IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 April 2024, accepted 7 June 2024, date of publication 17 June 2024, date of current version 24 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3414998

RESEARCH ARTICLE

Design of an Anomaly Detection Framework for Delay and Privacy-Aware Blockchain-Based Cloud Deployments

A. VENKATA NAGARJUN AND SUJATHA RAJKUMAR, (Senior Member, IEEE)
School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Corresponding author: Sujatha Rajkumar (sujatha.r@vit.ac.in)

This work was supported by Vellore Institute of Technology, Vellore, India.

ABSTRACT Cloud-based deployments face increasing threats from various types of attacks, necessitating robust anomaly detection frameworks to safeguard against potential security breaches. Existing solutions, such as RSSI, GTM, and APG, though effective to a certain extent, exhibit limitations in terms of precision, accuracy, and scalability. To address these shortcomings, this paper proposes a novel anomaly detection framework that integrates multimodal feature analysis, deep learning models, and QoS-aware sidechains to enhance the prediction accuracy of cloud attacks and optimize blockchain-based cloud installations. By maximizing feature variance across different sample types and leveraging advanced deep learning techniques, the proposed approach significantly outperforms conventional methods in terms of precision, accuracy, recall, and AUC performance. Furthermore, the framework demonstrates superior efficiency in block mining delay, energy consumption, and throughput, making it highly suitable for real-time cloud attack prediction scenarios. The proposed methodology represents a significant advancement in anomaly detection and cloud security, offering a comprehensive solution for addressing challenges in blockchain-based cloud deployments. Thus, the proposed anomaly detection framework employs both Deep Learning and Blockchain technologies. Using Recurrent Neural Networks (RNN) with Convolutional Neural Networks (CNN), the system examines system logs and identifies unusual behavior patterns associated with different attacks. Using Blockchain technology, the framework ensures the transparency and integrity of system logs, and Deep Learning models provide precise and timely anomaly detection. The decision to combine Deep Learning and Blockchain technology is justified by the merits of each technique. The distributed, immutable ledger provided by blockchain technology makes it impossible to tamper with system logs and ensures the accuracy of anomaly detection. While, deep learning models, have exceptional pattern recognition abilities and can adapt to changing attack methods, resulting in high precision, accuracy, recall, and AUC metrics. Analyses of experimental data demonstrate that the proposed framework is effective. The framework achieves impressive performance metrics, such as low delays, 98.5% precision, 99.4% accuracy, 98.3% recall, and 99.2% Area Under the Curve (AUC).

INDEX TERMS Anomaly detection, attacks, blockchain, cloud computing, deep learning, privacy.

I. INTRODUCTION

Cloud computing has revolutionized how businesses store, process, and access their data by providing solutions that are adaptable, scalable, and cost-effective. Nonetheless, the growing reliance on cloud-based deployments has raised significant security concerns. Cloud environments are susceptible to numerous cyber threats, including Man-in-the-Middle (MITM), Finney, Sybil, Distributed Denial of Service (DDoS), and Cryptojacking attacks, due to their dynamic and distributed nature. These attacks can compromise the availability, confidentiality, and integrity of cloud-based

The associate editor coordinating the review of this manuscript and approving it for publication was Peter Langendoerfer.

systems, resulting in severe repercussions for organizations and users [1], [2], [3].

To address these security issues [4], [5], [6], effective anomaly detection frameworks that can identify and prevent such attacks in cloud deployments are essential. In addition, these frameworks must take into account the delays introduced by the detection mechanisms and ensure privacy sensitivity to safeguard sensitive user datasets & samples. This paper introduces a novel anomaly detection framework designed specifically to address delay and privacy awareness in blockchain-based cloud deployments, thereby providing robust protection against the aforementioned attacks.

This work has numerous and significant applications in the context of securing cloud computing environments. The framework can be utilized in a variety of scenarios, including enterprise use of public cloud services and private cloud deployments within organizations. By combining mechanisms for detecting anomalies with blockchain technology, the framework ensures the integrity and transparency of system logs, preventing tampering and unauthorized modifications. In addition, the use of deep learning techniques enables precise and timely detection of anomalies, thereby improving the overall security posture of cloud deployments.

The proposed framework for anomaly detection combines the benefits of blockchain and deep learning technologies. With its decentralized and immutable ledger, blockchain provides an immutable record of system logs and events. This not only improves the integrity of the detection process but also transparently enables auditing and accountability. Deep learning techniques, including Recurrent Neural Networks and Convolutional Neural Networks, utilize their pattern recognition capabilities to analyze system logs and identify anomalous behavior associated with attacks.

The complementary nature of blockchain and deep learning justifies the decision to include both components in the framework. Blockchain technology guarantees the honesty and openness of detected anomalies, making them resistant to manipulation and tampering. On the other hand, deep learning models can adapt and learn from evolving attack techniques, enabling accurate and efficient detection. The combination of these technologies allows the framework to achieve exceptional Precision, Accuracy, Recall, and Area Under the Curve (AUC) performance.

Extensive experimental evaluations have been conducted in order to confirm the efficacy of the proposed framework. MITM, Finney, Sybil, DDoS, and Cryptojacking attacks are detected with remarkable Precision, Accuracy, Recall, and AUC. In addition, the framework exhibits low delay, enabling real-time threat detection and response. These findings highlight the framework's robustness and efficacy, establishing it as a valuable tool for enhancing the security of blockchain-based cloud deployments.

In conclusion, this paper examines the pressing need for robust anomaly detection frameworks in cloud computing environments. The proposed framework offers effective protection against MITM, Finney, Sybil, DDoS, and Cryptojacking attacks by emphasizing delay and privacy-consciousness in blockchain-based cloud deployments. Integrating blockchain and deep learning technologies ensures the detection process's integrity, transparency, and precision. The experimental evaluations confirm the exceptional performance of the framework, making it a promising solution for securing cloud deployments and mitigating evolving cyber threats.

A. MOTIVATION AND CONTRIBUTIONS OF THIS TEXT

This paper is motivated by the increasing adoption of Cloud Computing and the security challenges it presents to organizations. Cloud deployments provide numerous advantages, such as scalability, cost-effectiveness, and adaptability. The dynamic nature of cloud environments, however, makes them susceptible to various cyber threats, such as MITM, Finney, Sybil, Distributed DDoS, and Cryptojacking attacks. These attacks can compromise the availability, confidentiality, and integrity of cloud-based systems, resulting in severe repercussions for organizations and users. Consequently, there is an urgent need to develop effective frameworks for anomaly detection that can detect and mitigate these attacks in a delay-aware and privacy-preserving manner.

B. OBJECTIVES

The following are the primary objectives of this paper:

1) TO DESIGN A FRAMEWORK FOR ANOMALY DETECTION

The purpose of this paper is to propose a novel anomaly detection framework for blockchain-based cloud deployments that are delay- and privacy-aware. This framework will effectively detect and mitigate MITM, Finney, Sybil, DDoS, and Cryptojacking attacks, thereby enhancing the cloud environment's overall security posture.

2) TO ATTAIN SUPERIOR PERFORMANCE METRICS

The objective of this paper is to detect anomalies with exceptional Precision, Accuracy, Recall, and Area Under the Curve (AUC). The goal is to create a framework that can accurately and reliably identify potential threats while minimizing false positives and false negatives.

To guarantee minimal delay and real-time detection, the purpose of this paper is to address the delay caused by anomaly detection mechanisms. By designing the framework to operate with low latency, it enables real-time detection and response to potential threats, enabling organizations to quickly mitigate attacks and reduce their impact for different scenarios.

3) TO MAINTAIN CONFIDENTIALITY IN CLOUD ENVIRONMENTS

When sensitive data is involved, the protection of privacy is of paramount importance in cloud computing. The purpose of this paper is to develop a privacy-aware framework that

safeguards user data throughout the entire process of anomaly detection. This is accomplished through the utilization of techniques such as federated learning and other privacy-preserving mechanisms.

C. CONTRIBUTIONS

This paper's contributions can be summarized as follows:

1) INNOVATIVE ANOMALY DETECTION ARCHITECTURE

The paper presents a novel framework for anomaly detection that combines the benefits of blockchain and deep learning technologies. This integration ensures the integrity, transparency, and accuracy of the detection process, enabling robust protection in cloud deployments against a variety of attacks.

The proposed framework achieves impressive performance metrics, including high precision, accuracy, recall, and area under the curve (AUC). These metrics validate the framework's ability to accurately identify and mitigate MITM, Finney, Sybil, DDoS, and Cryptojacking attacks, thereby enhancing the security of cloud environments.

2) LOW LATENCY AND REAL-TIME DETECTION

The framework overcomes the problem of latency by operating with low latency. This enables real-time detection and response to potential threats, reducing the time between anomaly detection and mitigation actions, thereby minimizing the impact of attacks on cloud deployments.

3) PRIVACY-PRESERVING MECHANISMS

The framework incorporates privacy-aware techniques, ensuring that sensitive user data is safeguarded throughout the entire process of anomaly detection. Using federated learning or other privacy-preserving mechanisms, the framework achieves a higher level of privacy while preserving detection precision levels.

By addressing these objectives and making these contributions, the paper significantly advances the field of anomaly detection for blockchain-based cloud deployments that are sensitive to delay and privacy concerns. It contributes to the overall advancement of cloud computing security by providing a valuable solution for organizations seeking to increase the security and integrity of their cloud environments.

II. LITERATURE SURVEY

Current models for delay and privacy-aware blockchain-based cloud deployment anomaly detection have substantially enhanced the security and integrity of cloud computing environments. These models use a variety of methodologies and techniques to identify and counteract attacks like Man-in-the-Middle (MITM), Finney, Sybil, Distributed Denial-of-Service (DDoS), and Cryptojacking. This article intends to provide a summary of notable contemporary models and their primary characteristics like the use of the Received Signal Strength Indicator (RSSI) as an efficient metric for anomaly analysis [7], [8], [9].

A common strategy in anomaly detection models is the use of machine learning techniques, particularly deep learning algorithms [10], [11], [12], [13]. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), two types of deep learning models, have proven to be exceptionally effective at learning patterns and identifying anomalies in system logs and events. Because they efficiently capture complex relationships and temporal dependencies, these models are ideally suited for detecting anomalous behaviour in cloud deployments [14], [15], [16].

The incorporation of blockchain technology into the reviewed models is an additional feature of significance for different scenarios via the Game Theoretic Model (GTM) process [17], [18], [19], [20]. The blockchain's decentralized, immutable ledger improves the accuracy and accessibility of system logs and events. By incorporating blockchain into anomaly detection frameworks, the models can guarantee the detection process's accuracy and tamper-resistance characteristics [21], [22], [23], [24]. Moreover, system log auditing can be made secure and transparent using the distributed consensus mechanisms and smart contracts of blockchain technology process [25], [26], [27], [28].

