al (2019) A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J 6(5):9042–9053
6. Nathiya T, Suseendran G (2019) An effective hybrid intrusion detection system for use in security monitoring in the virtual network layer of cloud computing technology. Data management, analytics and innovation. Springer, Singapore, pp. 483-497
7. Brown IL (2018) An appropriate technology system for emergent beekeepers: Field testing and development towards implementation. Diss. University of Johannesburg (South Africa)
8. Sharafaldin I et al (2019) Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. 2019 International Carnahan Conference on Security Technology (ICCST). IEEE
9. Jayasinghe U et al (2019) TrustChain: A privacy preserving blockchain with edge computing." Wireless Communications and Mobile Computing 2019
10. Borangiu T et al (2019) Digital transformation of manufacturing through cloud services and resource virtualization. Comput Ind 108:150–162
11. Alashhab ZR et al (2021) "Impact of coronavirus pandemic crisis on technologies and cloud computing applications." Journal of Electronic. Sci Technol 19(1):100059
12. Nguyen DC et al (2021) Federated learning meets blockchain in edge computing: Opportunities and challenges. IEEE Internet of Things Journal
13. Zhang K, Jacobsen H-A (2018) Towards Dependable, Scalable, and Pervasive Distributed Ledgers with Blockchains (Technical Report)
14. Velmurugadass P et al (2021) Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm. Materials Today: Proc 37:2653–2659
15. Datta P et al (2020) A secured smart national identity card management design using blockchain. 2020 2nd international conference on advanced information and communication technology (ICAICT). IEEE
16. Kumar R, Bhalaji N (2021) Blockchain based chameleon hashing technique for privacy preservation in E-governance system. Wirel Pers Commun 117(2):987–1006
17. Kerr M, Han F, van Schyndel R (2018) A blockchain implementation for the cataloguing of cctv video evidence. 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE
18. Das S, Namasudra S (2022) A Novel Hybrid Encryption Method to Secure Healthcare Data in IoT-enabled Healthcare Infrastructure. Comput Electr Eng 101:107991
19. Arif YM et al (2020) Blockchain-based data sharing for decentralized tourism destinations recommendation system. Int J Intel Eng Syst 13(6):472–486
20. Firdaus M, Rhee K-H (2021) On blockchain-enhanced secure data storage and sharing in vehicular edge computing networks. Appl Sci 11(1):414
21. Lee JY (2019) A decentralized token economy: How blockchain and cryptocurrency can revolutionize business. Bus Horiz 62(6):773–784
22. Albanese G et al (2020) "Dynamic consent management for clinical trials via private blockchain technology." Journal of Ambient Intelligence and Humanized. Computing 11(11):4909–4926
23. Swetha MS et al (2020) Blockchain enabled secure healthcare Systems. 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT). IEEE
24. Khraisat A et al (2019) A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics 8(11):1210
25. Liang C et al (2020) Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics 9(7):1120
26. Alkadi O et al (2020) A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet Things J 8(12):9463–9472
27. Thilagam T, Aruna R (2021) Intrusion detection for network based cloud computing by custom RC-NN and optimization. ICT Express 7(4):512–520
28. Derhab A et al (2020) Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering.Wireless Communications and Mobile Computing 2020.
29. HaddadPajouh H et al (2018) A deep recurrent neural network based approach for internet of things malware threat hunting. Futur Gener Comput Syst 85:88–96

30. Mansour RF (2022) Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment. Sci Rep 12(1):12937
31. Belhadi A et al (2023) Group intrusion detection in the Internet of Things using a hybrid recurrent neural network. Clust Comput 26(2):1147–1158