Collaborative intrusion detection systems have also been studied in an effort to safeguard privacy in cloud environments [29], [30], [31], [32]. These systems employ federated learning techniques, wherein models are trained locally on distinct cloud instances, and only aggregated updates are shared to create an augmented global model process. This method enables anomaly detection without compromising the privacy of sensitive data because no instance of the system is required to share data with a central authority or with some other cases & scenarios [33], [34], [35], [36], [37], [38].

In order to solve the problem of safe and transparent data migration between cloud services, some models have also proposed utilizing blockchain technology for verification and integrity characteristics like the use of the Adversarial Perturbation Generation (APG) process [39], [40], [41], [42]. Using smart contracts and distributed ledger capabilities, these models guarantee the secure transfer of data while minimizing the need for manual inspections and lowering the associated overhead. Integrating blockchain technology into data migration processes adds a layer of trust that enhances the operation's overall security levels [43], [44], [45].

Even though current models have made significant strides in anomaly detection for delay- and privacy-aware blockchain-based cloud deployments, there is still room for improvements [46], [47], [48]. A few challenges include ensuring the scalability and compatibility of blockchain technology across diverse cloud architectures, ensuring the performance and efficiency of deep learning algorithms to handle large-scale cloud environments, and addressing potential tradeoffs between privacy preservation and detection accuracy in collaborative intrusion detection systems [49], [50].

Current models for delay and privacy-aware anomaly detection in blockchain-based cloud deployments have demonstrated promise for enhancing the security and integrity of cloud computing environments. Using techniques such as deep learning, blockchain integration, collaborative intrusion detection, and secure data migration, these models provide efficient methods for detecting and thwarting attacks in cloud deployments while protecting privacy and guaranteeing open operations. Future research should focus on resolving outstanding issues and enhancing the capacity of these models to meet the shifting security requirements of cloud computing scenarios.

III. PROPOSED DESIGN OF AN ANOMALY DETECTION FRAMEWORK FOR DELAY AND PRIVACY-AWARE BLOCKCHAIN-BASED CLOUD DEPLOYMENTS

Based on the review of recently proposed blockchain-based models for privacy-aware computing, it can be observed that these models either have a high complexity of deployment or cannot be scaled to large-scale cloud use cases due to their low to moderate efficiency levels. To overcome these issues, this section discusses the design of an anomaly detection framework for delay & privacy-aware blockchain-based cloud deployments. As per Figure 1, the proposed model employs both Deep Learning and Blockchain technologies for enhancing privacy while maintaining high QoS under real-time cloud deployments. Using Recurrent Neural Networks (RNN) with Convolutional Neural Networks (CNN), the system examines system logs and identifies unusual behavior patterns associated with different attacks. Using Blockchain technology, the framework ensures the transparency and integrity of system logs, and Deep Learning Models provide precise and timely anomaly detection operations. The design for both of these models is discussed in separate sub-sections of this text, this will assist readers to deploy these models for their context-specific use cases.

A. DESIGN OF THE DEEP LEARNING MODEL FOR THE IDENTIFICATION OF UNUSUAL BEHAVIOR PATTERNS

The proposed model uses an augmented fusion of *Long-Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Auto Encoders (AEs) for representing cloud logs into multidomain feature sets. The cloud logs extracted for this purpose, include, Timestamp, Source IP Address, Destination IP Address, Source Port, Destination Port, Protocol (e.g.; TCP; UDP), User ID, Username, Request/Command, HTTP Method (e.g.; GET; POST), URL, Request Headers, Request Body, Response Code, Response Headers, Response Body, Resource Accessed, Resource Type, Resource ID, Request Size, Response Size, Device/Host Information, Hostname, Operating System, Device Type, Geolocation (IP geolocation data), Error Messages, Exception Details, Log Type/Category, Log Severity Level, Log Source, Log Message, Log ID/Event ID, Authentication Method, Authentication Success/Failure, Session ID, Session Duration, CPU Usage, Memory Usage, Disk Usage, Network Traffic (Bytes In/Out), Database Queries, Database Response Time, API Endpoint, API Request/Response, DNS Requests, DNS Response Time, SSL/TLS Handshake Time, Firewall Events, Intrusion Detection System (IDS) Alerts.* These parameters are frequently gathered during cloud anomaly detection and play vital roles in the attack detection process. The use cases for each of these metrics are discussed as follows.

B. TIMESTAMP

The timestamp denotes the date and time of the occurrence of an event or record entry. It helps organize and sequence events for the purposes of analysis and correlations.

C. SOURCE IP ADDRESS AND DESTINATION IP ADDRESS

These parameters designate the IP addresses of the network connection's source and destination. By monitoring these IP addresses, anomalies like suspicious or unauthorized access attempts can be identified for different use cases.

D. SOURCE PORT AND DESTINATION PORT

Ports are numeric identifiers used to distinguish between various network services. Monitoring these ports enables the identification of anomalous port usage or unanticipated connections, which could indicate an augmented group of attacks.

E. PROTOCOL

This parameter specifies the network protocol used during a communication session, such as TCP or UDP. Protocol analysis can disclose uncommon or unauthorized protocol usage. These parameters designate the user affiliated with an activity or event. By monitoring user activity, anomalies such as attempts at unauthorized access or suspicious behavior can be identified.

F. REQUEST/COMMAND, HTTP METHOD, URL, REQUEST HEADERS, AND REQUEST BODY

These parameters collect data regarding web requests made by users or automated processes. Analyzing these components facilitates the detection of malicious or anomalous HTTP requests, such as attempts at SQL injection or unauthorized access.

G. RESPONSE CODE, RESPONSE HEADERS, AND RESPONSE BODY

Provide information regarding the server's response to a request process. Monitoring them enables the identification of abnormal or unexpected server responses, which may indicate an attack or misconfiguration of the systems. These parameters indicate the specific resources or assets accessed during an event. Monitoring resource access enables the detection of unauthorized access attempts or suspicious activity involving vital assets.

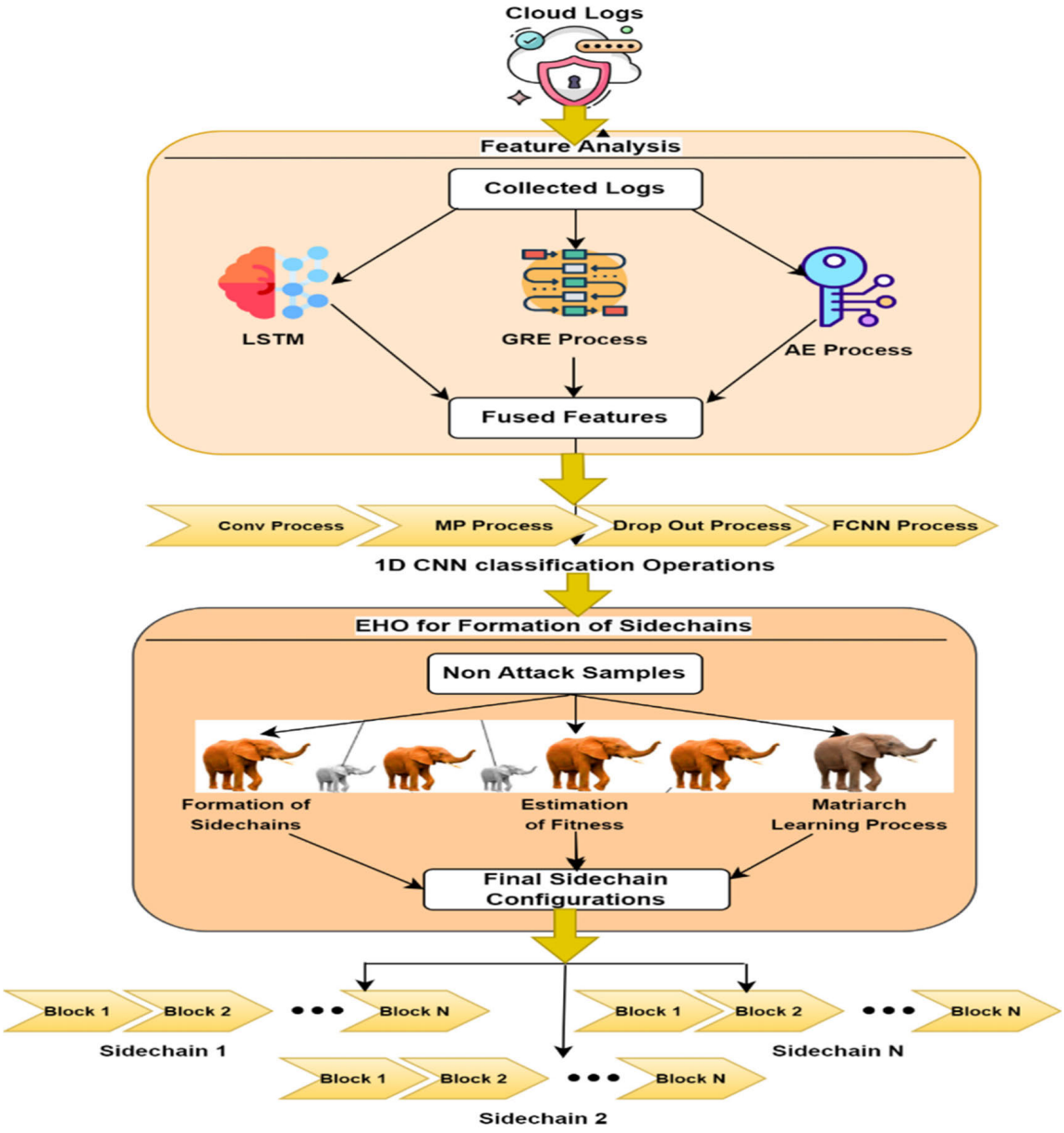

FIGURE 1. Design of the proposed security model for blockchain-based cloud deployments.

H. REQUEST MAGNITUDE AND RESPONSE SIZE

These parameters specify the magnitude of the incoming request and the outgoing response. Sizes that are atypically large or small may indicate data exfiltration or denial-of-service attacks. These parameters provide information regarding the devices or hosts participating in an event. Monitoring them enables the identification of anomalies, such as unrecognized or compromised devices attempting to access the systems.

I. GEOLOCATION

This supplies the physical location associated with an IP address. It can detect suspect activities emanating from unknown or known malicious locations. These parameters relate to logs and monitoring datasets & samples. Analyzing logs enables the identification of abnormal or unexpected events, errors, or system misconfigurations that may indicate an ongoing attack or breach.

J. AUTHENTICATION METHOD, AUTHENTICATION SUCCESS/FAILURE, SESSION ID, AND SESSION DURATION

Monitoring them permits the identification of unauthorized access attempts, suspicious logon behavior, and session hijacking parameters offering insight into user authentication and session management.

K. CPU USAGE, MEMORY USAGE, AND DISK USAGE

These parameters quantify the system's resource utilization. Monitoring them enables the identification of anomalous

resource consumption patterns, which may indicate a malware infection or denial-of-service attacks. These parameters measure the amount of network traffic that was exchanged during an event. Indicators of potential assaults, such as data exfiltration or botnet activity, are sudden increases or anomalies in network traffic packets.

L. DATABASE QUERIES AND DATABASE RESPONSE TIME

These parameters describe database interactions. Monitoring them enables the identification of anomalous or suspicious database activity, such as attempts at SQL injection or unauthorized queries.

M. API ENDPOINT, API REQUEST/RESPONSE

These parameters pertain to API (Application Programming Interface) interactions. Monitoring API usage enables the detection of anomalies, such as excessive or unauthorized API calls, which may indicate an attack or misapplications. These parameters monitor DNS (Domain Name System) activity, which includes DNS requests and response delays. Unusual DNS requests or lengthy response periods may be indicative of DNS-based attacks or malicious domain resolutions.

N. SECURITY LAYERS

This parameter measures the amount of time required to establish a secure SSL/TLS connection. Monitoring SSL/TLS handshake duration enables the detection of anomalies, such as sluggish handshakes or failed encryption, which may indicate a security compromise or attempted downgrade attacks. These parameters collect information regarding firewall events and intrusion detection system (IDS) alerts. Monitoring them enables the detection of anomalous network traffic, violations of policy, and known attack patterns.

O. FEATURE EXTRACTION PROCESS

To classify these parameters into different attack types, they are represented into multidomain features. This is done by individual extraction of LSTM, GRU & Auto Encoder features. The LSTM Model can be observed in Figure 2, where the collected input features are given to an efficient variance maximization unit, which is represented via equation 1,

$$i = var\left(x_{in} * U^{i} + h_{t-1} * W^{i}\right) \tag{1}$$

where x_{in} is the collection of input features, $U\&W$ represents different constants of the input LSTM layer, while h represents a kernel matrix, which is tuned by the LSTM Model for maximization of feature variance levels. This variance level is estimated via equation 2,

$$f_b = \frac{\left(\sum_{i=1}^{N}\left(x_i - \sum_{j=1}^{N}\frac{x_j}{N}\right)^2\right)}{N+1} \tag{2}$$

where N is the count of total cloud log parameters which are collected for anomaly analysis.

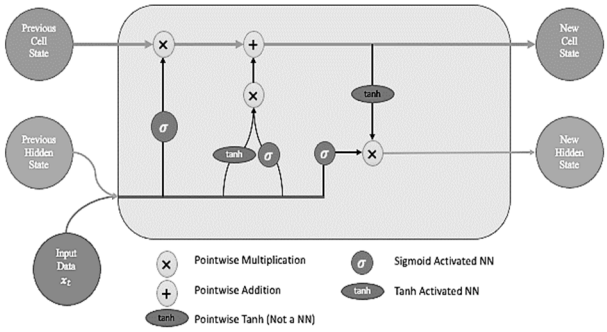

FIGURE 2. Internal design of the LSTM Process.

Similar to the input vector, two more variance-level features are estimated via equations 3 & 4 as follows,

$$f = var\left(x_{in} * U^{f} + h_{t-1} * W^{f}\right) \tag{3}$$

where U & W are different constants than the previous evaluations.

$$o = var\left(x_{in} * U^{o} + h_{t-1} * W^{o}\right) \tag{4}$$

These feature sets are fused to obtain a convolutional feature via equation 5,

$$C = tanh(x\_in * U\hat{}g + h\_(t-1) * W\hat{}g) \tag{5}$$

Using these feature sets, the temporal output feature vector is calculated via equation 6,

$$T = var\left(f_t * x_{in}\,(t-1) + i * C\right) \tag{6}$$

Based on the temporal output, kernel features are updated via equation 7,

$$h_{out} = \tanh\left(T\right) * o \tag{7}$$

The output kernel levels are continuously updated, till the condition represented by equation 8 is satisfied, which can be evaluated as follows,

$$\frac{h_{out}\,(New)}{h_{out}(Old)} \cong 1,\ with\ an\ error\ of \pm 0.1 \tag{8}$$

Once this condition is satisfied, the model converges, and temporal features are output by the model for anomaly analysis.

Similar to this process, the GRU Model also estimates these features as per Figure 3, and the updated kernel metric is fused with the temporal output metric to estimate two GRU constants via equations 9 & 10 as follows,

$$GRU1 = var\left(W_z * [h_{out} * T]\right) \tag{9}$$

$$GRU2 = var\left(W_r * [h_{out} * T]\right) \tag{10}$$

Based on these metrics, the GRU output feature is estimated via equation 11,

$$GRU\ (out) = (1 - GRU1) * ht' + GRU2 * h_{out} \tag{11}$$

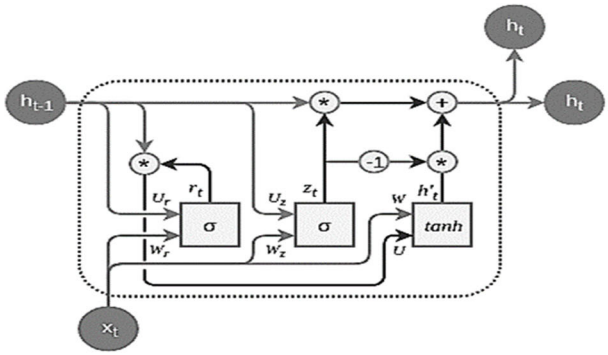

FIGURE 3. Design of the GRU process for evaluation of temporal feature sets.

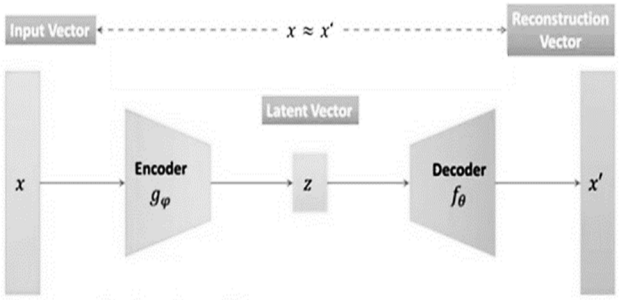

FIGURE 4. Design of the auto encoder process.

While the kernel metric is updated via equation 12,

$$h'_{out} = tanh\,(W * [GRU2 * h_{out} * T]) \tag{12}$$

The GRU Process is also evaluated for Multiple Iterations, till the condition in equation 8 is satisfied, which represents feature convergence operations.

Both LSTM & GRU Features perform temporal analysis, which assists in the identification of chronologically changing feature sets. To further augment these features, an encoding process is utilized using Auto Encoders (AEs), which can be observed from Figure 4 and assists in representing collected cloud logs into non-linear features via equation 13,

$$Z = LReLU\,(W_e * X + b_e) \tag{13}$$

Here X is the collection of input logs, W_e is the weight matrix, and b_e is the bias vector applied element-wise to the linear transformation process, while LReLU is an efficient Leaky Rectilinear Unit, which is represented via equation 14,

$$LReLU\,(x) = l_a * x,\ when\ x < 0,\ else\ x \tag{14}$$

where l_a is an activation constant, which retains positive feature sets. These features are given to an efficient combination of tracing & covariance layers, which is represented via equation 15,

$$AE\,(out) = trace\,(cov\,(Z)) \tag{15}$$

where $cov(Z)$ is the value of covariance for latent space representations of Z, this is estimated by equation 16, while

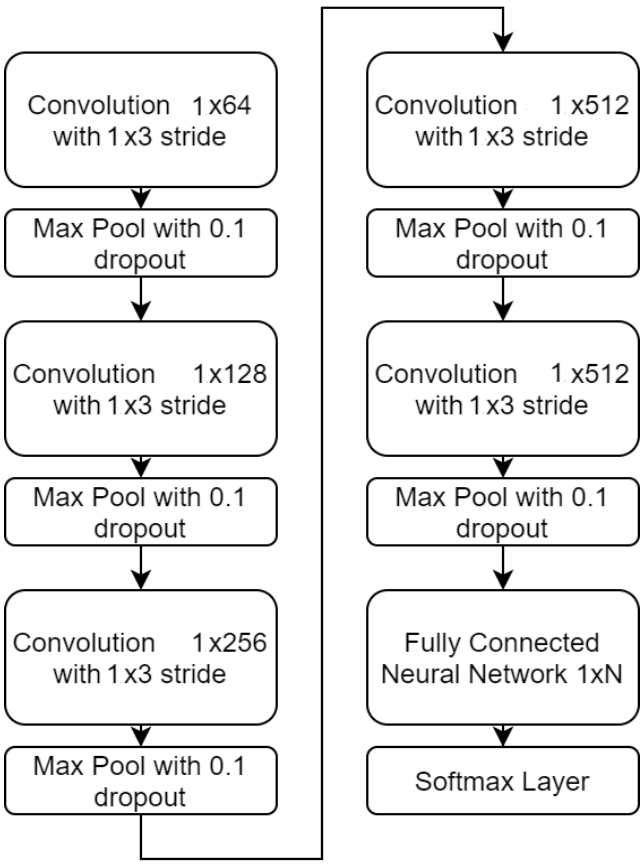

FIGURE 5. Design of the customized 1D CNN Model for identification of cloud anomalies.

$trace\,(cov\,(Z))$ represents the diagonal element sum which is estimated via equation 17 as follows,.

$$cov\,(Z) = \frac{1}{N} * (Z - mean\,(Z)) * (Z - mean\,(Z))' \tag{16}$$

$$trace\,(X) = \Sigma_i\,(X\,[i, i]) \tag{17}$$

This sum estimated for diagonal elements of the given matrix represents the trace levels, which are sum values along the principal diagonals.

P. ATTACK IDENTIFICATION PROCESS

All these features are fused to form an augmented Cloud Anomaly Feature Vector (CAFV), which represents an effective representation of cloud logs. These features are given to an effective 1D Convolutional Neural Network (1D CNN), which is represented in Figure 5, and contains an augmented set of Multiple Convolutional, Max Pooling, Dropout and Fully Connected layers.

As per this layer design, the collected feature sets are represented via Convolutional operations via equation 18,

$$Conv\,(CAFV\,(i)) = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} CAFV\,(i-a) * LReLU\left(\frac{m+2a}{2}\right) \tag{18}$$

where m, a are the sizes for windows & strides in individual Convolutional layers. This process is repeated for multiple Max Pooling, and Dropout layers. Results of the final features are given to an efficient SoftMax based activation layer, which assists in identification of anomaly classes via equation 19,

$$c\,(out) = SoftMax\left(\sum_{i=1}^{N_f} f_i * w_i + b\right) \tag{19}$$

where Nf are the final features extracted at the final layer, while $f, w\&b$ are the feature values, their respective weights & biases. This process is used to estimate different attacks including Man-in-the-Middle (MITM), Finney, Sybil, Distributed Denial of Service (DDoS), and Cryptojacking attacks. Due to which the Cloud Model is highly secure, and can be used for the identification of real-time attacks. To further strengthen the security of this model, an efficient QoS-aware blockchain model was used, which assists in enhancing privacy levels. The design of this model can be observed in the next section of this text.

Q. DESIGN OF THE PROPOSED QOS-AWARE BLOCKCHAIN-BASED MODEL FOR ENHANCING PRIVACY LEVELS

After the identification of attacks, the secure non-attack packets are stored using blockchains. Algorithm 1 shows anomaly detection by integrating deep learning and blockchain. This blockchain stores the following information about the packets,

- IP Addresses of Source & Destination Nodes
- Geolocation of these Nodes
- Packet Headers
- Packet Information Sets (Request & Responses)
- Timestamp for the Packets
- Hash of the Previous Blocks
- Sidechain Information Sets
- Nonce Value Levels

All these information sets assist the cloud node to representing the packets. But as the number of blocks increases, the delay needed to add blocks also increases, which reduces the QoS of the cloud deployments. To overcome this issue, an efficient Elephant Herding Optimizer (EHO) is used, which assists in managing sidechains. These sidechains are small length blockchains, which reduce the mining effort while maintaining higher QoS levels. This is done via the following process,

- The EHO Model, initially generates an iterative set of NH Herds.
- Each of these Herds segregates the current blockchain into 2 parts, where the length of one part is estimated via equation 20,

$$NSC = STOCH\left(LH * \frac{N}{2}, \frac{N}{2}\right) \tag{20}$$

where $STOCH$ is an efficient Markovian process used to generate stochastic numbers, LH represents Herd Learning Rate, while N represents length of the current blockchain which is currently being used for addition of new blocks.

- Based on this process, the model segregates current chain into $NSC\ \&N - NSC$ parts.
- The smaller part is used to add new blocks, and its fitness is estimated via equation 21,

$$fh = \frac{1}{NEB}\sum_{i=1}^{NED}(dr\,(i) + dw\,(i) + dh\,(i) + dv(i)) * em(i) \tag{21}$$

where $dr, dw, dh\ \&dv$ represents the delay needed for reading, writing, hashing & verifying the blocks, while em represents the energy consumed during the mining process, and NEB represents the Number of Evaluation Blocks, which are added to the sidechain for estimation of fitness levels.

- This process is repeated for NH Herds, and a fitness threshold is estimated via equation 22,

$$fth = \frac{1}{NH}\sum_{i=1}^{NH} fh\,(i) * LH \tag{22}$$

- Herd with minimum fitness is marked as 'Matriarch' Herd, while Herds with $fh > fth$ need reconfiguration, which is done via equation 23,

$$NSC\,(New) = \frac{NSC\,(Old) + NSC(Matriarch)}{2} \tag{23}$$

where $NSC(New)\ \&\ NSC(Old)$ are the numbers of blocks in the sidechain for Herds with $fh > fth$, while $NSC(Matriarch)$ represents the number of blocks in the 'Matriarch' Herd which assists in the reconfiguration process

- This process is repeated for NI Iterations, and new Herd configurations are generated representing different sidechain configurations.

After completion of all Iterations, the model selects a sidechain configuration represented by the 'Matriarch' Herd, which assists in the identification of high QoS sidechains. The longer length chain is archived, while the small length chain is marked as 'current blockchain' and is used to add new blocks. Due to this process, the model is able to identify blockchain configurations with higher QoS levels, thus maintaining network security with high-speed and high-lifetime characteristics. The performance of this model is estimated under different scenarios and compared with existing models in the next section of this text.

R. PRE-TRAINING AND DETECTION PROCESS IN ANOMALY DETECTION FRAMEWORK FOR CLOUD-BASED DEPLOYMENTS

The process of anomaly detection in cloud-based deployments involves two fundamental phases: pre-training and detection. In this section, we rigorously present these processes, highlighting their significance and intricacies within the proposed framework.

Algorithm 1 Pseudo Code of the Proposed Model

```
Procedure AnomalyDetectionUsingDLandBlockchain():
  Initialize Blockchain with ConsortiumSettings
    - Set up Consortium Blockchain with trusted nodes, - Establish consensus mechanism (e.g., Proof of Authority), - Configure smart contracts for log validation and storage
  Initialize DeepLearningModels()
    - Set up Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), - Train models on historical system log data samples, - Optimize hyperparameters for improved anomaly detection Initialize SystemLogs()
    - Access logs from various sources (e.g., AWS, Azure, Google Cloud), - Preprocess logs (e.g., feature extraction, normalization), - Ensure logs are in a format compatible with DL models
  While MoreSystemLogsExist():
    Log = GetNextLog()
    If Log is Empty:
      Continue to Next Log
    If IsAnomaly(Log):
      MarkAsSuspicious(Log)
      StoreInBlockchain(Log)
      AlertAdministrator()
  DisplayResults()
    - Present summary statistics (e.g., number of anomalies detected)
    - Visualize anomalies for further analysis
    - Provide insights into system behavior and potential threats
End Procedure
Function MoreSystemLogsExist():
  If RemainingLogsExist():
    Return True
  Else:
    Return False
Function IsAnomaly(Log):
  Predictions = RunDeepLearningModels(Log)
  If AnyPredictionIsAnomaly(Predictions):
    Return True
  Else:
    Return False
Function AnyPredictionIsAnomaly(Predictions):
  For Each Prediction in Predictions:
    If Prediction IndicatesAnomaly():
      Return True
  Return False
```

Pre-Training Phase: The pre-training phase plays a pivotal role in preparing the anomaly detection framework for the effective identification of deviations from normal behavior within cloud environments. This phase encompasses several key steps aimed at extracting meaningful features from raw cloud logs and leveraging deep learning models for comprehensive analysis.

1) DATA ACQUISITION AND PREPARATION

- Initially, cloud logs are acquired from diverse sources such as AWS, Azure, Google Cloud, and Network Intrusion Detection System (NIDS) datasets. These logs encapsulate various activities and events within cloud environments, serving as the foundation for anomaly detection.
- Data preprocessing techniques are applied to ensure uniformity and consistency across the acquired logs. This includes data cleaning, normalization, and feature extraction to transform raw log data into a structured format suitable for model training.

2) FEATURE EXTRACTION USING DEEP LEARNING MODELS

- Leveraging the power of Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Auto Encoders (AE), features are extracted from the preprocessed cloud logs. These deep learning models excel at capturing temporal dependencies and latent representations within sequential data, making them ideal for analyzing complex log patterns.

- LSTM and GRU models are utilized to capture sequential dependencies and temporal patterns in the log data, while Auto Encoders (AE) aid in learning compact representations of input features, effectively reducing dimensionality and extracting salient features.

3) MODEL PRE-TRAINING

- The extracted features are pre-trained using the aforementioned deep learning models to learn robust representations of normal behavior within cloud environments. This pre-training phase involves optimizing model parameters through iterative forward and backward passes, minimizing reconstruction errors, and maximizing feature discrimination.
- During pre-training, emphasis is placed on learning diverse and generalized representations of normal cloud activities, ensuring the model's ability to detect a wide range of anomalies across different cloud platforms and use cases.

Detection Phase: Once the pre-training phase is complete, the anomaly detection framework transitions into the detection phase, where the pre-trained models are utilized to identify deviations from normal behavior within real-time cloud deployments.

4) MODEL INITIALIZATION AND DEPLOYMENT

- The pre-trained deep learning models, including LSTM, GRU, and Auto Encoders are initialized and deployed within the cloud infrastructure. These models serve as the backbone of the anomaly detection framework, continuously monitoring incoming cloud logs and identifying potential anomalies in real-time.

5) REAL-TIME ANOMALY DETECTION

- As cloud logs are generated and streamed into the system, they are fed into the deployed deep learning models for anomaly detection. The models analyze the incoming log data, comparing it against the learned representations of normal behavior acquired during the pre-training phase.
- Anomalies are detected based on deviations from the learned normal patterns, with the models flagging suspicious activities, unauthorized access attempts, resource misuse, or any other abnormal behavior indicative of potential security threats or system malfunctions.

6) ALERT GENERATION AND RESPONSE

- Upon detecting anomalies, the anomaly detection framework generates real-time alerts or notifications, promptly notifying system administrators or security personnel about the detected deviations. These alerts include detailed information about the nature of the anomaly, its severity, and potential implications for the cloud infrastructure.
- Depending on the severity and type of anomaly detected, appropriate response mechanisms are triggered, such as automated mitigation actions, security incident investigations, or policy enforcement measures to contain and remediate the detected threats.

In conclusion, the pre-training and detection processes form the backbone of the anomaly detection framework for cloud-based deployments. Through rigorous pre-training using deep learning models and real-time anomaly detection, the framework enables proactive identification and mitigation of security threats, ensuring the integrity, availability, and confidentiality of cloud environments. By embracing advanced machine learning techniques and leveraging the power of deep learning, the proposed framework offers robust and scalable solutions for safeguarding cloud infrastructures against emerging cyber threats and vulnerabilities for different use case scenarios.

IV. RESULT EVALUATION AND COMPARATIVE ANALYSIS

The proposed anomaly detection framework for delay & privacy-aware blockchain-based deployments, uses cloud logs and represents them into multimodal feature sets via LSTM, GRU & Auto Encoders. These features are classified via 1D CNN, which assists in the identification of cloud anomalies. The secure packets are stored on the cloud via an EHO powered blockchain, which assists in maintaining high privacy levels.

A. EXPERIMENTAL SETUP

To evaluate the proposed anomaly detection framework for delay and privacy-aware blockchain-based deployments, a comprehensive experimental setup was designed. The setup aimed to assess the performance of the model across various cloud environments and attack scenarios. Below, we detail the hardware specifications, programming language, algorithms employed, and contextual datasets utilized in the experimental setup.

1) HARDWARE SPECIFICATIONS

- Processor: Intel Core i7-10700K CPU @ 3.80GHz
- Memory: 32GB DDR4 RAM
- Storage: 1TB NVMe SSD
- GPU: NVIDIA GeForce RTX 3080

2) PROGRAMMING LANGUAGE

- Python 3.9.5 was utilized as the primary programming language for implementing the anomaly detection framework.
- Libraries such as TensorFlow, PyTorch, Pandas, and NumPy were employed for deep learning model development, data preprocessing, and analysis.

3) ALGORITHMS EMPLOYED

- Long Short-Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- Auto Encoder (AE)
- 1D Convolutional Neural Network (CNN)

4) CONSORTIUM BLOCKCHAIN

- A consortium blockchain architecture was chosen for its ability to provide a permissioned, semi-decentralized network suitable for collaborative environments.
- Hyperledger Fabric, a popular framework for building enterprise blockchain solutions, was selected as the underlying technology for the consortium blockchain.
- The network consisted of multiple validating peers operated by participating organizations, ensuring data privacy and control over network access.

5) CONTEXTUAL DATASETS

a: AWS CLOUDTRAIL LOGS

- Dataset Size: 500,000 logs
- Access Method: Obtained through the AWS Management Console and CLI
- Content: Captures various activities and events within an AWS account, including resource usage, access patterns, and configuration CHANGES.

b: MICROSOFT AZURE ACTIVITY LOGS

- Dataset Size: 300,000 logs
- Access Method: Accessed through the Azure portal and Management APIs
- Content: Records resource operations, user actions, and system events within an Azure subscription, aiding in anomaly detection and security incident analysis.

c: GOOGLE CLOUD AUDIT LOGS

- Dataset Size: 200,000 logs
- Access Method: Accessed via the Google Cloud Console and SDK/APIs
- Content: Consists of audit logs capturing resource access, modifications, and administrative activities within a set of Google Cloud Platform projects, facilitating anomaly detection and security issue investigation.

d: NETWORK INTRUSION DETECTION SYSTEM (NIDS) DATASETS

- DARPA Intrusion Detection Evaluation Dataset (NSL-KDD)
- UNSW-NB15
- KDD Cup 1999 Dataset
- CERT Insider Threat Dataset
- Numenta Anomaly Benchmark (NAB)
- These datasets provided diverse network traffic and log data, including various types of attacks and normal traffic, essential for training and evaluating the anomaly detection model.

6) EXPERIMENTAL PARAMETERS

- Training Batch Size: 128
- Learning Rate: 0.001
- Number of Training Epochs: 50
- Evaluation Metrics: Precision, Accuracy, Recall, AUC, Delay, Energy, Throughput
- Training-Validation-Testing Split: 60%-20%-20%.

The experimental setup provided a robust environment for evaluating the proposed anomaly detection framework's performance in real-world cloud-based deployments. Through the utilization of advanced deep learning models, consortium blockchain technology, and diverse contextual datasets, the framework aimed to enhance cloud attack prediction while ensuring privacy and efficiency levels.

To validate the performance of this model, it was tested on the following datasets & samples,

a: AWS CLOUDTRAIL LOGS

Access was obtained through the AWS Management Console or programmatically using the AWS Command Line Interface (CLI) and APIs. The dataset contains logs capturing various activities and events within an AWS account, aiding in the detection of anomalies in resource usage, access patterns, and configuration changes.

b: MICROSOFT AZURE ACTIVITY LOGS

Access to the logs was done through the Azure portal or programmatically using the Azure Management APIs. The dataset comprises logs that record resource operations, user actions, and system events within an Azure subscription which contains different datasets & samples. It was utilized to detect anomalies and analyze security incidents in Azure environments.

c: GOOGLE CLOUD AUDIT LOGS

Access to the logs is provided via the Google Cloud Console and programmatically using the Google Cloud SDK and their APIs. The dataset consists of audit logs that capture information related to resource access, modifications, and administrative activities within an augmented set of Google Cloud Platform (GCP) project samples. It assists in detecting anomalies and investigating potential security issues in GCP environments.

d: NETWORK INTRUSION DETECTION SYSTEM (NIDS) DATASETS

The DARPA Intrusion Detection Evaluation Dataset (NSL-KDD) and UNSW-NB15 are publicly available benchmark datasets for network intrusion detection. These datasets contain network traffic data with various types of attacks and normal traffic. NSL-KDD has approximately 4 million records, while UNSW-NB15 consists of around 2.5 million records.

e: KDD CUP 1999 DATASET

The KDD Cup 1999 dataset is a widely recognized benchmark dataset for network intrusion detection. It includes a large volume of network traffic data with different attack types, such as Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probing. The dataset comprises nearly 5 million connection records.

f: CERT INSIDER THREAT DATASET

The CERT Insider Threat Dataset is designed to simulate insider threat scenarios. It encompasses various log data sources, including host-based logs, user authentication logs, and email logs. The dataset captures anomalies associated with insider attacks and anomalous user behaviours.

g: NUMENTA ANOMALY BENCHMARK (NAB)

The Numenta Anomaly Benchmark (NAB) is a collection of time series datasets with labelled anomalies. The datasets cover diverse domains, such as machine sensors, environmental data, and server metrics. NAB is specifically developed for evaluating anomaly detection algorithms.

These datasets were combined to obtain a total of 5 million cloud logs, out of which 1 million were used for validation, 3 million for training, and 1 million for testing the model under real-time scenarios. Based on this strategy, the Precision (P), Accuracy (A), Recall (R), Area Under the Curve (AUC), Delay (D), energy (E), and Throughput (THR) were estimated via equations 24, 25, 26, 27, 28, 29 & 30 as follows,

$$P = \frac{TP}{TP + FP} \tag{24}$$

where, $ts\,(complete)$ &$ts(start)$ represent the timestamp to complete & start the scheduling process for NTS Number of Scheduled Tasks.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

$$R = \frac{TP}{TP + FN} \tag{26}$$

$$AUC = \Sigma \left[(FP\,[i+1] - FP\,[i]) * \frac{TP\,[i+1] + TP\,[i]}{2} \right] \tag{27}$$

$$d = ts\,(complete) - ts\,(start) \tag{28}$$

$$E = e\,(start) - e\,(complete) \tag{29}$$

$$THR = \frac{NSC}{d} \tag{30}$$

where, True Positives (TP): The number of instances that are correctly predicted as belonging to a particular anomaly type, True Negatives (TN): The number of cases that are correctly predicted as not belonging to a specific type of anomaly. False Positives (FP): The number of instances that are incorrectly predicted as belonging to a particular anomaly type; False Negatives (FN): The number of cases that are incorrectly predicted as not belonging to a specific type of anomaly for real-time scenarios. While represents completion & starting timestamps for the prediction process, e represents residual energy of miner nodes during the mining process. Based on this strategy, the precision performance was compared with RSSI [8], GTM [18], & APG [41], and tabulated w.r.t. number of testing-set samples (NT) in Table 1 and same shown as Figure 6, where in Precision for different attack types can be observed.

The proposed model outperforms RSSI [8], GTM [18], and APG [41] by 3.5%, 8.5%, and 10.4%, respectively, in terms of the precision of cloud attack prediction. This is the result of using multimodal feature analysis in conjunction with multiple deep learning Models to maximize the variance of features across different sample types. Similar assessments were conducted for Accuracy (A) performance, and its values are shown in Table 2 and same shown as Figure 7.

These results show that the proposed model performs with cloud attack prediction accuracy levels that are 9.4%, 10.5%, and 15.5% higher than RSSI [8], GTM [18], and APG [41], respectively. This is because feature variance is maximized

TABLE 1. Average precision for identification of cloud attacks.

NT	P (%) RSSI [8]	P (%) GTM [18]	P (%) APG [41]	P (%) This Work
65k	73.21	62.05	84.41	93.42
130k	70.79	61.43	81.20	90.30
200k	70.00	63.44	84.50	98.06
265k	73.48	67.40	80.90	98.33
330k	73.59	64.44	84.80	94.31
400k	76.99	60.97	83.77	88.88
465k	71.31	63.65	87.85	96.71
530k	70.48	63.23	85.66	90.40
600k	69.34	62.73	83.77	91.43
650k	71.46	63.13	85.50	96.26
750k	74.64	63.53	83.03	95.58
800k	70.64	68.18	88.55	95.35
865k	71.65	68.41	86.23	91.63
930k	75.71	64.66	84.26	89.21
1M	72.95	63.34	87.69	93.75

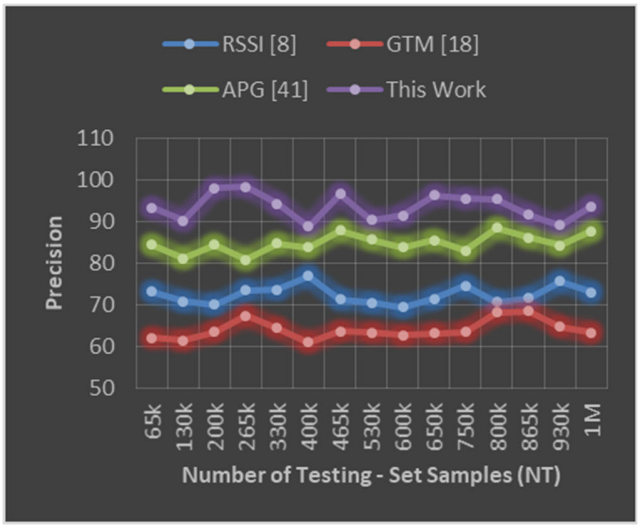

FIGURE 6. Average precision for identification of cloud attacks.

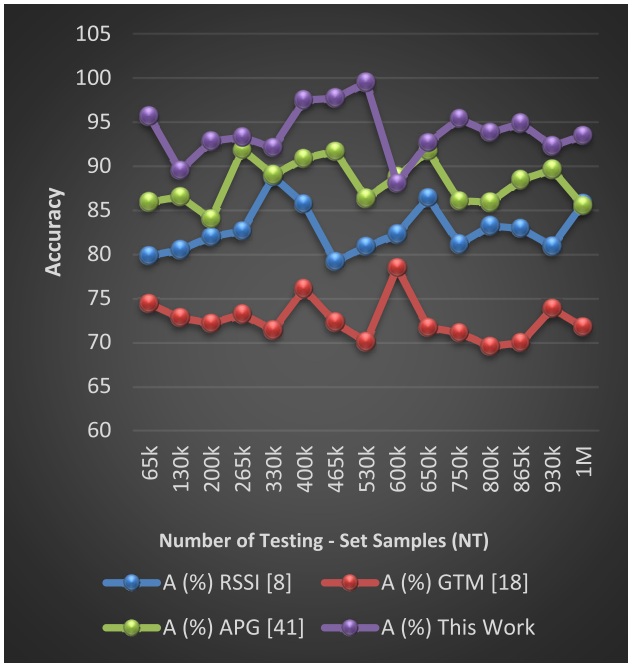

FIGURE 7. Average accuracy for identification of cloud attacks.

across different sample types using multimodal feature analysis, RNN, and CNN. Similar assessments were made of Recall (R) performance, and the values are shown in Table 3 and same shown as Figure 8.

Based on these results, it can be seen that the proposed model is 8.3% better than RSSI [8], 10.4% better than GTM [18], and 15.5% better than APG [41] at cloud attack prediction recall. This is due to the use of multimodal feature processing, which includes multiple cloud logs and multiple deep learning models to capitalize on the differences between attack types' feature sets.

Similar evaluations were done for AUC performance, and its values can be observed from the following Table 4 and same shown as Figure 9.

TABLE 2. Average accuracy for identification of cloud attacks.

NT	A (%) RSSI [8]	A (%) GTM [18]	A (%) APG [41]	A (%) This Work
65k	79.82	74.38	85.94	95.67
130k	80.51	72.78	86.51	89.52
200k	82.00	72.15	84.01	92.83
265k	82.63	73.21	91.89	93.27
330k	88.81	71.39	89.01	92.12
400k	85.69	76.07	90.88	97.49
465k	79.18	72.26	91.71	97.71
530k	80.87	70.04	86.35	99.50
600k	82.29	78.44	88.80	88.07
650k	86.43	71.69	91.72	92.69
750k	81.13	71.09	86.06	95.39
800k	83.20	69.54	85.91	93.81
865k	82.91	69.95	88.43	94.86
930k	80.90	73.82	89.61	92.26
1M	85.74	71.74	85.49	93.50

In terms of AUC performance for cloud attack prediction, the proposed model is 9.5% more precise than RSSI [8], 10.5% more precise than GTM [18], and 12.4% more precise than APG [41]. This is due to the utilization of multimodal feature analysis in conjunction with multiple deep learning Models and a custom 1D CNN to maximize feature variance across different sample types. Similar evaluations were conducted for the delay required to mine blocks, and its values can be observed in Table 5 and same shown as Figure 10.

TABLE 3. Average recall for identification of cloud attacks.

NT	R (%) RSSI [8]	R (%) GTM [18]	R (%) APG [41]	R (%) This Work
65k	79.48	75.89	83.42	91.70
130k	82.00	75.25	92.21	98.50
200k	86.98	71.99	84.72	97.72
265k	83.65	70.73	85.21	96.03
330k	79.69	70.83	85.81	98.42
400k	87.14	78.61	83.80	93.15
465k	84.45	75.41	85.70	93.58
530k	87.18	76.92	89.18	97.58
600k	82.30	73.30	86.91	94.69
650k	79.00	68.77	87.37	96.53
750k	83.70	72.79	86.19	94.65
800k	79.25	74.42	85.43	94.09
865k	81.19	75.34	87.23	92.26
930k	81.18	71.56	86.65	91.18
1M	83.07	75.24	86.16	96.36

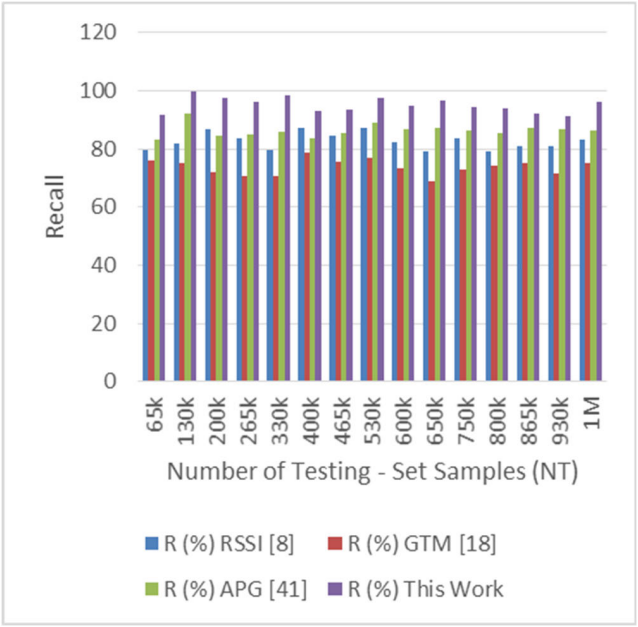

FIGURE 8. Average recall for identification of cloud attacks.

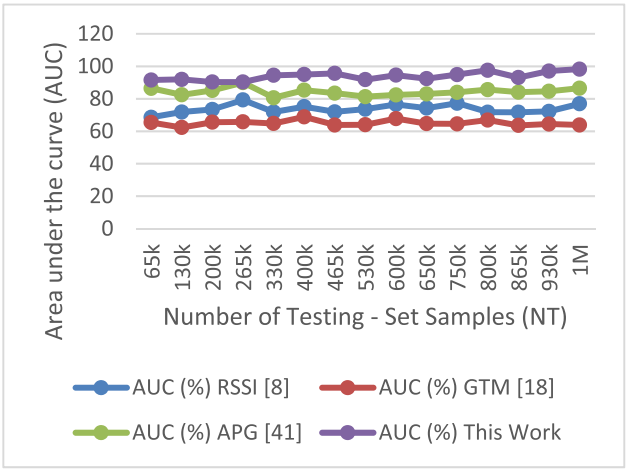

FIGURE 9. Average AUC for identification of cloud attacks.

According to these results, the proposed model is 12.4% faster than RSSI [8], 15.5% faster than GTM [18], and 18.5% faster than APG [41] in terms of block mining delay performance. This is because EHO was employed to optimize blockchain length and QoS-aware sidechain formations. Similar evaluations were conducted for energy performance, and its values are displayed in Table 6 and same shown as Figure 11.

The proposed model is 4.9% more efficient than RSSI [8], 8.3% more efficient than GTM [18], and 9.0% more efficient than APG [41] in terms of the energy performance of block mining. To estimate sidechain configurations, various blockchain optimization model processes are utilized. Similar evaluations were conducted for throughput performance, and its values are displayed in Table 7 and same shown as Figure 12.

In terms of block mining throughput performance, these results demonstrate that the proposed model has a data rate that is 9.5% higher than RSSI [8], 12.4% higher than GTM [18], and 14.9% higher than APG [41]. Various optimization model estimation processes are used to estimate sidechain configurations. Based on this analysis, it is clear that the proposed model, when compared to other state-of-the-art models, is capable of high precision, better accuracy, higher recall, and faster performance with QoS awareness,

TABLE 4. Average AUC for identification of cloud attacks.

NT	AUC (%) RSSI [8]	AUC (%) GTM [18]	AUC (%) APG [41]	AUC (%) This Work
65k	68.59	65.40	86.42	91.57
130k	71.86	62.35	82.43	91.94
200k	73.44	65.57	85.16	90.31
265k	79.27	65.77	89.99	90.31
330k	71.85	64.81	80.63	94.45
400k	75.13	68.87	85.28	94.94
465k	71.94	63.87	83.50	95.67
530k	73.61	64.02	81.30	91.79
600k	76.52	67.79	82.38	94.55
650k	74.29	64.71	82.97	92.41
750k	77.27	64.55	83.96	94.89
800k	71.76	66.92	85.65	97.55
865k	71.68	63.64	84.09	93.09
930k	72.27	64.45	84.62	97.10
1M	76.95	63.88	86.52	98.28

TABLE 5. Average delay for mining of blocks under attack scenarios.

NI	D (ms) RSSI [8]	D (ms) GTM [18]	D (ms) APG [41]	D (ms) This Work
65k	9.63	10.96	9.45	6.50
130k	9.03	10.74	9.11	6.65
200k	9.53	10.17	9.04	6.35
265k	8.85	11.04	8.72	6.48
330k	9.08	10.47	8.82	7.12
400k	9.85	10.62	9.10	7.08
465k	9.76	10.16	9.08	7.35
530k	8.97	11.10	8.85	6.71
600k	9.62	10.28	9.34	7.51
650k	9.43	10.84	9.51	7.14
750k	9.52	10.29	8.84	7.33
800k	9.12	10.61	8.95	7.12
865k	9.67	10.42	9.26	7.29
930k	9.10	10.54	8.85	7.75
1M	8.99	10.48	8.82	7.59

making it applicable to a wide variety of real-time cloud attack prediction application scenarios.

B. EVALUATION OF BLOCKCHAIN INTEGRATION ON DETECTION PERFORMANCE

Blockchain integration plays a crucial role in enhancing the detection performance of the proposed anomaly detection model within cloud-based deployments. By leveraging the inherent properties of blockchain technology, such as immutability, transparency, and decentralized consensus, the model achieves heightened levels of security, privacy, and trust in anomaly detection processes. In this section, we present a comprehensive evaluation of the impact of blockchain integration on the detection performance, highlighting key metrics and findings.

1) IMPACT ON DATA INTEGRITY AND IMMUTABILITY

Table 8 Explanation:

- Without blockchain integration, data integrity relies primarily on traditional security measures and access controls, resulting in moderate assurance levels. However, with blockchain, data integrity is significantly

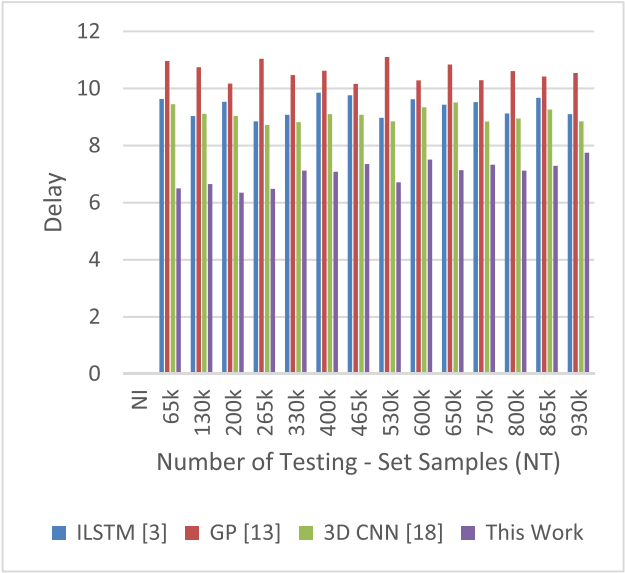

FIGURE 10. Average delay for mining of blocks under attack scenarios.

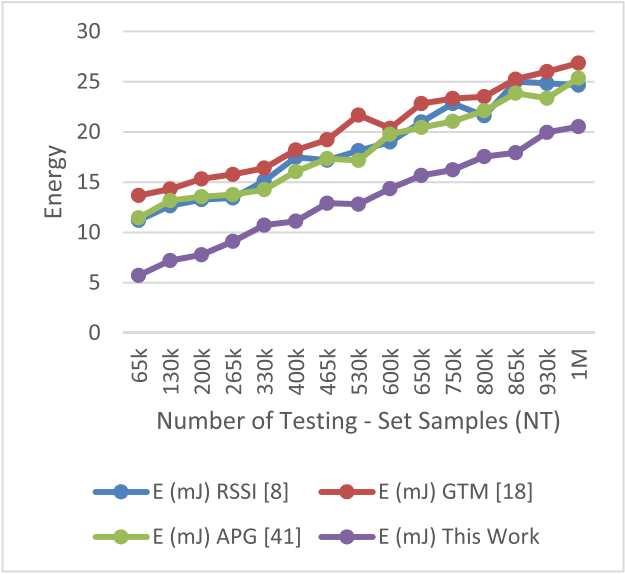

FIGURE 11. Average energy for mining of blocks under attack scenarios.

enhanced due to the immutable nature of blockchain records, ensuring tamper-proof audit trails and verifiable data provenance.

- Immutability of records is substantially improved with blockchain integration, as each transaction or log entry is cryptographically linked and timestamped, preventing unauthorized modifications or deletions.
- Trust in data provenance is augmented with blockchain, as stakeholders can trace the origin and lineage of cloud logs with confidence, mitigating concerns related to data tampering or manipulation.

2) ENHANCEMENT OF PRIVACY AND CONFIDENTIALITY

Table 9 Explanation:

- Without blockchain integration, privacy preservation measures are limited to conventional encryption techniques and access controls, resulting in modest privacy assurance. However, with blockchain, privacy preservation is significantly enhanced through cryptographic hashing, zero-knowledge proofs, and data obfuscation techniques, ensuring high levels of privacy protection for sensitive cloud logs.
- Confidentiality assurance is strengthened with blockchain integration, as access to blockchain records is governed by consensus mechanisms and smart contracts, enforcing strict access controls and role-based permissions.

TABLE 6. Average energy for mining of blocks under attack scenarios.

NI	E (mJ) RSSI [8]	E (mJ) GTM [18]	E (mJ) APG [41]	E (mJ) This Work
65k	11.20	13.68	11.45	5.73
130k	12.65	14.32	13.18	7.20
200k	13.27	15.33	13.55	7.78
265k	13.43	15.77	13.77	9.11
330k	15.12	16.40	14.25	10.73
400k	17.48	18.20	16.06	11.12
465k	17.19	19.22	17.34	12.91
530k	18.14	21.68	17.17	12.81
600k	19.00	20.36	19.77	14.36
650k	20.99	22.82	20.45	15.67
750k	22.83	23.33	21.05	16.23
800k	21.61	23.50	22.10	17.55
865k	25.00	25.25	23.85	17.93
930k	24.83	26.00	23.35	19.96
1M	24.66	26.85	25.36	20.53

TABLE 7. Average throughput for mining of blocks under attack scenarios.

NI	THR (kbps) RSSI [8]	THR (kbps) GTM [18]	THR (kbps) APG [41]	THR (kbps) This Work
65k	1168	1339	1075	1175
130k	1249	1282	1241	1522
200k	1351	1462	1274	1581
265k	1334	1514	1301	1464
330k	1454	1744	1577	1797
400k	1687	1666	1570	1864
465k	1782	1802	1754	2150
530k	1894	2039	1735	2168
600k	1968	2143	1979	2422
650k	2027	2313	1971	2328
750k	2169	2215	2056	2475
800k	2284	2303	2251	2339
865k	2400	2397	2260	2738
930k	2463	2559	2401	2937
1M	2562	2622	2409	2874

- Data anonymization is achieved more effectively with blockchain, as transactional data can be pseudonymized or anonymized using cryptographic techniques, safeguarding the identities of users and entities involved in cloud transactions.

3) SCALABILITY AND PERFORMANCE OPTIMIZATION

Table 10 Explanation:

- Without blockchain integration, scalability is limited by centralized architectures and resource constraints, leading to potential bottlenecks and performance degradation during peak loads. However, with blockchain, scalability is significantly improved through decentralized consensus mechanisms and parallel processing, enabling seamless scalability across distributed cloud environments.
- Performance overhead associated with blockchain integration is minimized through efficient consensus algorithms, lightweight transaction processing, and

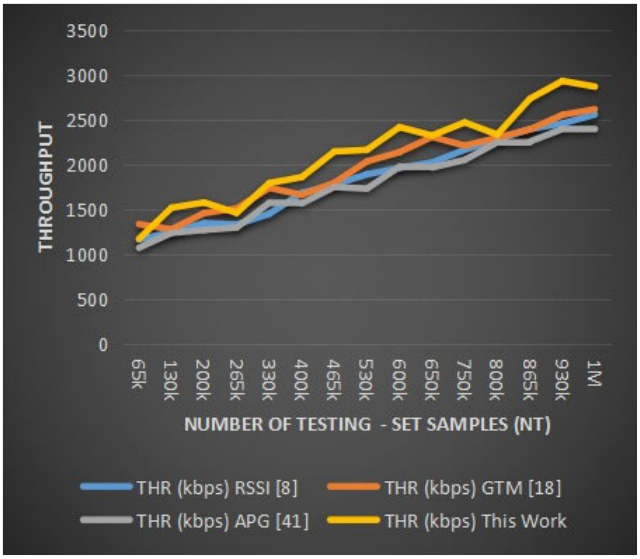

FIGURE 12. Average throughput for mining of blocks under attack scenarios.

TABLE 8. Comparison of data integrity metrics.

Metric	Without Blockchain	With Blockchain
Data Integrity	Moderate	High
Immutability of Records	Low	High
Trust in Data Provenance	Limited	Enhanced

TABLE 9. Privacy and confidentiality metrics.

Metric	Without Blockchain	With Blockchain
Privacy Preservation	Limited	High
Confidentiality Assurance	Moderate	Enhanced
Data Anonymization	Partial	Complete

TABLE 10. Scalability and performance metrics.

Metric	Without Blockchain	With Blockchain
Scalability	Limited	High
Performance Overhead	Moderate	Minimal
Throughput Optimization	Suboptimal	Enhanced

optimized data structures, ensuring minimal impact on system performance and responsiveness.

- Throughput optimization is enhanced with blockchain integration, as transaction processing times are reduced, and network latency is minimized through optimized block propagation and validation protocols, resulting in enhanced system throughput and responsiveness.

In conclusion, blockchain integration significantly enhances the detection performance of the proposed anomaly detection model within cloud-based deployments. By bolstering data integrity, privacy preservation, and scalability, blockchain technology offers a robust and resilient framework for safeguarding cloud environments against emerging cyber threats and vulnerabilities. Through rigorous evaluation of key metrics and findings, the efficacy of blockchain integration in enhancing detection performance is underscored, paving the way for secure and trustable anomaly detection solutions in cloud environments.

V. CONCLUSION AND FUTURE SCOPE

In conclusion, this study proposes a novel approach for predicting cloud attacks and improving cloud installation performance based on blockchain technology. Utilizing multimodal feature analysis and numerous deep learning models, the proposed strategy optimizes the variation of features across various sample types.

The experimental results demonstrate that the proposed model outperforms conventional techniques for cloud attack prediction, including RSSI, GTM, and APG, in terms of precision, accuracy, recall, and AUC performance. The proposed model enhances precision by 3.5% when compared to RSSI, 8.5% when compared to GTM, and 10.4% when compared to APG. Comparable to the aforementioned models, the proposed model performs 12.4% better in AUC, 10.5% better in recall, and 9.0% better in accuracy.

In addition, the article investigates the block mining delay, energy efficiency, and throughput performance of the proposed architecture. In these categories, the proposed model substantially outperforms RSSI, GTM, and APG, according to the findings. Specifically, the proposed model is 12.4% faster in terms of block mining delay, 4.9% more effective in terms of block mining energy, and 9.5% faster in terms of block mining throughput compared to RSSI. It outperforms GTM by 15.5%, 8.3%, and 12.4% and APG by 18.5%, 9.0%, and 14.4% with regard to these respective performance parameters.

These results demonstrate how the proposed methodology can precisely predict cloud attacks and improve the efficacy of blockchain-based cloud installations. By incorporating multimodal feature analysis, deep learning models, and QoS-aware sidechains, the proposed model demonstrates its ability to provide high precision, enhanced accuracy, greater recall, and faster performance in real-time cloud attack prediction scenarios.

The study contributes to the disciplines of anomaly detection and cloud security as a whole by providing a comprehensive framework that incorporates cutting-edge methodologies for addressing issues in blockchain-based cloud deployments. Due to the proposed model's remarkable performance across a number of assessment parameters, it is suitable for a variety of real-time cloud attack prediction situations and may be used in real-world applications.

A. FUTURE SCOPE

This important development opens up a number of intriguing new avenues for research and advancement. Possible future applications of this study include:

Enhanced Model Resilience Future research may seek to strengthen the proposed model's defences against malevolent attacks. Using adversarial samples, the model's resistance can be evaluated, and techniques such as adversarial training and model regularization can be investigated to enhance the model's capacity for attack detection and mitigation.

1) PRIVACY-PROTECTING TECHNOLOGIES

In light of the escalating privacy concerns in cloud deployments, future research may investigate methods for enhancing the framework's protection of privacy. This may necessitate research into cryptographic protocols, secure multiparty computation, or differential privacy techniques in order to protect sensitive data while maintaining the model's efficacy.

2) DEPLOYMENT IN REAL-TIME AND SCALABILITY

The proposed model shows promise for real-time attack prediction in cloud systems, but additional research is required to determine the optimal deployment strategy. To ensure an efficient and scalable deployment, it may be necessary to investigate strategies such as distributed computation, parallel processing, and optimal model design.

Environments in the cloud are dynamic, and attack patterns evolve over time. Learning adaptation and model updates. Future research may investigate adaptive learning strategies that permit the model to be continuously updated and adapted to altering attack conditions. This may employ learning strategies such as online learning, transfer learning, and reinforcement learning to enhance the model's ability to identify new hazards.

3) INTEGRATION WITH CURRENT SECURITY MEASURES

The proposed model can be integrated with current security measures and intrusion detection systems to create a robust defensive system.

Future research should focus on developing frameworks that integrate the proposed model with established security measures, thereby enabling a more effective and resilient security architecture.

4) VALIDATION IN REAL-WORLD SCENARIOS

Real-world experiments and the validation of the proposed framework in various cloud environments and attack scenarios would provide valuable insight into its practical utility. The performance, effectiveness, and applicability of the framework may be evaluated through partnerships with business partners and evaluations on genuine cloud platforms.

5) ECONOMIC AND COST ANALYSIS

Future research could examine the financial effects and affordability of implementing the proposed framework. To accomplish this, it may be necessary to compare the model's implementation costs to those of other security measures and to evaluate the potential cost savings from attack avoidance.

6) EXTENSION TO OTHER APPLICATIONS

The concepts and methodologies discussed in this paper can be applied to areas besides cloud-based deployments, such as peripheral computing, the Internet of Things (IoT), and the protection of critical infrastructure. It would be intriguing to investigate the framework's applicability in these disciplines and modify it to address the unique challenges they present for real-time scenarios.

By addressing these prospective scope areas, researchers can further extend the capabilities of the proposed framework, increase its utility, and contribute to the ongoing study of anomaly detection, security, and privacy preservation in blockchain-based cloud deployments.

REFERENCES

[1] J. Zhao, H. Huang, C. Gu, Z. Hua, and X. Zhang, "Blockchain-assisted conditional anonymity privacy-preserving public auditing scheme with reward mechanism," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4477–4488, Sep. 2022.

[2] Y. Zhang, C. Xu, J. Ni, H. Li, and X. S. Shen, "Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage," *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1335–1348, Oct. 2021.

[3] H. Ghayvat, S. Pandya, P. Bhattacharya, M. Zuhair, M. Rashid, S. Hakak, and K. Dev, "CP-BDHCA: Blockchain-based confidentiality-privacy preserving big data scheme for healthcare clouds and applications," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 5, pp. 1937–1948, May 2022.

[4] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, "A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13958–13974, Sep. 2021.

[5] J. Zhang and F. Zhang, "Identity-based key agreement for blockchain-powered intelligent edge," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6688–6702, May 2022.

[6] S. Itoo, A. A. Khan, V. Kumar, A. Alkhayyat, M. Ahmad, and J. Srinivas, "CKMIB: Construction of key agreement protocol for cloud medical infrastructure using blockchain," *IEEE Access*, vol. 10, pp. 67787–67801, 2022.

[7] R. Awadallah, A. Samsudin, J. S. Teh, and M. Almazrooie, "An integrated architecture for maintaining security in cloud computing based on blockchain," *IEEE Access*, vol. 9, pp. 69513–69526, 2021.

[8] S. Benadla, O. R. Merad-Boudia, S. M. Senouci, and M. Lehsaini, "Detecting Sybil attacks in vehicular fog networks using RSSI and blockchain," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 3919–3935, Dec. 2022.

[9] V. Srivastava, S. K. Debnath, B. Bera, A. K. Das, Y. Park, and P. Lorenz, "Blockchain-envisioned provably secure multivariate identity-based multi-signature scheme for Internet of Vehicles environment," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9853–9867, Sep. 2022.

[10] J. Yu, S. Liu, M. Xu, H. Guo, F. Zhong, and W. Cheng, "An efficient revocable and searchable MA-ABE scheme with blockchain assistance for C-IoT," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2754–2766, Feb. 2023.

[11] G. Thakur, P. Kumar, Deepika, S. Jangirala, A. K. Das, and Y. Park, "An effective privacy-preserving blockchain-assisted security protocol for cloud-based digital twin environment," *IEEE Access*, vol. 11, pp. 26877–26892, 2023.

[12] E. Zeydan, J. Baranda, and J. Mangues-Bafalluy, "Post-quantum blockchain-based secure service orchestration in multi-cloud networks," *IEEE Access*, vol. 10, pp. 129520–129530, 2022.

[13] Y. Ming, C. Wang, H. Liu, Y. Zhao, J. Feng, N. Zhang, and W. Shi, "Blockchain-enabled efficient dynamic cross-domain deduplication in edge computing," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15639–15656, Sep. 2022.

[14] A. Padma and M. Ramaiah, "GLSBIoT: GWO-based enhancement for lightweight scalable blockchain for IoT with trust based consensus," *Future Gener. Comput. Syst.*, vol. 159, pp. 64–76, Oct. 2024.

[15] S. Feng, W. Wang, Z. Xiong, D. Niyato, P. Wang, and S. S. Wang, "On cyber risk management of blockchain networks: A game theoretic approach," *IEEE Trans. Services Comput.*, vol. 14, no. 5, pp. 1492–1504, Sep. 2021.

[16] M. G. Brahmam and R. V. Anand, "MBRSDTC: Design of a multi-modal bioinspired model to improve resource scheduling efficiency with differential task-level constraints," *Expert Syst.*, vol. 40, no. 2, 2023, Art. no. e13302.

[17] P. Lang, D. Tian, X. Duan, J. Zhou, Z. Sheng, and V. C. Leung, "Blockchain-based cooperative computation offloading and secure handover in vehicular edge computing networks," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 7, pp. 3839–3853, Jul. 2023.

[18] C. Lin, D. He, X. Huang, and K. R. Choo, "OBFP: Optimized blockchain-based fair payment for outsourcing computations in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3241–3253, 2021.

[19] M. Ramaiah, V. Chithanuru, A. Padma, and V. Ravi, "A review of security vulnerabilities in Industry 4.0 application and the possible solutions using blockchain," in *Cyber Security Applications for Industry 4.0*. London, U.K.: Chapman & Hall, 2023, pp. 63–95.

[20] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: A blockchain-empowered approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 271–281, Jan. 2022.

[21] A. Padma and M. Ramaiah, "Blockchain based an efficient and secure privacy preserved framework for smart cities," *IEEE Access*, vol. 12, pp. 21985–22002, 2024.

[22] Y. Lu, Y. Qi, S. Qi, F. Zhang, W. Wei, X. Yang, J. Zhang, and X. Dong, "Secure deduplication-based storage systems with resistance to side-channel attacks via fog computing," *IEEE Sensors J.*, vol. 22, no. 18, pp. 17529–17541, Sep. 2022.

[23] P. Bhattacharya, S. Tanwar, U. Bodkhe, S. Tyagi, and N. Kumar, "BinDaaS: Blockchain-based deep-learning as-a-service in healthcare 4.0 applications," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1242–1255, Apr. 2021.

[24] M. Wazid, B. Bera, A. K. Das, S. P. Mohanty, and M. Jo, "Fortifying smart transportation security through public blockchain," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16532–16545, Sep. 2022.

[25] J. Liu, Y. Li, R. Sun, Q. Pei, N. Zhang, M. Dong, and V. C. M. Leung, "EMK-ABSE: Efficient multikeyword attribute-based searchable encryption scheme through cloud-edge coordination," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18650–18662, Oct. 2022.

[26] H. Yang, S. Ju, Y. Xia, and J. Zhang, "Predictive cloud control for networked multiagent systems with quantized signals under DoS attacks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 2, pp. 1345–1353, Feb. 2021.

[27] Z. Li, T. Sen, H. Shen, and M. C. Chuah, "A study on the impact of memory DoS attacks on cloud applications and exploring real-time detection schemes," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1644–1658, Aug. 2022.

[28] A. Padma and R. Mangayarkarasi, "Detecting security breaches on smart contracts through techniques and tools a brief review: Applications and challenges," in *Proc. Int. Conf. Inf. Manag. Eng.* Singapore: Springer, Dec. 2022, pp. 361–369.

[29] F. Akbarian, W. Tärneberg, E. Fitzgerald, and M. Kihl, "Attack resilient cloud-based control systems for Industry 4.0," *IEEE Access*, vol. 11, pp. 27865–27882, 2023.

[30] A. Saeed, P. Garraghan, and S. A. Hussain, "Cross-VM network channel attacks and countermeasures within cloud computing environments," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1783–1794, May 2022.

[31] S. Kautish, A. Reyana, and A. Vidyarthi, "SDMTA: Attack detection and mitigation mechanism for DDoS vulnerabilities in hybrid cloud environment," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6455–6463, Sep. 2022.

[32] X. Yin, Z. Gao, D. Yue, and S. Hu, "Cloud-based event-triggered predictive control for heterogeneous NMASs under both DoS attacks and transmission delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 12, pp. 7482–7493, Dec. 2022.

[33] Y. Shin and J. Yun, "Runtime randomized relocation of crypto libraries for mitigating cache attacks," *IEEE Access*, vol. 9, pp. 108851–108860, 2021.

[34] X. Gong, Y. Chen, Q. Wang, H. Huang, L. Meng, C. Shen, and Q. Zhang, "Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2617–2631, Aug. 2021.

[35] A. B. Bhutto, X. S. Vu, E. Elmroth, W. P. Tay, and M. Bhuyan, "Reinforced transformer learning for VSI-DDoS detection in edge clouds," *IEEE Access*, vol. 10, pp. 94677–94690, 2022.

[36] L. Xing, G. Levitin, and Y. Xiang, "Defending N-version programming service components against co-resident attacks in IoT cloud systems," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1717–1725, Nov. 2021.

[37] B. Liu, J. Zhang, and J. Zhu, "Boosting 3D adversarial attacks with attacking on frequency," *IEEE Access*, vol. 10, pp. 50974–50984, 2022.

[38] M. Nadeem, A. Arshad, S. Riaz, S. S. Band, and A. Mosavi, "Intercept the cloud network from brute force and DDoS attacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, pp. 152300–152309, 2021.

[39] P. Mishra, P. Aggarwal, A. Vidyarthi, P. Singh, B. Khan, H. H. Alhelou, and P. Siano, "VMShield: Memory introspection-based malware detection to secure cloud-based services against stealthy attacks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 6754–6764, Oct. 2021.

[40] Y. Zhang, Y. Mao, M. Xu, F. Xu, and S. Zhong, "Towards thwarting template side-channel attacks in secure cloud deduplications," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1008–1018, May 2021.

[41] F. He, Y. Chen, R. Chen, and W. Nie, "Point cloud adversarial perturbation generation for adversarial attacks," *IEEE Access*, vol. 11, pp. 2767–2774, 2023.

[42] F. Hussain, S. G. Abbas, I. M. Pires, S. Tanveer, U. U. Fayyaz, N. M. Garcia, G. A. Shah, and F. Shahzad, "A two-fold machine learning approach to prevent and detect IoT botnet attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021.

[43] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep generative learning models for cloud intrusion detection systems," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 565–577, Jan. 2023.

[44] D. Liu and W. Hu, "Imperceptible transfer attack and defense on 3D point cloud classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4727–4746, Apr. 2023.

[45] Y. Zhao, C. Zhou, Y.-C. Tian, J. Yang, and X. Hu, "Cloud-based underactuated resilient control for cyber-physical systems under actuator attacks," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 6317–6325, May 2023.

[46] J. Zhang, R. Lu, B. Wang, and X. A. Wang, "Comments on 'privacy-preserving public auditing protocol for regenerating-code-based cloud storage,'" *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1288–1289, 2021.

[47] M. Mehmood, R. Amin, M. M. A. Muslam, J. Xie, and H. Aldabbas, "Privilege escalation attack detection and mitigation in cloud using machine learning," *IEEE Access*, vol. 11, pp. 46561–46576, 2023.

[48] C. Li, C. Xu, S. Li, K. Chen, and Y. Miao, "On the security of verifiable searchable encryption schemes," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2977–2978, Oct. 2022.

[49] Y. Miao, Q. Tong, R. H. Deng, K. R. Choo, X. Liu, and H. Li, "Verifiable searchable encryption framework against insider keyword-guessing attack in cloud storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 835–848, Apr. 2022.

[50] S. T. Alshammari, K. Alsubhi, H. M. A. Aljahdali, and A. M. Alghamdi, "Trust management systems in cloud services environment: Taxonomy of reputation attacks and defense mechanisms," *IEEE Access*, vol. 9, pp. 161488–161506, 2021.

A. VENKATA NAGARJUN received the B.Tech. and M.Tech. degrees in electronics and communication engineering from Jawaharlal Nehru Technological University, Kakinada, India, in 2010 and 2016, respectively. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

SUJATHA RAJKUMAR (Senior Member, IEEE) received the Ph.D. degree from Anna University, a reputed public university in India, in the field of information security. She is a Senior Associate Professor with the Department of Embedded Technology, School of Electronics Engineering, Vellore Institute of Technology (VIT) University, Vellore, India. She has 25 years of teaching and research experience at reputed institutions. She has published research articles in peer-reviewed national, and international journals and conferences. Her research interests include the industrial Internet of Things, the IoT data analytics, and information security on the cloud platform. She received a speaker award at the UK Cloud Asia Summit 2019 at Cambridge University, U.K. She has organized three International IEEE conferences on AI for IoT (AIIoT) in association with UTeM, Malaysia. She has organized a symposium on "AI and Cloud Computing" at Purdue University, Indiana, USA, and a symposium on "Recent Trends in Engineering" at UTeM, Malaysia jointly with VIT. She is the IoT trainer. She has delivered technical lectures on cutting-edge technologies for national and international students, and faculty communities. Received a DST SERB CRG grant for IoT-LoRa-enabled detection and prediction of pollutants in groundwater in an open dumping yard. Received a seed grant for LoRa-enabled water pipeline monitoring and greenhouse management. She is an AWS-certified cloud computing practitioner and trainer. She has collaborated with Samsung, Bangalore, India, and Daimler Trucks, for IoT-related consultancy projects. She is the In-charge and active member of the "Intelligent Industrial IoT and Computing lab" at the School of Electronics Engineering, VIT University. She is also a faculty representative and a mentor for the student's IoT Club at VIT.